Dealing with uncertainty in network optimisation

Marta Pascoal

Departamento de Matemática, Universidade de Coimbra

PhD program seminar, November 17, 2010

Contents

Network optimisation is a branch of optimisation the problems of which are modelled over a valued graph, that is, a network. We briefly present introductory concepts in this field and discuss connections between network optimisation and related subjects. Classical methods applied in this area assume that deterministic information is associated with the graph structure, however, in real problems these parameters are often incomplete, inaccurate or stochastic. We describe some of the possible formulations of network optimisation problems when uncertainty is present.

Discrete mathematics

- Euler circuits: In order to minimise cost, how should garbage collection routes be designed for 100 000 households?
- Traveling salesman problem: In a wharehouse for documentation storage a list of tasks is issued daily. The tasks can be locations for new documents to be dropped, locations of documents to be picked up, and a set of documents to be moved from one location to another. How can they compute a good delivery plan for each day?
- Telecommunications routing: How do telecommunication companies determine how to route millions of long-distance calls using the land lines, repeater amplifiers, and satellite terminals?

The modelling can be done using graphs, where the nodes represent objects and the arcs represent relationships between them. Matrices are a data structure used to store the graph, in order to take advantage of the power of computers to search for a solution.
Many algorithms can be developed for finding solutions for the problems.

Discrete mathematics

The nature of the problems associated with the afore mentioned applications of Discrete Mathematics involves questions such as:

- Existence of solutions
- Number of solutions
- Algorithms for generating solutions
- Optimisation

Optimisation

Optimisation models intend to solve a problem in the "best" way.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be an objective function and $x \in \mathbb{R}^n$ be the decision variables:

- Unconstrained problem,

$$\begin{array}{ll} \text{minimise} & f(x) \\ \text{subject to} & x \in \mathbb{R}^n \end{array},$$

  or simply, $\min f(x)$.

- Constrained problem,

$$\begin{array}{ll} \text{minimise} & f(x) \\ \text{subject to} & x \in S \end{array},$$

  where $S = \{x \in \mathbb{R} : \; g_i(x) = 0, \; i \in E \text{ and } g_i(x) \geq 0, \; i \in I\}$ is the feasible region, or set of constraints.

Optimisation

Nonlinear optimisation (aka nonlinear programming):

- any of the functions (objective or constraints) may be nonlinear.

Linear optimisation (or linear programming):

- all functions (objective and constraints) are linear.

Integer programming

- some of the decision variables are integer.

Network optimisation (or network programming)

- at least part of the constraints define a graph.

Given

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in S \end{array}$$

- A feasible solution is a point that satisfies the constraints, ie, any $x \in S$.
- An optimal solution is $x^* \in S$ such that $f(x^*) \leq f(x)$, for all $x \in S$ (and, in this case, a global minimiser).

Nonlinear optimisation

Assuming f is differentiable, then the optimal solution is a stationary point of f, that is, a point x_* such that

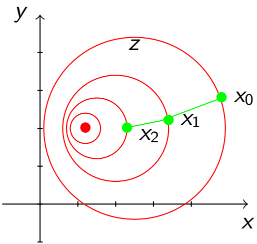
$$\nabla f(x_*) = 0,$$

and thus we apply iterative methods to approximately solve this system of nonlinear equations.

Nonlinear optimisation

Algorithm:

Take an initial guess of the solution, x_0
$k \leftarrow 0$
While x_k *is not optimal Do*
$x_{k+1} \leftarrow$ *new estimate of the solution, "better than"* x_k
$k \leftarrow k + 1$



Questions:

- Does the method converge? How fast? ...

Theorem (Sufficiency conditions)

Let $\bar{x}$ be a feasible solution of $\min\{f(x) : Ax \geq 0\}$. *Suppose there exists a vector $\bar{\lambda}$ such that*

- $\nabla f(\bar{x}) = A^T \bar{\lambda}$, $\bar{\lambda} \geq 0$, *"strict complementarity holds", and*
- $Z^T \nabla^2 f(\bar{x}) Z$ *is positive definite,*

where Z_+ is a basis for the null space of a "certain" sub-matrix of A. Then $\bar{x}$ is a strict local minimiser.

