Scientific Research Publishing

American Journal of Operations Research, 2025, 15(4), 125-145
https://www.scirp.org/journal/ajor
ISSN Online: 2160-8849
ISSN Print: 2160-8830

Comparing Modeling Approaches for Distributed Contested Logistics

Jack R. Hernon[1], Thomas Kendall[1], Joseph B. Maxwell[1], Gregory Bremser[1], Ira Crofford[1], Katherine Winters[2], James E. Richards[2], Russell J. Nelson[1]

[1]Department of Mathematical Sciences, United States Military Academy, West Point, NY, USA
[2]US Army Engineering Research and Development Center, Vicksburg, MS, USA
Email: jack.hernon@westpoint.edu, gregory.bremser@westpoint.edu

How to cite this paper: Hernon, J.R., Kendall, T., Maxwell, J.B., Bremser, G., Crofford, I., Winters, K., Richards, J.E. and Nelson, R.J. (2025) Comparing Modeling Approaches for Distributed Contested Logistics. *American Journal of Operations Research*, **15**, 125-145.
https://doi.org/10.4236/ajor.2025.154007

Received: March 14, 2025
Accepted: July 7, 2025
Published: July 10, 2025

Abstract

In any military operation, reliable logistics is essential to maintaining a combat-effective force. Without the continual resupply of ammunition, food, and other materiel, forces cannot sustain their operations. Currently, logistics routes are made manually based on the judgment of individual logisticians. To automate this process, we develop two models that incorporate Geographic Information System (GIS) data to generate distributed logistics networks between two locations. These models build upon existing approaches to non-military transportation problems. The first model, the Route Selection Algorithm (RSA), modifies the K Shortest Paths algorithm and employs a Mixed Integer Linear Program (MILP) to generate multiple routes that minimize tactical risk while maximizing route dissimilarity. The second model, the Route Generation Algorithm (RGA), iteratively determines the optimal path and then applies a penalty factor to previously used arcs, discouraging their selection in future routes to enable route dissimilarity. Both models return multiple dissimilar routes with minimized risk that provide commanders with several viable resupply options. After a small-scale model comparison of 100 simulated scenarios, our results indicate that the RSA produces routes with lower risk, while the RGA generates routes with higher dissimilarity and has a lower runtime. These models serve as initial formulations that can be further refined into a robust, comprehensive risk-avoidance model to be used by military logisticians. This paper introduces a novel comparison between two routing algorithms and presents an innovative method for quantifying the risks posed by enemy forces and weapons systems in military transportation problems.

Keywords

Vehicle Routing Problem, Risk Modeling, Python

1. Introduction

One of the key tenets of success in military operations is establishing reliable, robust, and secure logistics, defined as "the processes, resources, and systems involved in generating, transporting, sustaining, and redeploying or reallocating materiel and personnel." [1] During conflict, reliable logistics enables armies in combat to be resupplied and allows them to continue conducting operations. Conversely, unreliable logistics will force an army to halt their operations, even when opportunities to easily advance are present. Given the criticality of logistics, we want to evaluate the way the US Army currently creates logistics networks to determine if they can be improved.

1.1. Current US Army Logistics

Currently, US Army logistics networks follow a fairly standardized methodology. Materiel will arrive at an Aerial Port of Debarkation (APOD) or Sea Port of Debarkation (SPOD), where it is under the control of the Theater Sustainment Command (TSC) [2]. From there, the materiel is transported to a Central Receiving and Shipping Point (CRSP), run by a sustainment brigade, where it is temporarily stored before being moved closer to the supported unit [3]. After that, the materiel is then moved to a Combat Sustainment Support Battalion (CSSB), a battalion within the sustainment brigade, which stores the materiel until requested [3]. Once the materiel is requested, it is sent to a Brigade Support Battalion (BSB), which is a battalion within the supported Brigade Combat Team (BCT), located at the BCT's Brigade Support Area (BSA) [4]. The flow of materiel from an APOD or SPOD to a BSB is shown below in **Figure 1**. Collocated within the BSA is an element from the Forward Support Company (FSC) of the supported battalion [2]. These locations are called the Field Train Command Posts (FTCP) [4]. Upon request, the FSCs will then send the materiel from the FTCP to a Combat Train Command Post (CTCP), where the rest of the FSC is located, which typically serves as the main supply point for the supported battalion [4]. Finally, the materiel is then moved from the CTCP to the Logistics Resupply Point (LRP), where each individual supported company can receive the necessary equipment and distribute it as needed. [4] The flow of materiel from a CSSB to the LRP is illustrated in **Figure 2**. This is the typical process as described in Army doctrine; however, it is possible that the true flow will differ slightly. Depending on the commander's needs, it is possible that materiel could be moved to multiple CTCPs or CSSBs before being moved to a subordinate unit, or several steps in this process could be skipped entirely [5].

Problem Statement

Currently, the risk of an enemy exploiting a logistics route between two logistics nodes is evaluated through human intuition [5]. Typically, for routes between two locations that contain a large amount of throughput, there will exist between three and five possible routes, one of which is selected for use daily based off the current risk posed by enemy forces. The selected route, if no significant changes to the

risk level occur, does not need to change every day. However, this method of minimizing the risk posed by certain routes relies on human intuition and estimation, which can be fallible or lead to suboptimal results.

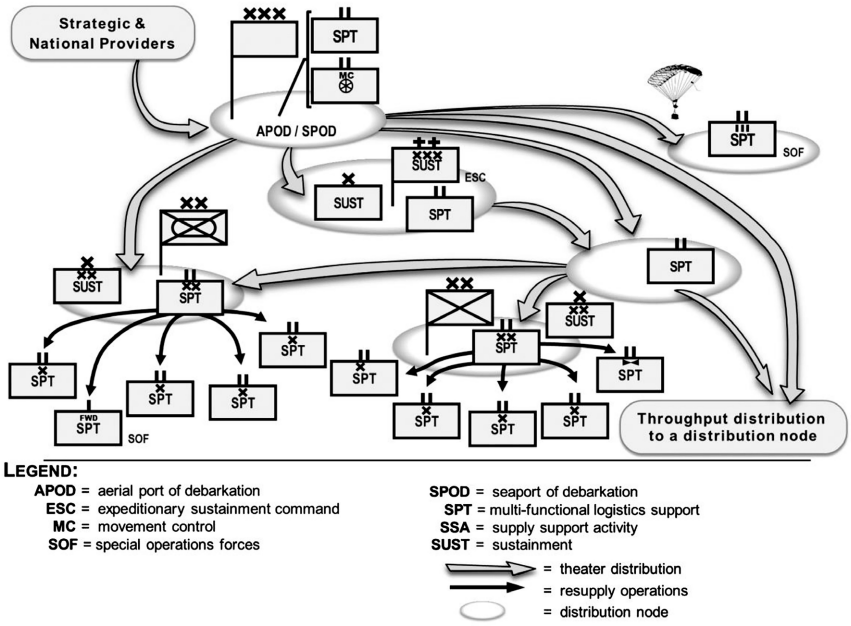

Figure 1. This graphic shows the flow of materiel from the APOD/SPOD to a BSB [4].

