Optimization Letters
https://doi.org/10.1007/s11590-021-01712-5

ORIGINAL PAPER

Shortest path network problems with stochastic arc weights

Jeremy D. Jordan[1] · **Stan Uryasev**[2]

Received: 9 December 2019 / Accepted: 9 February 2021

Abstract
This paper presents an approach to shortest path minimization for graphs with random weights of arcs. To deal with uncertainty we use the following risk measures: Probability of Exceedance (POE), Buffered Probability of Exceedance (bPOE), Value-at-Risk (VaR), and Conditional Value-at-Risk (CVaR). Minimization problems with POE and VaR objectives result in mixed integer linear problems (MILP) with two types of binary variables. The first type models path, and the second type calculates POE and VaR functions. Formulations with bPOE and CVaR objectives have only the first type binary variables. The bPOE and CVaR minimization problems have a smaller number of binary variables and therefore can be solved faster than problems with POE or VaR objectives. The paper suggested a heuristic algorithm for minimizing bPOE by solving several CVaR minimization problems. Case study (posted at web) numerically compares optimization times with considered risk functions.

Keywords Conditional value at risk · Buffered probability of exceedance · Shortest path · Stochastic arc costs

1 Introduction

The *shortest path problem* finds a path between two vertices (nodes) in a weighted graph, which minimizes the total weight of arcs (edges) in the path. Let $G = (V, A)$ be a graph with a set of nodes, V, and a set of arcs, A. We denote by $w(v_i, v_j)$ the

✉ Jeremy D. Jordan
jeremy.jordan@afit.edu

Stan Uryasev
stanislav.uryasev@stonybrook.edu

1 Air Force Institute of Technology, Wright-Patterson AFB, USA

2 Stony Brook University, Stony Brook, USA

weight of arc connecting node v_i and v_j. The weight of path $d = \{v_0, v_1, \dots, v_k\}$ is the sum of the weights of arcs in the path:

$$w(d) = \sum_{i=1}^{k} w(v_{i-1}, v_i).$$

Depending on an application, the weight may be travel time, cost, length etc.

The deterministic shortest path problem finds a path from a starting point to a final point with a minimal weight. There are many efficient algorithms for finding shortest paths: Dijkstra [1], Bellman-Ford [2, 3], Lawler [4], Floyd-Warshall [5, 6]. A "breadth-first search" algorithm [7, 8] is frequently used for unweighted graphs, that is, graphs where each arc has a unit weight. Paper [9, 10] considered constrained shortest-path problems. Maximum-flow network-interdiction problem was solved in [11].

In contrast to the deterministic case, weights of arcs in a stochastic weighted graph are considered to be random. The problem of determining the probability distribution of the shortest path length was studied in [12] and [13]. Murthy and Sarkar [14] considered the problem, where an optimal path maximizes the quadratic expected utility. Papers [14, 15] studied stochastic shortest path problems with different types of cost functions. Xiaoyu [16] considered the so called "expected shortest path", "α-shortest path", and the "most shortest path" problems. This paper presents three types of models: "expected value model", "chance-constrained programming", and "dependent-chance programming". These models were solved with a hybrid intelligent algorithm integrating stochastic simulation and genetic optimization. Paper [17] considered a network flow problem with uncertain arc failures. Paper [18] optimizes networks with chance constraints having stochastic right-hand sides (discrete distributions). Various variants of "robust shortest path problem" were considered in [19–27]. Paper [28] provides a survey of robust discrete optimization under discrete and interval uncertainty.

This paper considers shortest path optimization problem for stochastic graphs with the following objectives: Probability of Exceedance (POE), Buffered Probability of Exceedance (bPOE), Value at Risk (VaR), and Conditional Value-at-Risk (CVaR).

CVaR is a convex function (see, [29, 30]) and bPOE optimization (see, [31]) can be reduced to a convex problem. Therefore, CVaR and bPOE definitions do not involve binary variables. For stochastic shortest path problems with CVaR and bPOE objective, the number of binary variables is equal to the number of arcs in the graph. However, for problems with VaR and POE objectives, binary variables are needed to define VaR and POE (in addition to the binary variables corresponding to arcs). For stochastic problems with discrete scenarios the number of additional variables for defining VaR and POE is equal to the number of scenarios. For problems with a large number of scenarios, this leads to significant computational advantages of CVaR and bPOE formulations versus VaR and POE formulations. Optimization problems with CVaR and bPOE can be solved much faster than problems with VaR and POE.

This paper is focused on minimization problem statements with POE, bPOE, VaR, and CVaR functions. POE, VaR and CVaR functions are quite popular in optimization literature. It is straight forward to formulate shortest path problems with these functions. bPOE function is relatively recently introduced in the literature as an alternative to POE. Since bPOE is an inverse function of CVaR, solution of the problem with bPOE objective can be obtained by solving CVaR with some (unknown) value of confidence level. This paper suggests an iterative algorithm for finding this confidence level by solving several CVaR optimization problems.

We use Portfolio Safeguard (PSG), see [32], and Gurobi packages to solve minimization problems. PSG has precoded POE, bPOE, VaR, and CVaR functions; therefore, PSG optimization codes consist only from several lines. PSG generates a Gurobi code and calls Gurobi (as a sub-solver) for MILP optimization.

We have done a case study and posted it at web. We numerically compared performance of shortest path problems with POE, bPOE, VaR, and CVaR objectives. CVaR and bPOE minimization can be done significantly faster than POE and VaR minimization. Also, we suggested a new heuristic and found that bPOE minimization can be done with two–three CVaR minimization.

The paper is organized as follows. Section 2 introduces risk measures: POE, bPOE, VaR, and CVaR. In addition, it considers a "toy graph" explaining how shortest path problems with the considered risk measures are formulated. Section 3 provides mathematical problem statements for the shortest path problems. Section 4 conducts a case study and demonstrates that the suggested approaches are quite efficient. Finally, Sect. 5 concludes the paper.

2 Preliminaries

The following subsection introduces risk measures.