Theorem (Sufficiency conditions)

Let $\bar{x}$ be a feasible solution of $\min\{f(x) : g(x) \geq 0\}$. *Suppose there exists a vector $\bar{\lambda}$ such that*

- $\nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}) = 0$, $\bar{\lambda} \geq 0$, $\bar{\lambda}^T g(\bar{x}) = 0$,
- $Z_+(\bar{x})^T \nabla^2_{xx} \mathcal{L}(\bar{x}, \bar{\lambda}) Z_+(\bar{x})$ *is positive definite,*

where Z_+ is a basis for the null space of the Jacobian matrix of "some of the constraints". Then $\bar{x}$ is a strict local minimiser.

Linear optimisation

The feasible points of

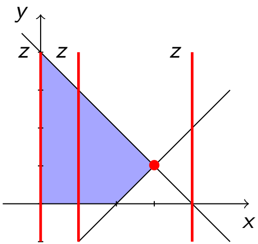
$$\begin{array}{ll} \min & c^T x \\ \text{s. t.} & Ax = b \\ & x \geq 0 \end{array}$$

can be rewritten as basic solutions, $x = \begin{pmatrix} B^{-1}b & 0 \end{pmatrix}^T$, associated with each partition $A = [B \mid N]$ and B is not singular.
Such solutions correspond to vertices of the polyhedron defined by $Ax = b$, one of them being an optimal solution (when the problem is finite).

Linear optimisation

Simplex method:

> *Take an initial basic solution,* x_0
> $k \leftarrow 0$
> *While* x_k *is not optimal Do*
> $x_{k+1} \leftarrow$ *basic solution, adjacent to* x_k *and "better than"* x_k
> $k \leftarrow k+1$



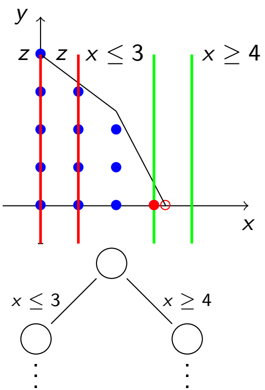
Questions:

- How to deal with degeneracy and "cycling"?

Integer programming

Branch-and-bound method:

Solve the LP relaxation of the problem (P)
`While` *the solution is fractional* `Do`
Let $x_i = \epsilon \notin \mathbb{Z}$
Consider (P) and $x_i \geq \lfloor \epsilon \rfloor$
Consider (P) and $x_i \geq \lceil \epsilon \rceil$



Questions:

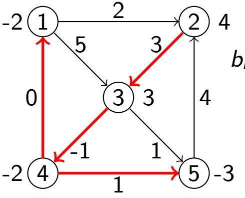
- Relies on having good lower bounds on the integer solutions.
- How fast is the method and up to what size of problems can it solve? ...

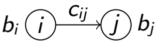
Network optimisation

A network is a graph the arcs or nodes of which are valuated.

If all functions are linear the network simplex method can be applied, however these problems are highly degenerate.
Methods that use the solutions' graph structure are usually more efficient.

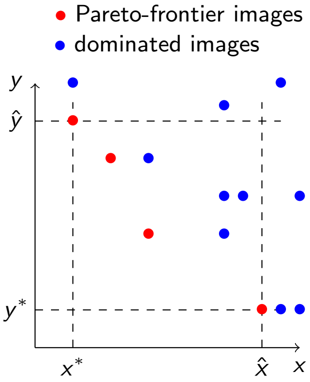
- Network flow problem
- Transportation problems
- Assignment problems
- Minimum spanning tree problem
- Maximum flow problem
- Shortest path problem, . . .





Multi-criteria optimisation

The objective function is $f : \mathbb{R}^n \to \mathbb{R}^r$, thus usually the functions f_i are conflicting and there is no "optimal solution".



Instead, a partial order is defined in order to compare solutions.

- Given $a, b \in S$, $a \prec b$ *iff* $f(a) \leq f(b)$ and $a \neq b$.
- If $a \prec b$ it is said that a dominates b.
- When there is no $b \in S$ that dominates a this is called a non-dominated solution.

Uncertainty

Classical formulations assume deterministic and fixed parameters. However, in a more precise representation of reality those parameters may be subjective, or may be subject to sources of uncertainty, imprecision and inaccurate determination,

- parameters may vary dynamically, or may result from a measuring instrument or from a statistic measure (which usually involve some imprecision),
- knowing a parameter accurately can be costly,
- a decision group may not agree on the values that each parameter should take, . . .

