A bi-objective uncapacitated facility location problem

Young-Soo Myung a,*, Hu-gon Kim b, Dong-wan Tcha b

a Department of Business Administration, Dankook University, Cheonan, Chungnam 330-714, South Korea
b Department of Management Science, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong, Yusong-Gu, Taejon 305-701, South Korea

Received 1 June 1995; accepted 1 May 1996

Abstract

We consider a bi-objective model for uncapacitated facility location where one objective is to maximize the net profit and the other to maximize the profitability of the investment. We first characterize the structure of the model having both a linear and a fractional objective function. In order to generate efficient solutions for the model, we develop a heuristic procedure which has computational advantages over existing methods. A numerical example is presented to illustrate the solution process and computational tests on large scale problems are also provided. © 1997 Elsevier Science B.V.

Keywords: Location; Bi-objective; Profitability

1. Introduction

Facility location problems have been extensively studied during the past two decades, which is well surveyed in [1,2,10,15]. The typical objective in most facility location models, especially those in the private sector, is to maximize the profit which is calculated by subtracting the total cost from the total revenue. Provided that demands are fully satisfied, this objective is the same as cost minimization. In these models, costs include a fixed charge for establishing each facility and the variable costs associated with production and transportation. Usually, the fixed cost means an investment of a single cash outlay at the initial time period, whereas the revenue and the variable costs occur periodically in a stream-like manner. Therefore, the cash flows differing in their temporal occurrences should be adjusted to the same base period using the discount factor.

The objective of profit maximization, though the most widely used decision criterion in business decision-making environments, is not necessarily the best one. In reality, investment decisions are often based not so much on the absolute size of the profit as on the profitability of each investment alternative. Particularly under a limited budget for investments, the profitability of each investment is the main factor to take into account in selecting the most desirable one. The profitability of an investment is usually measured by the ratio of the level of the profit gained to the size of the investment, which is referred to as the profitability index (PI), or the rate of return [3].

The objective of maximizing profit may conflict with the one of maximizing profitability, since the marginal return on an investment generally decreases as its size increases. Here, we consider a bi-objective uncapacitated facility location problem (BUFLP). The problem is to select the location of uncapacitated facilities in such a way as (i) to maximize the profit and (ii) to maximize the profitability of the
total investment. Though there exist many multiobjective location models in the literature, no models have addressed the above two objectives as yet. For an excellent survey of multiobjective location studies, see Current et al. [5].

Our model is a static one, and thus can be seen irrelevant to the profitability analysis since the rate of return of an investment is defined on the associated cash flows over multiple periods. However, for the dynamic locational environment where each customer’s demands are specified period by period, the static model is still appropriate if either each customer’s period-demands are the same over all periods or the locating decision should be made only at the initial time period. Furthermore, even the general dynamic location model without the above two restrictions can also be formulated as a static UFLP [9]. Therefore, we can apply the solution approach developed here for solving the more general models.

Our model has two objective functions, linear and fractional. Since it is difficult to generate all the efficient (non-dominated) solutions, we only aim at generating an approximation of the efficient frontier. The heuristic developed for the purpose solves a parametric single objective UFLP. The paper is organized as follows. In the next section, we present a mathematical programming formulation for the proposed BUFLP. Exploiting the structural properties of the model, we develop the above-mentioned heuristic in Section 3. Also, we compare our method with the existing methods, demonstrating that our method has computational advantages over them. Section 4 provides an illustrative example, and in Section 5, computing experiments on large scale problems are summarized. Finally, some concluding remarks are given in the last section.

2. Model formulation

To formulate the proposed model, we use the following notation: $I = \{1, \ldots, m\}$ is the set of potential sites for facilities; $J = \{1, \ldots, n\}$ is the set of customers; $D_j$ is the demand of customer $j$; $p$ is the unit price of the commodity; $t_{ij}$ is the nonnegative variable production and transportation cost per unit of customer $j$’s demand supplied from facility $i$; $f_i$ is the positive fixed cost for establishing facility $i$; $\pi_o$ is the positive minimum required profit level; $y_i = 1$ if facility $i$ is established and 0 otherwise; and $x_{ij}$ is the fraction of customer $j$’s demand which is supplied from facility $i$.

