DEPARTMENT OF OPERATIONS RESEARCH
UNIVERSITY OF AARHUS

Working Paper no. 2001/4

Bicriteria Network Location (BNL) problems with criteria dependent lengths and minisum objectives

Anders J.V. Skriver
Kim Allan Andersen
Kaj Holmberg

ISSN 1600-8987

Department of Mathematical Sciences
Telephone: +45 8942 3188
E-mail: institut@imf.au.dk

Building 530, Ny Munkegade
DK-8000 Aarhus C, Denmark
URL: www.imf.au.dk

Bicriteria Network Location (BNL) problems with criteria dependent lengths and minisum objectives

Anders J.V. Skriver and Kim Allan Andersen*
Department of Operations Research
University of Aarhus
Ny Munkegade, Building 530
DK - 8000 Århus C
Denmark

Kaj Holmberg
Department of Mathematics, Division of Optimization
Linköping Institute of Technology
S-581 83 Linköping
Sweden

April 26, 2001

Abstract

We present a new model, which is a generalization of the bicriterion median problem. We introduce two sum objectives and criteria dependent edge lengths. For this $\mathcal{NP}$ complete problem a solution method finding all the efficient solutions is presented. The method is a two-phases approach, which can easily be applied as an interactive method.

In Phase 1 the supported solutions are found, and in Phase 2 the unsupported solutions are found. This method can be thought of as a general approach to BOCO (Bi-objective Combinatorial Optimization) problems.

Keywords: MCDM, biobjective optimization, facility location, networks, MOCO.

1 Introduction

We begin by a motivating example. Assume we have to locate a money reserve, considering the two objectives of minimizing the transportation costs and the risk of having the transports robbed. The depot serves a number of clients varying in size, and we are given a connected network and interpret each of the n nodes as the clients. A relevant (node) weight for a client with respect to transportation costs is the number of monthly deliveries, and a weight for the risk objective is the maximum value of a money-transport. The

*Corresponding author. Email: kima@imf.au.dk

edge-lengths with respect to transportation costs could be the distance, and for the risk objective the edge-length could be the probability of an assault. If we assume that the cost of opening the new facility is independent of location, this particular cost is unimportant. A solution to this problem consists of two decisions. The first (and probably the most important) one is to decide where to locate the new facility (depot), and the second one consists in determining how to route the flow from the new facility to the nodes. The flow problem consists of $n-1$ Bicriterion Shortest Path (BSP) problems, which is a $\mathcal{NP}$ complete problem.
If each edge has only one length, we have the usual median problem. Now that we have one length for each criterion, the BSP problem becomes a subproblem. Therefore, this refinement has severe consequences on the complexity of the problem.

Before presenting the ideas behind the proposed solution method, some concepts from bicriterion analysis are reviewed. For a textbook introduction see Steuer [7] or Ehrgott [4]. Suppose we want to simultaneously minimize two functions $f^1(x)$ and $f^2(x)$ over some feasible set S. In our case S is a finite set of solutions.

$$\begin{array}{ll} \min & f^1(x) \\ \min & f^2(x) \\ \text{s.t.} & \\ & x \in S \end{array} \tag{1}$$

It is generally accepted, that solving (1) means finding the set of **efficient** (or Pareto optimal) solutions. A solution $x \in S$ is called efficient if one of the objective function values cannot be improved without worsening the other. Let $f(x) = (f^1(x), f^2(x))^t$, where t denotes transpose. The mathematical definition of efficiency is as follows.

Definition 1 *A point $x \in S$ is **efficient** iff there does not exist a point $\bar{x} \in S$ such that $f(\bar{x}) \leq f(x)$ with at least one strict inequality. Otherwise x is **inefficient**.*

Efficient points are defined in decision space. There is a natural counterpart in criterion space $Z = \{z \in \mathbb{R}^2 | \exists x \in S, z = f(x)\}$.

Definition 2 *$z(x) \in Z$ is a **nondominated** criterion vector iff x is an efficient solution. Otherwise $z(x)$ is a **dominated** criterion vector.*

In Definition 2 we have used that $z(x) = f(x)$. The set of efficient (E) solutions is denoted S^E, and the set of nondominated (ND) criterion vectors is denoted Z^{ND}, and is given by $Z^{ND} = z(S^E)$.

