Minimum Cost-Reliability Ratio Path Problem

Abdallah W. Aboutahoun
Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt
tahoun44@yahoo.com

Abstract: *The problem of finding a minimal cost-reliability ratio path is considered. The optimal solution to this problem is shown to map into an extreme supported non-dominated objective point in the objective space of the biobjective shortest path problem. Different forms of reliability are presented. We assume that this reliability does not change over time. We employ a parametric network simplex algorithm to compute all extreme supported non-dominated objective points. A sufficiency conditions introduced by Ahuja [1] are used to reduce the path enumeration. Our algorithm is based on the method of Sedeño-Noda and González-Martín. A numerical example is provided to illustrate the algorithm.*

[Abdallah W. Aboutahoun. **Minimum Cost-Reliability Ratio Path Problem.** *Life Sci J* 2012; 9(3):1633-1645] (ISSN: 1097-8135). http://www.lifesciencesite.com. 239

Keywords: biobjective shortest path problem; extreme efficient solution; extreme non-dominated solution.

1 Introduction

Suppose that the network is described by a connected, directed graph $G(V,E)$, where $V = \{1, 2, \dots, n\}$ is the node set and $E = \{e = (i,h): i, h \in V\}$ is the edge set. The nodes are assumed to be perfectly reliable. Associated with each edge $e \in E$ are two attributes. The first attribute is the edge cost c_{ij}. The second attribute is the probability $0 < p_{ij} < 1$, that when attempting to traverse edge (i,j) it is found in an operational state. The reliability measures the probability that the edge will be operational. The reliability of a directed path is de.ned as the product of the reliability of edges in the path $\left[i.e., \ R(P) = \prod_{(i,j)\in P} p_{ij}\right]$. We assume that this probability does not change over time.

Let s and t be two given and distinguished nodes of $G(V,E)$. A path P from s to t in $G\ (V,E)$ or simply path is a sequence of non-repeated nodes and connecting arcs, joining the initial node s to the terminal node t. We consider the problem of determination of a directed path P from a source node s to a destination node t for which

$$\frac{\sum_{(i,j)\in P} c_{ij}}{\prod_{(i,j)\in P} p_{ij}}$$

is minimum among all such paths. We refer to this problem as the Minimum Cost -Reliability Ratio Path Problem (MCRRPP) [1].

Ahuja [1] observed that the optimum solution of the MCRRPP is an efficient extreme solution of the bicriterion path problem. He employed the parametric programming to enumerate these efficient extreme solutions and a sufficiency condition is used to cut down the enumeration substantially. The algorithm is shown to be pseudo-polynomial. Chandrasekaran [6] provided a polynomial bounded algorithm to solve minimal ratio spanning trees. Chandrasekaran et al. [7] presented a polynomial algorithm consisting of an indirect search in the set of efficient extreme points for computing the solution to the cost-reliability ratio spanning tree problem. Aneja and Nair [3] considered a finite serial multistage system where the measure of effectiveness of the system is a ratio of two return functions. The numerator of the ratio is an additive return function whereas the denominator is a multiplicative one. They considered two-criterion dynamic program and showed that the optimal solution of the ratio dynamic program is a non-dominated solution of the two criteria program. Martins [12] presented a polynomial algorithm to determine a path between a specified pair of nodes, which minimizes the cost/capacity ratio.

This paper is organized as follows. Section 2 presents concepts, definitions and problem properties. In Section 3, we present an algorithm to solve MCRRPP. A numerical example is presented in Section 4. In Section 5 we conclude with some comments.

2 The problem and Properties

Let Φ be the set of all directed paths in $G(V,E)$ from the source s to the destination t. For each $P \in \Phi$ define

$$C(P) = \sum_{(i,j)\in P} c_{ij}$$
$$R(P) = \prod_{(i,j)\in P} p_{ij} \qquad (1)$$
$$D(P) = \sum_{(i,j)\in P} d_{ij}$$

where, $d_{ij} = -\ln p_{ij}$, $0 < p_{ij} \le 1$ then $d_{ij} > 0$, $\forall (i,j) \in E$ and $R(P) = \prod_{(i,j)\in P} p_{ij} = e^{-\sum_{(i,j)\in P} d_{ij}}$ for all $P \in \Phi$ which means that $R(P) = e^{-D(P)}$.

Now the problem we consider is

$$\min_{P\in\Phi} z(P) = \frac{C(P)}{R(P)} = C(P)e^{D(P)} \quad (2)$$

Associated with (2), we define the biobjective shortest path (BSP) problem as follows:

$$\min_{P\in\Phi}[C(P), D(P)] \quad (3)$$

and

$$\min_{P\in\Phi}[C(P), -R(P)] \quad (4)$$

The mathematical programming formulation of the BSP (3) is

$$\min F(x) = \begin{cases} f_1(x) = \sum_{(i,j)\in P} c_{ij}x_{ij} \\ f_2(x) = \sum_{(i,j)\in P} d_{ij}x_{ij} \end{cases} \quad (5)$$

$$s.t.$$

$$\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} 1 & if\ i = s \\ 0 & if\ i \neq s,t \\ -1 & if\ i = t \end{cases}$$

$$x_{ij} \in \{0,1\}, \forall (i,j) \in E$$

where s is the designated source node and t is the designated terminal node. Let X be the set of all feasible solutions to (5) and it is also called the feasible set in the decision space. So, the problem (5) can be stated as follows:

$$\min F(x) = (f_1(x), f_2(x)) \quad (6)$$
$$s.t.\ x \in X$$

Now, we introduce general definitions and a classification of efficient solutions. We will follow the terminology of Raith and Ehrgott [24], Eusébio and Figueira [13], Raith and Ehrgott [25], and Hamacher et al. [18].

Definition 1 A feasible solution $\tilde{x} \in X$ is called efficient if there does not exist any $x \in X$ with $(f_1(x), f_1(x)) \leq (f_1(\tilde{x}), f_2(\tilde{x}))$ and $(f_1(x), f_2(x)) \neq (f_1(\tilde{x}), f_2(\tilde{x}))$. Otherwise x is inefficient.

Let C be $p \times n$ criterion matrix whose rows are the c^i, the composite objective function is written $\lambda^T Cx$. The following theorem shows that the set of efficient solutions in X can be obtained by solving a parametric problem.

Theorem 1 $x \in X$ is efficient if and only if there exists

$$\lambda \in \Omega = \left\{\lambda \in R^p: \lambda_i > 0, \sum_{i=1}^{p} \lambda_i = 1\right\}$$

such that x minimizes the weighted-sum linear programming problem $\min\{\lambda^T Cx: x \in X\}$ (see [13] and [29], p.215).

Efficiency is defined in the decision space. There is a natural counterpart in the objective space. The objective space is denoted by Y and is given by

$$Y = \{F(x) \in R^2: F(x) = (f_1(x), f_2(x)), x \in X\}$$

Definition 2 $F(x) \in Y$ is a non-dominated (ND) point if and only if x is an efficient solution to (6). Otherwise $F(x)$ is a dominated point.