Some possible models

- "Approximate" model. The "most likely" value is assigned to each parameter, it is complemented by sensitivity analysis.
- Probabilistic model. Requires probability distributions (and correlation among parameters) for the unknown values and leads to a solution based on its expected cost, and perhaps its variance.

Both prematurely focus on a single solution, which is as arbitrary as the "most likely" values or the probabilities chosen, and the DMs tend to disregard other potential solutions.

- Robust model. We admit multiple instances of the model, each defined by an acceptable combination of values for the parameters. The "robustness analysis" approach, where we focus on identifying conclusions that are valid for every instance of the model is then appropriate.

The relative robust path problem

Models,

- scenario model, a set of alternative graphs are considered at the same time, and
- interval data model, where an interval of possible values is associated with each arc.

Given the interval data model,

- a relative robust shortest path (or simply robust shortest path) is a path which minimises the maximum deviation from the optimal shortest path over all realisations of arc costs.

Let $(\mathcal{N}, \mathcal{A})$ be a directed network where:

- $\mathcal{N}$ denotes the set of n nodes, and $\mathcal{A}$ denotes the set of m arcs,
- arc (i, j) is associated with the interval $[l_{ij}, u_{ij}]$.

Let:

- a path from s to t be a sequence $p = \langle v_1, v_2, \ldots, v_{\ell(p)} \rangle$, where $s = v_1$, $t = v_{\ell(p)}$, and $(v_k, v_{k+1}) \in \mathcal{A}$, $k = 1, \ldots, \ell(p) - 1$,
- $\mathcal{P}$ be the set of paths in $(\mathcal{N}, \mathcal{A})$,
- a scenario r be a snapshot of the arc costs, i.e., $c_{ij}^r \in [l_{ij}, u_{ij}]$, for any $(i, j) \in \mathcal{A}$.

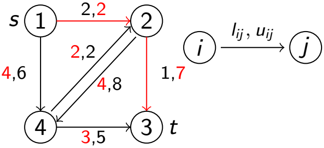
The robust deviation for path p in a scenario r is the difference between the cost of p in r and the cost of the shortest path in scenario r.

A path is a robust shortest path if it has the smallest (among all paths) maximum robust deviation (among all scenarios).

Note

Given $p \in \mathcal{P}$, the scenario r which maximises the robust deviation for p is the one where $c^r_{ij} = u_{ij}$, for $(i,j) \in p$, and $c^r_{ij} = l_{ij}$ for $(i,j) \notin p$.

The scenario derived from path p is called the scenario induced by p.



Mixed integer programming formulation

Let us consider the variables,

- $y_{ij} = \begin{cases} 1, & (i,j) \text{ is on the robust shortest path from } s \text{ to } t \\ 0, & \text{otherwise} \end{cases}$
- x_i, the cost of the shortest path from s to i in the scenario induced by the robust shortest path (defined by y).

Mixed integer programming formulation

$$\min \quad \sum_{(i,j)\in\mathcal{A}} u_{ij}y_{ij} - x_i \tag{1}$$

$$\text{s. t.} \quad x_j \leq x_i + l_{ij} + (u_{ij} - l_{ij})y_{ij}, \ \forall (i,j) \in \mathcal{A} \tag{2}$$

$$\sum_{(j,k)\in\mathcal{A}} y_{jk} - \sum_{(i,j)\in\mathcal{A}} y_{ij} = \begin{cases} 1, & j = s \\ 0, & j \in \mathcal{N} - \{s,t\} \\ -1, & j = t \end{cases} \tag{3}$$

$$x_s = 0 \tag{4}$$

$$y_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{A} \tag{5}$$

$$x_j \geq 0, \ \forall j \in \mathcal{N} \tag{6}$$

- Constraints (2) link the variables x and y.
- The remaining are the usual constraints for the classic shortest path problem formulation.

Notation

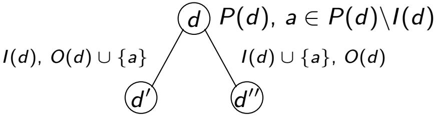
- $T(d)$: search-tree nodes contained in the subtree rooted in search-tree node d,
- $RC(p)$: robustness cost of path p.
- Let r denote the scenario where $c_{ij}^r = u_{ij}$, $\forall (i,j) \in B$, and $c_{ij}^r = l_{ij}$, $\forall (i,j) \in \mathcal{A} \setminus B$.
  $SP(B, I, O)$: is the cost of p, the path with the minimum cost in scenario r among those which include the arcs in set I and do not contain any arc from the set O; or $+\infty$ if such a path does not exist.