Let
\[
IN(y) = \sum_{i \in I} f_i y_i,
\]
\[
CI(x) = \sum_{i \in I} \sum_{j \in J} (p - t_{ij}) D_j x_{ij}.
\]

$CI(x)$ equals the total revenue minus the total of variable costs. Then $CI(x) - IN(y)$ is the net profit, and the PI is represented by $(CI(x) - IN(y))/IN(y)$. For expositional brevity, however, we represent the PI as $CI/IN$ instead.

We now present the following bi-objective 0-1 integer programming model for our BUFLP:

\[
\text{(BP)} \quad \max \left[ CI(x) - IN(y), CI(x)/IN(y) \right], \quad \text{s.t.} \quad CI(x) - IN(y) \geq \pi_o, \quad (1) \\
\sum_{i \in I} x_{ij} \leq 1, \quad j \in J, \quad (2) \\
x_{ij} \leq y_i, \quad i \in I, \quad (3) \\
x_{ij} \geq 0, \quad i \in I, \quad j \in J, \quad (4) \\
y_i = 0 \text{ or } 1, \quad i \in I. \quad (5)
\]

Inequality (1) is introduced to avoid trivial cases. Note first from the positiveness of $\pi_o$ and $f_i$’s that (1) forces any feasible solution of (BP) to satisfy $IN(y) > 0$, legitimizing the fractional objective. Besides in real-world business environments, any investment of not anticipating a certain level of net profit does not have to be seriously considered.

Any feasible vector $(x, y)$ of (BP) is an efficient solution, if there exists no feasible solution $(x', y')$ satisfying either
\begin{align*}
(i) \quad & CI(x) - IN(y) \leq CI(x') - IN(y') \quad \text{and} \\
& CI(x)/IN(y) < CI(x')/IN(y'), \\
(ii) \quad & CI(x) - IN(y) < CI(x') - IN(y') \quad \text{and} \\
& CI(x)/IN(y) \leq CI(x')/IN(y').
\end{align*}

Since (BP) has 0-1 variables and includes a fractional objective function, it is not trivial to find an efficient solution of (BP). In the next section, we consider how to generate efficient solutions.
3. Generating efficient solutions

The complexity of the BUFLP practically prohibits the identification of the complete set of its efficient solutions. We will thus be satisfied with obtaining an approximation of the efficient solution set. Among such approximating techniques, the constraint method and the weighting method, are most well-known. For details, refer to Cohon [4]. We briefly explain how the two methods can be applied to the BUFLP.

The constraint method optimizes one objective while putting the other objective as a constraint. The following problem then arises.

\[(CP) \quad \text{max} \ CI(x) - IN(y), \]
\[\text{subject to } (1)-(5) \text{ and } CI/IN \geq r \text{ (or } CI - rIN \geq 0),\]

where \( r \) is the rate of return. If \((CP)\) is solved for every positive value of \( r \), a rich set of efficient solutions can be generated. For this purpose, however, we need to solve a parametric integer programming problem which contains parametric coefficient terms at constraints. This parametric problem has a complex structure of having possible discontinuity in its objective function value, making its parametric analysis extremely difficult. Moreover, at each step we must solve a UFLP with an additional constraint.

Another alternative to approximate the efficient frontier is to use the weighting method, which was recommended by Zadeh [17] and has been applied to some multiobjective combinatorial optimization problems [6,7]. The weighting method is to weight the objectives for generating efficient solutions, by which we have the following problem:

\[(WP) \quad \text{max} \ CI(x) - IN(y) + wCI(x)/IN(y), \]
\[\text{subject to } (1)-(5).\]

From the fact that any optimal solution to \((WP)\) with a positive weight \( w \), is an efficient solution of \((BP)\), the weighting method generates efficient solutions by solving \((WP)\) for all positive weights. Thus we need to solve a parametric integer programming problem where the objective function has a fractional term. However, we can hardly expect an algorithm which solves such a parametric problem in reasonable time.

In this section, we show that such approximation can also be achieved by solving the parametric objective UFLP which is easier to handle than the above parametric problems. We develop an algorithm which solves our parametric problem efficiently. Furthermore, we will show that our algorithm generates all the efficient solutions which the weighting method can.