The criterion vectors can be partitioned into two kinds, namely supported and unsupported. Define the weighted objective function $W(x, \lambda)$ as:

$$W(x, \lambda) = \lambda f^1(x) + (1 - \lambda) f^2(x), \ \lambda \in (0; 1). \tag{2}$$

The function $W(x, \lambda)$ is a convex combination, or **weighted sum**, of the two objective functions. Optimizing this function over the feasible set S parametrically in $\lambda \in (0, 1)$ will give all the **supported** nondominated solutions to (1). The method is therefore often referred to as the **weighting method.**

It is important to note that each unsupported nondominated criterion vector is dominated by a convex combination of some set of nondominated criterion vectors. Supported nondominated (SND) criterion vectors are denoted Z^{SND} and the corresponding set of solutions are denoted S^{SE}.

The solution method proposed is a variant of the two-phases approach due to Ulungu and Teghem [9] and Visée *et al.* [10]. In Phase 1 all (or a representative subset of) the supported extreme solutions are found by using the weighting method. In Phase 2 a search between the supported solutions is conducted to find unsupported efficient solutions. The procedure is explained in details in Section 3.

The remaining parts of the paper is organized as follows. In Section 2 the bicriterion problem is presented, and some properties of the problem is given. In Section 3 the solution procedure is outlined, and an example is presented. In Section 4 the generalization to more than two criteria is discussed, and finally Section 5 contains the conclusions.

2 Problem formulation

We are given a connected directed network $G(\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ where $|\mathcal{V}| = n$ nodes, and edge set $\mathcal{E} = \{(v_i, v_j), (v_k, v_l), \ldots, (v_p, v_q)\}$ with $|\mathcal{E}| = m$ edges. The underlying graph is denoted by G, and edges may be referred to by $e \in \mathcal{E}$, by $(v_i, v_j) \in \mathcal{E}$ or simply by $(i, j) \in \mathcal{E}$, where node i is the tail and node j is the head. Each node v_i carries two weights $(w_i^1, w_i^2)^t$, where $w_i^q \in \mathbb{R}_+, \ q = 1, 2$, so we may refer to the matrix of weights by $W_{2\times n}$. Each edge $e \in \mathcal{E}$ has length $l(e) = (l^1(e), l^2(e)) \in \mathbb{R}_+^2$. Let us define a matrix of edges $E_{m\times(4)}$ with the following entries. E_{i1} is the tail of edge e_i, E_{i2} is the head, $E_{i3} = l^1(e_i)$ is the length with respect to criteria one and $E_{i4} = l^2(e_i)$ is the length with respect to criteria two.

Notice that an undirected network can be modeled as a directed network with the double

amount of edges. Define binary decision variables as follows:

$$x_i = \begin{cases} 1 & \text{if the facility is located in node } i \\ 0 & \text{else} \end{cases}$$

$$y_{ijk} = \begin{cases} 1 & \text{if edge } (i,j) \text{ is used in the path to node } k \\ 0 & \text{else} \end{cases}$$

We examine the so-called median objectives or weighted sum objectives:

$$f^q(y) = \sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij}^q w_k^q y_{ijk} \quad q = 1,2$$

Combining the coefficients to $c_{ijk}^q = l_{ij}^q w_k^q$, we get

$$f^q(y) = \sum_{k=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ijk}^q y_{ijk} \quad q = 1,2 \tag{3}$$

There are two types of constraints. The first constraint ensures that exactly one facility is located and the second set of constraints ensures the existence of paths from the facility to the remaining nodes. This leads to the following problem:

$$\begin{aligned} &\min \quad f^1(y) \\ &\min \quad f^2(y) \\ &\text{s.t.} \\ &\qquad \sum_{i=1}^{n} x_i = 1 \\ &\sum_{j=1}^{n} y_{jik} - \sum_{j=1}^{n} y_{ijk} = -x_i \qquad i \neq k, \quad \forall\, i,k \\ &\qquad x_i \in \{0,1\} \qquad \forall i \\ &\qquad y_{ijk} \in \{0,1\} \qquad \forall\, i,j,k \end{aligned} \tag{4}$$

Notice that we have omitted the following redundant constraints

$$\sum_{j=1}^{n} y_{jik} - \sum_{j=1}^{n} y_{ijk} = 1 - x_i \quad \forall\, i, \text{ where } i = k.$$

The reason being that this part of the constraint matrix consists of n totally unimodular sub-matrices forming the n sets of paths, see (5). Notice that one path is non-existing, since the node in which the new facility is located, ships nothing through the network.

