SYM-AM-25-337

Excerpt from the Proceedings
of the
Twenty-Second Annual Acquisition Research Symposium and Innovation Summit

Thursday, May 8, 2025 Sessions
Volume II

Mixed Risk Course of Action

Published: May 5, 2025

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

Mixed Risk Course of Action

Harrison Schramm—is a Senior Lecturer for the Department of Defense Management at the Naval Postgraduate School in Monterey, CA. [Harrison.schramm@nps.edu]

Abstract

Commanders are looking for courses of action. However, translating this desire to an actionable system requires nuanced approach to both mathematics and government-industry relations. This talk will be in three parts; first describing the specific problem that INDOPACOM needs to address as an executive (read: unclassified, non-nerdy) level. The second part will be about how NPS and our Industry Partners leveraged A collaborative research and development agreement (CRADA) in order to bring the capabilities of both the university (i.e. NPS) and industry to bear to create a solution, as well as likely transition pathways into acquisition. The third part will focus on the technical aspects of the solution, and will be around a patent-pending innovation generated by faculty at NPS.

Introduction

Finding the "shortest distance" between two points is a common problem both in military and historical applications. "Shortest Distance" typically is taken to mean linear distance (i.e., lines you draw on a map) or functions that are linear and additive, such as cost and time. This is of course not the only operationally relevant way to measure distance. Another approach has been to measure distance via risk, which is multiplicative, not additive. Our novel contribution is to combine these two methods, with the unexpected outcome of creating a system for developing a small number of logistically- and risk- balanced courses of action from a very large universe of possibilities.

This work was supported by US INDOPACOM.

Generally, optimization problems are typically solved by creating graphs where vertices represent elements from a set of numbers, and edges connect pairs of elements that sum to a desired value, usually by analyzing additive structures within a set. In mathematics, "graph theory used in additive math" refers to the application of graph theoretic concepts and techniques to solve problems in additive number theory, particularly within the field of "additive combinatorics," where the focus is on studying the structure of sets of integers under addition, often leveraging the visual representation and analysis capabilities of graphs to identify patterns and prove theorems about sums of integers. The most commonly used technique for finding the shortest path is the principle behind Dijkstra's algorithm:

$$\mathrm{D} = \min\left(\mathrm{d}(\mathrm{v}), \left(d[\mathrm{u}] + \mathrm{w}(\mathrm{v}, \mathrm{u})\right)\right),$$

where $\mathrm{d}([\mathrm{v}]$ is the current calculated shortest distance from a source node to node "v", and $\left(d[\mathrm{u}] + \mathrm{w}(\mathrm{v}, \mathrm{u})\right)$ calculates the potential new shortest distance $\mathrm{d}(\mathrm{v})$ which allows one to find the shortest path between one vertex and all other vertices in a weighted graph, provided the edge weights are non-negative; essentially, it states that at each step, one should select the unvisited vertex with the smallest distance from the starting point to build the shortest path. Edge weights are typically assigned a numerical value, a weight, to represent a metric such as distance, cost, or time between the connected vertices. Edges are lines or connections linking two "vertices," representing the relationship between different points within the graph structure, which define how different nodes are connected within the network.

Typically, an optimized solution is an additive combinatory solution utilizing weighted edges. Weighted edges are an additive combinatory means, where the total weight of a path or subgraph is calculated by adding up the weights of all the individual edges included in that path or subgraph; meaning, the weights are combined through addition, and therefore “additive.”

Risk is typically a unitless parameter representing a probability or likelihood of an event occurring and a potential consequence. Solutions to probabilities of events occurring simultaneously require multiplication to find the combined probability; essentially, multiplying the individual probabilities of each event to get the probability of all events occurring in sequence. Weighted edges found using multiplicative methods may not be trivially nor linearly combined with weighted edges found with additive combinatorial methods. Therefore, a system and method are needed to determine the shortest path using a mix of risk and linear costs.

The problem of determining the shortest path while maintaining acceptable survivability risk using unitless and multiplicative parameters in weighted graphs is solved by incorporating a convex and risk distance parameter to provide the shortest path based on acceptable risk.

Figure 1 is a block diagram of our automated mixed course of action (COA) generation system. As shown in Figure 1, an automated mixed COA system may include one or more risk data input sources, map sources, a risk mix manager, a risk assessment engine, and a COA communications output. Risk data input sources may include but should not be limited to user sensors, feedback, social media, marketing analysis, user behavior data, government data, and databases. The risk mix manager may identify, assess, and provide options for reduced risks that could impact a COA. The risk assessment engine may identify and assess potential risks that could negatively impact the user. The primary purpose is to determine the probability and impact of the identified risks and provide strategies to minimize or avoid the risk. The risk assessment engine may estimate the potential impact of the risk, monitor and report on the risk, and types of risk analysis using qualitative risk and/or quantitative risk analysis.

Figure 1: Block Diagram of Our Automated Mixed Course of Action (COA) Generation System

Figure 2 is a functional example showing a system generating a course of action based on mixed risk. This app, implemented in R/markdown is a demonstration of *risk-adjusted shortest paths*. We have a somewhat messy 6-node network. The two boxes on the right hand side of the “Demo” page show two measures on the same network; the top is the distance in *additive units*: you might think of this as being miles, but it could also be money or time. The lower plot is risk, measured in probability of survival, that is, 1 is certain survival, and 0 is certain death. The *risk-scored distance* is shown in the left plot.

