Shortest Path with Variable Edge Failure Financial Optimization

Austin Griffith

Georgia Institute of Technology, Quantitative Finance

1 Problem Setup

What is an optimal investment policy when charges and fees, interest and changes of stock quotations are taken into account? Questions such as this can be answered using a shortest path optimization. For the sake of this exercise, a generic derivation will be used, though additional dimensions in the form of variables can be added for the application to more complex problems.

Let us assume we are given a directed graph $G = (V, E)$. Let s and t both exist on V, which will represent the nodes that populate the graph. In this case, s is 0, and t is the final node value. The nominal measure on a given path (i, j) is c_{ij}. If the edge fails (gets congested), then the weight becomes $c_{ij} + d_{ij}$. To add complexity to this problem, let us assume there are at most L allowable simultaneous edge failures on this directed graph.

The goal is to minimize the worst-case cost from s to t. Once this is completed, congestions, L, can be increased to observe how the optimal solution and path change.

2 Derivation

The initial setup of the shortest path involves creating a simple minimization mixed integer linear program, using the binary variable x.

$$\min c^T x$$
$$\text{s.t.} \quad x \in \{0,1\}$$
$$\text{for} \quad c_j \in [\overline{c_{ij}}, \overline{c_{ij}} + d_{ij}]$$

Since we're minimizing the worst case scenario, we have to also consider the summation of edge failures.

$$\sum_{j=1}^{n} d_j x_j$$

In this derivation, n is the largest node value. This worst case scenario is determined by the maximum allowable edge failures. Therefore, this summation can be turned into a maximization problem, with decision variable S, and variable scalar L.

$$\max_{S \in 1 \ldots n, |S|=L} \sum_{j \in S} d_j x_j$$

From here, this maximization of the worst scenario can now be added to our simple linear program defined at the beginning. This allows us to solve for the optimal worst-case travel scenario.

$$\min_{x \in \{0,1\}} [c^T x + \max_{S \in 1 \ldots n, |S|=L} \sum_{j \in S} d_j x_j]$$

In its current state as a minimzation of a maximization, the program is unsolvable. To deal with this, we first need to solve the inner problem. The initial setup for the inner problem solution is shown below.

$$\max \sum_{j=1}^{n} x_j d_j s_j$$
$$\text{s.t.} \quad \sum_{j=1}^{n} s_j = L$$
$$s_j \in \{0,1\} \quad \forall j$$

Now that the inner problem has been laid out, we need to solve for the dual. This will set the maximization problem into a minimization, which can then be reinserted into the initial minimization program in a solvable form. In order to solve for the dual, there has to be a relaxation of the binary variable s.

$$\max \sum_{j=1}^{n} x_j d_j s_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} s_j = L$$

$$0 \leq s_j \leq 1 \quad \forall j$$

Now we can formulate the dual of the relaxation. The dual will be of the following form.

$$\text{objective function} \leq \text{relaxation}$$

$$\sum_j s_j \left(\lambda_0 + \lambda_1\right) \leq L\lambda_0 + \sum_j \lambda_j$$

The constraints of the new dual mixed integer linear program (MILP) are as follows:

$$\lambda_j \geq 0$$

$$\lambda_0 \quad \text{u.r.}$$

$$\lambda_0 + \lambda_j \geq x_j d_j \quad \forall j$$

$$\lambda_j \geq x_j d_j - \lambda_0$$

To make the combined equation simpler, we can comibine terms. The final two inequalities can be simplified into a pointwise maximum of the two right hand side values.

$$\lambda_j = max\left(x_j d_j - \lambda_0, 0\right)$$

Now that the inner problem has been converted to a minimization problem through duality, we can now add it back into the main MILP.

$$\min_{x,\lambda_0,z} \bar{c}^T x + L\lambda_0 + \sum_{j=1}^{n} max\left(x_j d_j - \lambda_0, 0\right)$$

$$x_j \in \{0, 1\}$$

$$\lambda_0 \quad \text{u.r.}$$

3 Final Formulation

Now that the derivation is complete, we can state the final equation for the shortest path model with respect to the problem stated at the beginning of the paper.

$$
\begin{gathered}
\min_{x,\lambda_0,z} \bar{c}^T x + L\lambda_0 + \sum_{j=1}^{n} z_j x_j \\
\text{s.t.} \quad x \in \{0,1\} \\
\lambda_0 \quad \text{u.r.} \\
z_j \geq 0 \quad \forall j \\
z_j \geq d_j - \lambda_0 \quad \forall j \\
\sum_j x_{0j} = 1 \\
\sum_i x_{i0} = 0 \\
\sum_i x_{it} = 1 \quad \text{for} \quad t = \text{end node} \\
\sum_j x_{tj} = 0 \quad \text{for} \quad t = \text{end node} \\
\sum_{i \neq 0, j \neq t} (x_{ij} - x_{ji}) = 0
\end{gathered}
$$

In this final formulation, x is a binary decision variable, representing whether a path edge will be taken. λ_0 is a single continuous, unrestricted decision variable that amplifies the magnitude of the failure value L. The final decision variable, z, is used to determine which edge will fail, so as to minimize the worst-case scenario. The summations of x values at the end are used to determine the start and end location of the path, and ensure the MILP doesn't end on a node somewhere in the middle.

From here, one could attempt to solve for the solution analytically. Assuming the data set for c_{ij} is small enough, one could relax the decision variables, and then attempt a branch and bound method to reach a solution. However, if the problem contains dozens of nodes and thousands of paths, the optimal path should be solved for using an optimization package such as Gurobi or PuLP.