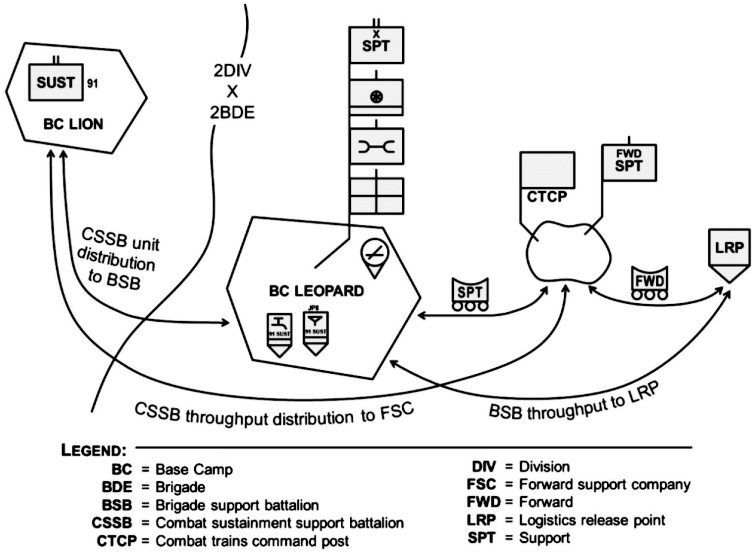

Figure 2. This graphic shows the flow of materiel from a CSSB to the LRP [9].

We propose that we automate the process used to create logistics routes to comprehensively account for enemy locations and time constraints. These routes would account for the risk posed by enemy forces in the area, create spatially dissimilar routes to ensure diverse route options, and meet mission time constraints. This would ensure more accurate estimations of risk for the transportation of materiel in a contested environment and free up valuable time for logisticians to refocus

on other matters, such as managing and compiling supply requests for different units.

1.2. Literature Review

To better understand this problem, it is necessary to explore current research and similar applications in transportation problems. Such problems focus on the use of different algorithms to generate routes in a particular environment, which will optimize according to different objectives, such as time or distance. Multiple approaches have been used to model similar problems that involve minimizing objectives in routing.

A model from Dadkar *et al.* uses a two-stage formulation to minimize exposure to toxic waste in rail travel [6]. First, a K shortest path (KSP) heuristic is used to identify routes that minimize a combined time and risk metric. Second, a mixed integer linear program (MILP) is used to select a dissimilar subset. A different model from Chang *et al.* uses a similar approach but deals with stochastic travel times. This model calculates the expected travel time and selects routes with a specified precision [7]. Another paper from Liu *et al.* uses a heuristic algorithm instead of an MILP to create routes that optimize a combined shortest time and dissimilarity metric [8]. Another model from Kang *et al.* incorporates Value at Risk (VAR) into a KSP routing model for hazardous waste transportation [9].

Additionally, Xie *et al.* explores a similar problem, where they determine the most cost effective set of rail yards to open to transfer toxic waste [10]. In this model, they create Origin-Destination (OD) segments between two rail yards, and then use a MILP to select the subset of OD pairs to minimize the cost and number of rail transfers. A similar paper from Wan and Lo uses a MILP to create OD pairs between nodes in a civilian rail network with multiple constraints [11]. Furthermore, a different model from Kendall *et al.* creates land routes to avoid a stationary enemy location [12]. In this model, a soldier moves to an objective by either walking, crouching, or crawling, and must avoid audio and visual detection from a stationary enemy. They calculate the probability of detection resulting from traversing along each pair of arcs, and then use an MILP to identify the route that minimizes the risk of detection.

There are other papers in the literature that have looked at similar problems. Erkut and Verter attempt to find dissimilar routes for hazardous waste transportation [13]. Their work examines multiple algorithms to determine which method produces the best routes while considering urban and rural areas, factors that may be worth exploring for future work. Their study also highlights the challenge of modeling transport risk, as different risk models often result in different optimal routes. This is an important consideration for military logistics, where the quantification of risk can directly impact route selection and operational security.

Thyagarajan *et al.* explore methods to determine dissimilar paths for military aircraft during mission entry, with the aim of avoiding mission detection [14]. Their work generates a network based on geographic information and then ap-

plies p-dispersion heuristics and the tabu search procedure to find a set of dissimilar routes. The focus on route dispersion aligns with risk mitigation strategies in military logistics, where reducing predictability can be critical for mission success.

2. Materials and Methods

We create two different models and compare their performances across various metrics. One approach, which we will call the Route Selection Algorithm (RSA), is based on the model by Dadkar *et al.* We first use Yen's Algorithm to identify many possible routes to minimize the risk posed by enemy forces and then use an MILP to select a subset of dissimilar routes to return to the user [6]. The second approach, which we will call the Route Generation Algorithm (RGA), is based on Hu and Chiu's model. It will use Dijkstra's algorithm to identify the optimal route according to a preference index, and then apply a penalty to any arc previously used [15].

These two approaches represent novel use cases for the previous uses in both models. The original work in the model by Dadkar *et al.* focused on mitigating exposing surrounding populations to toxic waste transported via rail while balancing transit time [6]. The original work in Hu and Chiu's model focused on minimizing transit time and route similarity in a civilian transportation context [15]. Both of these models were focused on civilian and commercial applications of these approaches. Our work is unique from these original models by incorporating the risk posed by enemy forces to serve as an additional factor to consider. The modifications we introduce in our RSA and RGA models introduce this new factor, creating a military application for these approaches.

2.1. Risk Modeling

Both approaches require us to quantify the risk posed by enemy forces and enemy spotters. To model this, we first define risk as the probability that enemy actions prevent friendly forces from traversing a route or arc due to sufficient damage. We assume that enemies need to have both visual observation and that their weapons be within the maximum effective range of a node or arc to pose this risk; having one without the other would prevent coordinated fires, and thus be ineffective in impacting friendly travel.

First, we need to determine the risk posed at each individual node. To do this, we need to calculate the probability of an enemy visually detecting or accurately targeting a friendly force at each node. For both cases, we use EQN 1 shown below. $P(D_i)$ represents the probability that friendly forces at node i are detected, and $P(K_i)$ represents the probability that friendly forces at node i are accurately targeted by a weapons system. M is the maximum range of the spotter or weapons system, and d_i is the distance from the node to a given enemy weapon or spotter. This is a logistic curve, and we make several assumptions: there is a 99% accuracy rate at the minimum distance, a 50% accuracy rate at half the maximum distance, and a 1% accuracy rate at the maximum range. This parametrization is intended to be somewhat arbitrary. There are too many factors, such as the

type of weapons system, weather, and visibility due to the time of day that would influence the ability of a weapons system to accurately target or an observer to observe friendly locations. Therefore, we choose to make a standardized equation and apply it to all types of weapons systems and observers. This is intended to act as a stand in; a more accurate model for a specific enemy weapons system could easily be substituted if available.

