Submitted to *INFORMS Journal on Computing*
manuscript JOC-2015-01-OA-008.R2

Online Supplement to "Branch-and-bound for bi-objective integer programming"

Sophie N. Parragh and Fabien Tricoire
Institute of Production and Logistics Management, Johannes Kepler University Linz,
Altenberger Straße 69, 4040 Linz, Austria
{sophie.parragh,fabien.tricoire}@jku.at, http://www.jku.at/plm

In this online supplement, we present algorithms and model formulations that are used in order to produce experimental results in the article.

Key words: branch-and-bound, integer programming, bi-objective optimization, branch-and-price, orienteering
History:

---

1. Algorithms for criterion space search methods

Algorithms 4 and 5 outline the ϵ-constraint method and the balanced box method, respectively. The set P denotes the set of Pareto optimal solutions. Function *solveMIP* can either be a call to Gurobi or CPLEX or to a tailor-made branch-and-bound algorithm. The value of ϵ is supposed to be valid for the problem at hand; for instance with pure integer problems with integer coefficients for the objective function, $\epsilon = 1$ is such a valid value.

Algorithm 4 ϵ-constraint algorithm

```
P ← ∅
ε-constraint ← f₂ ≤ ∞
while MIP is feasible do
    x ← solveMIP(lexmin(f₁, f₂), ε-constraint)
    P ← P ∪ x
    ε-constraint ← f₂ ≤ f₂(x) − ε
end while
return P
```

2. UBOFLP: Model formulation

In the UBOFLP, we are given a set V of potential facilities and a set N of locations. Furthermore, we denote by N_i the set of locations that can be covered by facility i because they are within a certain radius. Each facility has opening costs F_i and each location has a weight or demand W_j. The considered objectives simultaneously minimize the opening costs of facilities and maximize the total covered demand.

Using binary decision variables $y_i \in \{0,1\}$ equal to 1 if facility i is opened and 0 otherwise, and $x_{ij} \in \{0,1\}$ equal to 1 if location j is assigned to facility i, we formally define the UBOFLP as follows:

$$\min \sum_{i \in V} F_i y_i \tag{1}$$

$$\max \sum_{i \in V} \sum_{j \in N_i} W_j x_{ij} \tag{2}$$

subject to:

$$x_{ij} \leq y_i \qquad \forall i \in V, j \in N, \tag{3}$$

$$\sum_{i \in V} x_{ij} = 1 \qquad \forall j \in N, \tag{4}$$

$$y_i \in \{0,1\} \qquad \forall i \in V, \tag{5}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V, j \in N. \tag{6}$$

3. SSUFLP: Model formulation

In the SSUFLP, we are given a set V of potential facilities and a set N of locations that have to be assigned to one of the facilities each. Each facility has opening costs F_i and it costs

Algorithm 5 Balanced box method

$P \leftarrow \emptyset$, $Rectangles \leftarrow \emptyset$
$x^T \leftarrow solveMIP(lexmin(f_1, f_2))$
$x^B \leftarrow solveMIP(lexmin(f_2, f_1))$
$P \leftarrow P \cup \{x^T, x^B\}$
$Rectangles \leftarrow Rectangles \cup \{rectangle\, z^T = (f_1(x^T), f_2(x^T)), z^B = (f_1(x^B), f_2(x^B))\}$
while $Rectangles \neq \emptyset$ **do**
  $Rectangles.pop(rectangle(z^1, z^2))$
  $rectangle^B \leftarrow rectangle((z_1^1, (z_2^1 + z_2^2)/2), z^2)$
  $\bar{x}^1 \leftarrow solveMIP(lexmin(f_1, f_2), rectangle^B constraints)$
  $\bar{z}^1 = (f_1(\bar{x}^1), f_2(\bar{x}^1))$
  if $\bar{z}^1 \neq z^2$ **then**
    $P \leftarrow P \cup \{\bar{x}^1\}$
    $Rectangles \leftarrow Rectangles \cup rectangle(\bar{z}^1, z^2)$
  end if
  $rectangle^T \leftarrow rectangle(z^1, (\bar{z}_1^1 - \epsilon, (z_2^1 + z_2^2)/2))$
  $\bar{x}^2 \leftarrow solveMIP(lexmin(f_2, f_1), rectangle^T constraints)$
  $\bar{z}^2 = (f_1(\bar{x}^2), f_2(\bar{x}^2))$
  if $\bar{z}^2 \neq z^1$ **then**
    $P \leftarrow P \cup \{\bar{x}^2\}$
    $Rectangles \leftarrow Rectangles \cup rectangle(z^1, \bar{z}^2)$
  end if
end while
return P