2.1 Risk measures

Let F_X be a cumulative distribution function of a random variable X modeling loss:

$$F_X(x) = P\{X \leq x\} \quad \text{for} \quad x \in (-\infty, \infty).$$

The random variable X can be alternatively described by a quantile function $q_\alpha(X)$, which is the inverse of the cumulative distribution:

$$F_X^{-1}(\alpha) = q_\alpha(X) = \inf \left\{ x : \alpha \leq F_X(x) \right\} \text{for} \alpha \in (0, 1).$$

For a given threshold $x \in (-\infty, \infty)$ the POE with threshold x equals:

$$p_x(X) = P(X > x) = 1 - F_X(x).$$

This probabilistic characteristic, $p_x(X)$, describes the upper tail of the distribution. In financial applications quantile is called Value-at-Risk (VaR). Both POE and VaR are "optimistic" risk measures which are based on the lower bound of outcomes in the tail. Both characteristics do not provide information about large losses, which may occur with low probability. CVaR and bPOE are conservative counterparts of VaR and POE risk measures. CVaR, denoted by $\bar{q}_\alpha(X)$, for a distribution which does not jump at the VaR equals the expected loss exceeding VaR, see [29],

$$\bar{q}_\alpha(X) = E\big[X : X \geq q_\alpha(X)\big].$$

These formulas hold for continuous random variables, whereas in the general case, when F_X has a jump at the VaR, $q_\alpha(X)$,the definition of CVaR is more complicated:

$$\bar{q}_\alpha(X) = \min_u\left(u + \frac{E[X-u]^+}{1-\alpha}\right),$$

where $[\cdot]^+ = \max\{\cdot, 0\}$. CVaR has many alternative names: expected shortfall, tail VaR, average VaR, and superquantile (as defined in [33, 34]). This paper uses CVaR and VaR terminology. CVaR has exceptional mathematical properties, compared to VaR. For example, VaR is difficult to optimize for discrete distributions, because it is non-convex, non-smooth, and has multiple local extrema. CVaR is an attractive measure of risk since it is convex in random variables, and therefore is much easier to solve. Moreover, for many cases it can be optimized with linear and convex optimization algorithms.

Paper [31] defined bPOE as one minus inverse of CVaR:

$$\bar{p}_x(X) = 1 - \bar{q}_x^{-1}(X) \text{ for } EX \leq x < \sup X.$$

bPOE can be calculated with the following formula for $EX \leq x < \sup X$ (see [31]):

$$\bar{p}_x(X) = \min_{a\geq 0} E[a(X-x)+1]^+.$$

If $x < EX$, then $\bar{p}_x(X) = 1$; if$\sup X < x$, then$\bar{p}_x(X) = 0$.

There are two possibilities to assign the value of bPOE at the point $x = \sup X$. The first possibility is to set

$$\bar{p}_x(X) = 0 \text{ for } x = \sup X,$$

which corresponds to so called *lower bPOE*. The second possibility is to set

$$\bar{p}_x(X) = \min_{a\geq 0} E[a(X-x)+1]^+ \text{ for } x = \sup X,$$

which corresponds to so called *upper bPOE*. This means that the lower and upper bPOE may differ only in one point $x = \sup X$, if $\sup X = \max X$ and F_X is discontinuous in point $\max X$.

Further in this paper we will use lower bPOE because constraint on lower bPOE can be equivalently replaced by constraint on CVaR in optimization problems. We will call lower bPOE for simplicity by bPOE (without mentioning that it is lower bPOE).

bPOE provides more information about the magnitude of outcomes in the tail of the distribution, compared to POE. By definition, POE equals the probability of exceeding a threshold, while upper bPOE equals the probability of the upper tail with the specified mean value. See, for instance [35], how bPOE was used for analysis of hurricanes.

2.2 "Toy" example problem

This section considers a small "toy" graph with only 3 paths, see Fig. 1. Costs (weights) of arcs are discretely distributed random values. To demonstrate how the solution may depend upon a risk function in objective of the optimization problem, we manually found paths with the lowest maximum, average, standard deviation, VaRs and CVaRs (with different confidence levels), and POE and bPOE (with different thresholds).

The set of vertices of the graph is denoted by $V = \{1, 2, 3, 4, 5\}$. Each arc is denoted by an ordered pair $\{i, j\}$ and A is the set of arcs,

$$A = \{\{1, 2\}, \{2, 4\}, \{4, 5\}, \{1, 5\}, \{1, 3\}, \{3, 5\}\}.$$

Each arc $\{i, j\}$ is characterized by the random cost, ξ_{ij}. The arc $\{1, 2\}$ has the following random cost,

$$\xi_{12} = \begin{cases} 1 \ \textit{with probability} = 0.25, \\ 2 \ \textit{with probability} = 0.5, \\ 3 \ \textit{with robability} = 0.25. \end{cases}$$

Similarly, random costs for other arcs are described in Fig. 1.We treat values of ξ_{ij} like random scenarios for arc $\{i, j\}$, and set of values ξ_{ij} for all ordered pair $\{i, j\}$ like random scenarios for stochastic weighted graph.We consider that probability distributions of weights for each arc are independent of each other. There are three paths from the origin vertex 1 to the final vertex 5: 1–2–4–5; 1–5; 1–3–5. We want to find a path with a minimal risk function (characteristic). Since the graph is very simple,

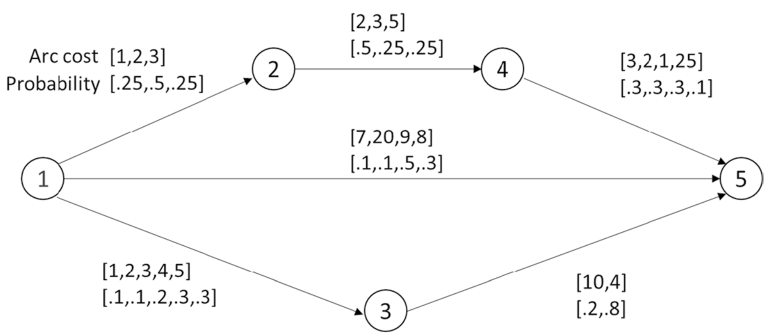

Fig. 1 Stochastic weighted graph

we can directly estimate value of risk function for each path and find the path with the minimal value of the function.