For each node i, we calculate $P(D_i)$ and $P(K_i)$. It should be noted that each arc does not contain just two nodes; each arc is broken up into roughly equidistant segments by internal nodes to capture the non-linearity of the arcs. The output of this equation is a value between 0 and 1. A 0 represents the enemy spotter or weapons system is out of range, and cannot observe or target friendly forces, and a 1 represents extremely close proximity to enemy forces, which would almost guarantee an engagement.

$$P(D_i), P(K_i) = \frac{1}{1+e^{\frac{2\ln(99)}{M}\left(d_i - \frac{M}{2}\right)}} \tag{1}$$

Following this, we need to calculate the probability of detection or accurate targeting from all enemy forces at a single node. To accomplish this, we calculate $P(D_n)$ and $P(K_n)$, which represent the probability that at least one enemy will observe or accurately target the friendly force located at node n. These are depicted below in Equation (2) and (3) below. In these equations, s represents all enemy observers with a nonzero value of $P(D_i)$ at node n, and e represents all enemy weapons systems with a nonzero value of $P(K_i)$ at node n. These equations allow us to calculate the probability that there will be accurate targeting or detection from all possible enemy spotters or weapons systems at node n. This allows us to condense the individual probabilities from all enemy elements into two single probabilities for each node.

$$P(D_n) = 1 - \prod_{i=1}^{s} \left(1 - P(D_i)\right) \tag{2}$$

$$P(K_n) = 1 - \prod_{i=1}^{e} \left(1 - P(K_i)\right) \tag{3}$$

Finally, we calculate the probability that an enemy force will both detect and accurately target friendly forces along a given arc. This is shown below in Equation (4). There, $R_{x,y}$ is the probability that the enemy force will successfully target a friendly force traveling along the arc contained by nodes x and y. In this equation, n represents all the nodes within that arc. $R_{x,y}$ will take a value between 0 and 1, where 0 represents a completely safe route, while 1 represents an extremely high chance of destruction due to enemy actions.

$$P(R_{x,y}) = 1 - \prod_{i=1}^{n} \left(1 - P(D_n) \cdot P(K_n)\right) \tag{4}$$

2.2. RSA Model Formulation

Below, we outline the three-step process in the RSA (Route Selection Algorithm)

model, which we base off the model created by Dadkar *et al.* [6]. First, we use a modification of an existing KSP algorithm to generate K routes that minimize exposure to enemy risk. Then, we quantify the dissimilarity between each pair of routes. Finally, we use an MILP to minimize the maximum similarity between the N returned routes to ensure spatial dissimilarity between selected routes. This process is shown below in Figure 3.

Route Identification

For our RSA model, we first implement a KSP algorithm to generate many different viable routes. A KSP algorithm will find the K best routes in a given network, not just the best route. One common KSP algorithm is Yen's K shortest path algorithm [16]. This algorithm was first created in 1971 by Dr. Jin Yen and has since been adapted to many different applications within vehicle transportation problems [16]. This algorithm works by iterating through Dijkstra's algorithm multiple times. First, it finds the most optimal route. It then sequentially splits this route at a node. The first segment of the route is untouched. The second route is removed from the network, and Dijkstra's algorithm is used to calculate the optimal route from the end of the first segment to the destination. This process runs sequentially until the K best routes have been identified.

In our model, we make two substantial changes. First, we minimize according to the risk posed by the route, not the time it takes to traverse it. Second, we implement a user-defined time constraint, which is not present in Yen's Algorithm [16]. This allows the commander to limit the total travel time for each generated route, simulating an operational time constraint. This algorithm will create K candidate routes, where K is a hyperparameter.

Dissimilarity Index Calculation

Once we generate K candidate routes, we need to define a way to compare the dissimilarity of each pair of routes. This is accomplished below in Equation (5), where r and r^* represent two different routes, $L(r)$ represents the length of route r, and $L(r \cap r^*)$ represents the length of all arcs contained in both r and r^* [17]. The output of $S(r, r^*)$ is a similarity index between 0 and 1, where 0 indicates routes with no shared segments and 1 indicates identical routes. This metric is calculated for each possible pair of routes generated from the KSP algorithm.

$$S(r, r^*) = \frac{\frac{L(r \cap r^*)}{L(r)} + \frac{L(r \cap r^*)}{L(r^*)}}{2} \tag{5}$$

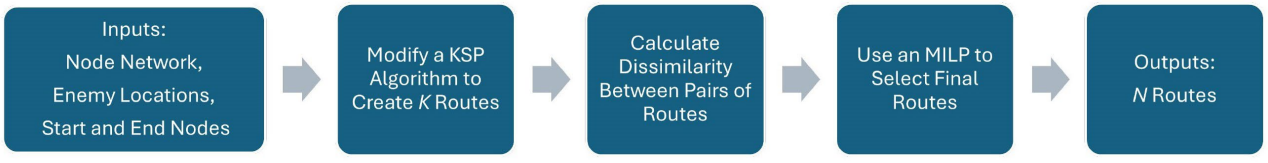

Figure 3. This graphic shows the process used in the RSA model.

Selecting Dissimilar Routes

Finally, we select a subset of N routes based on the calculated similarity metrics. To do this, we use a MILP to minimize the maximum similarity between any two selected routes. First, we define the variable x_r below in Equation (6). This models whether a route is selected to be returned to the user.

$$x_r = \begin{cases} 1 & \text{if route } r \text{ is in the final subset of routes,} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Next, we define the MILP in Equations (7), (8), and (9) below. In Equation (7), we minimize s, which represents the largest similarity index between any two selected routes. In Equation (8), we select a total of N routes to return to the user. In Equation (9), we ensure that the value of s is the largest shared similarity index between any two selected routes. This ensures that all N routes have a baseline level of dissimilarity with the selected other routes. This prevents a scenario where certain returned routes are very dissimilar, while others are extremely similar, and ensures a diverse network of routes are returned.

$$\text{Minimize} \quad s \tag{7}$$

$$\text{subject to} \quad \sum_r x_r = N \tag{8}$$

$$S\left(r, r^*\right)\left(x_r + x_{r^*} - 1\right) \le s, \quad \forall r, r^* \tag{9}$$

2.3. RGA Model Formulation

Below, we will outline the process used in our RGA model. We base this approach on a model by Hu and Chiu, where they generate spatially dissimilar routes that minimize travel time in an interstate system [15]. In this algorithm, they use Dijkstra's algorithm to first find the route that minimizes the time traveled between two nodes [15]. Then, they apply a penalty term to each arc contained in that route and recalculate the optimal route between the same start and end notes [15]. Unlike Yen's Algorithm, this algorithm does not remove any arcs from the network; instead, it just penalizes previously used arcs. This creates routes that tend to be suboptimal, but these routes are much more dissimilar than those created in Yen's Algorithm. Additionally, the runtime for this algorithm is substantially lower.

This model does not require a three-step process like the RSA model; rather, we generate a certain number of routes initially that do not require further selection. To generate each route, we use Dijkstra's algorithm and minimize according to a preference index, which we define as risk. We then apply a penalty factor to each arc that is used to generate subsequent routes. This process is shown below in **Figure 4**.

RGA Route Generation

For our RGA model, we first create a preference index. After each iteration, we apply a penalty factor to the preference index to indicate that the arc is less preferable in order to reduce similarity among generated routes. The risk index cannot be used because the preference index needs to be modified once an arc is

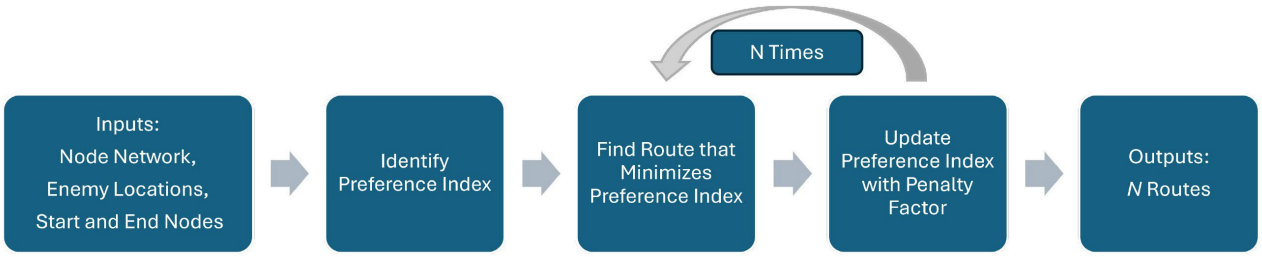

Figure 4. This graphic shows the process used in the RGA model.

used; we want to prevent the risk index from being directly modified to ensure that it reflects the risk posed purely by enemy forces. As a result, we set the initial values of the preference index equal to the values of the risk index; this ensures that we account for the risk when generating the routes, while ensuring that we do not modify the risk index values directly.