Let $X_E \subseteq X$ be the set of all efficient solutions of the BSP (6) and $Y_{ND} \subseteq Y$ be the set of all ND objective points. We distinguish two different types of ND objective points, supported and non-supported ND objective points. Let

$$Y^{\geq} = conv(Y_{ND}) + R^p_{\geq}$$

where $conv$ is the convex hull operator and $R^p_{\geq} = \{y \in R^p: y \geq 0\}$ is the Pareto cone and $conv(Y_{ND}) + R^p_{\geq} = \{y \in R^p: y = y' + y'', y' \in conv(Y_{ND}), y'' \in R^p_{\geq}\}$. The non-dominated frontier of Y is defined as the set [see Ehrgott [22] and Hamacher et al. [18]]

$$\left\{y \in conv(Y_{ND}): conv(Y_{ND}) \cap \left(y + \left(-R^p_{\geq}\right)\right) = \{y\}\right\}$$

Definition 3 (Supported ND solution Y_{SN}). Let y denote an ND objective solution. Then, if y belongs to the efficient frontier of Y, y is a supported ND objective solution. Otherwise, y belongs to the interior of $Y^{\geq}$ and it is a non-supported ND objective solution.

The efficient frontier is piecewise linear and convex. Its breakpoints are the extreme ND objective points which are images of extreme efficient solutions in the decision space.

Definition 4 (Extreme supported ND solution Y_{XSN}). Let $y \in Y_{SN}$. Then, y is an extreme supported solution if it is an extreme point of $Y^{\geq}$. Otherwise, y is a non-extreme supported solution.

All supported ND objective points are located on the "lower-left boundary" of $conv(Y_{ND})$, *i.e.* they are ND points of $Y^{\geq}$. The supported and the non-supported efficient solutions are defined to be the inverse images of the supported and the non-supported of ND objective points. They can be distinguished as follows:

- Supported efficient solutions are those efficient solutions that can be obtained as optimal solutions to a (single objective) weighted sum problem

$$\min_{x \in X} \lambda f_1(x) + (1-\lambda) f_2(x) \qquad (7)$$

for some $\lambda > 0$. The set of all supported efficient solutions is denoted by X_{SE}, its non-dominated image is Y_{SN}.

- Supported efficient solutions which define an extreme point of $Y^{\geq}$ are called extreme supported efficient solutions and is denoted by X_{XSE}.
- The remaining efficient solutions in $X_{NE} := X_E \backslash X_{SE}$ are called non-supported efficient solutions. They cannot be obtained as solutions of a weighted sum problem as their images lie in the interior of $Y^{\geq}$. The set of non-supported non-dominated points is denoted by Y_{NN}. Note that this definition implies $Y_{NN} \subset int\left(conv(Y_{ND}) + R^p_{\geq}\right)$. There is no known characterization of non-supported efficient solutions that leads to a polynomial time algorithm for their computation.

The two objective functions f_1 and f_2 do generally not attain their individual optima for the same values of $\tilde{x}$. We will assume in the following that there exists no $\tilde{x}$ such that $\tilde{x} \in \arg\min\{f_1\}$ and $\tilde{x} \in \arg\min\{f_2\}$ for a problem of the form (5).

The solution of the BSP contain both non-supported and supported non-dominated vectors / efficient solutions, which can be geometrically characterized as follows: the non-supported non-dominated vectors are located inside the feasible region in the objective space, while the supported vectors are found on the boundaries of the convex hull of this feasible region. Supported non-dominated vectors correspond to the optimal solutions of a sequence of single objective parametric network flow problems.

All the previous terminology can be summarized in Table 1.

Table 1: Classification of efficient and non-dominated in the decision and objective spaces

Decision Space	Objective Space
X: set of all feasible solutions	$Y = F(X)$: image of X under the objective function (objective space)
X_E: set of all efficient solutions	Y_{ND}: set of all non-dominated objective solutions
$X_{SE} \subseteq X_E$: set of all supported efficient solutions	$Y_{SND} \subseteq Y_{ND}$: set of all supported non-dominated objective solutions
$X_{XSE} \subseteq X_{SE}$: set of all extreme supported efficient solutions	$Y_{XSND} \subseteq Y_{SND}$: set of all extreme supported non-dominated objective solutions
$X_{NE} := X_E \backslash X_{SE}$: set of all non-supported efficient solutions	$Y_{NND} \subseteq Y_{ND} \backslash Y_{SND}$: set of all non-supported non-dominated objective solutions

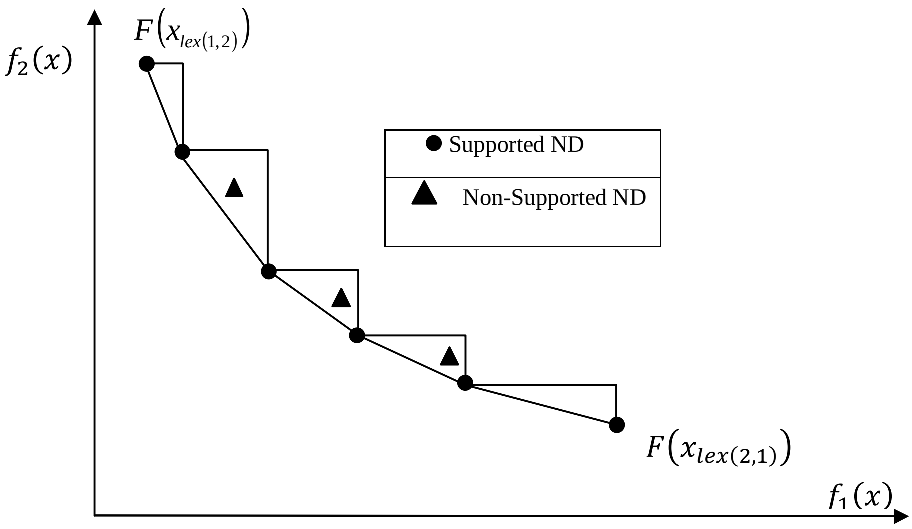

Figure 1: All non-dominated points in the objective space

Theorem 2 An optimal solution P^* of the MCRRPP maps into a supported extreme non-dominated point of $conv(Y)$.
Proof As we mentioned above $R(P) = e^{-D(P)}$: It easy to see that the optimal solution P^* of the MCRRPP maps into a non-dominated objective point of (3). Otherwise, let $\tilde{P}$ such that $C(\tilde{P}) < C(P^*)$ and $D(\tilde{P}) < D(P^*)$ with strict inequality holding at least at one of these two places. This implies

$$\frac{C(\tilde{P})}{e^{-D(\tilde{P})}} < \frac{C(P^*)}{e^{-D(P^*)}}$$

since $C \geq 0$. That is

$$\frac{C(\tilde{P})}{R(\tilde{P})} < \frac{C(P^*)}{R(P^*)}$$

which contradict the optimality of P^*.
Suppose that P^* maps into $y^* \in conv(Y) + R^2_{\geq}$. We want to show that y^* is an extreme point of $Y^{\geq}$.
Suppose the contrary, y^* is not an extreme point of $Y^{\geq}$. Then there exist two extreme points y^1 and y^2 (corresponding to two efficient extreme paths P^1 and P^2), such that $y^* = \alpha y^1 + (1-\alpha)y^2; 0 < \alpha < 1$, where

$$y^1 = (C(P_1), D(P_1))$$
$$y^2 = (C(P_2), D(P_2))$$

and

$$y^* = (C(P^*), D(P^*))$$

Assume that

$$\frac{C(P_1)}{R(P_1)} = m_1 \leq \frac{C(P_2)}{R(P_2)} = m_2$$

Now,

$$D(P^*) = \alpha D(P_1) + (1-\alpha)D(P_2)$$

and by convexity of e^{-x}, we have

$$e^{-D(P^*)} = e^{-(\alpha D(P_1)+(1-\alpha)D(P_2))} < \alpha e^{-D(P_1)} + (1-\alpha)e^{-D(P_2)} = \alpha\frac{C(P_1)}{m_1} + (1-\alpha)\frac{C(P_2)}{m_2} = \frac{[\alpha C(P_1) + (1-\alpha)C(P_2)]}{m_1} = \frac{C(P^*)}{m_1}$$