To understand the structure of the constraint matrix of (4), we write it out. We define the vector $\mathbf{y}_{ijk}$ (in bold) as the vector of all combinations of i and j, but with a fixed k. This way $\mathbf{y}_{ij1}$ contains all edge variables for node 1 and so forth. The matrix M_k is

the totally unimodular sub-matrix forming paths from node x_i to node k. These matrices have dimension $(n-1) \times n^2$. I_{-k} is an $(n-1) \times n$ identity matrix with the k'th row deleted.

$$\begin{bmatrix} 1 \cdots 1 & 0 & \cdots & 0 & \cdots & 0 \\ I_{-1} & M_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ I_{-k} & 0 & \cdots & M_k & \cdots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ I_{-n} & 0 & \cdots & 0 & \cdots & M_n \end{bmatrix} \begin{bmatrix} x \\ \mathbf{y}_{ij1} \\ \vdots \\ \mathbf{y}_{ijk} \\ \vdots \\ \mathbf{y}_{ijn} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5}$$

It turns out that this matrix is not totally unimodular.

Theorem 1 *The constraint matrix in (5) is not totally unimodular.*

An example of a sub-matrix of (5) with determinant two is given in the appendix. Since the constraint matrix is not totally unimodular, solving the LP relaxation of (4) is not guaranteed to return integer solutions, as is often the case in network problems.
Weighting the two objective functions in (4), using the weights λ and $1-\lambda$, $\lambda \in (0;1)$, results in the weighted version of (4)

$$\begin{aligned} \min \quad & \lambda f^1(y) + (1-\lambda) f^2(y) \\ \text{s.t.} \quad & \\ & \sum_{i=1}^{n} x_i = 1 \\ & \sum_{j=1}^{n} y_{jik} - \sum_{j=1}^{n} y_{ijk} = -x_i \qquad i \neq k \quad \forall\, i,k \\ & x_i \in \{0,1\} \qquad \forall i \\ & y_{ijk} \in \{0,1\} \qquad \forall\, i,j,k \end{aligned} \tag{6}$$

In Section 3.4 we describe how problem (6) can be solved in $O(n^4)$ running time using Benders' decomposition for a fixed λ.

3 Solution procedure

In this section the solution procedure for solving the bicriterion problem (4) is outlined. Before stating the procedure it may be helpful to consider a naive method. One possible way of solving the problem could be to solve problem (6) n times, namely one time for each possible location of the new facility. Suppose that the location of the new facility is fixed at a specific node, say node i (so $x_i = 1$). Using the weighting method, the supported efficient solutions (paths) with respect to node i can be revealed. We call these efficient

solutions *locally* efficient with respect to node i. Given $\lambda \in (0,1)$ and x the corresponding locally efficient solution can be found in $O(n^3)$ running time, since we are faced with $n-1$ shortest path problems.

Finding the locally unsupported efficient solutions that are in fact globally efficient, constitutes a more difficult problem. These cannot be found using the weighting method. This fact is known from studying the BSP problem alone [5].

We thus have three types of efficient solutions:

- supported efficient solutions
- locally supported efficient solutions
- (locally) unsupported efficient solutions

The reason why locally supported efficient solutions are interesting, is that they may be unsupported efficient solutions in the main problem (4). These three kinds of solutions are illustrated in Example 3.1.

3.1 Example

We examine the network presented in Figure 1 with the following weights and edges. Each column of W consists of the two node-weights.

$$W = \begin{bmatrix} 200 & 300 & 500 & 100 & 400 & 500 & 400 \\ 7 & 4 & 2 & 6 & 6 & 2 & 8 \end{bmatrix}$$

The first two columns of E are the tail and head nodes. The next two columns are the two edge-lengths.

$$E = \begin{bmatrix} 1 & 2 & 78 & 22 \\ 1 & 3 & 24 & 72 \\ 1 & 4 & 26 & 71 \\ 1 & 5 & 13 & 71 \\ 1 & 7 & 86 & 12 \\ 2 & 3 & 98 & 29 \\ 2 & 5 & 17 & 90 \\ 3 & 5 & 29 & 97 \\ 3 & 6 & 87 & 28 \\ 3 & 7 & 7 & 69 \\ 4 & 5 & 4 & 77 \\ 4 & 7 & 89 & 5 \\ 5 & 6 & 17 & 92 \\ 5 & 7 & 40 & 74 \\ 6 & 7 & 69 & 12 \end{bmatrix}$$