Following that, we need to identify the penalty factor, α. This is a value greater than 0, and if an arc is used, the preference index for that arc is multiplied by $(1+\alpha)$ to induce a penalty for using the arc in subsequent routes. If an arc has a risk value of 0, we assign it a preference index of 0.01. This is because if it remains at 0, then any penalty factor will not change the preference index; this would eliminate the intent of penalizing reused arcs.

For this model, we generate routes through a cyclic process as depicted in **Figure 4**. First, we find a route that minimizes according to the preferred metric by using Dijkstra's algorithm. We then update all preference indices that have been used in the route by the penalty factor. This is repeated N times to generate a total of N routes.

2.4. Dataset Used

For our dataset, we use publicly available data from Open Street Maps. This data represents an approximately 10 km by 10km road network in the town of Britburg, Germany and some of the surrounding countryside. This is intended to be a sample use case in a relatively small environment. In total, this area contains 1609 road arcs and 1314 road nodes. Additionally, we assume that each enemy weapon systems has a range of 5 km and each enemy observer has a range of 1.6 km. We do not consider the impact of buildings or terrain on these ranges.

2.5. Hyperparameter Value Selection

Before these two algorithms can be compared, we first must select values for the hyperparameters in each model. For RSA, we need to choose a value of K, or the number of routes returned by the KSP algorithm. For RGA, we need to choose a value of α, the penalty factor. We base the selection of these hyperparameters depend on the metrics for the returned routes from each model. This means that we examine how the mean similarity scores, mean risk scores, and mean routing algorithm run times vary across the returned N routes for each different hyperparameter value to determine which values of K and α to use in our final model.

Each scenario has 4 random enemy locations, 2 random spotter locations, and return 3 final routes. Additionally, for each iteration within the scenario, the route start and end location remain the same. These are kept constant to prevent any confounding variables from obscuring the results from the different hyperparameter values. We test out a total of 10 different values of both K and α for 10 trials each, for a total of 100 trials per model.

RSA

For the RSA model, we need to identify a value of K to use. We first explore how varying K will impact the mean similarity of the N selected routes. For clarity, it should be noted that we only examine returned routes, not all K routes generated by the KSP algorithm. The results of our simulation are shown in Figure 5; as the value of K increases, the mean similarity score for all routes decreases. This indicates that the routes become more dissimilar as the value of K increases, which is ideal; we want to ensure that the final routes are dissimilar.

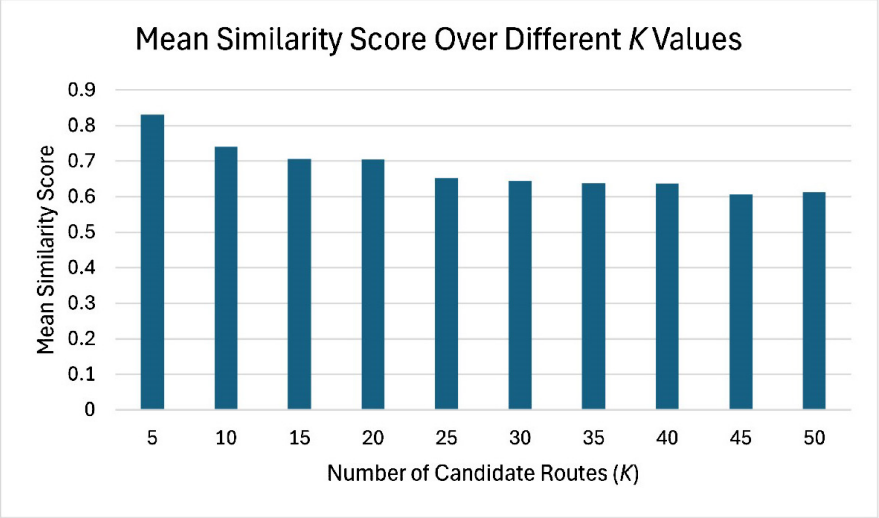

Figure 5. This figure shows the mean similarity scores for different values of *K*.

Additionally, we explore how the mean route risk score changes as K increases. The results from our simulation are shown below in Figure 6; as the value of K increases, the mean risk score for the N selected routes marginally increase. At $K = 5$, the mean risk score is 12.82, while at $K = 50$ it is 13.05. It should be noted that the vertical axis does not start at 0; it starts at 12.5 to highlight the slight increase in risk from each increase in K. This indicates that as the value of K increases, the final routes incur slightly more risk. This is not ideal, as risk should be avoided, but the magnitude of change is much less compared to the change in mean similarity scores.

Furthermore, as the value of K increases, the overall runtime of the route identification algorithm also increases. This is shown below in Figure 7. The mean runtime appears to increase linearly as the value of K increases. This is because a higher value of K will explore more possible routes, which increases the over-

all runtime. This is not ideal; a longer runtime will cause the user to wait longer for the results.

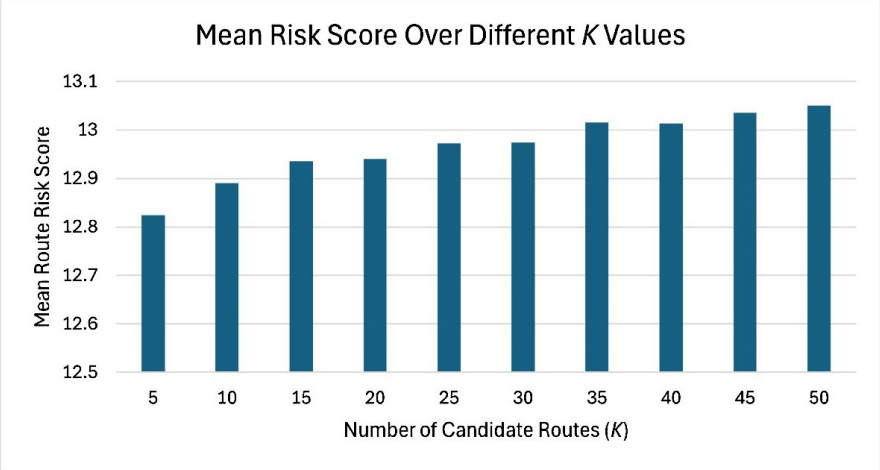

Figure 6. This figure shows the mean risk scores for different values of *K*.