That is, $\frac{C(P^*)}{R(P^*)} > m_1$, contradicting the optimality of P^* ■
The problem, thus, reduces to searching through shortest paths which correspond to non-dominated extreme points of the set $Y^{\geq}$ in the biobjective space.

Definition 5 A function $f: S \subseteq R \to R$ is unimodal on an interval S if there exists a $x^* \in S$ at which f attains a minimum and f is nondecreasing on the interval $\{x \in S: x \geq x^*\}$ whereas it is nonincreasing on the interval $\{x \in S: x \leq x^*\}$.
It is well known that the efficient frontier obtained by joining the points P_{k-1} to P_k for all $k = 2, \dots, w$, is a piecewise linear convex function and typically is of the form as shown in Fig. 1. Let $P_1, P_2, \dots, P_w$ be the set of all ND extreme points of (3) in the increasing order of their $D(P_i)$ value. Let $C_{min} = C(P_w)$ and $C_{max} = C(P_1)$. Further, let L_k denote the line passing through P_{k-1} and P_k. The equation of L_k is given by $y = a_k - b_k x$, where $b_k = \frac{D(P_k)-D(P_{k-1})}{C(P_{k-1})-C(P_k)}$ and $a_k = D(P_k) + b_k C(P_k)$. For any point $(x, y) \in L_k$ define $h_k(x) = xe^y = xe^{a_k - b_k x}$. It is easy to see that $h_k(x)$ is a unimodular function and achieves its maximum at $x^* = \frac{1}{b_k}$.
Let $x_i = C(P_i), \forall i = 1, 2, \dots, w$. Further, let

$$z_k^* = \min_{1 \leq i \leq k}\{z(P_i)\}$$

and P^* be the path for which this minimum is attained.

Theorem 3 If $h_k(C_{min}) \geq z_k^*$, then P^* is an optimum solution of the MCRRPP.
Proof Since the efficient frontier is piecewise linear and convex, it follows that $a_k - b_k x_i < D(P_i), \forall\, i = k+1, k+2, \dots, l$, then

$$h_k(x_i) = x_i e^{a_k - b_k x_i} < C(P_i)e^{D(P_i)} = z(P_i), \forall\, i = k+1, \dots, l$$

Since the function $h_k(x)$ achieves its maximum at $x = \frac{1}{b_k}$, so we consider two cases, the first case when $x_k \leq \frac{1}{b_k}$, and by the nature of the function $h_k(x)$

$$z_k^* \leq h_k(C_{min}) \leq h_k(x_i) < z(P_i), \quad \forall i = k+1, k+2, \dots, m$$

The second case when $x_k > \frac{1}{b_k}$, let $\tilde{x}_k$ be such that $h_k(x_k) = h_k(\tilde{x}_k)$

$$z_k^* \leq h_k(C_{min}) \leq h_k(x_k) \leq h_k(x_i), \quad \forall \tilde{x}_k \leq x_i \leq x_k$$

and the proof is complete ■

The paths are enumerated in the order $P_w, P_{w-1}, \dots, P_1$ by the parametric analysis which we are going to explain in Section 4. We can use the following condition as a termination condition. Let $\tilde{P}^*$ be the minimum of $\tilde{z}^* = \min_{k \leq i \leq w}\{z(P_i)\}$.

Theorem 4 If $h_{k+1}(C_{max}) \geq \tilde{z}_k^*$, then $\tilde{P}^*$ is an optimum solution of the MCRRPP
Proof The proof is similar to the previous Theorem ■

2.1 A different measure for reliability

In this section we are presenting different measure for the reliability of a path $R(P) = \prod_{(i,j)\in P} p_{ij}$. We assume that this probability does not change over time. Although there are no limitations regarding the number of edges that can be in a failed state, we assume that failures occur independently and they are unrecoverable. Reliability of a path refers to the probability of traversal, *i.e.*, the probability that all edges along the path are operational. We model the operational probability of an edge as an exponential function of physical distance. A realistic assumption regarding p_{ij} is that failures that prohibit the use of the edge for traversal are generated according to a Poisson process with constant rate $\lambda_{ij}, (i,j) \in E$, modeling p_{ij} as an exponential function of the physical distance. The failure rate λ_{ij} represents the average number of failures per unit length. We represent the relationship between edge lengths, operational probability and failure rate, using the exponential model introduced by Melachrinoudis and Helander [19], as $p_{ij} = e^{-\lambda_{ij} d_{ij}}$.

Suppose we know for each edge $(i,j) \in E$ its failure rate λ_{ij} and distance d_{ij}. The operational probabilities are calculated by using the exponential model $p_{ij} = e^{-\lambda_{ij} d_{ij}}$. We use the logarithmic transformation between operational probability and edge length, which was proposed by Melachrinoudis and Helander [19], to calculate for each edge $(i,j) \in E$ the "artificial" edge length $d_{ij}^A = -\ln p_{ij} = \lambda_{ij} d_{ij}$ and to define a new network $\breve{G}(V,E)$ with the same sets of edge attributes costs, c_{ij} and operational probabilities, $p_{ij} = e^{-d_{ij}^A}$ but its edge distances are $d_{ij}^A, (i,j) \in E$. Due to the logarithmic transformation, the most reliable route between nodes i and j on $G(V,E)$ is the shortest path between nodes i and j on $\breve{G}(V,E)$.

Let Φ be the set of all directed paths in $\breve{G}(V,E)$. For each $P \in \Phi$ define

$$C(P) = \sum_{(i,j)\in P} c_{ij}$$

$$\breve{R}(P) = \prod_{(i,j)\in P} p_{ij}$$

$$\breve{D}(P) = \sum_{(i,j)\in P} d_{ij}$$

where, $d_{ij}^A = -\ln p_{ij}$, $0 < p_{ij} \le 1$ then $d_{ij}^A > 0$, $\forall (i,j) \in E$ and $\breve{R}(P) = \prod_{(i,j)\in P} p_{ij} = e^{-\sum_{(i,j)\in P} d_{ij}^A}$ for all $P \in \Phi$ which means that $\breve{R}(P) = e^{-\breve{D}(P)}$.

Now, the problem we consider is

$$\min_{P\in\Phi} z(P) = \frac{C(P)}{\breve{R}(P)} \qquad (8)$$

Associated with (8) we define the following biobjective shortest path (BSP) problem

$$\min_{P\in\Phi}[C(P), \breve{D}(P)] \qquad (9)$$

and

$$\min_{P\in\Phi}[C(P), -\breve{R}(P)]$$

Theorem 5 An optimal solution P^* of the problem (8) maps into a supported extreme non-dominated point of (9).