Consider the following parametric objective UFLP:

\[(P^\theta) \quad z(\theta) = \text{max} \ CI(x) - \theta IN(y), \]
\[\text{subject to } (x, y) \in F,\]

where \( F = \{(x, y) \mid (x, y) \text{ satisfies (2), (3), (4), and (5)}\}. \) Let \((x(\theta), y(\theta))\) denote the optimal solution vector for \((P^\theta)\) and let \( \theta_p = \max\{CI(x)/IN(y) \mid (x, y) \in F\}. \) The following two lemmas are important in figuring out the structural properties of \((P^\theta)\).

**Lemma 1.** \( z(\theta) \) is piece-wise linear, convex and nonincreasing for \( \theta \geq 0 \), and \( z(\theta_p) = 0. \)

**Lemma 2.** For \( 1 < \theta_1 < \theta_2 < \theta_p \), the following relation holds:

(i) \( IN(y(\theta_1)) \geq IN(y(\theta_2)), \)
(ii) \( CI(x(\theta_1)) - IN(y(\theta_1)) \geq CI(x(\theta_2)) - IN(y(\theta_2)), \)
(iii) \( CI(x(\theta_1))/IN(y(\theta_1)) \leq CI(x(\theta_2))/IN(y(\theta_2)). \)

**Proof.** From the optimality of \((x(\theta_1), y(\theta_1))\) and \((x(\theta_2), y(\theta_2))\),

\[\theta_1 \{IN(y(\theta_1)) - IN(y(\theta_2))\} \leq CI(x(\theta_1)) - CI(x(\theta_2)) \leq \theta_2 \{IN(y(\theta_1)) - IN(y(\theta_2))\}.\]

From the fact that \( \theta_1 < \theta_2, IN(y(\theta_1)) - IN(y(\theta_2)) \geq 0. \) Since \( \theta_1 \geq 1, \)

\[CI(x(\theta_1)) - CI(x(\theta_2)) \geq \theta_1 \{IN(y(\theta_1)) - IN(y(\theta_2))\} \geq IN(y(\theta_1)) - IN(y(\theta_2)).\]
Finally, from the fact that $CI(x(θ_2)) - θ_2IN(y(θ_2)) ≥ 0$, $CI(x(θ_2))/IN(y(θ_2)) ≥ θ_2$. So,

$$CI(x(θ_1)) - CI(x(θ_2)) ≤ θ_2 \{IN(y(θ_1)) - IN(y(θ_2))\} \leq CI(x(θ_2))/IN(y(θ_2)) \{IN(y(θ_1)) - IN(y(θ_2))\},$$

which shows that (iii) holds. □

Now we present the key result on which our solution procedure is based.

**Theorem 3.** For any $1 < θ < θ_p$, if $(x(θ), y(θ))$ satisfies (1), then it is an efficient solution of (BP).

**Proof.** For some $1 < θ < θ_p$, suppose that $(x(θ), y(θ))$ satisfies (1) but is not an efficient solution of (BP), i.e. there exists at least one $(x', y') ∈ F$ satisfying either

(i) $CI(x(θ)) - IN(y(θ)) ≤ CI(x') - IN(y')$ and $CI(x(θ))/IN(y(θ)) ≤ CI(x')/IN(y')$, or

(ii) $CI(x(θ)) - IN(y(θ)) < CI(x') - IN(y')$ and $CI(x(θ))/IN(y(θ)) ≤ CI(x')/IN(y')$.

Then we can show that $CI(x(θ)) - θIN(y(θ)) < CI(x') - θIN(y')$, which contradicts the optimality of $(x(θ), y(θ))$.

**Case 1:** $IN(y') = IN(y(θ))$. From (i) and (ii), $CI(x(θ)) < CI(x')$. Thus

$$CI(x(θ)) - θIN(y(θ)) < CI(x') - θIN(y').$$

**Case 2:** $IN(y') < IN(y(θ))$. The following relation holds:

$$θ \{IN(y') - IN(y(θ))\} < IN(y') - IN(y(θ)) \leq CI(x') - CI(x(θ)).$$

The first strict inequality holds from $θ > 1$, and the second inequality by (i) and (ii).