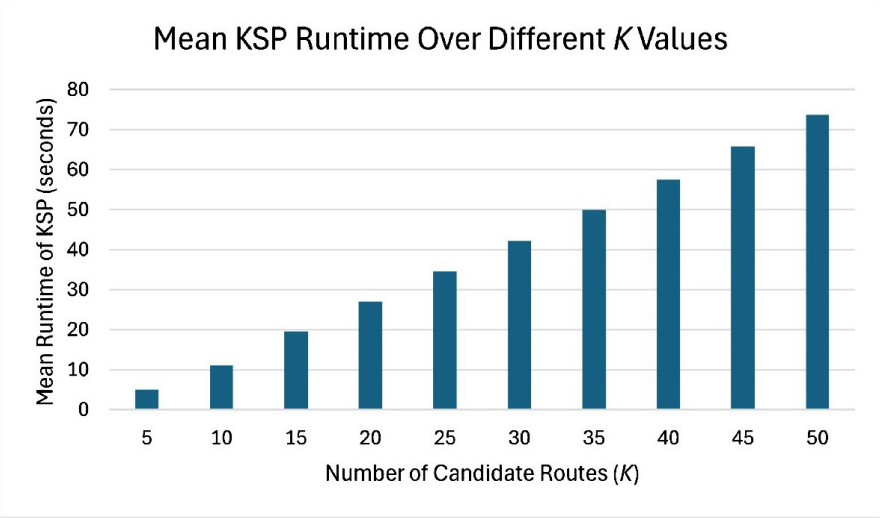

Figure 7. This figure shows the mean routing algorithm runtimes for different values of *K*.

Based on these results, we choose to use a value of $K = 50$. This is because this hyperparameter value will generate enough diversity in the routes while not having an unreasonable runtime. In Figure 5, the average mean similarity score decreases from around 0.8 at $K = 5$ to under 0.6 at $K = 50$. In Figure 6, the mean risk score increases from 12.82 to 13.05 as K increases from 5 to 50. In Figure 7, the mean runtime increased from around 4 seconds at $K = 5$ to 74 seconds at $K = 50$. This means that a large value of K is needed to ensure we have sufficiently diverse routes; a smaller value would result in routes that are just too similar. Additionally, 74 seconds is unlikely to threaten mission completion as it is unlikely to significantly delay mission planning. Though there is a slight increase in risk, this is not significant enough to outweigh the benefits of increased route

dissimilarity. Therefore, we choose a value of $K = 50$ because it will maximize the dissimilarity of our candidate routes among the values of K we examine.

RGA

To identify the best value of α, we explore the effect of different values of α on the mean route similarity, mean route risk scores, and mean routing algorithm runtime. First, we explore the relationship between α and the mean similarity score. The results are shown below in Figure 8. This indicates that as the value of α increases, the mean similarity score decreases; this is ideal for our model.

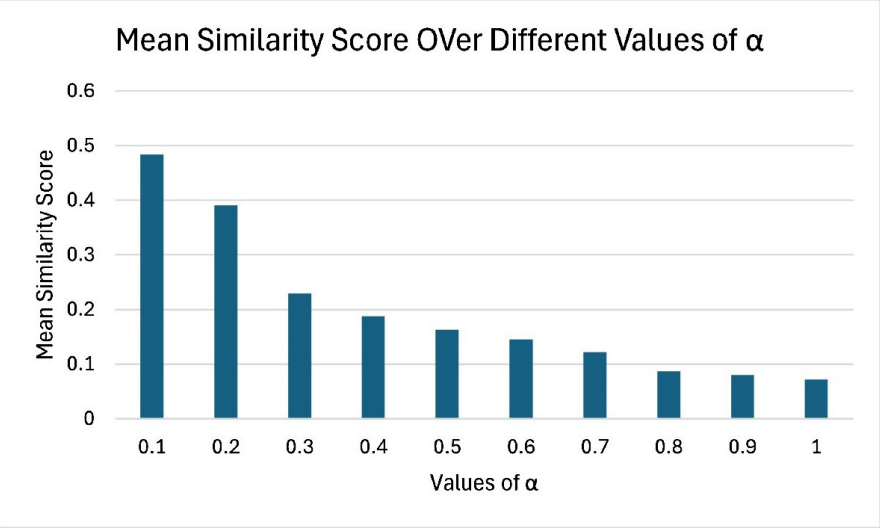

Figure 8. This figure shows the relationship between α and the mean similarity score.

Additionally, we explore the relationship between α and the mean risk score of the chosen routes. The results are shown below in Figure 9. This indicates that as the value of α increases, the mean risk score of the chosen routes appears to increase linearly. This is not ideal, as we want to avoid risk as much as possible.

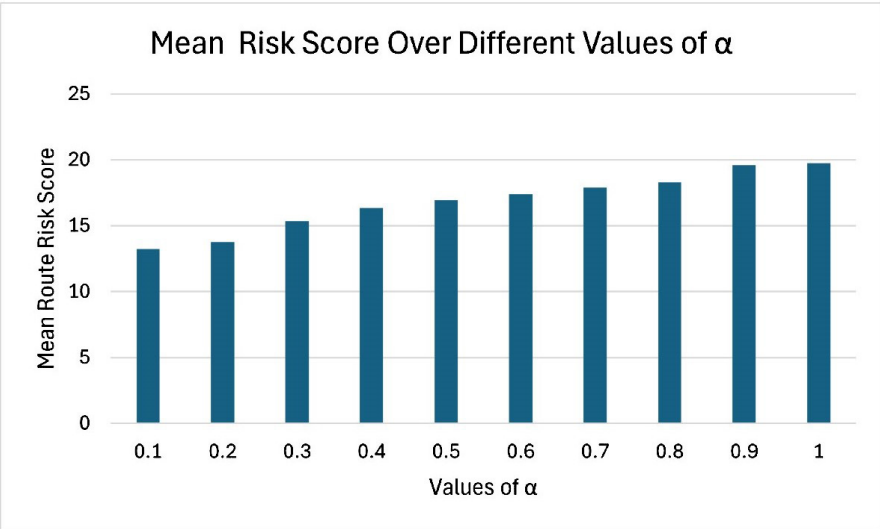

Figure 9. This figure shows the relationship between α and the mean risk score.

Furthermore, we explore the relationship between α and the mean routing runtime. The results are shown below in Figure 10. This indicates that as the value of α increases, the mean routing runtime generally decreases; however, the decrease is so slight, that it does not appear to be significant because they only vary by several hundredths of a second. Since the change is so small, this does not significantly affect our model.

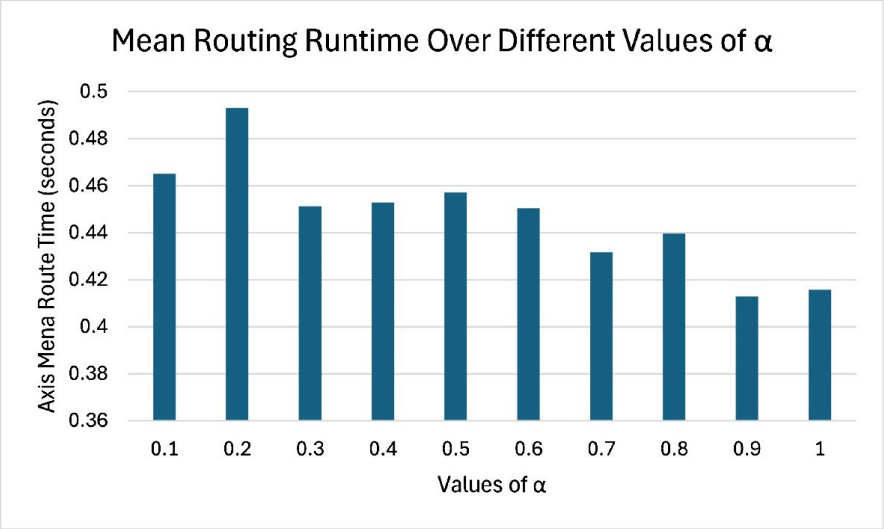

Figure 10. This figure shows the relationship between α and the mean routing runtime.