Further we present Cumulative Distribution Functions (CDFs) of costs for three paths, see Fig. 2.

For each path we calculated the following risk functions of random costs: maximum, average, standard deviation, VaRs (quantiles), CVaRs, (with different confidence levels), POEs and bPOEs with different thresholds. Results of calculations are presented in Table 1, where we use the following notations:

- VaR_0.8,VaR_0.9, VaR_0.95, and VaR_0.99 denote VaR with confidence levels 0.8, 09, 0.95, and 0.99, accordingly;
- CVaR_0.8,CVaR_0.9, CVaR_0.95, and CVaR_0.99 denote CVaRs with confidence levels 0.8, 09, 0.95, and 0.99, accordingly;
- POE_8, POE_9, and POE_15 denote POE with thresholds 8, 9, and 15, accordingly;
- bPOE_8, bPOE_9, and bPOE_15 denote bPOE with thresholds 8, 9, and 15, accordingly.

In Table 1, the minimum values of each risk function are highlighted. For instance, the path 1–2–4–5 has minimal POE with threshold = 8; the path 1–5 has minimal VaR with confidence level 0.8 and 0.9; the path 1–3–5 has minimal VaR

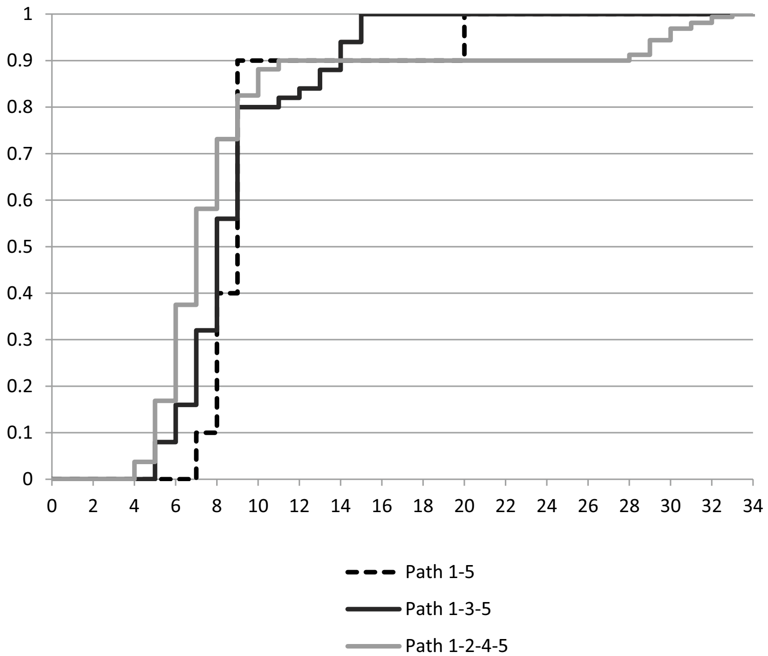

Fig. 2 Probability distributions of cost for paths 1–5, 1–3–5, and 1–2–3–5

Table 1 Risk functions values for three paths (minimum values are italic)

Statistical characteristics	Paths		
	1–5	1–3–5	1–2–4–5
Maximum	20	*15*	33
Average	9.6	*8.8*	9.3
Standard deviation	3.53	*2.72*	7.09
VaR_0.8	*8.8*	9	9
VaR_0.9	*9*	13	11
VaR_0.95	14.5	*14.2*	29.5
VaR_0.99	18.9	*14.8*	31.7
CVaR_0.8	14.5	*13.6*	19.97
CVaR_0.9	20	*14.6*	30
CVaR_0.95	20	*15*	31.1
CVaR_0.99	20	*15*	32.6
POE_8	0.6	0.44	*0.27*
POE_9	*0.1*	0.2	0.18
POE_15	0.045	*0*	0.097
bPOE_8	1	1	1
bPOE_9	1	*0.95*	1
bPOE_15	0.183	*0*	0.352

Col. #1 = risk function and its parameter; col. #2 = risk function value on path 1–5; col. #3 = risk function value on path 1–3–5; col. #4 = risk function value on path 1–3–5

with confidence level 0.95. The table shows that decision makers with different risk preferences select different minimum risk paths.

3 Stochastic shortest path problem formulations with different risk functions

In the previous section we considered the graph with random arc costs, which has only 3 paths and demonstrated that the even in this simple case shortest path may depend upon risk preferences of the decision maker. In this section we consider shortest path optimization problems with Average, POE, bPOE, VaR and CVaR risk functions.

To define a path in the optimization problem statement, we use a vector of binary variables:

$$\boldsymbol{x} = \left\{ x_a | a \in A \right\},$$

where A is a set of ordered pairs $\{i,j\}$—arcs, a is one pair from set A; $x_a = 1$ if the arc a belongs to a path, and $x_a = 0$, otherwise. We suppose that node = 1 is the source node and the last node = n is the sink node.

If costs (weights) of arcs are positive and we solve deterministic or stochastic shortest path problem, the set of constraints which determines a path from source node 1 to sink node n is the following:

$$\sum_{a \in A_1^-} x_a = 1 \tag{1a}$$

$$\sum_{a \in A_i^-} x_a - \sum_{a \in A_i^+} x_a = 0, \quad 2 \le i \le n-1, \tag{1b}$$

$$x_a \in \{0, 1\}, \quad \forall a \in A, \tag{1c}$$

where $A_i^- \in A, A_i^+ \in A$ are subsets of out coming and incoming arcs for node i. Equation for node n is not needed because it is linear dependent of constraints (1a)–(1b). Further we will call constraints (1) as "path constraints".

We consider finite set of scenarios $k = 1, \dots, K$ for stochastic weighted graph with equal probabilities $\frac{1}{K}$ as a joint distribution of arcs' costs $\{\xi_a^k | a \in A\}$.

The following Problem 1 minimizes the expected shortest path, see [16].

Problem 1 (Minimization of average)

$$\min_{\boldsymbol{x}} E\left[\sum_{a \in A} \xi_a x_a\right] \tag{2}$$

subject to:

path constraints (1).