**Case 3:** $IN(y') > IN(y(θ))$. From (i) and (ii), $CI(x')/CI(x(θ)) ≥ IN(y')/IN(y(θ))$. Thus

$$CI(x') - CI(x(θ)) ≥ CI(x(θ))/IN(y(θ)) \{IN(y') - IN(y(θ))\}. \quad (6)$$

Suppose that $CI(x(θ)) - θIN(y(θ)) ≥ CI(x') - θIN(y')$. Then,

$$θ \{IN(y') - IN(y(θ))\} ≥ CI(x') - CI(x(θ)). \quad (7)$$

From (6) and (7), $CI(x(θ))/IN(y(θ)) ≤ θ$ and $z(θ) = CI(x(θ))/IN(y(θ)) ≤ 0$. From Lemma 1, $θ ≥ θ_p$, which is a contradiction. □

Therefore, if we find an optimal solution $(x(θ), y(θ))$ of $(P^θ)$ for any $θ > 1$ such that $z(θ) > 0$ and $CI(x(θ))/IN(y(θ)) ≤ π_0$, then $(x(θ), y(θ))$ is an efficient solution of (BP). The following theorem shows the relation between $(P^θ)$ and (WP).

**Theorem 4.** If $(x, y)$ is an optimal solution to (WP) for any positive weight $w$, then $(x, y)$ is an optimal solution of $(P^θ)$ for some $1 ≤ θ ≤ θ_p$.

**Proof.** Suppose that $(x, y)$ is an optimal solution to (WP) for some positive weight $w$ but not an optimal solution of $(P^θ)$ for any $1 ≤ θ ≤ θ_p$. From the property of the weighting method, $(x, y)$ is an efficient solution of (BP). By slightly abusing the notation of $θ_p$, we suppose that $z(θ)$ consists of $p$ line segments over the range of $1 < θ ≤ θ_p$ and define

$$θ_k = \begin{cases} 1, & \text{for } k = 0, \\
θ \text{ value of the } k\text{th break point,} & \text{for } k = 1, \ldots, p
\end{cases}$$

and let $K = \{1, \ldots, p\}$.

Let $(x^k, y^k)$ be an optimal solution of $(P^θ)$ with $θ_{k-1} ≤ θ ≤ θ_k$ for each $k ∈ K$. If $(x^k, y^k)$ satisfies (1), it is an efficient solution of (BP) by Theorem 3. Since $(x, y)$ is not an optimal solution of $(P^θ)$ at $θ = 1$,

$$CI(x) - IN(y) < CI(x^1) - IN(y^1). \quad (8)$$

Moreover,

$$CI(x) - IN(y) ≥ CI(x^p) - IN(y^p). \quad (9)$$

Otherwise, $(x, y)$ is dominated by $(x^p, y^p)$ and cannot not be an efficient solution, because $CI(x)/IN(y) ≤ CI(x^p)/IN(y^p)$. Therefore, from (8) and (9), $p ≥ 2$ and by (ii) of Lemma 2, there exist $(x^k, y^k)$ and $(x^{k+1}, y^{k+1})$ such that
\[ CI(x^{k+1}) - IN(y^{k+1}) \leq CI(x) - IN(y) \]
\[ \leq CI(x^k) - IN(y^k). \]

To prove the theorem, we aim at showing that for every positive \( w \),
\[ CI(x) - IN(y) + w \cdot \frac{CI(x)}{IN(y)} \]
\[ < \max [CI(x^k) - IN(y^k) + w \cdot \frac{CI(x^{k+1})}{IN(y^{k+1})}, \]
\[ CI(x^{k+1}) - IN(y^{k+1}) + w \cdot \frac{CI(x^{k+1})}{IN(y^{k+1})}], \]
which contradicts the fact that \((x, y)\) is an optimal solution to \((WP)\) with some positive \( w \). Consider a nonnegative \( \lambda \) such that \( 0 \leq \lambda \leq 1 \) and
\[ \lambda \{CI(x^k) - IN(y^k)\} + (1 - \lambda) \{CI(x^{k+1}) - IN(y^{k+1})\} = CI(x) - IN(y), \]
then,
\[ IN(y) - \{\lambda IN(y^k) + (1 - \lambda) IN(y^{k+1})\} = CI(x) - \{\lambda CI(x^k) + (1 - \lambda) CI(x^{k+1})\}. \]