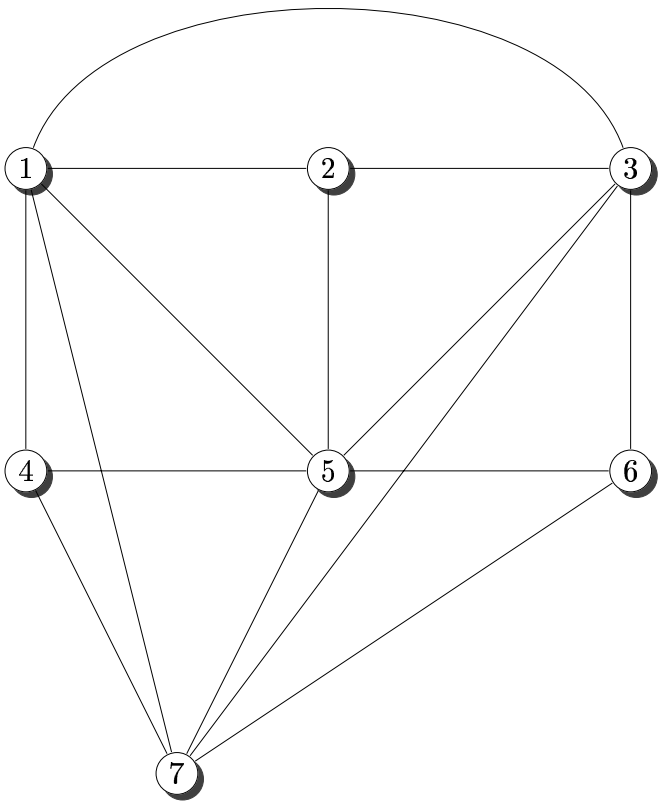

Figure 1: Network for Example 3.1.

The resulting 11 nondominated criterion vectors are presented in Table 1. These criterion vectors are visualized in Figure 2 and it is seen that there are 6 supported and 5 unsupported criterion vectors. Of the 5 unsupported solutions, only one, $(89200, 1868)$, is locally unsupported. The other four unsupported solutions are locally supported by the nodes indicated in Figure 2. The last nondominated solution, $(89200, 1868)$, is dominated by a convex combination of the following two locally supported solutions:

$$\frac{9}{11}(91200, 1684) + \frac{2}{11}(80200, 2587) = (89200, 1848.18)$$

There are a total of 2128 feasible criterion vectors, using only efficient paths between nodes. All these vectors are illustrated in Figure 3.

3.2 Two-phases approach

The procedure that we propose instead of the naive method, is a variant of the two-phases approach due to Ulungu and Teghem [9] and Visée *et al.* [10], and may be stated generically as:

- **Phase 1**: Find all (or a representative subset of) the supported solutions.

Node	f^1	f^2
5	45500	3025
5	47100	2289
1	78200	2062
7	89200	1868
7	91200	1684
1	92600	1506
7	97200	1376
1	107500	1182
7	111600	1112
7	129300	856
7	203800	798

Table 1: Nondominated values for Example 3.1.

- **Phase 2:** Conduct a search between the supported solutions in order to find unsupported nondominated solutions.

3.3 Phase 1

As explained in Section 2 all supported solutions to (4) may be obtained by solving the weighted program (6) parametrically in $\lambda \in (0,1)$. We will do that by employing NISE (Non-Inferior Set Estimation), a method presented in Cohon [3]. NISE guides the choice of $\lambda \in (0,1)$.

First, the weighted program (6) is solved using $\lambda = 1$ and $\lambda = 0$. This results in the minimum values f^{1*} and f^{2*} of the two objectives f^1 and f^2 respectively. Say there are alternative optima for the problem with $\lambda = 1$, then we choose a solution with the lowest objective function value of the second objective f^2. This automatically gives upper bounds, $\bar{f}^2$ and $\bar{f}^1$, on the other objective. The initial nondominated criterion vectors (in Z^{SND}) are $E_1 = (f^{1*}, \bar{f}^2)$ and $E_2 = (\bar{f}^1, f^{2*})$.

Next we find the outward normal, $\bar{n} = (\bar{n}_1, \bar{n}_2)$, to the line between the two initial points, E_1 and E_2. Using $\lambda = \frac{\bar{n}_1}{\bar{n}_1 + \bar{n}_2}$ in solving (6), may result in two cases. We either get a new unique solution E_3, or we get E_1 or E_2 again. In the first case, the point E_3 is in Z^{SND}, and we continue by examining the two line-segments $E_1 - E_3$ and $E_2 - E_3$. In the latter case we know that there does not exist a supported (extreme) criterion vector between E_1 and E_2. The procedure proceeds until no new supported criterion vectors are found, or until a desired number of solutions are found. The outward normal to the line-segment between two points can easily be found as differences between the objective function values.