Proof Similar to Theorem 1

Theorem 6 If all edge failure rates are equal, the optimal solution P^* of (2) is the same as the optimal solution of (8)

Proof Let $\lambda_{ij} = \lambda$, $\forall (i,j) \in E$. The network $\breve{G}(V,E)$ has the same topology as $G(V,E)$ and its edge lengths have been scaled by λ, *i.e.*, $d_{ij}^A = \lambda d_{ij}$. Let P^* be the optimal solution of the problem $\min_{P\in\Phi} z(P) = \frac{C(P)}{\breve{R}(P)}$. Since $\breve{D}(P) = \sum_{(i,j)\in P} d_{ij}^A = \sum_{(i,j)\in P} \lambda d_{ij} = \lambda D(P)$, hence,

$\min_{P\in\Phi} \frac{C(P)}{\breve{R}(P)} =$

$\min_{P\in\Phi} \frac{C(P)}{e^{-\breve{D}(P)}} = \min_{P\in\Phi} \frac{C(P)}{e^{-\lambda D(P)}} = \min_{P\in\Phi} \frac{C(P)}{(R(P))^{\lambda}}$,

which proves that P^* is also the optimal solution of $\min_{P\in\Phi} \frac{C(P)}{R(P)}$ ■

Table 2: Classification of BSP algorithms and references

Two Phase Method	Path/tree	Mote et al. [16]
Biobjective Label Correcting	Node-selection	Skriver and Andersen [27], Brumbaugh-Smith and Shier [4]
Biobjective Label Setting	Label-selection	Hansen [10]
Kth Shortest Path	Ranking	Clímaco and Martins [8]
Near Shortest Path	Ranking	Carlyle and Wood [5]

3 Solution Method

3.1 A brief review of solution methods for the BSP problem

In this section we give a brief review of different methods to solve BSP exactly. Three main approaches are considered. The two phase method, the biobjective labeling methods, and ranking methods. Climaco and Martin [8] and Mote et al. [16] fall in the path/tree handling procedure. Hansen [10], Brumbaugh-Smith and Shier [4] and Skriver and Andersen [27] fall in the labeling procedure.

In table 2 the references that fall in the main approaches to solve BSP are listed.

Our review is based on Skriver [26] and Raith and Ehrgott [25]

1) Two phase method
   In the existing literature all algorithms, except perhaps the Parametric Approach by Mote et al. [16], have been proven slower than the Label Correcting approach [27]. In phase I, all the extreme supported efficient solutions (efficient solutions which define extreme points of the convex hull of the set of feasible objective vectors) are computed. In the second phase the remaining efficient solutions are computed with one of the enumerative approaches mentioned before. The enumerative methods can be employed in a very effective way as enumeration can be restricted to small areas of the objective space [see [25]].
2) Biobjective label correcting
   Label correcting differs in whether they employ label-selection or node-selection. Skriver and Andersen [27] have claimed that the node-selection algorithms outperform the path/tree algorithms (two phase method) because the number of non-dominated values is always smaller than (or equal to) the number of efficient paths. A stronger argument is that the node-labeling algorithm only finds the list of non-dominated values at the terminal node, and not the actual efficient paths.
3) Biobjective label setting
   Biobjective label setting approaches always employ label-selection. In particular, a lexicographically smallest label with respect to all nodes is selected among all tentative labels in each iteration. Guerriero and Musmanno [9] investigated label correcting and label setting methods for the multicriteria shortest path tree problem. There are problem instances where label-selection is superior and others where node-selection is superior. Furthermore, label setting is superior for some instances, and label correcting is superior for others.
4) Ranking methods
   Starting with the optimal value for one objective, the second-best solution, the third-best solution, etc. is obtained until the *k*-best solution is reached. For BSP, the process continues until it is guaranteed that all non-dominated points have been found. *K*th shortest path methods have been found not to be competitive with label correcting methods. On the basis of computational tests, Carlyle and Wood [5] conclude that their near shortest path routine solves the k-shortest path problem faster than other algorithms dedicated to solving the k-shortest path problem [25].

A label correcting algorithm with node-selection is identified as the most successful approach to solve BSP problems by Skriver and Andersen [27] and label setting as in Guerriero and Musmanno [9]. Raith and Ehrgott [25] conclude that two phase method is competitive with other commonly applied approaches to solve the BSP problem. The two phase method works well with both a ranking, a label correcting, and a label setting approach in phase 2, but the label correcting and setting approaches appear to be preferable as they are more stable. The purely enumerative near shortest path approach is a very successful approach to solve some problem instances, but the run-time on others is very long.

Skriver and Andersen [27] argued that the parametric approach is slower, due to the structure of the algorithm. The approach is to use the weighting method to find the efficient extreme paths, and then use backtracking of spanning trees to search for non-extreme efficient paths. The weighting method means solving LP problems, but for the shortest-path problem that is done by Dijkstra's shortest-path algorithm (or a similar algorithm). It turns out that Dijkstra's algorithm is actually a slower approach in practice than the Label correcting routine. On top of this comes the fact, that the weighting method of the parametric approach by far is faster than the backtracking part. When we are backtracking, we might have to evaluate all the edges in all the spanning trees in the worst case, resulting in an exponentially growing number of comparisons.

We are going to use in this paper phase I in the two phase method. The backtracking part which makes the two phase method slower than the labeling algorithms will not be used here. Since the non-extreme efficient paths need not be generated.

Skriver and Andersen [27] presented a label correcting algorithm for solving the BSP. They imposed some simple domination conditions, which reduced the number of iterations needed to find all the efficient (Pareto optimal) paths in the network. Guerriero and Musmanno [9] developed a solution of the multicriteria shortest path problem. They present

a class of labeling methods to generate the entire set of Pareto-optimal path-length vectors from an origin node s to all other nodes in a multicriteria network.
Raith and Ehrgott [25] compared different strategies for solving the BSP problem. They considered a standard label correcting and label setting method, a purely enumerative near shortest path approach, and the two phase method, investigating different approaches to solving problems arising in phases 1 and 2. In particular, they investigated the two phase method with ranking in phase 2. In order to compare the different approaches, they investigated their performance on three different types of networks. They were able to show that the two phase method is competitive with other commonly applied approaches to solve the BSP problem. The two phase method works well with both a ranking, a label correcting, and a label setting approach in phase 2, but the label correcting and setting approaches appear to be preferable as they are more stable.
Raith and Ehrgott [24] presented an algorithm to compute a complete set of efficient solutions for the biobjective integer minimum cost flow problem. They used the two phase method, with a parametric network simplex algorithm in phase 1 to compute all non-dominated extreme points. In phase 2, the remaining non-dominated points (non-extreme supported and non-supported) are computed using a $k-$ best flow algorithm on single-objective weighted sum problems. Eusébio and Figueira [13] presented an algorithm for finding all the non-dominated solutions and corresponding efficient solutions for biobjective integer network flow problems. The algorithm solves a sequence of $\varepsilon-$ constraint problems and computes all the non-dominated solutions by decreasing order of one of the objective functions.
Mote et al. [16] developed an algorithm to solve the BSP. This algorithm first relaxes the integrality conditions and solves a simple bicriterion network problem. The bicriterion network problem is solved parametrically, exploiting properties associated with adjacent basis trees. Consider the following biobjective linear programming formulation which is to send 1 unit of flow from the source s to every other node along efficient paths.