For Problem 1, Kulkarni [36] developed an analytical method for finding shortest path for the case of independent and exponentially distributed costs of arcs. Using analytical method developed in [36], Davis et al. [37] derived a closed-form formula for the expected length of a shortest path, when arc lengths are independent and exponentially distributed random values.

Xiaoyu [16] considered "the most shortest path model" which maximizes probability that the travel time is shorter than time T. We reformulated this problem using POE in the objective function as follows: find a path which minimizes probability that the cost of path is larger than fixed value C, see Problem 2.

Problem 2 (Minimization of POE)

$$\min_{x} p_C\left(\sum_{a \in A} \xi_a x_a\right) \tag{3}$$

subject to:

path constraints (1).

Problem 2a (MILP formulation of POE minimization Problem 2)

$$
\begin{aligned}
&\min_{x,\zeta} \sum_{k=1}^{K} \frac{\zeta^k}{K}\\
&C \geq \sum_{a\in A} \xi_a^k x_a - M\zeta^k, \quad \zeta^k \in \{0,1\}, \quad k = 1,\dots,K,\\
&\text{path constraints (1)},
\end{aligned}
$$

where M is a large positive number; $\boldsymbol{\zeta} = (\zeta^1,\dots,\zeta^K)$ is a vector of additional Boolean variables ζ^k; $\frac{1}{K}$ is probability of one scenario.

Similar to POE minimization, we minimized bPOE, see Problem 3.

Problem 3 (Minimization of bPOE)

$$
\min_x \bar{p}_C\left(\sum_{a\in A} \xi_a x_a\right) \tag{4}
$$

subject to:

path constraints (1).

Further, we minimize threshold C under constraint on probability that path cost is greater than C, see Problem 4. This model, called α-shortest path model, see [16] minimizes VaR which is denoted by $q_\alpha\left(\sum_{a\in A} \xi_a x_a\right)$.

Problem 4 (Minimization of VaR)

$$
\min_{x,C} C \tag{5}
$$

subject to:

$$
p_C\left(\sum_{a\in A} \xi_a x_a\right) \leq 1-\alpha, \tag{6}
$$

path constraints (1).

Problem 4a (MILP formulation of VaR minimization Problem 4)

$$
\begin{aligned}
&\min_{x,\zeta,C} C,\\
&\sum_{k=1}^{K} \frac{\zeta^k}{K} \leq 1-\alpha,\\
&C \geq \sum_{a\in A} \xi_a^k x_a - M\zeta^k, \quad \zeta^k \in \{0,1\}, \quad k = 1,\dots,K,
\end{aligned}
$$

path constraints (1),

where M is a large positive number; $\boldsymbol{\zeta} = (\zeta^1, \dots, \zeta^K)$ is a vector of additional Boolean variables ζ^k; $\frac{1}{K}$ is probability of one scenario.

Problem 5 (Minimization of CVaR)

$$\min_{x} \bar{q}_\alpha \left(\sum_{a \in A} \xi_a x_a \right)$$

subject to:

$$\text{path constraints (1).}$$

Paper [16] solved Problems 1, 2, 4 with a hybrid approach integrating stochastic simulation and genetic algorithm. The approach uses stochastic simulation to estimate risk functions in these problems.

Rockafellar and Uryasev [29, 30] described how CVaR optimization can be reduced to minimization of linear function with linear constraints. Using this technique Problem 5 can be reduced to the mixed-integer linear problem (MILP), see Problem 5a.

Problem 5a (MILP formulation of CVaR minimization Problem 5)

$$\min_{\boldsymbol{x},\boldsymbol{u},C} \left\{ C + \frac{1}{1-\alpha} \sum_{k=1}^{K} \frac{u_k}{K} \right\},$$

subject to:

$$u_k \geq \sum_{a \in A} \xi_a^k x_a - C, u_k \geq 0, \quad k = 1, \dots, K,$$
$$\text{path constraints (1),}$$

where K is a number of scenarios; $\frac{1}{K}$ is the probability of one scenario; $\boldsymbol{u} = (u_1, \dots, u_K)$ is a vector of additional continues variables u_k.

POE minimization, Problem 2, can be formulated as an MILP (Problem 2a). This Problem 2a has additional Boolean variables $\zeta^1, \dots, \zeta^K$ and additional constraints, compared to bPOE minimization (Problem 3), which has binary variables only in path constraints (1). bPOE minimization Problem 3 can be considered as a proxy for POE minimization Problem 2a. However, bPOE minimization can be done significantly faster than POE minimization.

VaR minimization, Problem 4, can be reduced to MILP, Problem 4a, as described by Pavlikov et al. [38], however, the number of binary variables is significantly larger, compared to MILP formulation of CVaR minimization, Problem 5a. Therefore, Problem 5a can be solved significantly faster than Problem 4a for large dimensions.

CVaR minimization Problem 5a can be considered as a proxy for VaR minimization Problem 4 because of the following considerations. The constraint (9) in VaR minimization Problem 4 is nonconvex w.r.t. variables x_a. This nonconvex probabilistic constraint can be approximately replaced by the bPOE constraint,

$$\bar{p}_C\left(\sum_{a\in A}\xi_a x_a\right) \leq 1-\alpha.$$

This bPOE constraint can be replaced equivalently by the convex CVaR constraint,

$$\bar{q}_\alpha\left(\sum_{a\in A}\xi_a x_a\right) \leq C.$$

Therefore, VaR minimization Problem 4 can be approximately solved with fast CVaR minimization.