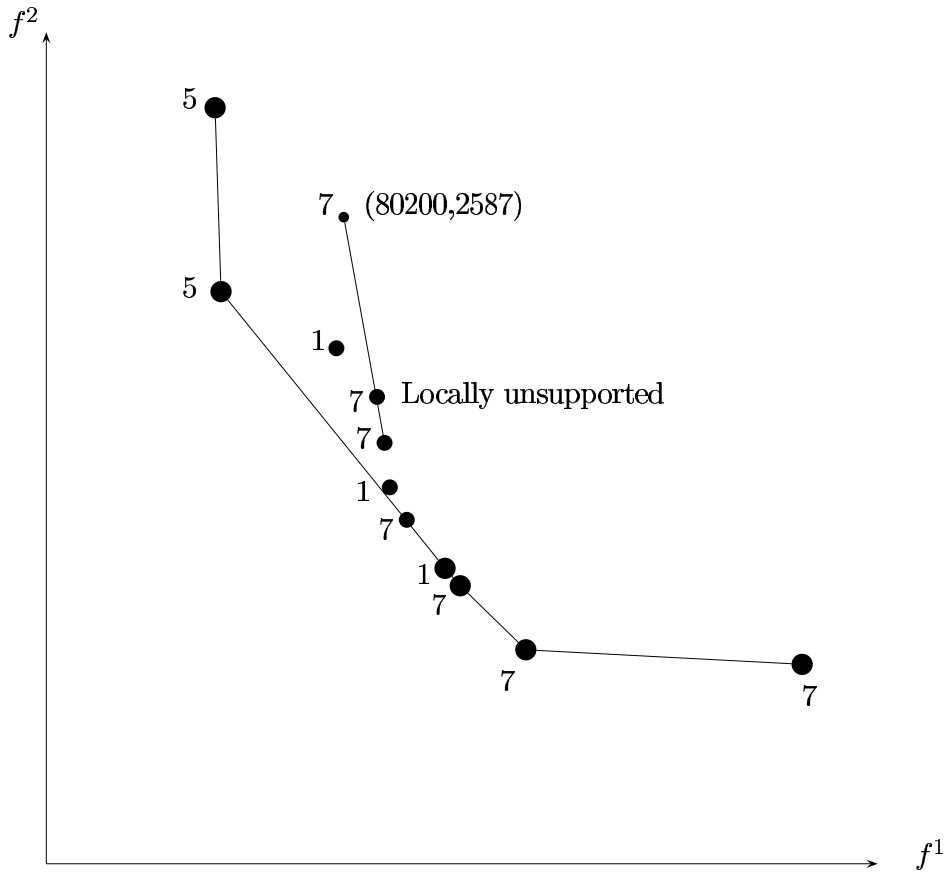

Figure 2: Nondominated vectors for Example 3.1. Large dots illustrate the supported solutions, and only one solution is locally unsupported. The numbers indicate the location node.

3.4 Benders' decomposition in Phase 1

In this section we present how Benders' decomposition can be used to find the supported solutions given a weight λ in Phase 1. Let λ be fixed and define

$$c_{ijk}(\lambda) = \lambda w_k^1 l_{ij}^1 + (1-\lambda) w_k^2 l_{ij}^2 \quad (\geq 0 \text{ since } l, w \geq 0).$$

When x is fixed, we can use the path constraints being totally unimodular, and relax the integrality constraints on y. Fixing x means locating the facility at a particular node. For a fixed $\overline{x}$ satisfying $\sum_i x_i = 1, \;\; x_i \in \{0,1\}$, we get the following **Benders' subproblem**:

$$\begin{aligned} \min \quad & \sum_{k,i,j} c_{ijk}(\lambda) y_{ijk} \\ s.t \quad & \\ & \sum_j y_{jik} - \sum_j y_{ijk} \;=\; -\overline{x}_i \quad i \neq k \quad \forall i,k \\ & 0 \leq y_{ijk} \;\leq\; 1 \qquad\quad \forall i,j,k \end{aligned} \tag{7}$$

