CACE STUDY: Shortest Path in a Stochastic Weighted Graph using Average, CVaR, POE, bPOE (avg, cvar, pr_pen, bpoe)

Background

The case study is based on paper by Jordan and Uryasev [13]. We consider four stochastic formulations of the shortest path problem. The shortest path problem finds a shortest route from a starting point to a final point. Such problems are usually formulated with a graph consisting of a set of vertices (nodes) and arcs (edges) connecting vertices. The deterministic shortest path problem finds a path from a starting point to a final point with the minimal weight. There are many efficient algorithms for finding shortest path: Dijkstra [1], Bellman-Ford [2], [3], Lawler [4], Floyd-Warshall [5], [6].

Weights of arcs in a stochastic weighted graph are random. The problem of determining the probability distribution of the shortest path length was studied in [7] and [8]. Paper [9] considers the problem, where the optimal path maximizes a quadratic expected utility. Papers [10], [11] study stochastic shortest path problems with different types of cost functions. Paper [12] considers the so called expected shortest path, α-shortest path, and the "most shortest" path problems.

This case study solves the following problems. We find a path with the minimal:

1) Expected cost (length);
2) Conditional Value-at-Risk (CVaR) of the path cost with a specified confidence level;
3) Probability of Exceedance (POE) of the path cost with a specified threshold;
4) Buffered Probability of Exceedance (bPOE) of the path cost with a specified threshold.

We started with a deterministic dataset including a distance matrix with pairwise distances between 58 nodes. The data are actual flying distances between 58 U.S. Air Force bases located throughout the world. In practice, these distances may vary due to a variety of factors including weather, airspace considerations, operational concerns, etc. 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 each group by $(1 + \xi_i), i = 1,2,3$. We generated 1000 scenarios of the distance matrix.

References

1. Dijkstra, E.W. (1959). A note on two problems in connexion with graphs, Numer. Math 1, 269-271.
2. Bellman R.E. (1958). On a routing problem, Quart. Appl. Math 16, 87-90.
3. Ford, L.R., and D. R. Fulkerson. (1962). Flows in Networks. Princeton University Press.
4. Lawler, E.L. (1976). Combinatorial Optimization: Networks and Matroids. Holt, Rinehart, and Winston.
5. Floyd, R.W. (1962). Algorithm 97 (SHORTEST PATH). Communications of the ACM, 5(6):345.
6. Warshall, S. (1962). A theorem on boolean matrices. Journal of the ACM, 9(1):11–12.
7. Frank, H. (1969). Shortest paths in probability graphs, Oper. Res. 17, 583–599.
8. Mirchandani, B.P. (1976). Shortest distance and reliability of probabilistic networks, Comp. Oper. Res. 3, 347–676.
9. Murthy, I., and S. Sarkar. (1996). A relaxation-based pruning technique for a class of stochastic shortest path problems, Transp. Sci. 30 (3), 220–236.
10. Loui, P. (1983). Optimal paths in graphs with stochastic or multidimensional weights, Commun. ACM 26, 670–676.
11. Murthy, I., and S. Sarkar. (1996). A relaxation-based pruning technique for a class of stochastic shortest path problems, Transp. Sci. 30 (3), 220–236.
12. Xiaoyu Ji. (2005). Models and algorithm for stochastic shortest path problem. Applied Mathematics and Computation. 170(1) 503-514
13. Jordan J. and S. Uryasev. (2019). Shortest Path Network Problems with Stochastic Arc Costs. Optimization Letters, Submitted for Publication.

Notations

V = set of vertices (nodes);

A= set of arcs (edges) connecting vertices;

I= number of nodes in the set V;

v_i = i-th node in the set V, $(v_i \in V)$, $i = 1, \dots, I$;

$G = (V, A)$ = weighted graph;

$\{v_i, v_j\} \in A$ = arc connecting nodes v_i and v_j;

$\boldsymbol{x} = \{x_{\boldsymbol{ij}} | \{i, j\} \in A\}$ = vector of binary variables, where $x_{\boldsymbol{ij}} = 1$ if the arc $\{i, j\}$ belongs to a path, and$x_{\boldsymbol{ij}} = 0$, otherwise;

ξ_{ij} = random weight of the arc connecting nodes v_i and v_j, $(v_i, v_j \in V)$, $i, j = 1, \dots, I$. Depending on an application, the weight can be travel time, cost, etc.;

$d = \{v_0, v_1, \dots, v_k\}$ = path, connecting nodes v_0 and v_k in the graph G, with arcs $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{k-1}, v_k\} \in A$;

$\xi(d) = \left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right]$ = random cost of a path;

$E\left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right]$ = average (expected) cost of a path;

$\bar{p}_T\left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right]$ = Buffered Probability of Exceedance T by cost of a path;

$\bar{q}_\alpha\left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right]$ = Conditional Value-at-Risk of cost of a path with confidence level α.

$p_T\left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right]$ = Probability of Exceedance T by cost of a path;

The first problem finds a path with the minimal expected cost (length).

Problem 1

$$\min_{\boldsymbol{x}} E\left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right] \tag{1}$$

subject to:

$$\sum_{(1,j)\in A} x_{1j} - \sum_{(j,1)\in A} x_{j1} = 1, \tag{2}$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, \quad 2 \le i \le n-1, \tag{3}$$

$$\sum_{(n,j)\in A} x_{nj} - \sum_{(j,n)\in A} x_{jn} = -1, \tag{4}$$

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

The following problem finds a path with a minimal CVaR of the path cost.