Mafusalov and Uryasev [31] demonstrated that bPOE minimization for stochastic linear functions with continuous variables can be reduced to a linear program. Therefore, bPOE minimization can be done significantly faster than POE minimization. We observe this, also, with a deterministic variant of bPOE, which is called Cardinality of Upper Average (CUA). This is an alternative to Cardinality of Upper Tail (CUT) which is the deterministic variant of POE, see results of numerical experiments in [39]. However, the change of variables in [31] is not applicable to bPOE minimization with binary variables (Problem 3). The approach is valid only for continuous variables. Paper [31] demonstrates that a solution of Problems 3 can be obtained by solving Problem 5 with an appropriate confidence level α corresponding to a threshold C. However, this confidence level α is not known in advance. We used AORDA PSG package [32] to minimize bPOE, which has the following heuristic for finding an appropriateα. Initially, Problem 5a is solved with $\alpha = 0.5$ and bPOE is calculated at the optimal point. Further, the confidence level is set to $\alpha = 1 - \bar{p}_C\left(\sum_{a\in A}\xi_a x^*_a\right)$, where $x^*_a, a \in A$, are components of optimal point. Then, Problem 5a is solved with the new confidence level. Again, bPOE is calculated at the new optimal point and the confidence level is set to $\alpha = 1 - \bar{p}_C\left(\sum_{a\in A}\xi_a x^*_a\right)$; and so on. The sequence of optimal values of CVaR quickly converges to the correct threshold of bPOE. Also, bPOE converges to a stationary point which is a solution of Problem 3. bPOE usually stabilizes after several iterations of the algorithm. In particular, Problems 3 with data considered in the case study converges after 2–3 iterations (see the following Sect. 4).

4 Case study

Problems are solved with AORDA PSG package [32] in combination with MIP capabilities of Gurobi optimization package. PSG has precoded Average, POE, bPOE, VaR, CVaR, and other risk functions, which considerably simplify programming of the optimization problems. PSG solves a sequence of sub-problems (linear, quadratic, or MIP) and it can call external Gurobi solver for these sub-problems, if needed. The case study described in this section (data, codes, and solutions) are posted online at this link:

http://uryasev.ams.stonybrook.edu/index.php/research/testproblems/logistics/cace-study-shortest-path-in-a-stochastic-weighted-graph-using-average-cvar-poe-and-bpoe-performance-functions/

We started with a deterministic dataset including a distance matrix with pairwise distances between 58 nodes. The data (available from open sources) are flying distances between 58 U.S. Air Force bases located throughout the world. The distances are used for estimating fuel costs, maintenance costs over time, etc., when calculating resource requirements for aircraft movement between bases. In practice, these distances may vary due to a variety of factors including weather, airspace considerations, operational concerns, etc. Therefore, we have generated a stochastic variant of this dataset as follows. We divided all arcs in three groups and considered three independent standardly distributed normal random values $\xi_i, i = 1, 2, 3$, corresponding to each group. Distances in these three groups were randomized by these three random values. For randomization we used the truncated normal distributions with the lower and upper bounds, -0.9 and 0.9, i.e., $\xi_i \in [-0.9, 0.9], i = 1, 2, 3$. Random samples of weights in i-th group were obtained by multiplying deterministic weights corresponding to arcs in i-th group by $\left(1 + \xi_i\right), i = 1, 2, 3$. We generated 1000 scenarios of the distance matrix.

We selected starting node 1 and final node 5 and optimized Problems 1 (average minimization), 2a (probability minimization), 3 (bPOE minimization), 4a (VaR minimization), 5a (CVaR minimization) with several values of parameters of risk functions. Table 2 presents calculation results.

We started with minimization of the average flying time (Problem 1); see optimal objective in col. #2 and solution time in col. #3 of Table 2.

We optimized POE (Problem 2a) with thresholds 4413.2, 4730.8, 5513.7, 5611.493. These thresholds are VaR values on optimal CVaR points, see col. #8, Table 2. PSG reduced the POE minimization problem to a linear MIP (this reduction was requested with MIP option). Maximal solution time was set to 3600 s.; two problems were not solved during this time (see col. #4, Table 2). For two problems the MIP solver has found an exact solution, but has spent much more time than bPOE or CVaR optimization.

We optimized VaR (Problem 4a) with confidence levels 0.9, 0.95, 0.99, 0.995, which are typical for risk management problems. PSG reduced the VaR minimization problem to a linear MIP. Maximal solution time was set to 3600 s. and MIP solver has not solved VaR minimization problems during allocated time (see col. #4, Table 2).

CVaR minimization (Problem 5a) was solved with $\alpha = 0.9$, 0.95, 0.99, 0.995, which are typically used in risk management (see α in col. #2 and optimal CVaR value in col. #3 in Table 2). We observe that optimal CVaR values monotonically increase as function of confidence level α. We note that CVaR with $\alpha = 0$ is actually the Average value. Therefore the lowest possible CVaR objective equals the minimum of Average = 3149.32. The largest value of CVaR for the considered confidence levels, equals 5859.67 for $\alpha = 0.995$.

Table 2 Optimization results for Problems 1, 2a, 3, 4a, 5a

Optimization results				Calculated values of risk functions on optimal points							
				CVaR		VaR		POE		bPOE	
Risk measure (objective) #1	Parameter of risk measure #2	Optimal objective #3	Time (s) #4	Parameter #5	Value #6	Parameter #7	Value #8	Parameter #9	Value #10	Parameter #11	Value #12
Average	N/A	3149.32	0.06	0.9	4928.942	0.9	4413.2	4413.2	0.1	4928.942	0.1
…	…	…	…	0.95	5324.89	0.95	4730.8	4730.8	0.05	5324.89	0.05
…	…	…	…	0.99	6233.35	0.99	5542.0	5513.7	0.011	5714.379	0.027
…	…	…	…	0.995	6696.6	0.995	6031.7	5611.493	0.008	5859.674	0.02
POE	4413.2	0.1	> 3600	0.9	4928.942	0.9	4413.2	4413.2	0.1	4928.942	0.1
POE	4730.8	0.05	> 3600	0.95	5324.89	0.95	4730.8	4730.8	0.05	5324.890	0.05
POE	5513.7	0.01	1893	0.99	5714.379	0.99	5513.708	5513.7	0.01	5714.379	0.01
POE	5611.493	0.005	300.7	0.995	5859.674	0.995	5611.493	5611.493	0.005	5859.670	0.005
bPOE	4928.942	0.1	89.70	0.9	4928.942	0.9	4413.2	4413.2	0.1	4928.942	0.1
bPOE	5324.89	0.05	85.37	0.95	5324.89	0.95	4730.8	4730.8	0.05	5324.89	0.05
bPOE	5714.379	0.01	101.44	0.99	5714.379	0.99	5513.708	5513.7	0.01	5714.379	0.01
bPOE	5859.674	0.005	94.65	0.995	5859.674	0.995	5611.493	5611.493	0.005	5859.674	0.005
VaR	0.9	4413.2	> 3600	0.9	4928.942	0.9	4413.2	4413.2	0.1	4928.942	0.1
VaR	0.95	4730.8	> 3600	0.95	5324.89	0.95	4730.8	4730.8	0.05	5324.890	0.05
VaR	0.99	5513.708	> 3600	0.99	5714.379	0.99	5513.708	5513.7	0.01	5714.379	0.01
VaR	0.995	5611.493	> 3600	0.995	5859.674	0.995	5611.493	5611.493	0.005	5859.670	0.005
CVaR	0.9	4928.942	21.4	0.9	4928.942	0.9	4413.2	4413.2	0.1	4928.942	0.1
CVaR	0.95	5324.890	22.99	0.95	5324.89	0.95	4730.8	4730.8	0.05	5324.89	0.05
CVaR	0.99	5714.379	29.67	0.99	5714.379	0.99	5513.708	5513.7	0.01	5714.379	0.01
CVaR	0.995	5859.674	31.91	0.995	5859.674	0.995	5611.493	5611.493	0.005	5859.674	0.005