Both \((x^k, y^k)\) and \((x^{k+1}, y^{k+1})\) are the optimal solutions of \((P^\theta)\) at \( \theta = \theta^k \) but \((x, y)\) is not. So,
\[ CI(x) - \theta^k IN(y) < \lambda \{CI(x^k) - \theta^k IN(y^k)\} \]
\[ + (1 - \lambda) \{CI(x^{k+1}) - \theta^k IN(y^{k+1})\} \]
and by rearranging the above inequality,
\[ CI(x) - \{\lambda CI(x^k) + (1 - \lambda) CI(x^{k+1})\} \]
\[ < \theta^k [IN(y) - \{\lambda IN(y^k) + (1 - \lambda) IN(y^{k+1})\}] . \]

From (10) and the fact that \( \theta^k > 1 \),
\[ IN(y) - \{\lambda IN(y^k) + (1 - \lambda) IN(y^{k+1})\} > 0. \]

In addition, the following holds:
\[ \lambda CI(x^k)/IN(y^k) \]
\[ + (1 - \lambda) CI(x^{k+1})/IN(y^{k+1}) > 1. \]

Note that \( CI(x^k)/IN(y^k) > 1 \) and \( CI(x^{k+1})/IN(y^{k+1}) > 1 \), because
\[ z(\theta^k) = CI(x^k) - \theta^k IN(y^k) \]
\[ = CI(x^{k+1}) - \theta^k IN(y^{k+1}) \]
\[ \geq 0 \]
and \( \theta^k > 1 \). From (12), (13), and (14),
\[ CI(x) - \{\lambda CI(x^k) + (1 - \lambda) CI(x^{k+1})\} \]
\[ < \left\{ \left[ \frac{CI(x^k)}{IN(y^k)} + (1 - \lambda) \frac{CI(x^{k+1})}{IN(y^{k+1})} \right] \right\} \]
\[ \times [IN(y) - \{\lambda IN(y^k) + (1 - \lambda) IN(y^{k+1})\}]. \]

Therefore,
\[ CI(x) - \left\{ \left[ \frac{CI(x^k)}{IN(y^k)} + (1 - \lambda) \frac{CI(x^{k+1})}{IN(y^{k+1})} \right] \right\} \]
\[ \times [IN(y) - \{\lambda IN(y^k) + (1 - \lambda) IN(y^{k+1})\}] \]
\[ = \lambda (1 - \lambda) (IN(y^k) - IN(y^{k+1})) \]
\[ \times \left( \frac{CI(x^k)}{IN(y^k)} - \frac{CI(x^{k+1})}{IN(y^{k+1})} \right), \]
\[ \leq 0. \]

The last inequality holds by (i) and (iii) of Lemma 2 and \( 0 \leq \lambda \leq 1 \). So,
\[ \frac{CI(x)}{IN(y^k)} < \frac{CI(x^k)}{IN(y^k)} + (1 - \lambda) \frac{CI(x^{k+1})}{IN(y^{k+1})}. \]

The above fact, along with (11), shows that (10) holds for any positive \( w \).

It is not difficult to see that the inverse direction of the theorem does not hold, which means that our heuristic approximates the efficient frontier better than the weighting method.

Now we present an algorithm which generates efficient solutions based on \((P^\theta)\). \((P^\theta)\) is a parametric objective integer programming problem for which there is a usual solution scheme, first suggested by Geoffrion and Nauss [11] and formalized by Jenkins [12]. The scheme, exploiting the convexity of the objective value, will be well tailored to our parametric objective UFLP. In addition, the following observations are very useful for saving the computational burden.
Table 1

Data for example

\[
(p - t_{ij})D_j \\
\begin{array}{cccc}
  i \backslash j & 1 & 2 & 3 \\
  1 & 30 & 0 & 0 & 20 \\
  2 & 0 & 30 & 0 & 25 \\
  3 & 0 & 0 & 30 & 27 \\
\end{array}
\]

Lemma 5. For some \( \theta' \geq 1 \), if an optimal solution of \( (P^\theta) \) does not satisfy (1), then neither does any optimal solution of \( (P^\theta) \) with \( \theta \geq \theta' \).