Based on these results, we choose to use an α value of 0.6. This is because it appears to balance the dissimilarity and risk score of the routes created. Based on Figure 8, the mean similarity score will vary between 0.5 and less than 0.1. At $\alpha = 0.6$, the mean similarity score is approximately 0.15. In Figure 9, the mean risk score varies between 13 and 20; at $\alpha = 0.6$, this score is approximately 17. In Figure 10, the mean routing runtime did not significantly change over different values of α. Given we want to achieve a low mean similarity score and a low mean risk score, $\alpha = 0.6$ appears to be an appropriate balance between these two competing factors.

3. Model Comparison

To evaluate our different approaches, we need to determine how each model compares in terms of runtime, mean similarity, and mean risk. First, we run 100 simulations on both models using publicly available GIS data. Each scenario has 2 randomly placed enemy weapon systems, 4 randomly placed enemy observers, and returns the $N = 3$ routes with the lowest risk scores. For the 100 iterations, the route start and end location remain the same. Next, we examine and collect the results from each model. Furthermore, the same preprocessing and risk calculation methods are used for both models. All scenarios used a K value of 50 for the RSA model, and α of 0.6 for the RGA model. Finally, we compare these results for statistical significance and form our final conclusions.

3.1. RSA Results

The results of the RSA model are shown in the appropriate row in Table 1. This indicates that the mean route risk score, out of all N selected routes, is 18.15. The mean similarity score between selected routes is 0.536. The mean total runtime for each iteration is 72.99 seconds. Finally, the mean routing runtime, or the mean runtime of just the route generation and route selection parts of the model, but not any preprocessing, is 62.00 seconds. An example output is shown below in Figure 11. In this image, the black lines represent the existing road network, the red circles represent the radius of the enemy weapon systems, the blue circles represent the range of the enemy spotters, and the pink circles represent the start and end nodes. The chosen routes are shown in green, purple, and orange.

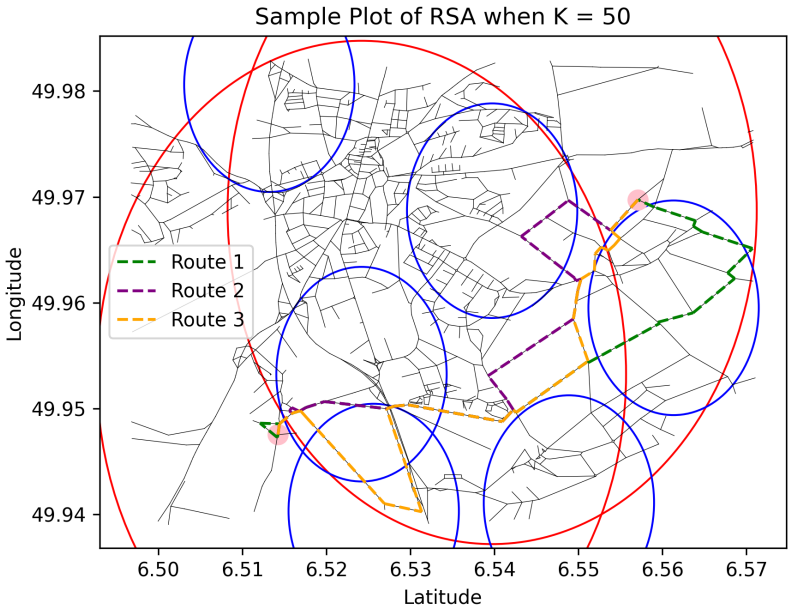

Figure 11. This figure shows a sample output of the RSA model. It is sample 95 from our 100 random scenarios.

Table 1. The above contains summary statistics from both models.

Model	Mean Risk Score	Mean Similarity Score	Mean Total Runtime	Mean Routing Runtime
RSA	18.15	0.536	72.99 (s)	62.00 (s)
RGA	22.58	0.135	12.97 (s)	0.467 (s)

3.2. RGA Results

The results of the RGA model are shown in the appropriate row in Table 1. This shows that the mean route risk score, out of all generated routes, is 22.58. The mean similarity score between selected routes is 0.135. The mean total runtime

for each iteration is 12.97 seconds. Finally, the mean routing runtime, or the mean runtime of just generating the three routes without preprocessing, is 0.467 seconds. An example output is shown below in Figure 12. The colors and shapes represent the same values shown in Figure 11.

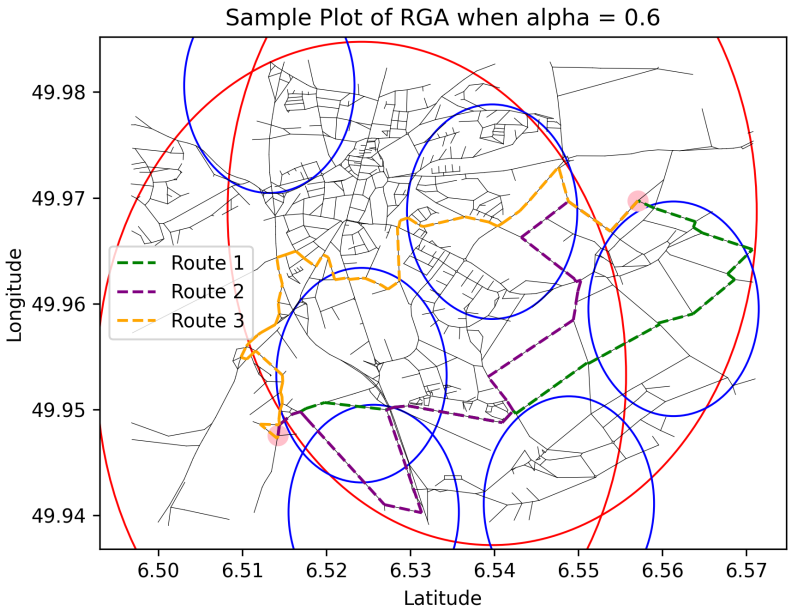

Figure 12. This figure shows a sample output of the RGA model. It is iteration 95 from our 100 random scenarios.

3.3. Model Comparison Analysis

To better understand the true difference between the results of the RSA and RGA models, we will conduct *t*-tests to determine if the difference in mean risk scores and mean similarity scores are statistically significant.

First, we examine the mean risk scores. A boxplot of the different mean risk scores between the models is shown below in Figure 13. Based on the plot, there appears to be a difference in the mean risk score between these models; the RGA Model has generally higher scores than the RSA model. This observation is validated with a Welch's two-sample *t*-test on these mean risk scores. The results return a *p*-value of 3.62×10^{-6}, which indicates that there is very strong evidence to suggest that the different means observed between these two models are statistically significant. Additionally, we calculate the 95% confidence interval for the true difference in the mean risk scores between the RGA and RSA models to be (2.60, 6.26). This represents we are 95% confident that the true difference in the mean risk scores between models is within that confidence interval. This further indicates that these two models will result in statistically significantly different mean risk scores, with the RGA model creating routes with a higher risk than the RSA models.