$$\min F(x) = \begin{cases} f_1(x) = \sum_{(i,j)\in P} c_{ij}x_{ij} \\ f_2(x) = \sum_{(i,j)\in P} d_{ij}x_{ij} \end{cases} \quad (10)$$

s. t.

$$\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} n-1 & if\ i = s \\ -1 & if\ i = t \end{cases}$$

$$x_{ij} \geq 0\ and\ integer\ , \forall (i,j) \in E$$

4 The Algorithm

The shortest path problem has been studied extensively and many polynomial and strongly algorithms for solving it have been proposed [see, [2]]. We present here a brief review of the primal simplex algorithm for the shortest path problem. Like minimum cost flow problem, the shortest path problem has a spanning tree solution. Because node s is the only source node to every other node is demand node, the tree path from the source node to every other node is a directed path. This implies that the spanning tree must be a directed out tree rooted at node s. Any spanning tree for the shortest path problem contains a unique directed path from node s to every other node. The single-objective shortest path simplex (SPS) algorithms maintain a basic solution at each stage. Every basic feasible solution corresponds to a spanning tree T of the network $G(V,E)$. Every feasible basis tree T is a directed-out (spanning tree) rooted at node s, and it represents nondegenerate solution, $i.e., x_{ij} > 0$ for all $(i,j) \in T$ because $x_{ij} = |N_j|$, where N_j denoted the set of nodes in the subtree of T rooted at j.
A dual variable associated with each node of $G(V,E)$ is a function $\pi: V \rightarrow R$. For a given dual variable π, the reduced dual of an arc (i,j) is defined as $\tilde{c}_{ij} = c_{ij} - \pi_i + \pi_j$. The SPS algorithm finds the optimal basis tree that is a tree of shortest paths and the optimal node potentials (dual variables) $\pi_i, i \in V$. These dual variables are defined by requiring that $\pi_s = 0$ and that $\tilde{c}_{ij} = 0$ for each arc in the spanning tree T.
At each iteration, the SPS algorithm selects an eligible arc to enter the basis. There are different rules for the selection of entering arcs. The process of moving from one feasible basis tree to another feasible basis tree is called a simplex pivot. On a simplex pivot an arc $(p,q) \notin T$ is added to T creating a unique cycle and an arc $(i,j) \in T$ is deleted yielding a new basis tree. A new basic feasible solution is obtained by replacing arc (p,q) by $(pred(q),q)$ in T and updating the node potentials $\pi_i, \forall i \in V$. In each step in the network simplex algorithm, a non-basic arc (p,q) with a negative reduced cost to introduce into the spanning

tree. The addition of arc (p, q) to the tree creates a cycle which we orient in the same direction as arc (p, q). Let w be the apex of this cycle. In this cycle, every arc from node q to node w is a backward arc and every arc from node w to node p is a forward arc: Consequently, the leaving arc would lie in the segment from q to w. In fact, the leaving arc would be the arc $(pred(q), q)$ because this arc has the smallest flow value among all arcs in the segment from node q to node w.

According to the above discussion, if (p, q) is an entering arc on a simplex pivot and $p \notin N_q$, then the leaving arc is $(pred(q), q)$. If $p \in N_q$ then the network contains a negative cost cycle which yields unbounded solution. Let $NB = \{(i, j) \notin T: \tilde{c}_{ij} < 0\}$ be the set of all nonbasic arcs. The algorithm would then increase the potentials of nodes in the subtree rooted at node q by the amount $|\tilde{c}_{ij}|$ update the tree indices, and repeat the computations until all nontree arcs have nonnegative reduced costs. When the algorithm terminates, the final tree would be a shortest path tree ($i.e.$, a tree in which the directed path from node s to every other node is a shortest path).

4.1 Parametric Simplex

The optimal solution to the MCRRPP corresponds to an extreme supported non-dominated point of the BSP, so we present an algorithm that computes a complete set of extreme supported non-dominated points in the objective space. We will not compute the non-supported non-dominated points.

The two phase method [25] is based on computing supported and non-supported non-dominated points separately. In phase 1 extreme supported efficient solutions are computed, possibly taking advantage of their property of being obtainable as solutions to the weighted sum problem (4). The other approach is based on the network simplex method where extreme efficient solutions are generated in a right-to-left (or left-to-right) fashion. In phase 2 the remaining supported and non-supported efficient solutions can be computed with different enumerative approaches, as there is no theoretical characterization for their efficient calculation. It is expected that the search space for the enumerative approach in phase 2 is highly restricted due to information obtained in phase 1 so that the associated problems can be solved a lot quicker than by solving BSP with a purely enumerative approach only. The enumerative methods can be employed in a very effective way as enumeration can be restricted to small areas of the objective space. Phase 2 must determine $x \in X$ such that $F(x)$ is in the triangle defined by two consecutive non-dominated supported points in the objective space (see Fig. 1).

In this paper, according to Theorem 2, we need only to consider phase 1 to compute a complete set of extreme supported efficient solutions. We use a parametric simplex method proposed by Sedeño-Noda and González-Martín [14]. Initially, one of the two lexicographically optimal solutions, e.g., the $lex(1,2)$-best solution, is obtained with a single-objective network simplex algorithm with $lex(1,2)$ objective. The procedure generates a complete set of extreme efficient solutions moving in a right-to-left fashion. In the single-objective network simplex [2], each BFS is represented by a tree given by a set of basic arcs with flow $x_{ij} > 0$, since the variables in the minimum cost flow formulation of the shortest path problem have no upper bounds; all nontree (non-basic) arcs are at their lower bounds and have a flow of $x_{ij} = 0$. Let $L^t = \{(i, j) \in E: (i, j)\ is\ non-basic\ in\ BFS\ xt\ with\ xt{=}0.$

The efficient frontier is built in a right-to-left fashion, using network simplex algorithm for the single criterion optimization. Starting with lexicographical minimum for the second objective, the arc entering the basis is chosen upon a determination of the smallest ratio between reduced costs for the two criteria. The reduced costs of a given arc (i, j) are defined as follows:

$$\tilde{c}_{ij} = c_{ij} - \pi_i^c + \pi_j^c$$
$$\tilde{d}_{ij} = d_{ij} - \pi_i^d + \pi_j^d$$

In each iteration from the list St of arcs yielding the minimal ratio of the reduced costs one arc is chosen to enter the basic tree of the current efficient basic feasible flow x.

The algorithm starts with the extreme supported non-dominated point $y^{(0)} = (\tilde{y}_1, y_2^*)$ associated with the lexicographically minimum of $f_2(x)$, $(y_2^* = \min x \in X f2x,\ y1{=}\min x \in X{*} f1x,$ where $X{*}{=}x{*}{:}\ f2x{*}{=}y2{*}$ and ending with the minimum of $f1x$.