Structure of the search-tree node d

- $I(d)$: set of arcs which must appear in all of the paths associated with the nodes of $T(d)$;
- $O(d)$: set of arcs forbidden for all of the paths associated with the nodes of $T(d)$;
- $P(d)$: path associated with the search-tree node d, with the minimum cost in scenario u which contains the arcs in $I(d)$ and does not include the arcs in $O(d)$;
- $lb(d)$: lower bound for the robustness cost of the paths associated with the search-tree nodes of $T(d)$.

Branching strategy

- The root, r, of the search-tree is such that $I(r) = O(r) = \emptyset$. Initially r is the only node of the search-tree.
- At each iteration,
  $d \leftarrow$ the not yet scanned node with the smallest value of $lb(d)$,
  $a \leftarrow$ the first arc in $P(d) \backslash I(d)$.
- If $P(d) \neq I(d)$, the search-tree nodes, d', d'', are created,



If not dominated, d', d'' are inserted into the search-tree.

Property

Each set $I(d)$ contains a chain of arcs which form a sub-path starting from s.

Lower bound $lb(d) = SP(\mathcal{A}, I(d), O(d)) - SP(\mathcal{A} \setminus O(d), \emptyset, \emptyset)$

Lemma

$SP(\mathcal{A}, I(d), O(d)) \leq SP(\mathcal{A}, I(f), O(f))$, for all $f \in T(d)$.

$I(d) \subseteq I(f), \forall f \in T(D)$, and $O(d) \subseteq O(f), \forall f \in T(D)$, thus the path $P(f)$ is subject to more constraints than $P(d)$.

Lemma

$SP(\mathcal{A} \setminus O(d), \emptyset, \emptyset) \geq SP(P(f), \emptyset, \emptyset)$, for all $f \in T(d)$.

Similar.

Theorem

$SP(\mathcal{A}, I(d), O(d)) - SP(\mathcal{A} \setminus O(d), \emptyset, \emptyset) \leq RC(P(f))$, for all $f \in T(d)$.

It results from $RC(P(f)) = SP(\mathcal{A}, I(f), O(f)) - SP(P(f), \emptyset, \emptyset)$, together with the lemmas above.

Reduction rules

In order to speed up the evaluation of $SP(\mathcal{A}, I(d), O(d))$ it is enough to compute the shortest path from the last node of the chain starting in s to t, according to $I(d), O(d)$, and to add it the cost of the arcs in $I(d)$.

Theorem

Given a search-tree node d, if $(i,j) \in I(d)$ then $\forall (i,k) \in \mathcal{A} \setminus \{(i,j)\}$ and $\forall f \in T(d)$, $(i,k) \in P(f)$.

$(i,j) \in I(d) \subseteq P(f)$, *thus if* $(i,k) \neq (i,j)$ *then* $(i,j) \notin P(f)$.

Proposition (R1)

If $(i,j) \in I(d)$ then $\forall (i,k) \in \mathcal{A} \setminus \{(i,j)\}$, (i,k) can be inserted into $O(d)$.

Proposition (R2)

If $(i,j) \in I(d)$ then $\forall (k,j) \in \mathcal{A} \setminus \{(i,j)\}$, (k,j) can be inserted into $O(d)$.

B & B algorithm

$I(r) \leftarrow O(r) \leftarrow \emptyset$; $lb(r) \leftarrow 0$;
$S \leftarrow \{r\}$; $ub \leftarrow RC(P(r))$; $ubPath \leftarrow P(r)$;
`While` $S \neq \emptyset$ `Do`
  $d \leftarrow argmin\{lb(f): \ f \in S\}$; $S \leftarrow S \backslash \{d\}$;
  `If` $I(d) \neq P(d)$ `Then`
    $a \leftarrow$ first arc in $p(d)$ not contained in $I(d)$;
    $I(d') \leftarrow I(d)$; $O(d') \leftarrow O(d) \cup \{a\}$;
    `If` $RC(P(d')) < ub$ `Then`
      $ub \leftarrow RC(P(d'))$; $ubPath \leftarrow RC(P(d'))$;
      $S \leftarrow S \backslash \{f: \ f \in S$ and $lb(f) \geq ub\}$;
    $lb(d') \leftarrow SP(\mathcal{A}, I(d'), O(d')) - SP(\mathcal{A} \backslash O(d'), \emptyset, \emptyset)$;
    `If` $lb(d') < ub$ `Then` $S \leftarrow S \cup \{d'\}$;
    Apply the reductions rules R1 and R2;
    $I(d'') \leftarrow I(d) \cup \{a\}$; $O(d'') \leftarrow O(d)$;
    $lb(d'') \leftarrow SP(\mathcal{A}, I(d''), O(d'')) - SP(\mathcal{A} \backslash O(d''), \emptyset, \emptyset)$;
    `If` $lb(d'') < ub$ `Then` $S \leftarrow S \cup \{d''\}$;