Starting node = 1, final node = 58. Col. #1 = risk function (objective function); col. #2 = parameter of risk function; col. #3 = optimal value of objective function; col. #4 = solution time. Remaining columns #5 –12 contain values of risk functions calculated for parameters from col. #2 for optimal points of optimization problems with risk functions in col. #1

Let α be a confidence level of CVaR and C be a threshold of bPOE. If an optimal $CVaR_\alpha = C$, then the optimal $bPOE_C = 1 - \alpha$ (see, Mafusalov and Uryasev [31]). We minimized bPOE (Problem 3) with thresholds 4928.942, 5324.89, 5714.38, 5859.67, which are optimal CVaR values with $\alpha = 0.9, 0.95, 0.99, 0.995$ (see col. #2 in Table 2). We know that minimal bPOE with threshold 4928.942 should be equal to 0.1, since CVaR_0.9 = 4928.942. Similar, minimal bPOE with thresholds 5324.89, 5714.38, 5859.67 should be equal to 0.05, 0.01, 0.005, accordingly. Col. #3, Table 2, shows that minimal bPOE values were found correctly.

Col. #4 in Table 2 shows that bPOE and CVaR minimization are considerably faster than POE and VaR minimization. Also, for this specific instance, CVaR and bPOE minimization provide reasonable solutions which cannot be quickly improved. Also, we observe that minimization times for bPOE are about 3–4 times larger than for CVaR minimization. bPOE is minimized by running in cycle CVaR minimization sub-problems. Therefore, 3–4 CVaR minimizations find a confidence level corresponding to the minimal bPOE.

Table 3 shows optimal paths corresponding to minimization problems reported in Table 2. We observe that for low values of parameters of considered risk functions (objectives), the solution is trivial = 1–58. However, when risk functions target low probability tail, the shortest path includes six arcs = 1–2–43–13–44–51–58. Increased risk aversion leads to a different shortest path.

Table 3 Optimal paths for Problems 1, 2a, 3, 4a, 5a

Risk measure (objective) #1	Parameter of objective #2	Optimal path #3
Average	N/A	1–58
CVaR	0.90	1–58
CVaR	0.95	1–58
CVaR	0.99	1–2–43–13–44–51–58
CVaR	0.995	1–2–43–13–44–51–58
VaR	0.90	1–58
VaR	0.95	1–58
VaR	0.99	1–2–43–13–44–51–58
VaR	0.995	1–2–43–13–44–51–58
POE	4413.2	1–58
POE	4730.8	1–58
POE	5513.7	1–2–43–13–44–51–58
POE	5611.49	1–2–43–13–44–51–58
bPOE	4928.942	1–58
bPOE	5324.89	1–58
bPOE	5714.379	1–2–43–13–44–51–58
bPOE	5859.674	1–2–43–13–44–51–58

Starting node = 1, final node = 58. Col. #1 = risk function (objective); col. #2 = parameter of risk function; col. #3 = optimal path

Table 4 illustrates the bPOE minimization algorithm described in the end of the Sect. 3. We show how many solutions of CVaR optimization Problem 5a is needed for finding minimal bPOE in Problem 3. We observe that algorithm converges after 2–3 iterations.

Further, we solved a set of optimization problems for the same graph with starting node = 15 and final node = 45. Such selection is motivated by the fact that the arc, connecting these two nodes, has maximum deterministic weight. Deterministic weight of 15–45 arc is 3.29 times bigger than the deterministic weight of 1–58 arc. We minimized: Average, CVaR with confidence 0.9 and bPOE with threshold 15,000. Table 5 shows optimization results. First, we note that all optimal paths include at least two arcs (col. #7, Table 5). Also, optimal paths are different for Average, CVaR, and bPOE risk functions. We observe that minimization times for CVaR and bPOE are significantly larger compared to the previous case (initial node = 1, final node = 58). bPOE minimization time is about three times larger than CVaR minimization time (col. #4, Table 5), which means that CVaR was minimized 3 times to minimize bPOE.

Table 5 shows that bPOE minimum value with the threshold 15,000 equals 0.0193. Therefore, bPOE minimization with this threshold can be done by minimizing CVaR with confidence level 1–0.0193 = 0.9807. Average, by definition, is CVaR with confidence level 0. Therefore, results in Table 5 can be reproduced by minimizing CVaR. Table 6 shows that with CVaR minimization we can reproduce Average and bPOE minimization.