Our algorithm is based on the algorithms presented by Sedeño-Noda and González-Martín [14, 15] which is modified by Raith and Ehrgott [24]. These algorithms for solving the continuous biobjective minimum cost flow problem and the biobjective integer minimum cost flow problem.

Algorithm

1. Compute $y_1^{(0)} = (y_1^*, \tilde{y}_2) = lex\ \min_{x \in X} \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}$, and $y^{(0)} = (\tilde{y}_1, y_2^*) = lex\ \min_{x \in X} \begin{pmatrix} f_2(x) \\ f_1(x) \end{pmatrix}$
2. Let $x^{(0)}$ be the starting extreme supported efficient solution corresponding to $y_2^{(0)}$, $EX\_EFF = \{x^{(0)}\}$, and let C_{min} be the length of the shortest path from s to t corresponding to the spanning tree generated by solving $y_1^* = \min_{x \in X} f_1(x)$
3. Compute the reduced costs $\tilde{c}, \tilde{d}$ for $x^{(0)}$
4. Set $z^* = M$ (a large number)
5. Set $t = 1,\ k = 1$
6. Compute_Entering_Arcs($L^{t-1}, c, \pi^c, \pi^d, S^{t-1}, w^{t-1}$)
7. **While** $S^{t-1} \neq \emptyset$ **do**
8. **Begin**
9. $x^t = Compute\_New\_BFS(x^{t-1}, L^{t-1}, \pi^c, \pi^d, S^{t-1})$
10. Update $\tilde{c}, \tilde{d}$ and x^t
11. Compute_Entering_Arcs($L^{t-1}, c, \pi^c, \pi^d, S^{t-1}, w^{t-1}$)
12. **If** $w^t \neq w^k$ **then**
13. $w^k = w^t$ and $x^k = x^t$
14. $EX\_EFF = EX\_EFF \cup \{x^k\}$
15. Identify the unique directed path P_k in the feasible spanning tree of shortest paths T_k from s to t
16. Compute $z(P_k) = \frac{C(P_k)}{R(P_k)}$
17. **If** $z(P_k) \leq z^*$, **then** set $z^* = z(P_k)$ and $P^* = P_k$
18. **If** $h_k(C_{min}) \geq z^*$, **then** P^* is an optimum path with z^* as the objective function value. Go to 25.
19. **end if**
20. **end if**
21. $k = k + 1$
22. **end if**
23. $t = t + 1$
24. **end while**
25. $z(P^*) = \frac{C(P^*)}{R(P^*)}$ is the optimal solution

Procedure 1 Compute_Entering_Arcs $\left(L^t,\ \tilde{c},\ \tilde{d},\ \pi^c,\ \pi^d,\ S^t,\ w^t\right)$

1. **Begin**
2. $\tilde{c}_{ij} = c_{ij} - \pi_i^c + \pi_j^c$
3. $\tilde{d}_{ij} = d_{ij} - \pi_i^d + \pi_j^d$
4. $S^t = \emptyset$
5. Set $w^t = \left\{\frac{\tilde{d}_{ij}}{\tilde{c}_{ij}}: \tilde{d}_{ij} < 0\ and\ \tilde{c}_{ij} > 0,\ \forall (i,j) \in L^t\right\}$
6. Let $S^t \subseteq L^t$ be the set of non-basics arcs for which $\min w^t$ is attained
7. **end**

Procedure 2 Compute_New_BFS $(x^t,\ L^t,\ \pi^c,\ \pi^d,\ Pred,\ Depth,\ Thread,\ S^t)$

1. **While** $S^t \neq \emptyset$ **do**
2. Let (i,j) be the first arc in S^t; set $S^t := S^t - (i,j)$
3. **If** $\tilde{d}_{ij} < 0,\ \tilde{c}_{ij} > 0$ and $(i,j) \in L^t$ **then**
4. Perform simplex-pivot with entering arc (i,j)
5. Update $x^t,\ L^t,\ \pi^c,\ \pi^d,\ Pred,\ Depth,\ Thread,\ S^t$
6. **end if**
7. **end while**

The Compute_Entering_Arcs procedure calculates the set of arcs S^t those arcs that do not fulfill the optimality conditions with respect to the second objective: These make up the sequence of pivots to reach the adjacent extreme ND point in the objective space. One of the candidate arcs $(i,j) \in S^t$ is removed from S^t and enters the basis. By performing a simplex-pivot with entering arc (i,j), $i.e.$, introducing the arc (i,j) into the basis and removing the leaving arc $(pred(j),j)$ from the basis, the reduced costs may change. The reduced costs of all arcs remaining in S^t are updated according to the BFS obtained by pivoting (i,j) into x^t. As long as there are arcs remaining in S^t with $\tilde{d}_{ij} < 0$ and $\tilde{c}_{ij} > 0, \forall(i,j) \in L^t$.

The Compute_new_BFS procedure carries out these pivots updating the spanning tree structure. The next BFS x^{t+1} might de.ne an extreme ND point $\left(f_1(x^{t+1}), f_2(x^{t+1})\right) \in conv(Y)$. Denote by x^k the last extreme efficient solution that was found so far. If for the new minimal ratio w^{t+1} we have $w^{t+1} \neq w^t$, then x^{t+1} corresponds to an extreme efficient solution. On the other hand, if $w^{t+1} = w^t$ then x^{t+1} is not extreme, $i.e.$, $\left(f_1(x^{t+1}), f_2(x^{t+1})\right)$ corresponds to a supported non-extreme ND point.

It may be pointed out that every iteration in the above algorithm does not yield a new efficient path. The pivot operations may change the tree of shortest paths, but may not change the path from s to t. It can be easily seen that P_k from s to t changes if and only if the entering arc is incident on a node belonging to P_k. Also, to obtain the first efficient path, the algorithm selects arcs with $\tilde{d}_{ij} = 0$ if they exist, and performs the pivot operation.

In case of all edge failure rates are equal, the extreme supported ND points in the objective space on $G(V,E)$ are determined as follows

Step 1: Generate $\breve{G}(V,E)$ by finding "artificial" distances $d_{ij}^A = \lambda d_{ij}$ for each edge $(i,j) \in E$.
Step 2: Find the set of extreme supported non-dominated points in the objective space on $\breve{G}(V,E)$.
Step 3: Calculate the set of extreme supported non-dominated points in the objective space on $G(V,E)$ as follows $\left(C(P) = \sum_{(i,j)\in P} c_{ij}, D(P) = \sum_{(i,j)\in E} \frac{d_{ij}}{\lambda}\right)$.

Theorem 7 In the worst case, the algorithm generates the complete set of extreme efficient solutions of BSP.
Proof See Theorem 1 in [24]■

5 Numerical Example

The following example problem is provided to demonstrate the procedure presented in the previous section. The example problem network consists of 10 nodes and 21 directed arcs, Nodes 1 and 10, natively, are the origin and destination nodes. The arc costs c_{ij}, distances d_{ij}, artificial distances $\lambda_{ij} d_{ij}$ and reliability p_{ij} are given in Table 3.