The stochastic path problem

There are various types of path problems over random probabilistic networks, the nodes or arcs may be randomly connecting, or the values associated with each arc may be stochastic.

- determination of the probability distribution of the shortest path,
- maximisation of the expected value of a utility function,
- expected shortest path: minimisation of the path's expected value,
$$E\left(\sum_{(i,j)\in p} X_{ij}\right),$$
- most shortest path: maximisation of the probability that the optimal path length does not exceed a specified value, P_0,
$$P\left(\sum_{(i,j)\in p} X_{ij} \leq P_0\right),$$
- α-shortest path: path that satisfies some constraints with probability of at least α,
$$\min\left\{\bar{P} : P\left(\sum_{(i,j)\in p} X_{ij} \leq \bar{P}\right) \geq \alpha\right\}.$$

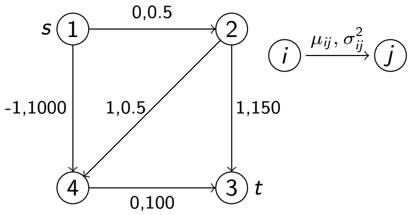
Let $(\mathcal{N}, \mathcal{A})$ be a directed network where:

- arc (i, j) is associated with:
  - the mean value, μ_{ij}, and
  - the variance, σ^2_{ij},

  of a real random variable (r.r.v.) X_{ij}, called the random cost of (i, j).

Let:

- the r.r.v. $X_p = \sum_{(i,j)\in p} X_{ij}$ is the random cost of p, $p \in \mathcal{P}$.



Assuming that $X_{ij} \sim N(\mu_{ij}, \sigma_{ij}^2)$ are independent, and that paths have no repeated nodes, by the stability of the normal distribution, $X_p \sim N\left(\mu_p, \sigma_p^2\right)$, where

$$\mu_p = \sum_{(i,j)\in p} \mu_{ij}, \;\; \sigma_p = \sum_{(i,j)\in p} \sigma_{ij}^2.$$

We consider a utility function $U : \mathcal{P} \rightarrow \mathbb{R}$, such that $U(p)$ depends on X_{ij}, for any $(i,j) \in p$.

In the stochastic shortest path problem we aim to

$$\max_{p\in\mathcal{P}} E(U(p)).$$

Let

$$U(X_p) = \begin{cases} a - bX_p, & X_p \leq d \\ 0, & X_p > d \end{cases}, \quad a, b, d > 0, \;\; a - bd \geq 0.$$

The density function of $p \in \mathcal{P}$ is

$$g_p(x) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left(-\frac{1}{2} \left(\frac{x - \mu_p}{\sigma_p} \right)^2 \right), \;\; x \in \mathbb{R},$$

thus

$$\begin{aligned} E(U(X_p)) &= \int_{\mathbb{R}} U(x) g_p(x) dx \\ &\cdots \\ &= (a - b\mu_p) G\left(\frac{d - \mu_p}{\sigma_p} \right) + b\sigma_p g\left(\frac{d - \mu_p}{\sigma_p} \right), \end{aligned}$$

with G the cumulative distribution function, and g the probability density function, of a standard normal r.r.v..

Thus, we aim to

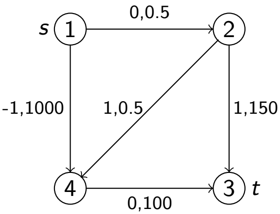
$$\max_{p \in \mathcal{P}} \left\{ (a - b\mu_p) G\left(\frac{d-\mu_p}{\sigma_p}\right) + b\sigma_p g\left(\frac{d-\mu_p}{\sigma_p}\right) \right\} \qquad (1)$$

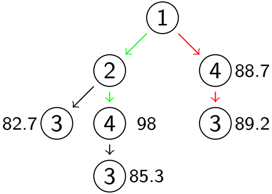
Principle of optimality

The optimal solution is formed by optimal subsolutions

Counter-example

This principle is not valid for (1).