Next, we examine the mean similarity scores. A boxplot of the mean similarity scores between the two models is shown below in Figure 14. Based on the results,

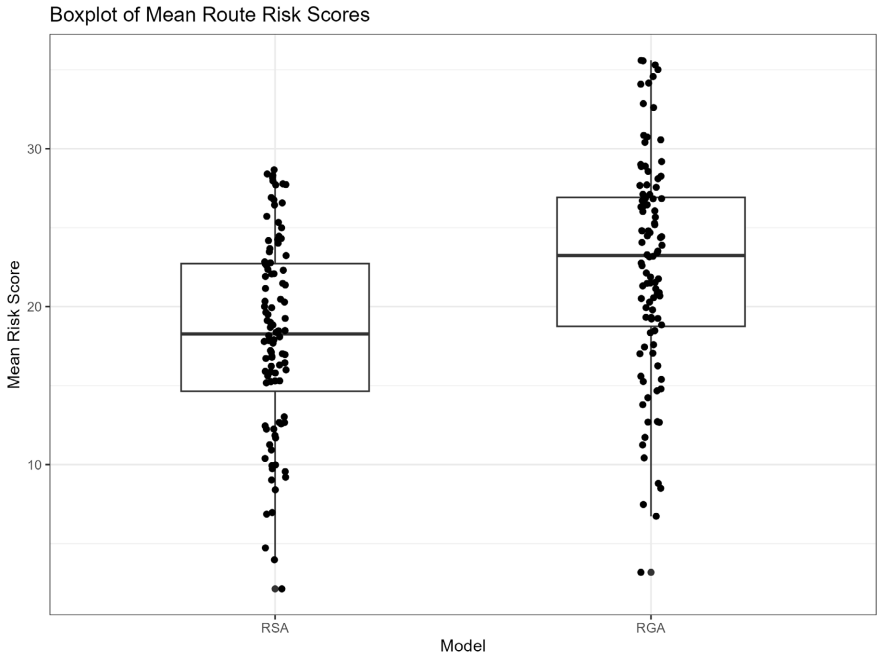

Figure 13. This boxplot compares the mean risk scores between the RSA and RGA models.

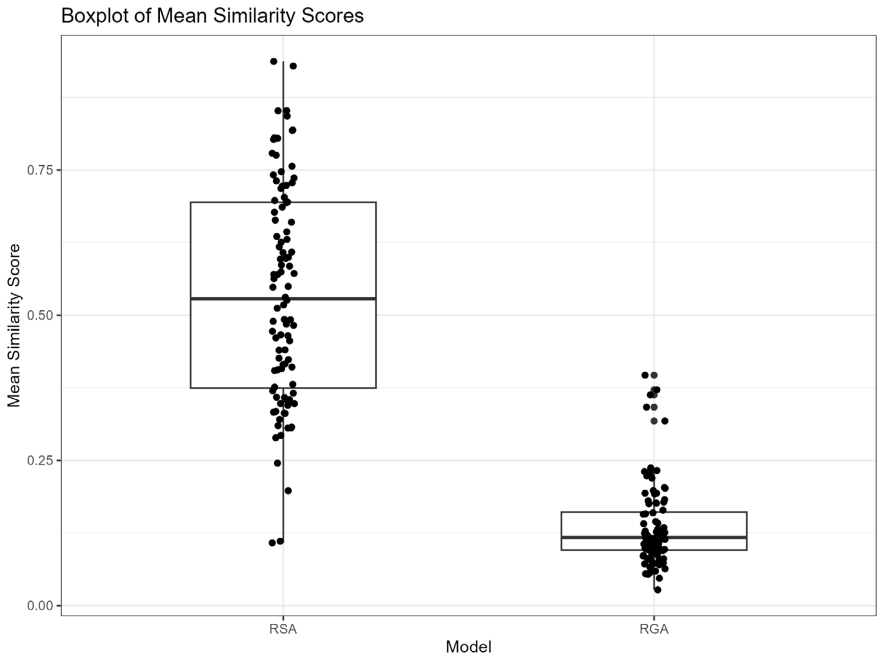

Figure 14. This boxplot compares the mean similarity scores between the RSA and RGA models.

the values in the RGA model appear to be significantly smaller than those in the RSA model. We further test this with a Welch's two-sample t-test to determine if the mean similarity scores are different between the models. The results give a p-value of 2.2×10^{-16}, which indicates there is overwhelming evidence to suggest these two models have different means. Furthermore, we calculate the 95% confi-

dence interval for the mean similarity score for the true difference in means to be (−0.441, −0.362). This indicates that the mean similarity score for the mean similarity score for the RGA is significantly lower than the score for the RSA model.

The difference in the mean routing runtime is likewise substantial. As depicted in Table 1, the mean routing runtime for the RSA model is 62.00 seconds, while it is 0.467 seconds for the RGA model. This indicates that the RGA is significantly less complex than the RSA model.

4. Discussion

Both models proved capable of successfully generating routes that minimized exposure to the areas where the enemy could both detect and effectively engage friendly forces. Additionally, the models generated routes that were dissimilar from each other to ensure route diversity. There were relatively low similarity scores, especially for the RGA model, which indicates that each generated route had substantial differences from the other generated routes.

Additionally, the results from the model comparison indicate that, although both models can generate the necessary routes, there exists a trade off between the models. The results of the *t*-tests indicate that the RGA model produces routes with a lower similarity, a quicker runtime, and higher risk. Likewise, the RSA model produces routes with a higher mean similarity score and runtime, but generates routes with a lower mean risk score. These differences do not indicate one model is exclusively "better" than the other, but rather indicate that the different models would be better for different uses. If a solution was needed very quickly, such as on a handheld tablet used to create new routes while under enemy fire, then the RGA model would be a better option. In the event where computation time is not an issue and the user prioritizes minimizing risk over dissimilarity, then the RSA model would be better. The right model choice depends entirely on the needs of the user in their specific situation. In the deliberate preliminary planning for an operation, there would be enough time to utilize the RSA model to ensure that the selected routes minimize as much risk as possible; if only 62 seconds are needed, then this would easily fall within a normal mission planning timeline. However, in the event that a decision must be made extremely quickly due to imminent enemy actions, then the RGA model would give an immediate answer to the user. Thus, our work provides multiple models that successful generate multiple routes for different situational needs.

Furthermore, it should be noted that the difference in routing times can be explained by the differing time complexities of the shortest path algorithms used in each model. In our RSA algorithm, Yen's algorithm is used, which has a time complexity shown in EQN 10, where K is the number of paths generated, N is the number of nodes in the network, and M is the number of arcs in the network [18]. In our RGA algorithm, Dijkstra's algorithm is used, which has a time complexity shown in EQN 11 [19]. This indicates that if the number of nodes in the network increases, then the time complexity will increase by an additional fac-

tor of N in the RSA model, resulting in it having a much longer runtime. Additionally, the number of nodes and arcs in the network will determine how these models scale in different environments, not the spatial size of the environment itself. It is possible to have a spatially larger area that has fewer nodes and arcs compared to the current one, which will result in both algorithms being completed in a quicker time. This could be achieved by limiting the arcs included in the dataset, such as only examining arcs that represent roads that can handle the pressure by large military vehicles.

$$O\left(KN\left(M + N\log N\right)\right) \tag{10}$$

$$O\left(M + N\log N\right) \tag{11}$$

5. Conclusions

Ultimately, our models can help address and automate a key planning task for military logistics. Route planning for military logistics is vital to ensuring operational success because it ensures that the necessary materiel will arrive reliably. A review of relevant work revealed several applications of dissimilar path routing, but none specifically addressed military logistics. We decided to create multiple models that generate routes that minimize the risk posed by enemy actions, can meet a time constraint, and maximize the dissimilarity between the routes. For risk, we calculated the probability that an enemy could detect and destroy forces along an arc. To create the routes, we used two approaches. Our RSA model used a KSP algorithm to generate many viable routes that minimizes risk subject to a time constraint. It then calculated how dissimilar each route was from each other, and finally a mixed integer linear program to maximize the dissimilarity between the selected routes. Our RGA model applied a penalty factor and iterated through several times to identify multiple routes to serve as a network. We determined the RSA model results in routes that incur less risk, and that the RGA model results in more dissimilar routes and has a shorter runtime.