c_{ij} to assign location j to facility i. The considered objectives simultaneously minimize the total opening costs of facilities and the total assignment costs.

Using binary decision variables $y_i \in \{0, 1\}$ equal to 1 if facility i is opened and 0 otherwise, and $x_{ij} \in \{0, 1\}$ equal to 1 if location j is assigned to facility i, we formally define the SSUFLP as follows:

$$\min \sum_{i \in V} F_i y_i \tag{7}$$

$$\min \sum_{i \in V} \sum_{j \in N} c_{ij} x_{ij} \tag{8}$$

subject to:

$$x_{ij} \le y_i \qquad \forall i \in V, j \in N, \tag{9}$$

$$\sum_{i \in V} x_{ij} = 1 \qquad \forall j \in N, \tag{10}$$

$$y_i \in \{0,1\} \qquad \forall i \in V, \tag{11}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in V, j \in N. \tag{12}$$

4. BITOPTW: Model formulation

The BITOPTW is defined on a directed graph $G = (V, A)$, where A is the set of arcs and V the set of vertices, representing the starting location (vertex 0), the ending location (vertex $n+1$) and n control points. Each control point i is associated with a score S_i, a service time d_i and a time window $[e_i, l_i]$. Each route $k \in K$, with $|K| = m$, has to start at location 0 and end at location $n+1$ and each arc (i, j) is associated with travel cost c_{ij} and travel time t_{ij}. The aim is to maximize the total collected score and to simultaneously minimize the total travel cost. Using binary decision variables $z_i \in \{0,1\}$ equal to 1 if location i is visited and 0 otherwise, and $y_{ijk} \in \{0,1\}$ equal to 1 if arc (i, j) is traversed by route k and 0 otherwise, and continuous variables B_{ik}, denoting the beginning of service at i by route k, we formally define the BITOPTW as follows:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} y_{ijk} \tag{13}$$

$$\max \sum_{i \in V \setminus \{0, n+1\}} S_i z_i \tag{14}$$

subject to:

$$\sum_{j \in V \setminus \{0\}} y_{0jk} = 1 \qquad \forall k \in K \tag{15}$$

$$\sum_{i \in V \setminus \{n+1\}} y_{i,n+1,k} = 1 \qquad \forall k \in K \tag{16}$$

$$\sum_{k \in K} \sum_{j \in V \setminus \{n+1\}} y_{jik} = z_i \qquad \forall i \in V \setminus \{0, n+1\} \tag{17}$$

$$\sum_{j \in V \setminus \{n+1\}} y_{jik} - \sum_{j \in V \setminus \{0\}} y_{ijk} = 0 \qquad \forall k \in K, i \in V \setminus \{0, n+1\} \tag{18}$$

$$(B_{ik} + d_i + t_{ij})y_{ijk} \leq B_{jk} \qquad \forall k \in K, (i,j) \in A \tag{19}$$

$$e_i \leq B_{ik} \leq l_i \qquad \forall k \in K, i \in V \tag{20}$$

$$y_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A \tag{21}$$

$$z_i \in \{0,1\} \qquad \forall i \in V \setminus \{0, n+1\} \tag{22}$$

Objective function (13) minimizes the total routing costs while objective function (14) maximizes the total collected profit. Constraints (15) and (16) make sure that each route starts at the defined starting point and ends at the correct ending point. Constraints (17) link the binary decision variables and (18) ensure connectivity for visited nodes. Constraints (19) set the time variables and (20) make sure that time windows are respected. We note that constraints (19) are not linear but they can easily be linearized using big M terms.