This means that we cannot use a classic labeling method. Instead,

$$\frac{\partial E(U(X_p))}{\partial \mu_p} = -bG\left(\frac{d-\mu_p}{\sigma_p}\right) - \frac{a-bd}{\sigma_p} g\left(\frac{d-\mu_p}{\sigma_p}\right) \leq 0, \text{ always}$$
$$\frac{\partial E(U(X_p))}{\partial \sigma_p^2} = g\left(\frac{d-\mu_p}{\sigma_p}\right) \frac{(a-bd)\mu_p + b\sigma_p^2 - d(a-bd)}{2(\sigma_p^2)^{3/2}}$$

If a, b, d are such that

1. $\frac{\partial E(U(X_p))}{\partial \sigma_p^2} > 0$, then the optimal path satisfies

$$X_p \sim N\left(\min \sum_p \mu_{ij}, \max \sum_p \sigma_{ij}^2\right).$$

2. $\frac{\partial E(U(X_p))}{\partial \sigma_p^2} < 0$, then the optimal path satisfies

$$X_p \sim N\left(\min \sum_p \mu_{ij}, \min \sum_p \sigma_{ij}^2\right).$$

For case 1. (case 2. is similar) we have,

Theorem

The bicriteria optimisation problem

$$\max_{p \in P} f(p), \quad \text{with } f(p) = \left(\min \sum_{p} \mu_{ij}, \max \sum_{p} \sigma_{ij}^2\right)$$

satisfies the principle of optimality if there are no positive or negative cycles in $(\mathcal{N}, \mathcal{A})$.

> *By contradiction, assuming that a non-dominated path contains a subpath that is dominated by another.*

Thus,

1. the subproblem can be solved by a labeling algorithm,
2. for the sake of space and time efficiency, the algorithm should eliminate the subsolutions that are "dominated" by others.

Algorithm

Find all non-dominated solutions of $\left(\min \sum_p \mu_{ij}, \max \sum_p \sigma^2_{ij}\right)$

Find all non-dominated solutions of $\left(\min \sum_p \mu_{ij}, \min \sum_p \sigma^2_{ij}\right)$

Select the previously obtained paths p*, such that* X_p *maximises*

$$(a - b\mu_p)G\left(\frac{d-\mu_{ij}}{\sigma_p}\right) + b\sigma_p g\left(\frac{d-\mu_{ij}}{\sigma_p}\right)$$

with the general bicriteria path algorithm,

```
For all i ∈ N Do L(i) ← ∅
L(s) ← {(0,0)}; X ← {s}
While X ≠ ∅ Do
    Select node i in X; X ← X − {i}
    For all (i,j) ∈ A Do
        For all π_x ∈ L(i) Do
            If π_x + (μ_ij, σ²_ij) is not dominated in L(j) Then
                Insert new label, π_x + (μ_ij, σ²_ij), into L(j)
                Delete all labels in L(j) that became dominated
        If L(j) has changed Then X ← X ∪ {j}
```

Further details at Office 4.4 or at: www.mat.uc.pt/∼marta/.

Thanks for the attention.
Questions?

Brumbaugh-Smith, J. and Shier, D. (1989).
An empirical investigation of some bicriterion shortest path algorithms.
European Journal of Operational Research, 43:216–224.

Dias, L. and Clímaco, J. (2000).
Shortest path problems with partial information: Models and algorithms for detecting dominance.
European Journal of Operational Research, 121:16–31.

Ji, X. (2005).
Models and algorithm for stochastic shortest path problem.
Applied Mathematics and Computation, 170:503–514, 2005.

Montemanni, R. and Gambardella, L. (2004).
An exact algorithm for the robust shortest path problem with interval data.
Computers & Operational Research, 31:1667–1680.

Montemanni, R. and Gambardella, L. (2005).
The robust shortest path problem with interval data via Benders decomposition.
4OR – Quart Jour of the Belgian, French and Italian Op Res Soc, 3:315–328.

Montemanni, R., Gambardella, L., and Donati, V. (2004).
A branch and bound algorithm for the robust shortest path problem with interval data.
Operations Research Letters, 32:225–232.

Rasteiro, D. and Anjo, A. (2004).
Optimal paths in probabilistic networks.
Journal of Mathematical Sciences, 120:974–987.