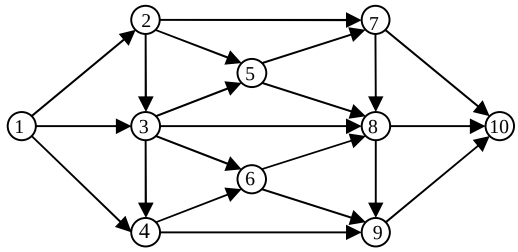

Figure 2: Example network consisting of 10 nodes and 21 directed edges

The example has seven efficient extreme supported points in the decision space, the images of these points corresponds to extreme supported non-dominated points in the objective space.

The algorithm begins by generating the lexicographical solutions $y_1^{(0)}$ and $y_2^{(0)}$ for BSP problem. This can be done by making use of the network simplex algorithm, solving the network parametric problem $\min_{x\in X} \lambda f_1(x) + (1-\lambda) f_2(x)$ with $\lambda = 1$ and $\lambda = 0$, respectively. The Compute_Entering_Arcs procedure computes the arcs that do not fulfill the optimality conditions with respect to the second objective. These arcs make a sequence of pivots to reach an adjacent supported extreme non-dominated point in the objective space. The Compute_New_BFS procedure updates the spanning tree structure, the tree indices and the potentials of the nodes with respect to the two objectives.

If $h_{k+1}(C_{min}) \geq z^*$, then P^* is an optimum path with z^* as the objective function value and the algorithm terminates. When $C(P), D(P) = (205, 230)$, $z(P) = \frac{C(P)}{e^{-D(P)}} = \frac{205}{e^{-230}} = 1.583 \times 10^{102}$. When the algorithm reaches the adjacent point $(70, 245)$, the objective value is $z(P) = \frac{170}{e^{-245}} = 4.2914 \times 10^{108}$.

Table 4 presents the spanning trees representing the seven supported extreme efficient points in the decision space. It also gives the objective values of $\left(f_1(x), f_2(x)\right)$ and the corresponding shortest path lengths from the source to destination $\left(C(P), D(P)\right)$. Figure 3 presents the set of all extreme supported ND points in the objective space.

Table 3: Arcs costs, distances and artificial distances for the network example

(i,j)	c_{ij}	d_{ij}	$p_{ij} = e^{-d_{ij}}$	λ	$p_{ij} = e^{-\lambda d_{ij}}$	λ_{ij}	$\lambda_{ij}d_{ij}$	$p_{ij} = e^{-\lambda_{ij}d_{ij}}$
$(1,2)$	**10**	**60**	8.7565×10^{-27}	0.006	0.69768	0.011921	0.71526	0.48906
$(1,3)$	**25**	**80**	1.8049×10^{-35}	0.006	0.61878	0.006412	0.51296	0.59872
$(1,4)$	**20**	**75**	2.6786×10^{-33}	0.006	0.63763	0.008084	0.6063	0.54536
$(2,3)$	**5**	**45**	2.8625×10^{-20}	0.006	0.76338	0.021228	0.95526	0.38471
$(2,5)$	**75**	**30**	9.3576×10^{-14}	0.006	0.83527	0.037815	1.1345	0.32160
$(2,7)$	**95**	**15**	3.059×10^{-7}	0.006	0.91393	0.085012	1.2752	0.27938
$(3,4)$	**65**	**10**	4.5400×10^{-5}	0.006	0.94176	0.138859	1. 3886	0.24943
$(3,5)$	**90**	**20**	2.0612×10^{-9}	0.006	0.88692	0.059195	1. 1839	0.30608
$(3,6)$	**75**	**15**	3.059×10^{-7}	0.006	0.91393	0.084184	1. 2628	0.28287
$(3,8)$	**60**	**85**	1.2161×10^{-37}	0.006	0.60050	0.005211	0.44294	0.64215
$(4,6)$	**40**	**120**	7.6676×10^{-53}	0.006	0.48675	0.001885	0.2262	0.79756
$(4,9)$	**35**	**110**	1.6889×10^{-48}	0.006	0.51685	0.002582	0.28402	0.75275
$(5,7)$	**25**	**160**	3.2575×10^{-70}	0.006	0.38289	0.000684	0.10944	0.89634
$(5,8)$	**145**	**50**	1.9287×10^{-22}	0.006	0.74082	0.018245	0.91225	0.40162
$(6,8)$	**130**	**80**	1.8049×10^{-35}	0.006	0.61878	0.006201	0.49608	0.60891
$(6,9)$	**55**	**140**	1.5804×10^{-61}	0.006	0.43171	0.000925	0.1295	0.87853
$(7,8)$	**140**	**35**	6.3051×10^{-16}	0.006	0.81058	0.028206	0.98721	0.37261
$(7,10)$	**65**	**170**	1.4789×10^{-74}	0.006	0.36059	0.000562	0.09554	0.90888
$(8,9)$	**10**	**60**	8.7565×10^{-27}	0.006	0.69768	0.011962	0.71772	0.48786
$(8,10)$	**70**	**185**	4.5240×10^{-81}	0.006	0.32956	0.000452	0.08362	0.91978
$(9,10)$	**150**	**45**	2.8625×10^{-20}	0.006	0.76338	0.021224	0.95508	0.38478

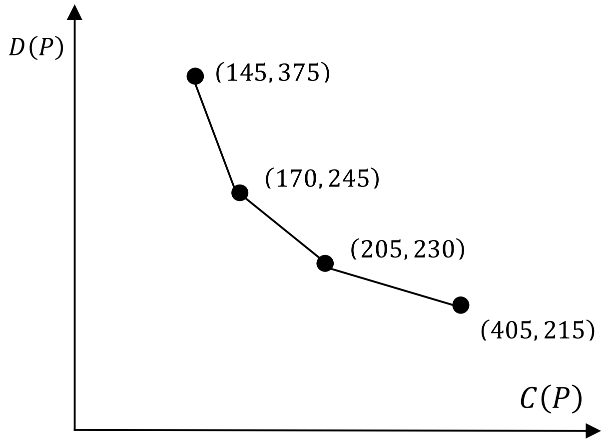

Figure 3: The set of extreme supported non-dominated points of min(C(P), D(P))

In case of considering the operational probability of each edge as $p_{ij} = e^{-\lambda_{ij}d_{ij}}$, two cases are considered:

Case 1: All failure rates are equal. Let $\lambda_{ij} = \lambda$. Then $\breve{G}(V,E)$ is a scaled version of $G(V,E)$, $i.e.$, the two networks share V and E and each edge in $\breve{G}(V,E)$ is λ times the original distance associated with each edge in $G(V,E)$, $d_{ij}^{A} = \lambda d_{ij}$. In this case, we get the same optimal solution as the case $p_{ij} = e^{-d_{ij}}$.

Case 2: Failure rates are not equal. In this case, we may get different optimal paths, $i.e.$, different extreme supported ND points for the BSP from that of operational probabilities $p_{ij} = e^{-d_{ij}}$. Table 3 gives the data for this case.