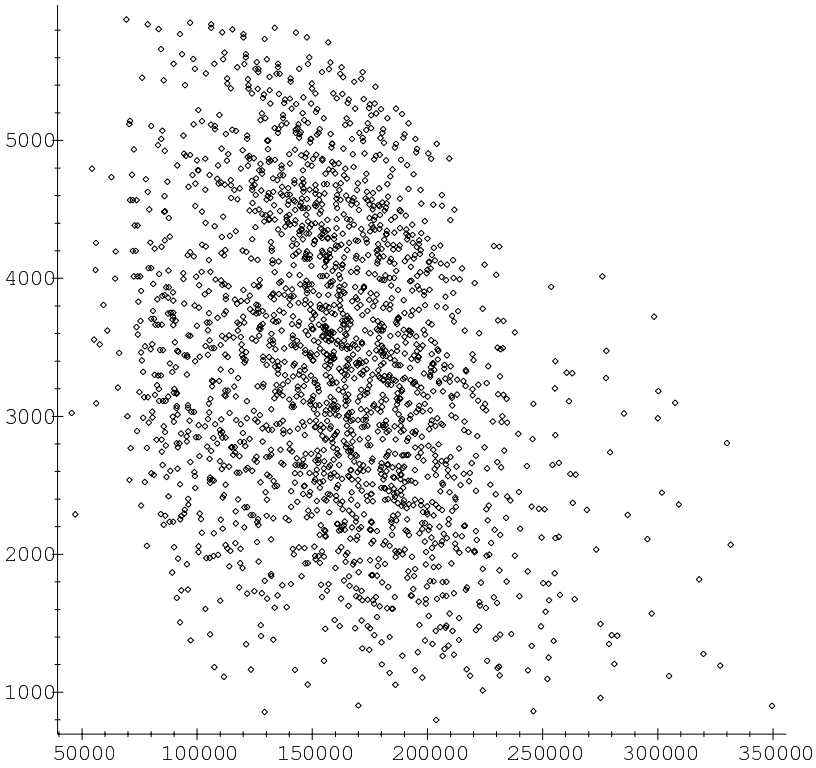

Figure 3: Illustration of 2128 criterion vectors for Example 3.1.

This linear programming problem has the following dual program:

$$\begin{aligned} \max \quad & \sum_{\substack{i,k \\ i \neq k}} \alpha_{ik}(-\overline{x}_i) + \sum_{k,i,j} \beta_{ijk} \\ s.t \quad & \\ & \alpha_{jk} - \alpha_{ik} + \beta_{ijk} \leq c_{ijk}(\lambda) \quad i \neq k \quad \forall i,j,k \\ & \beta \leq 0 \end{aligned} \tag{8}$$

The variables α are free variables corresponding to the path constraints in (7) and the β variables correspond to the upper bound on y. These dual variables can be found when the $n-1$ shortest path problems are solved in the Benders' subproblem, so we need not actually solve the dual problem (8). The dual leads to the following **Benders' master**

problem:

$$
\begin{array}{lll}
\min & v & \\
s.t & & \\
& v \geq -\sum\limits_{\substack{i,k \\ i \neq k}} \alpha_{ik}^{l} x_i + \sum\limits_{k,i,j} \beta_{ijk}^{l} & \forall l \\
& \sum_i x_i = 1 & \\
& x_i \in \{0,1\} & \forall i
\end{array}
\tag{9}
$$

where l is an index for the added inequalities.

The first time we generate a redundant inequality (or suggests a node picked earlier), the solution at hand is optimal (efficient). This is true because the subproblem (7) will return an earlier found solution.

Notice that Benders' master problem (9) is easy to solve in this case. It can be reformulated as a minimax problem. Let us rewrite the first constraint in (9), keeping in mind that only one x_i will be one.

$$
\begin{aligned}
v &\geq -\sum_i \sum_{\substack{k \\ i \neq k}} \alpha_{ik}^{l} x_i + \sum_{k,h,j} \beta_{hjk}^{l} \\
v &\geq \sum_i \left(-\sum_{\substack{k \\ i \neq k}} \alpha_{ik}^{l} + \sum_{k,h,j} \beta_{hjk}^{l} \right) x_i \\
v &\geq \sum_i c_i^{l} x_i
\end{aligned}
$$

where $c_i^{l} = -\sum\limits_{\substack{k \\ i \neq k}} \alpha_{ik}^{l} + \sum\limits_{k,h,j} \beta_{hjk}^{l}$. If we think of these c coefficients in a matrix, the optimal x_i is to find the column where the largest element c_i^{l} is as small as possible.