The results of this work will allow commanders to automate the route planning process for military logistics. Rather than developing multiple route options by hand, without a systematic method, these models could be used by logisticians to quickly create and identify multiple possible routes between two locations. These routes will avoid areas where enemy forces pose a risk, thus increasing the likelihood of supplies being delivered to the final destination. Additionally, the dissimilar routes will ensure that, if any unplanned changes occur to the operational environment such as damaged roads or enemy movements occur, there will likely be another unaffected route that can still be used. Another possible use case is to send multiple convoys along each selected route. Due to the dissimilarity of the routes, this would reduce the probability that an entire supply convoy would be destroyed; it is more probable that the supplies on at least one or two of the routes will make it to the final destination compared to sending a single convoy. This approach will create more optimal routes, save

planners a substantial amount of time, and allow for more distribution among resupply operations.

Future work on this topic will focus on increasing the fidelity of both models, as they both currently are an initial prototype and several significant improvements can be made. Incorporating elevation data will allow us to better model the speeds that the arcs can be traversed and the range of the spotters and enemy weapons systems. This addition would likely reduce the risk levels posed by enemy forces when major terrain features are present. Since enemy small arms or observers could not effectively target friendly forces if their line of sight was broken by a terrain feature, such as a hill, then this would reduce the maximum range the weapons system or observer is effective. This would allow nodes that would otherwise have a probability of observation or effective fire on the other side of the terrain feature to no longer have these probabilities. Such a change should reduce the overall risk of the routes created by reducing the resulting risk indices among such nodes, and allow the routes to possibly utilize these locations to their advantage. Additional work includes reexamining the destruction and detection functions to make them more accurately reflect current operations and testing these models in varied environments. This will allow us to better understand the limitations of each model when there are more arcs and nodes and different numbers of enemy forces and spotters. Furthermore, these models could incorporate mobile enemies to account for roaming foot patrols, enemy convoys, and enemy UAS capabilities. This would increase the fidelity of our model by better capturing the change in risk over time due to enemy movement, and possibly allow for better routes to be created partway through a mission. Finally, rail and riverine travel methods can be incorporated to ensure we account for all possible modes of transportation available.

Disclaimer

The authors' contributions were performed as part of their official duties as employees of the United States Government. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Army, the Department of Defense, or the United States Government.

Funding Attribution

This material is based upon work supported by the US Army Engineer Research and Development Center (ERDC) under MIPR W81EWF41104389. The research described and the resulting data presented herein, unless otherwise noted, are supported under PE 622144, Project CV3 "Engineer Enablers for Contested Maneuver, Logistics, and Sustainment", Task "Planning Logistics Analysis Network System".

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] (2018) Maneuver Self Study Program? U.S. Army Maneuver Center of Excellence (MCoE). https://www.benning.army.mil/mssp/Logistics/

[2] Headquarters, Department of the Army (2023) Combat Support Sustainment Battalion (ATP 4-93.1).

[3] Headquarters, Department of the Army (2014) Army Expeditionary Intermodal Operations (ATP 4-13).

[4] Headquarters, Department of the Army (2020) Materiel Management, Supply, and Field Services Operations (ATP 4-42).

[5] Bates, A. (2023) Interview. Conducted by CDT Jack Hernon.

[6] Dadkar, Y., Jones, D. and Nozick, L. (2008) Identifying Geographically Diverse Routes for the Transportation of Hazardous Materials. *Transportation Research Part E*: *Logistics and Transportation Review*, **44**, 333-349. https://doi.org/10.1016/j.tre.2006.10.010

[7] Chang, T., Nozick, L.K. and Turnquist, M.A. (2005) Multiobjective Path Finding in Stochastic Dynamic Networks, with Application to Routing Hazardous Materials Shipments. *Transportation Science*, **39**, 383-399. https://doi.org/10.1287/trsc.1040.0094

[8] Liu, H., Jin, C., Yang, B. and Zhou, A. (2018) Finding Top-K Shortest Paths with Diversity. *IEEE Transactions on Knowledge and Data Engineering*, **30**, 488-502. https://doi.org/10.1109/tkde.2017.2773492

[9] Kang, Y., Batta, R. and Kwon, C. (2014) Generalized Route Planning Model for Hazardous Material Transportation with Var and Equity Considerations. *Computers & Operations Research*, **43**, 237-247. https://doi.org/10.1016/j.cor.2013.09.015

[10] Xie, Y., Lu, W., Wang, W. and Quadrifoglio, L. (2012) A Multimodal Location and Routing Model for Hazardous Materials Transportation. *Journal of Hazardous Materials*, **227**, 135-141. https://doi.org/10.1016/j.jhazmat.2012.05.028

[11] Wan, Q.K. and Lo, H.K. (2003) A Mixed Integer Formulation for Multiple-Route Transit Network Design. *Journal of Mathematical Modelling and Algorithms*, **2**, 299-308. https://doi.org/10.1023/b:jmma.0000020425.99217.cd

[12] Kendall, T.P., Killian, D.T. and Koch, M.J. (2023) Optimized Tactical Route Planning. *Military Operations Research*, **28**, 5-21. https://doi.org/10.5711/1082598328405

[13] Erkut, E. and Verter, V. (1998) Modeling of Transport Risk for Hazardous Materials. *Operations Research*, **46**, 625-642. https://doi.org/10.1287/opre.46.5.625

[14] Thyagarajan, K., Batta, R., Karwan, M.H. and Szczerba, R.J. (2005) Planning Dissimilar Paths for Military Units. *Military Operations Research*, **10**, 25-42. https://doi.org/10.5711/morj.10.1.25

[15] Hu, X. and Chiu, Y. (2015) A Constrained Time-Dependent K Shortest Paths Algorithm Addressing Overlap and Travel Time Deviation. *International Journal of Transportation Science and Technology*, **4**, 371-394. https://doi.org/10.1016/s2046-0430(16)30169-1

[16] Yen, J.Y. (1970) An Algorithm for Finding Shortest Routes from All Source Nodes to a Given Destination in General Networks. *Quarterly of Applied Mathematics*, **27**, 526-530. https://doi.org/10.1090/qam/253822

[17] Akgün, V., Erkut, E. and Batta, R. (2000) On Finding Dissimilar Paths. *European Journal of Operational Research*, **121**, 232-246. https://doi.org/10.1016/s0377-2217(99)00214-3

[18] Bouillet, E., Ellinas, G., Labourdette, J. and Ramamurthy, R. (2007) Path Routing in Mesh Optical Networks. Wiley. https://doi.org/10.1002/9780470032985

[19] Fredman, M.L. and Tarjan, R.E. (1984) Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms. 25*th Annual Symposium on Foundations of Computer Science*, 1984, Singer Island, 24-26 October 1984, 338-346. https://doi.org/10.1109/sfcs.1984.715934