Additionally, we conducted numerical experiments for Shortest Path problems with data from OR-library posted by Beasley (see http://people.brunel.ac.uk/~mastj

Table 4 Convergence of solutions of Problems 5a to solution of Problems 3

#1 Iteration	#2 Confidence level of CVaR	#3 Optimal CVaR	#4 bPOE
Minimize bPOE with threshold: 4928.942			
1	0.500	3928.89	0.1
2	0.900	4928.942	0.1
Minimize bPOE with threshold: 5324.89			
1	0.500	3928.89	0.05
2	0.950	5324.89	0.05
Minimize bPOE with threshold: 5714.379			
1	0.500	3928.89	0.0271
2	0.973	5519.817	0.01
3	0.990	5714.379	0.01
Minimize bPOE with threshold: 5859.674			
1	0.500	3928.89	0.0199
2	0.980	5583.481	0.005
3	0.995	5859.674	0.005

Col. #1 = number of iterations; col. #2 = confidence level of CVaR; col. #3 = optimal CVaR; col. #4 = bPOE value

Table 5 Optimization results: starting node = 15 and final node = 45

Optimized risk function #1	Parameter #2	Optimal objective #3	Time (s) #4	Calculated risk function #5	Value #6	Optimal path #7
Average	N/A	11,033.8	0.06			15–14–45
				CVaR_0.9	17,406.3	
				bPOE_15000	0.301	
CVaR	0.9	14,059.8	2531.7			15–53–49–44–13–43–14–3–5–30–45
				Average	11,460.5	
				bPOE_15000	0.0308	
bPOE	15,000	0.0193	6269.9			15–53–51–44–43–2–14–33–9–40–34–45
				CVaR_0.9	14,122.8	
				Average	11,405.4	

Col. #1 = risk function (objective); col. #2 = parameter of risk function; col. #3 = optimal objective; col. #4 = solution time; col. #5 = calculated risk function; col. #6 = value of calculated risk function on optimal path; col. #7 = optimal path

Table 6 Minimizing CVaR with different confidence levels (reproduction results of Table 5)

Objective #1	Confidence level #2	Optimal objective value #3	Average #4	CVaR_0.9 #5	bPOE_15000 #6
CVaR	0	11,033.8	11,033.8	17,406.3	0.3010
CVaR	0.90	14,059.8	11,460.5	14,059.8	0.0308
CVaR	0.9807	15,000.0	11,405.4	14,122.8	0.0193

Col. #1 = objective (CVaR); col. #2 = confidence level of CVaR; col. #3 = optimal objective; col. #4 = Average at optimal point; col. #5 = $\text{CVaR}_{0.9}$ at optimal point; col. #6 = bPOE_{15000} at optimal point

jb/jeb/orlib/rcspinfo.html). We considered 4 data sets: rcsp1, rcsp7, rcsp16, rcsp24. Based on deterministic data posted at the website, 1000 scenarios of weighted graph were generated for each dataset.

For each dataset we optimized Average, CVaR, and bPOE of length of a shortest path. We set CVaR confidence level = 0.9. After minimizing CVaR we minimized bPOE with the threshold equal to optimal CVaR; see results in Table 7. We observe that bPOE minimization is done correctly since minimal bPOE is close to the theoretical optimum = 0.1 (see, col. #8 in Table 7). Table 7 shows that for graphs with the number of arcs from 955 to 4868, the CVaR optimization time rages from 3.88 to 18.19 s. Accordingly, bPOE minimization time is ranging from 7.66 to 26.83 s.

Table 7 Optimization results: starting node = 1, final node = last one.

Data set #1	Number of nodes #2	Number of arcs #3	Optimal average #4	Time (s) #5	Optimal $CVaR_{0.9}$ #6	Time (s) #7	Optimal $bPOE_{CVaR}$ #8	Time (s) #9
rcsp1	100	955	89.37	0.02	141.23	3.88	0.099956	7.66
rcsp7	100	999	3.27	0.02	6.0208	3.96	0.099985	6.39
rcsp16	200	1960	5.48	0.02	9.1514	13.38	0.100001	13.23
rcsp24	500	4868	3.28	0.05	5.3939	18.19	0.100007	26.83

Col. #1 = dataset in OR-library; col. #2 = number of nodes; col. #3 = number of arcs; col. #4 = optimal Average objective; col. #5 = minimization time with Average objective; col. #6 = optimal CVaR_0.9 objective; col. #7 = optimization time with CVaR_0.9 objective; col. #8 = optimal bPOE objective with threshold = minimal CVaR; col. #9 = minimization time with bPOE objective

5 Conclusions

We have formulated stochastic shortest path problems using recent developments in risk management theory. Considered in the literature minimization problems with POE and VaR objectives have many binary variables associated with definition of POE and VaR. Alternative minimization formulations with bPOE and CVaR and objectives have a significantly smaller number of binary variables. The new minimization formulations resulted in a significant saving of optimization time compared to problems with POE and VaR objective.

We want to note that CVaR and bPOE take into account outcomes in the tail of the distribution, while VaR and POE ignore these outcomes. CVaR by definition is the average of the worst-case (1-α)*100% outcomes (e.g., 5% outcomes). Consequently, CVaR takes into account outcomes in access of VaR. Risk bPOE function is inverse of CVaR and it is equal to the probability of tail with a known mean value (which is the threshold in bPOE). bPOE takes into account outcomes both in access and below of the threshold. So, largest outcomes are not ignored, compared to POE. Therefore, CVaR and bPOE functions have important conceptual advantages versus VaR and POE.

We have conducted a case study demonstrating effectiveness of the approach and posted results at the website. For solving optimization problems we have used PSG package which has precoded POE, VaR, bPOE and CVaR functions. Optimization codes in PSG language consist only of few lines. For solving MIPL sub-problems, PSG calls Gurobi (such option is available, if needed). We have found that CVaR and bPOE minimization frequently also lead to minimal VaR and POE. Therefore, CVaR and bPOE optimization is a good proxy for VaR and POE minimization.

Future research areas include application of bPOE and CVaR to more complicated network flow models, and exploring potential heuristics to optimize bPOE for very large networks.