Notice, that we have to solve problems (7) and (9) at most $n-1$ times. Since Benders' subproblem consists of $n-1$ shortest path problems, problem (7) can be solved in $O(n^3)$ running time. Therefore the overall running time in Phase 1, given λ, is $O(n^4)$ running time.

3.5 Phase 2

Here we can first find the locally supported nondominated vectors by using the weighting method for a fixed node(s).

To find locally unsupported efficient points of (4), we use the Tchebycheff theory. Let $z = (z^1, z^2)$ denote a fixed reference point with $z \leq z^* = \left(f^{1*}, f^{2*}\right)$, where z^* is the ideal

point. Then the augmented non-weighted Tchebycheff program (10) may be stated as

$$
\begin{array}{lrcll}
\min & \alpha + \rho \left(f^1(y) + f^2(y) \right) & & & \\
\text{s.t.} & & & & \\
& f^q(y) - \alpha & \leq & z^q & q = 1,2 \\
& \sum_{i=1}^{n} x_i & = & 1 & \\
& \sum_{j=1}^{n} y_{jik} - \sum_{j=1}^{n} y_{ijk} & = & -x_i & i \neq k \quad \forall i,k \\
& x_i \in \{0,1\} & & & \forall i \\
& y_{ijk} \in \{0,1\} & & & \forall i,j,k \\
& \alpha \in \mathbb{R}_+ & & &
\end{array}
\tag{10}
$$

where ρ is a small positive constant ensuring that the solution found is in fact efficient. A few comments are in order. Note that instead of solving the usual weighted Tchebycheff program as found in Steuer and Choo [8], we propose to solve the augmented non-weighted Tchebycheff program (10). It was shown by Alves and Climaco [1] that all nondominated solutions to (4) can be found using the non-weighted program for integer problems (IP), and in Alves and Climaco [2] this result was generalized to mixed integer problems (MIP). Note that the augmented Tchebycheff program (10) has the same constraints as our original problem (4), as well as two additional constraints. The two new constraints are the reference point constraints, linking the reference point to the objective function in (10). These two new constraints complicate the problem, since they destroy the nice structure of the constraint matrix. Using Lagrange relaxation of these constraints does not solve our problem, as described in Appendix 2. We simply end up with the weighting method. However, problem (10) is a one objective MIP, which can be solved by the usual IP methods, such as branch and bound.

Next we explain how to determine the appropriate reference point(s). Assume that we want to search for locally unsupported solutions between the two nondominated points E_1 and E_2. First, we determine a maximum deviation factor

$$\delta = \max \left\{ \delta^1, \delta^2 \right\}$$

where $\delta^q = \bar{f}^q - f^{q*}$ $q = 1,2$. This deviation factor is going to ensure that our reference point is below the ideal point z^*. Next we find reference points corresponding to our two nondominated solutions, E_1 and E_2:

$$z(E_i) = (E_i^1 - \delta, E_i^2 - \delta) \; i = 1,2$$

The search reference point z_{new} can then be determined as the maximum of the reference point coordinates, because this point has a maximum distance of δ to both $z(E_1)$ and $z(E_2)$:

$$z_{new} = \left(\max\left\{z^1(E_1), z^1(E_2)\right\}, \max\left\{z^2(E_1), z^2(E_2)\right\}\right).$$

Using z_{new} in (10) can result in two things. If a new solution is returned, this solution is nondominated and defines two new search areas. Otherwise one of the points E_1 or E_2 is returned, and no nondominated (unsupported) solutions exist between the two points.

For our Example 3.1 we find $\delta = \max\{203800 - 45500, 3025 - 798\} = 158300$. Next we search for locally unsupported solutions between the two points $E_1 = (78200, 2062)$ and $E_2 = (91200, 1684)$ (on either side of the single locally unsupported point in Figure 2). This leads to the reference point $z_{new} = (-67100, -156238)$, where $\alpha = 158300$ can find both E_1 and E_2. In this case $E_3 = (89200, 1868)$ is found with $\alpha = 158106$.

4 Generalization to multiple criteria

Most of the ingredients in our approach easily generalize to more than two criteria. However, the NISE procedure used in Phase 1 to find supported nondominated points in a "spread-out" way, does not generalize. In two dimensions we find upper bounds on the objectives by minimizing the other objective alone. Forming the hyperplane between these two upper bounds, and then moving this hyperplane, we are guaranteed not to miss any supported nondominated solution. In three dimensions we may set upper bounds as the highest value from minimizing the other two objectives. The problem is that we may have supported nondominated solutions above this hyperplane. In Solanki *et al.* [6] these difficulties are explained.