Problem 2.

$$\min_{\boldsymbol{x}} \bar{q}_\alpha\left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right] \tag{11}$$

subject to:

$$\sum_{(1,j)\in A} x_{1j} - \sum_{(j,1)\in A} x_{j1} = 1, \tag{12}$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, \quad 2 \le i \le n-1, \tag{13}$$

$$\sum_{(n,j)\in A} x_{nj} - \sum_{(j,n)\in A} x_{jn} = -1, \tag{14}$$

$$x_{ij} \in \{0,1\},\ \forall (i,j) \in A. \tag{15}$$

The following problem finds a path with minimal POE of the path cost with a specified threshold *T*.

Problem 3.

$$\min_{\boldsymbol{x}} p_T \left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right] \tag{6}$$

subject to:

$$\sum_{(1,j)\in A} x_{1j} - \sum_{(j,1)\in A} x_{j1} = 1, \tag{7}$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, \quad 2 \le i \le n-1, \tag{8}$$

$$\sum_{(n,j)\in A} x_{nj} - \sum_{(j,n)\in A} x_{jn} = -1, \tag{9}$$

$$x_{ij} \in \{0,1\},\ \forall (i,j) \in A. \tag{10}$$

Finale problem finds a path with the minimum bPOE of the path cost with a specified threshold *T*.

Problem 4.

$$\min_{\boldsymbol{x}} \bar{p}_T \left[\sum_{(i,j)\in A} \xi_{ij} x_{ij}\right] \tag{6}$$

subject to:

$$\sum_{(1,j)\in A} x_{1j} - \sum_{(j,1)\in A} x_{j1} = 1, \tag{7}$$

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, \quad 2 \le i \le n-1, \tag{8}$$

$$\sum_{(n,j)\in A} x_{nj} - \sum_{(j,n)\in A} x_{jn} = -1, \tag{9}$$

$$x_{ij} \in \{0,1\},\ \forall (i,j) \in A. \tag{10}$$

Below we show results of solving Problem 1, 2, 3, 4 with different parameters (confidence levels α and thresholds T) and calculating four functions CVaR, VaR, POE and bPOE with different parameters on optimal points of these Problems.

Table 1. Optimization results for Problems 1, 2, 3, 4: starting node 1 and final node 58.

Optimization Results Objective Functions				**Values of Risk Functions on Optimal Points**							
				CVaR		**VaR**		**POE**		**bPOE**	
Risk Measure	Parameter	Optimal Value	Time (sec)	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
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
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
POE	4413.2	0.1	>4000	0.9	4928.942	0.9	4413.2	4413.2	0.1	4928.942	0.1
POE	4730.8	0.05	>4000	0.95	5324.89	0.95	4730.8	4730.8	0.05	5324.890	0.05
POE	5513.7	0.01	>4000	0.99	5714.379	0.99	5513.708	5513.7	0.01	5714.379	0.01
POE	5611.493	0.005	>4000	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

To begin with, we select a starting node 1 and a final node 58. We solve with PSG optimization Problems 1, 2, 3, 4 and show results in Table 3. We select confidence levels 0.9, 0.95, 0.99, 0.995 for CVaR to observe sensitivity of the results to confidence level. Considered confidence levels are standardly used in risk management. CVaR is monotonic w.r.t. confidence level α. Table 3, column 3 shows that the lowest optimal values for CVaR equals 3149.32, because Average is actually CVaR with confidence level $\alpha = 0$. We have observed also that the largest value of CVaR equals 5859.67, for the confidence level α =0.995 . Therefore threshold levels for bPOE (which is the inverse function of CVaR) were selected in the range 3149.32 - 5859.67. In particular, we considere bPOE with thresholds 4928.942, 5324.89, 5714.38, 5859.67, which are equal to the optimal CVaR values with confidence levels 0.9, 0.95, 0.99, 0.995, see Table 3. This selection was done for verification purposes, because it is known that the minimal bPOE value with threshold 4928.942 should be equal to 0.1, since CVaR_0.9 = 4928.942 . Similar we know that the optimal values for bPOE with thresholds 5324.89, 5714.38, 5859.67 should be equal to 0.05, 0.01, 0.005, accordingly. We observe that the PSG solver correctly found the minimal bPOE values, as it is stated in the third column of Table 3. Regarding the computation times, Table 3, column 4 shows that computation times for bPOE minimization are about 3-4 times larger than CVaR minimization. The reason for this is that bPOE solution is obtained by running several CVaR optimization problems. So, after 3-4 runs of CVaR optimization the algorithm finds appropriate confidence lebel, which delivers minimal bPOE. We optimized also POE with thresholds 4413.2, 4730.8, 5513.7, 5611.493. These thresholds are POE values on optimal CVaR points, see Table 3. PSG automatically reduced the POE minimization problem to linear MIP. Maximal solution time was set to 4000 sec. We observe that during this time the MIP solver has not improved the solution coming from CVaR optimization. This observation confirms the fact that CVaR and bPOE minimization are considerably faster that POE minimization.

Table 2. Optimal paths for optimization problems: starting node 1 and final node 58.

Risk Function	Parameter	Optimal Path
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
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

Table 4 shows that for the small values of parameters of considered functions, the solution is trivial, 1-58 . However, when risk requirements are getting more stringent, the solution includes 6 arks, 1-2-43-13-44-51-58. This shows that risk averseness results in a different shortest path.