Proof. It is straightforward by (ii) of Lemma 2. \( \Box \)

By Lemma 1 and 5, once we find \( \theta' \) for which either \( z(\theta') \leq 0 \) or \( CI(x(\theta')) - IN(y(\theta')) < \pi_o \), we cannot find an efficient solution of \( (BP) \) by solving \( (P^\theta) \) for any \( \theta \geq \theta' \). Let \( \theta_f = \max\{\theta \mid CI(x(\theta)) - IN(y(\theta)) \geq \pi_o \} \) and \( \theta_{\text{max}} = \min\{\theta_p, \theta_f\} \). If \( (BP) \) is feasible, \( \theta_{\text{max}} \geq 1 \). From the above observations, our solution procedure then boils down to simply solving \( (P^\theta) \) for \( 1 \leq \theta \leq \theta_{\text{max}} \).

Now we elaborate on how to solve \( (P^\theta) \). Let \( UB(\theta) \) and \( LB(\theta) \) be the upper and lower bound functions for \( z(\theta) \). If the values of these two bound functions coincide, we then have \( z(\theta) \). The process is illustrated in Fig. 1. First \( (P^1) \) and \( (P^{\theta_{\text{max}}}) \) are solved, yielding the respective optimal solutions, \( (x(1), y(1)) \) and \( (x(\theta_{\text{max}}), y(\theta_{\text{max}})) \). Then \( ABC \) produced by \( CI(x(1)) - \theta IN(y(1)) \) and \( CI(x(\theta_{\text{max}})) - \theta IN(y(\theta_{\text{max}})) \) forms \( LB(\theta) \), and the straight line \( AC \) forms \( UB(\theta) \), from the convexity of \( z(\theta) \). Next \( (P^{\theta_1}) \) is solved where \( \theta_1 \) is determined such that \( CI(x(1)) - \theta_1 IN(y(1)) = CI(x(\theta_{\text{max}})) - \theta_1 IN(y(\theta_{\text{max}})) \). If \( z(\theta_1) \) coincides with either \( LB(\theta_1) \) or \( UB(\theta_1) \), then \( (P^{\theta_1}) \) is solved. Otherwise, \( LB(\theta) \) and \( UB(\theta) \) are updated and the procedure continues. In summary, this method solves \( (P^\theta) \) to optimality at each \( \theta \) for which \( LB(\theta) \) has a break point and does not coincide with \( UB(\theta) \). If \( (BP) \) is infeasible, \( z(1) \leq \pi_o \). Therefore, even when the original problem is infeasible, our solution process could find the infeasibility at its very first iterative step.

When incorporating the above framework into our procedure, one difficulty is that we cannot identify \( \theta_{\text{max}} \) until we solve \( (P^\theta) \). This problem is circumvented by finding an approximation of \( \theta_{\text{max}} \). For the purpose, we develop the following procedure by slightly modifying the method which is used to obtain \( \theta_p \) in the fractional objective programming.

Algorithm theta-max;
begin
\( \theta_a := 1; \)
solve \( (P^{\theta_a}) \);
while \( z(\theta_a) > 0 \) and \( CI(x(\theta_a)) - IN(y(\theta_a)) \geq \pi_o \) do
begin
calculate \( \theta \) such that \( CI(x(\theta)) - \theta IN(y(\theta)) = \pi_o \);
set \( \theta_a := \theta; \)
solve \( (P^{\theta_a}) \);
end;
end;

Note that if the obtained \( \theta_a \) satisfies \( CI(x(\theta_a)) - IN(y(\theta_a)) \geq \pi_o, \theta_{\text{max}} = \theta_a \). Otherwise, \( \theta_{\text{max}} \leq \theta_a \). In the latter case, \( \theta_a \) can still be updated during the procedure for solving \( (P^\theta) \). While performing the procedure \( \text{theta-max} \), \( z(\theta) \)'s for at least two values of \( \theta \), \( 1 \) and \( \theta_a \), are obtained, from which \( LB(\theta) \) and \( UB(\theta) \) are initialized. Let \( \Theta \) be a set of \( \theta \)'s with \( 1 \leq \theta \leq \theta_a \), for each of which \( LB(\theta) \) has a break point and does not coincide with \( UB(\theta) \). Now we formalize the whole solution process for \( (P^\theta) \) as follows:

Algorithm para;
begin
\( \text{theta-max}; \)
\( \text{construct } LB(\theta), UB(\theta) \) and the corresponding \( \Theta; \)
Table 2
Computational results for Kuhn-Hamburger problems

<table>
<thead>
<tr>
<th>Problem set</th>
<th>Fixed cost</th>
<th>No. of efficient solutions</th>
<th>CPU (seconds)¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>100-300</td>
<td>13</td>
<td>0.336</td>
</tr>
<tr>
<td></td>
<td>300-500</td>
<td>9</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>500-700</td>
<td>7</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>700-900</td>
<td>8</td>
<td>0.281</td>
</tr>
<tr>
<td></td>
<td>1000-2000</td>
<td>8</td>
<td>0.171</td>
</tr>
<tr>
<td>II</td>
<td>100-300</td>
<td>10</td>
<td>0.219</td>
</tr>
<tr>
<td></td>
<td>300-500</td>
<td>9</td>
<td>0.328</td>
</tr>
<tr>
<td></td>
<td>500-700</td>
<td>7</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>700-900</td>
<td>5</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>1000-2000</td>
<td>4</td>
<td>0.109</td>
</tr>
<tr>
<td>III</td>
<td>100-300</td>
<td>11</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>300-500</td>
<td>10</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>500-700</td>
<td>8</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>700-900</td>
<td>7</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>1000-2000</td>
<td>5</td>
<td>0.109</td>
</tr>
<tr>
<td>IV</td>
<td>100-300</td>
<td>10</td>
<td>0.281</td>
</tr>
<tr>
<td></td>
<td>300-500</td>
<td>9</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>500-700</td>
<td>7</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>700-900</td>
<td>6</td>
<td>0.172</td>
</tr>
<tr>
<td></td>
<td>1000-2000</td>
<td>4</td>
<td>0.172</td>
</tr>
</tbody>
</table>

¹Including output but excluding input time.

Table 3
Computational results

<table>
<thead>
<tr>
<th>Problem set</th>
<th>Problem size</th>
<th>Fixed cost</th>
<th>No. of efficient solutions</th>
<th>CPU (seconds)²</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>33×33</td>
<td>100-300</td>
<td>28</td>
<td>0.770</td>
</tr>
<tr>
<td></td>
<td></td>
<td>300-500</td>
<td>18</td>
<td>0.488</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000-2000</td>
<td>6</td>
<td>0.547</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2000-3000</td>
<td>7</td>
<td>0.391</td>
</tr>
<tr>
<td>II</td>
<td>42×42</td>
<td>50-100</td>
<td>10</td>
<td>0.711</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100-150</td>
<td>9</td>
<td>1.156</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200-250</td>
<td>4</td>
<td>0.438</td>
</tr>
<tr>
<td></td>
<td></td>
<td>500-600</td>
<td>3</td>
<td>0.391</td>
</tr>
<tr>
<td>III</td>
<td>57×57</td>
<td>100-200</td>
<td>37</td>
<td>2.094</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200-300</td>
<td>28</td>
<td>3.078</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000-2000</td>
<td>9</td>
<td>1.102</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2000-3000</td>
<td>6</td>
<td>0.938</td>
</tr>
<tr>
<td>IV</td>
<td>100×100</td>
<td>1000-2000</td>
<td>21</td>
<td>18.070</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2000-3000</td>
<td>15</td>
<td>14.453</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000-4000</td>
<td>10</td>
<td>14.219</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4000-5000</td>
<td>7</td>
<td>14.289</td>
</tr>
</tbody>
</table>