References

1. Dijkstra, E.W.: A note on two problems in connection with graphs. Numer. Math. **1**, 269–271 (1959)
2. Bellman, R.E.: On a routing problem. Q. Appl. Math **16**, 87–90 (1958)
3. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press (1962)
4. Lawler, E.L.: Combinatorial Optimization: Networks and Matroids. Holt, Rinehart, and Winston (1976)
5. Floyd, R.W.: Algorithm 97 (shortest path). Commun. ACM **5**(6), 345 (1962)
6. Warshall, S.: A theorem on Boolean matrices. J. ACM **9**(1), 11–12 (1962)
7. Moore, E.F.: The shortest path through a maze. In: Proceedings of the International Symposium on the Theory of Switching, pp. 285–292. Harvard University Press (1959)
8. Lee, C.Y.: An algorithm for path connection and its applications. IRE Trans. Electron. Comput. **10**(3), 346–365 (1961)
9. Carlyle, W.M., Royset, J.O., Wood, R.K.: Lagrangean relaxation and enumeration for solving constrained shortest-path problems. Networks **52**(4), 256–270 (2008)
10. Royset, J.O., Carlyle, W.M., Wood, R.K.: Routing military aircraft with a constrained shortest-path algorithm. Mil. Oper. Res. **14**(3), 31–52 (2009)

11. Royset, J.O., Wood, R.K.: Solving the bi-objective maximum-flow network-interdiction problem. INFORMS J. Comput. **19**(2), 175–184 (2007)
12. Frank, H.: Shortest paths in probability graphs. Oper. Res. **17**, 583–599 (1969)
13. Mirchandani, B.P.: Shortest distance and reliability of probabilistic networks. Comput. Oper. Res. **3**, 347–676 (1976)
14. Murthy, I., Sarkar, S.: A relaxation-based pruning technique for a class of stochastic shortest path problems. Transp. Sci. **30**(3), 220–236 (1996)
15. Loui, P.: Optimal paths in graphs with stochastic or multidimensional weights. Commun. ACM **26**, 670–676 (1983)
16. Xiaoyu, Ji.: Models and algorithm for stochastic shortest path problem. Appl. Math. Comput. **170**(1), 503–514 (2005)
17. Boginski, V.L., Commander, C.W., Turko, T.: Polynomial-time identification of robust network flows under uncertain arc failures. Optim. Lett. **3**(3), 461–473 (2009)
18. Elci, O., Noyan, N., Bulbul, K.: Chance-constrained stochastic programming under variable reliability levels with an application to humanitarian relief network design. Comput. Oper. Res. **96**, 91–107 (2018)
19. Yu, G., Yang, J.: On the robust shortest path problem. Comput. Oper. Res. **25**, 457–468 (1998)
20. Pessoa, A., Pugliese, L., Guerriero, F., Poss, M.: Robust constrained shortest path problems under budgeted uncertainty. Networks (2015). https://doi.org/10.1002/net.21615
21. Pascoal, M., Resende, M.: Reducing the minmax regret robust shortest path problem with finite multi-scenarios. In: Bourguignon, P., Jeltsch, R., Pinto, A., Viana, M. (eds.) CIM Series in Mathematical Sciences—Mathematics of Planet Earth: Energy and Climate Change (Dynamics, Games and Science), vol. 2. Springer (2014)
22. Pascoal, M., Resende, M.: Minmax regret robust shortest path problem in a finite multi-scenario model. Appl. Math. Comput. **241**, 88–111 (2014)
23. Murthy, I., Her, S.-S.: Solving min-max shortest path problems on a network. Naval Res. Log. **39**, 669–683 (1992)
24. Montemanni, R., Gambardella, L., Donati, V.: A branch and bound algorithm for the robust shortest path problem with interval data. Oper. Res. Lett. **32**, 225–232 (2004)
25. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Math. Program. Ser. B **98**, 49–71 (2003)
26. Bruni, M.E., Guerriero, F.: An enhanced exact procedure for the absolute robust shortest path problem. Int. Trans. Oper. Res. **17**, 207–220 (2010)
27. Catanzaro, D., Labbé, M., Salazar-Neumann, M.: Reduction approaches for robust shortest path problems. Comput. Oper. Res. **38**, 1610–1619 (2011)
28. Kasperski, A., Zieliński, P.: Robust discrete optimization under discrete and interval uncertainty: a survey. In: Doumpos, M., Zopounidis, C., Grigoroudis, E. (eds.) Robustness Analysis in Decision Aiding, Optimization, and Analytics. International Series in Operations Research & Management Science, vol. 241. Springer, Cham (2016)
29. Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. J. Risk **2**(3), 21–41 (2000)
30. Rockafellar, R.T., Uryasev, S.: Conditional value-at-risk for general loss distributions. J. Bank. Finance **26**(7), 1443–1471 (2002)
31 Mafusalov, A., Uryasev, S.: Buffered probability of exceedance: mathematical properties and optimization algorithms. SIAM J. Optim. **28**(2), 1077–1103 (2018)
32. Optimization and Risk Management: Case Studies with Portfolio Safeguard (PSG). AORDA (2010). ISBN: 0982821301
33. Rockafellar, R.T., Royset, J.O.: On buffered failure probability in design and optimization of structures. J. Reliab. Eng. Syst. Saf. **99**, 499–510 (2010)
34 Rockafellar, R.T., Uryasev, S.: The fundamental risk quadrangle in risk management, optimization and statistical estimation. Surv. Oper. Res. Manag. Sci. **18**, 33–53 (2013)
35. Davis, J.R., Uryasev, S.: Analysis of tropical storm damage using buffered probability of exceedance. Nat. Hazards **83**, 465–483 (2016)
36. Kulkarni, V.G.: Shortest paths in networks with exponentially distributed arc lengths. Networks **16**, 255–274 (1986)
37. Davis, R., Prieditis, A.: The expected length of a shortest path. Inf. Process. Lett. **46**, 135–141 (1993)

38 Pavlikov, K., Veremyev, A., Pasiliao, E.L.: Optimization of value-at-risk: computational aspects of MIP formulations. J. Oper. Res. Soc. **69**, 676–690 (2017). https://doi.org/10.1057/s41274-017-0197-4
39 Norton, M., Mafusalov, A., Uryasev, S.: Cardinality of upper average and its application to network optimization. SIAM J. Optim. **28**(2), 1726–1750 (2018)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.