This is determined by computing

$$\lambda LD + (1 - \lambda)\kappa(-ln(RD)),$$

where LD is the linear distance, RD is the Risk Distance, and λ is a convexity parameter,

$0 \leq \lambda \leq 1$. The **shortest path** is computed and displayed (somewhat primitively) at the bottom of the input box.

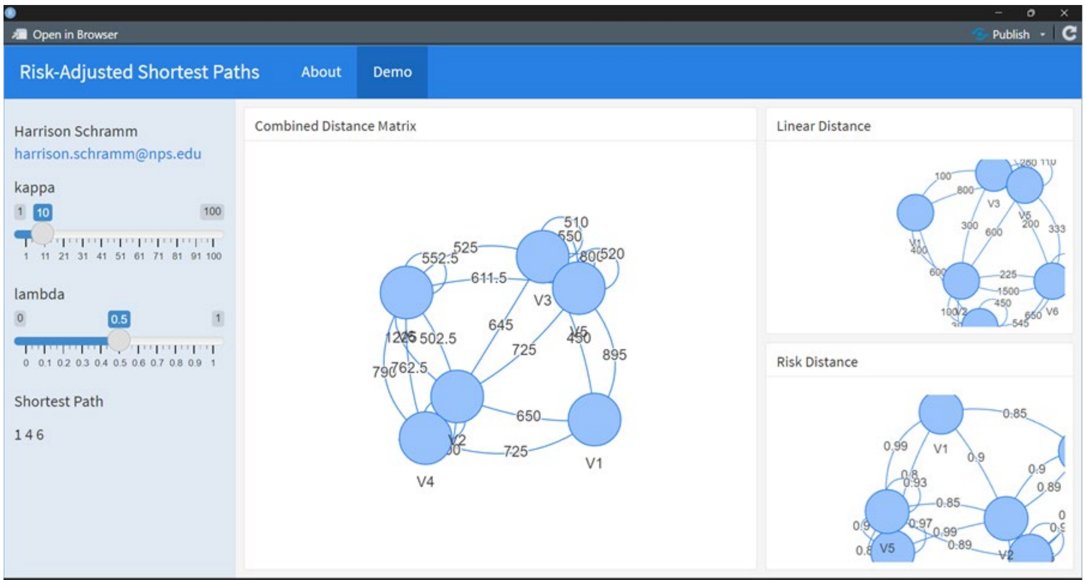

Figure 2: Risk-Adjusted Shortest Paths

A note about *κ*

Probabilities are unitless as are their functions. In order to add “risk” to “distance,” we need to introduce a parameter, κ that provides units. In addition to making the math work, it also has an important operational interpretation, in the “how many miles would you go out of your way to reduce risk?” In this example, we have benchmarked 10 miles per 1% of risk; so a route that has a 90% probability of survival inherits a 100 mile penalty.

Quantifying this risk in a way that is palatable to decision-makers is a key task for adoption. Our sense is that once κ is set, it will remain so for the duration of the exercise/scenario.

A note about *λ*

This parameter creates a *convex combination* of risk and distance; you might think of this as the proportion of froth and coffee in a latte, that is, a frothy latte would have a high value of λ. The key idea for this parameter—and the entire analytic approach for that matter—is to vary λ to create different solutions to the shortest path routing problem based on the relative importance of time and risk. In the current formulation, low values of λ emphasize speed while high values of λ emphasize low risk. Each solution provided by this approach is analytically defensible. Each solution produced by varying λ is a unique Course of Action, or COA. Trivially, it is possible that a network will have only one COA for all values of λ. It is unclear the upper limit of COAs that could be produced as a function of network complexity. That’s a math problem, not an operational problem.

Figure 3 shows minimal hardware implementation, consisting of a Raspberry Pi 5 computer, SixFab Cellular Modem, and Coral TPU. A $20 bill is included for size comparison.

Figure 3: Minimal Hardware Implementation

Beyond creating this algorithm, we have placed a functional version on a minimal hardware stack, consisting of a Raspberry Pi 5 with a quad core, 2.4 GHz processor, coupled with a SixFab cellular modem for on/off board communication and a Google Coral TPU for enhanced AI inference (note, the TPU is not necessary for the solution of the basic problem). This hardware implementation demonstrates the computational efficiency of the method, and the ability to be run on a hardware "stack" with a total price of less than $500. Such a system is readily and easily deployable at the tactical edge.

Future research: Future work on this project will involve implementation on a smaller technical stack, specifically the Raspberry Pi Zero 2W, and replacing the cellular modem with a LoRA radio.

References

Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2024). *rmarkdown: Dynamic documents for R* (R package version 2.28). https://github.com/rstudio/rmarkdown

Almende, B. V., & Thieurmel, B. (2022). *visNetwork: Network visualization using "vis.js" library* (R package version 2.1.2). https://CRAN.R-project.org/package=visNetwork

Csardi, G., & Nepusz, T. (2006). *The igraph software package for complex network research*. *InterJournal, Complex Systems*, 1695. https://igraph.org

Grimmett, G., & Stirzaker, D. (2001). *Probability and random processes*. Oxford University Press.

Newman, M. (2018). *Networks* (2nd ed.). https://doi.org/10.1093/oso/9780198805090.001.0001

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Raspberry Pi Foundation. (2025). *Raspberry Pi 5 Documentation*. https://www.raspberrypi.com/documentation/

Raspberry Pi. (n.d.). *Bookworm—The new version of Raspberry Pi OS*. https://www.raspberrypi.com/news/bookworm-the-new-version-of-raspberry-pi-os/

Ravindra K. A., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications*. Prentice-Hall.

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net