Using another way to set the weights in Phase 1 in order to find the supported nondominated solutions, will leave us with a similar problem in Phase 2. Near the borders of the efficient frontier it may be difficult to determine a reference point in order to search for unsupported solutions.

5 Concluding remarks

In this paper we present a new, interesting location problem. This formulation incorporates both the location and the routing aspects in a multiobjective setting. We also present a solution method for the problem, and illustrate the problem structure and solution procedure by an example. The presented method can easily be made interactive, since the procedures in both phases are easily made interactive.

Appendix 1

Proof of Theorem 1:
Consider the complete directed network with 4 nodes ($n = 4$). This includes both directed edges between all nodes given k: (i, j, k) and (j, i, k) $\forall i, j, k$ where $i \neq j$. From (5) choose the first 4 columns corresponding to the x variables. Choose also the three columns corresponding to y_{124},y_{132} and y_{143}. Next we specify the seven rows. Choose the first row corresponding to the sum of x_i constraint. From I_{-4} choose rows 1 and 2, from I_{-2} choose rows 1 and 2 and from I_{-3} choose rows 1 and 3. This lead to the following 7×7 matrix with determinant two:

$$\begin{vmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{vmatrix} = 2$$

Appendix 2

Lagrange relaxation in the augmented Tchebycheff problem
As we will show, this approach does not help! We end up with the weighting method, if we relax the reference point constraints.
Let β be the Lagrange multiplier on the reference point constraints of problem (10). We are then left with the constraints of our original problem (4), and the constraint $\beta \geq 0$. Let's assume that β is fixed at $\bar{\beta}$. $\bar{\beta}$ can then be updated using for example a subgradient. The new objective function is given by

$$f(x, y) = \alpha + \rho \left(f^1(y) + f^2(y)\right) + \beta^1(f^1(y) - \alpha - z^1) + \beta^2(f^2(y) - \alpha - z^2).$$

Rearranging terms, we get

$$f(x, y) = (1 - \beta^1 - \beta^2)\alpha + (\rho + \beta^1)f^1(y) + (\rho + \beta^2)f^2(y) - \beta^1 z^1 - \beta^2 z^2 \qquad (11)$$

Let's evaluate the optimal value of α. If $1 - \beta^1 - \beta^2 \geq 0$, we choose $\alpha = 0$, and if $1 - \beta^1 - \beta^2 < 0$, we choose $\alpha = \infty$. Neither solution is good, because $\alpha = 0$ makes no improvement when we update $\bar{\beta}$ using the usual sub-gradient direction

$$d = (f^1(y) - \alpha - z^1, f^2(y) - \alpha - z^2)^t$$

Since z is a reference point $f(y) > z$, and we will simply increase $\bar{\beta}$ until we get the situation where $\alpha = \infty$. We therefore conclude that $\beta^1 + \beta^2 = 1$, so α can be any positive number. Since ρ is almost zero, we recognize this to be the weighting method applied in Phase 1.

References

[1] M.J. Alves and J. Climaco. Using cutting planes in an interactive reference point approach for multiobjective integer linear programming problems. *European Journal of Operational Research*, 117:565–577, 1999.

[2] M.J. Alves and J. Climaco. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. *European Journal of Operational Research*, 124:478–494, 2000.

[3] J.L. Cohon. *Multiobjective Programming and Planning*. Academic Press, 1978.

[4] M. Ehrgott. *Multicriteria Optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems*. Springer, 2000.

[5] A.J.V. Skriver and K.A. Andersen. A label correcting approach for solving bicriterion shortest path problems. *Computers and Operations Research*, 27:507–524, 2000.

[6] R.S. Solanki, P.A. Appino, and J.L. Cohon. Approximating the noninferior set in multiobjective linear programming problems. *European Journal of Operational Research*, 68:356–373, 1993.

[7] R.E. Steuer. *Multiple criteria optimization: Theory, Computation, and Application*. Wiley, New York, 1986.

[8] R.E. Steuer and E.U. Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26:326–344, 1983.

[9] E.L. Ulungu and J. Teghem. The two-phases method: An efficient procedure to solve biobjective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20:149–165, 1995.

[10] M. Visée, J. Teghem, M. Pirlot, and E.L. Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12:139–155, 1998.