Table 4: The optimal shortest path spanning trees corresponding to the efficient extreme solutions in the decision space

	x^1	x^2	x^3	x^4	x^5	x^6	x^7
$(1,2)$	√	√	√	√	√	√	√
$(1,3)$	√	√	√	√	√		
$(1,4)$	√	√	√	√	√	√	√
$(2,3)$						√	√
$(2,5)$	√	√	√	√	√	√	√
$(2,7)$	√	√	√	√	√	√	√
$(3,4)$							
$(3,5)$							
$(3,6)$	√	√	√	√			
$(3,8)$			√	√	√	√	√
$(4,6)$					√	√	√
$(4,9)$		√	√	√	√	√	√
$(5,7)$							
$(5,8)$							
$(6,8)$							
$(6,9)$							
$(7,8)$	√	√					
$(7,10)$				√	√	√	
$(8,9)$	√						
$(8,10)$							√
$(9,10)$	√	√	√				
$f_1(x)$	1250	850	690	655	615	595	570
$f_2(x)$	970	1000	1055	1070	1170	1220	1350
$C(P)$	405	205	205	170	170	170	145
$D(P)$	215	230	230	245	245	245	375

6 Conclusions

In this paper, we considered the minimum cost-reliability ratio path Problem. The optimal solution of the MCRRPP corresponds to a supported extreme non-dominated point in the objective space of a biobjective shortest path. We used only phase 1 in the two phase method for BSP to generate the set of supported extreme non-dominated points in the objective space. We used the termination criterion presented by Ahuja [1] when the optimum solution is reached. In the worst case, starting with the supported extreme non-dominated point $(\tilde{y}_1, y_2^*) = lex\ \min_{x\in X}\begin{pmatrix} f_2(x) \\ f_1(x) \end{pmatrix}$, we will reach the point $(y_1^*, \tilde{y}_2) = lex\ \min_{x\in X}\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}$. In each simplex iteration, the basic entering arc is chosen to be an arc with the least ratio between improvement of $f_2(x)$ and the deterioration of $f_1(x)$, both expressed through reduced costs. Whenever, the shortest path from s to t changes, another non-dominated point is found.

In case of using the logarithmic transformation between operational probability and edge length, which was proposed by Melachrinoudis and Helander [19], to calculate for each edge the "artificial" edge length $d_{ij}^A = \lambda_{ij} d_{ij}$ and to define a new network $\breve{G}(V,E)$ with the same sets of edge attributes costs, c_{ij} and operational probabilities, $p_{ij} = e^{-d_{ij}^A}$, but its edge distances are $d_{ij}^A, (i,j) \in E$. The optimal solution of (8) corresponds to a supported extreme non-dominated point of (9).

An area of future research is the generation of test instances and tests the proposed algorithm. Also, a comparison of different solution strategies for the BSP and investigate their performance on different types of networks with the algorithm presented in this paper should be developed in the future.

Corresponding author
Abdallah W. Aboutahoun
Department of Mathematics, Faculty of Science, Alexandria University, Alexandria, Egypt
tahoun44@yahoo.com

References

1. Ahuja, P.K., (1988). Minimum cost-reliability ratio path problem, Computers & Operations Research, 15(1): 83-89
2. Ahuja, P.K., Magnanti, T.L. and Orlin, J.B. (1993). Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, New Jersey.

3. Aneja, Y.P. and Nair, K.P.K. (1984). Ratio dynamic programs. Operations Research Letters, 3:167-172.
4. Brumbaugh-Smith, J. and Shier, D. (1989). An empirical investigation of some bicriterion-shortest path algorithms. European Journal of Operational Research, 43:216-224.
5. Carlyle, W.M. and Wood, R.K., (2005). Near-shortest and k-shortest simple paths. Networks, 46(2):98-109.
6. Chandrasekaran, R. (1977). Minimum ratio spanning trees. Networks, 7: 335-342.
7. Chandrasekaran, R., Aneja, Y.P. and Nair, K.P.K., (1981) Minimum cost-reliability ratio spanning tree. In Studies on Graphs and Discrete Programming (Edited by P. Hansen). North-Holland, Amsterdam: 53-60.
8. Clímaco, J.C.N. and Martins, E.Q.V., (1982). A bicriterion shortest path problem. European Journal of Operational Research, 11:399-404.
9. Guerriero, F. and Musmanno, R. (2001). Label correcting methods to solve multicriteria shortest path problems. Journal of Optimization Theory and Applications, 111(3): 589.613.
10. Hansen, P. (1980). Bicriterion path problems. In: Multiple criteria decision making: theory and applications. Heidelberg:Springer: 109-127.
11. Martins, E.Q.V. (1984). On a multicriteria shortest path problem. European Journal of Operational Research, 16: 236-245.
12. Martins, E.Q.V., (1984). An algorithm to determine a path with minimal cost/capacity ratio, Discrete Applied Mathematics, 8:189-194
13. Eusébio, A. and Figueira, J.R., (2009). Finding non-dominated solutions in bi-objective integer network flow problems, Computers & Operations Research, 36:2554-2564.
14. Sedeño-Noda, A. and González-Martín, C., (2000). The biobjective minimum cost flow problem. European Journal of Operational Research, 124: 591-600.
15. Sedeño-Noda, A. and González-Martín, C., (2001). An algorithm for the biobjective integer minimum cost flow problem, Computers & Operations Research, 28:139-156.
16. Mote, J., Murthy, I. and Olson, D.L., (1991). A parametric approach to solving bicriterion shortest path problems, European Journal of Operational Research, 53:81-92.
17. Santiváñeza, J., Melachrinoudis, E. and Helander, M.E., (2009). Network location of a reliable center using the most reliable route policy, Computers & Operations Research, 36:1437-1460
18. Hamacher, H.W., Pedersen, C.R. and Ruzika, S., (2007). Multiple objective minimum cost flow problems: a review, European Journal of Operational Research, 176:1404-1422.
19. Melachrinoudis, E. and Helander, M.E., (1996). A single facility location problem on a tree with unreliable edges. Networks, 27:219-237.
20. Lee, H. and Pulat, P.S., (1993). Bicriteria network flow problems: integer case. European Journal of Operational Research, 66:148-157.
21. Pulat, P.S., Huarng, F. and Lee, H. (1992). Efficient solutions for the bicriteria network flow problem, European Journal of Operational Research, 19(7):649-655.
22. Ehrgott, M. and Gandibleux, X., (2009). Bounds sets for biobjective combinatorial optimization problems, Computers & Operations Research, 36:1299-1331
23. Ehrgott, M. and Skriver, A.J.V., (2007). Solving biobjective combinatorial max-ordering problems by ranking methods and a two-phases approach, European Journal of Operational Research, 34:2674-2694
24. Raith, A. and Ehrgott, M., (2009). A two-phase algorithm for the biobjective integer minimum cost flow problem, Computers & Operations Research, 36:1945-1954
25. Raith, A. and Ehrgott, M., (2009). A comparison of solution strategies for biobjective shortest path problems, Computers & Operations Research, 36:1299-1331
26. Skriver, A.J.V., (2000). A classification of bicriterion shortest path (BSP) algorithms, Asia Pacific Journal of Operational Research, 17:199-212.
27. Skriver, A.J.V., and Andersen, K.A. (2000). A label correcting approach for solving bicriterion shortest-path problems, Computers & Operations Research, 27:507-524
28. Przybylski, A., Gandibleux, X. and Ehrgott, M. (2008). Two phase algorithms for the biobjective assignment problem, European Journal of Operational Research, 185:509-533
29. Steuer, R.E., (1986). Multiple Criteria Optimization: Theory, Computation, and Applications, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York.

7/30/2012