²Including output but excluding input time.
Table 4

Efficient solutions for the (33 × 33) problem

<table>
<thead>
<tr>
<th>Efficient solutions</th>
<th>θ</th>
<th>CI - IN</th>
<th>CI/IN</th>
<th>No. of open facilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00 ≤ θ ≤ 1.12</td>
<td>28133</td>
<td>3.44</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1.12 ≤ θ ≤ 1.44</td>
<td>28059</td>
<td>4.28</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1.44 ≤ θ ≤ 2.04</td>
<td>27887</td>
<td>4.53</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2.04 ≤ θ ≤ 3.19</td>
<td>26740</td>
<td>5.90</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3.19 ≤ θ ≤ 3.90</td>
<td>25937</td>
<td>6.27</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>3.90 ≤ θ ≤ 7.34</td>
<td>25688</td>
<td>6.34</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>7.34 ≤ θ ≤ 9.45</td>
<td>17079</td>
<td>8.45</td>
<td>1</td>
</tr>
</tbody>
</table>

while Θ ≠ ∅ do
begin
select θ' ∈ Θ;
solve (Pθ');
if CI(x(θ')) - IN(x(θ')) < πo
then Θa := θ';
update LB(θ), UB(θ) and Θ;
end;
end;

4. An illustrative example

In this section, we provide an example to demonstrate the solution process developed in Section 3. The data for an example problem is given in Table 1. The required profit level, πo, is 1. Let I+ denote the set of open facilities.

(i) First we calculate θa and the initial LB(θ) and UB(θ) through the algorithm theta-max. We solve (Pθ) at θ = 1. The resulting optimal solution has I+ = {1,2,3}, z(1) = 18, and CI(x(1)) - IN(y(1)) > πo. We then calculate θa satisfying CI(x(1)) - θaIN(y(1)) = 0. So, θa = 1.25. Next, the optimal solution of (Pθ) at θ = 1.25 gives I+ = {1}, z(1.25) = 5, and CI(x(1.25)) - IN(y(1.25)) > πo. The resulting θa is equal to 1.5. Then we solve (Pθ) at θ = 1.5. Since z(1.5) = 0, theta-max terminates with θa = 1.5.

(ii) Now we solve (Pθ) for 1 ≤ θ ≤ θmax completely. θ = {15/13}. We obtain an optimal solution of (Pθ) at θ = 15/13 which provides I+ = {1,2} and θ = {10/9, 6/5}. Both z(10/9) and z(6/5) coincide with LB(θ). Since θmax = 1.5, we obtain three efficient solutions of (BP). In this example, these three solutions are the only efficient ones.

5. Computational results

Our solution procedure was coded in FORTRAN IV on a Pentium (90MHz) personal computer. Computational experiments were conducted on several problems with dimensions (m × n) of (25 × 50), (33 × 33), (42 × 42), (57 × 57) and (100 × 100) whose tij values are taken from the UFLP problems considered by Erlenkotter [8]. The first four sets of (25 × 50) problems contain the well known Kuehn-Hamburger problems [16]. The other four sets of problems have tij values taken from the data for the traveling salesman problems supplied by Karg and Thompson [13].

Tables 2 and 3 give the results for these problems with randomly generated fixed costs. The range of the fixed costs in each problem is also shown. We set πo equal to 1. When successively solving the UFLP, we use the dual-based algorithm by Erlenkotter [8]. Even though the CPU time is highly data-dependant, our algorithm solved the large scale problems within reasonable time even on a personal computer. Table 4 gives the details of a computing experiment on a (33 × 33) problem where fixed costs range between 2000 and 3000. For this problem, seven efficient solutions are obtained, in which profits lie from 28133 to 17079, and the rates of return extend from 344% to 845%. As θ increases, the number of open facilities for the obtained solution becomes smaller, as dictated by (i) of Lemma 2. However, note that the number of open facilities does not necessarily decrease as θ increases.

6. Conclusion

We have presented a bi-objective uncapacitated facility location problem having two objectives, one
maximizing the net profit and the other maximizing the profitability of an investment. This problem has too complex a structure to simply rely on the existing methods such as the constraint method and the weighting method for obtaining its efficient solutions. However, using the parametric objective UFLP, we have developed the computationally efficient solution method. Moreover, our algorithm generates all the efficient solutions that the weighting method can. As the model UFLP has been extended to a variety of general cases such as capacitated, dynamic, concave cost, multicommodity, etc. [14], so can our solution approach be applied to the bi-objective versions of those general models.

References