An Implementation of the A* Search Algorithm for Dynamic Message Routing in Homogeneous Communication Satellite Networks

Alexandra N. Ramotar[1] and Mackenzie H. Lau[2]
Georgia Institute of Technology, Atlanta, GA 30332

The number of payloads delivered to space has increased dramatically over the past decade. However, the cost to access space has remained high because of strict requirements on system performance and reliability. One possible approach to relaxing system requirements and increasing access to space is the implementation of an intermediate constellation of dedicated communication satellites to facilitate satellite-to-ground and satellite-to-satellite message routing. A critical characteristic of such a constellation would be the method used to route messages across the network to the destination. The routing method must be robust to unknowable initial conditions and the evolution of satellite positions over time, while also being capable of accommodating potential variations in constellation architecture. This research adapted, implemented, and tested a version of the A* search algorithm for dynamic message routing across a constellation of homogeneous communication satellites to investigate the applicability of such a method to aid in analyzing the performance of such constellations for potential deployment in support of existing future space missions. A parametric comparison to a best-first routing algorithm showed significant improvements in terms of network reliability and expected time required to deliver messages.

I. Nomenclature

A*	=	A-Star Search Algorithm
dt	=	time propagation variable
E(x)	=	expected value
i	=	number in range [1:*n*]
MADM	=	Multi-Attribute Decision Making
n	=	number of values
n_cases	=	number of cases
nt	=	number of trials
NOAA	=	National Oceanic and Atmospheric Administration
p(x)	=	probability of x
S	=	standard deviation
TOPSIS	=	Technique for Order of Preference by Similarity to Ideal Solution
x	=	value in data set
x_i	=	*x* value at i^{th} position in the data set
$\bar{x}$	=	mean value

II. Introduction

Satellite capabilities are crucial to modern communication systems and proliferate the lifespan of space assets. Satellite-to-ground communication allow current assets in space to communicate with controllers on the

[1] Undergraduate Researcher, School of Aerospace Engineering, Atlanta, GA 30332, AIAA Student Member 1180473
[2] Faculty Advisor, School of Aerospace Engineering, Atlanta, GA 30332

ground and reduce costs and requirements for future space missions. The routing algorithm for these signals can greatly affect the time until the message is received, and optimization of these algorithms is necessary when the information is time sensitive. We will compare two popular methods for optimizing satellite communication: a greedy, best-first algorithm, Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), and an optimal path planning algorithm, the A-Star (A*) Search Algorithm.

Satellite communication routing methods impact the performance of the constellation, by metrics such as the time to route a message between origin and destination (transit time) and the number of connections needed in the path (number of nodes). The routing algorithms can affect the time for messages to be transmitted and the amount of data used in the constellation based on the number of nodes. The NOAA-18 weather satellite was selected as a candidate because the supporting constellation could extend its service life. Satellites like NOAA-18 need the ability to receive and transmit data in optimal times where the data it has collected may provide necessary warning information to areas affected by weather events. Optimizing the routing transmission will accelerate this transmission process and provide necessary information for disaster mitigation, scientific investigation, resource management, development planning, cartography, and road planning.[3]

III. Background

A homogeneous satellite constellation works by each satellite having the capabilities to communicate with other satellites, but when information can't be transmitted directly, it has to be sent through a pathway of satellites. The constellation comes into play by allowing every satellite to be able to contact every other satellite, directly or indirectly. When multiple nodes are needed to route a message, there are decisions that need to be made, like which satellite to send to as intermediate nodes. The routing algorithm makes these decisions, and different algorithms may produce different results depending on what factors are prioritized, like minimizing transit time or reducing the number of connecting nodes. In a homogeneous satellite constellation, the number of connecting nodes is directly correlated to the transit time. The messages are assumed to travel at the speed of light and the message download time is assumed to be constant at all satellites.

A. TOPSIS

The baseline routing algorithm for the original constellation used TOPSIS to determine a path for the message to follow. TOPSIS is a form of Multi-Attribute Decision Making (MADM) and finds the best alternative by considering different weighted criteria that affect the alternatives.[4] TOPSIS is classified as a greedy, best-first algorithm because it prioritizes the path that appears to be the most promising: the shortest in this case. However, TOPSIS picks the one that appears to be the most promising, whether or not it actually is. TOPSIS takes into account the distance from the positive ideal solution and the negative ideal solutions are determined by their relative proximity to the positive. This relative proximity creates a priority order which can be used for preference.[5] TOPSIS factors in different parameters based on their given weights, and for this experiment the weight was given to minimizing transit time.

B. A-Star Search Algorithm

The A* Search Algorithm is a routing method by looking for the least costly path between two points. A* works by locating nodes and edges in a given "maze" architecture. It is an application of the Breadth Search algorithm, which starts at an origin node and searches radially outwards to all points in the map. A* works by having an origin and a known destination, and searches for the least costly path between these two locations based on the maze architecture.[6] The heuristic function estimates the cost to get to each node, and the node with the lowest cost is selected. The heuristic function of A* is analogous to the weighting scheme in TOPSIS, so it's crucial to

---

[3] Wiryadinata et al., *Image Data Acquisition for NOAA 18 and NOAA 19 weather satellites using QFH antenna and RTL-SDR* 2018

[4] Rao, *Comparison of different MADM methods for different decision making situations of the manufacturing environment* 2012

[5] Rahim et al., *TOPSIS method application for decision support system in Internal Control for selecting best employees* 2018

[6] AlShawi et al., *Lifetime enhancement in wireless sensor networks using Fuzzy Approach and A-star algorithm* 2012

select uniform and valuable parameters as the weight for the success of each respective algorithm. A* accounts for cases of non-uniform mazes as well, by keeping track of all visited nodes in the path. If the current node could have been reached at an earlier point and was adjacent to a previous node, then the path is modified to its least costly possibility.[7] A* has the constant ability to retrace the path and determine if the current path is still the best possible option. This is an important property to have in analyzing dynamic simulations, where the maze architecture is constantly changing and decisions need to be reevaluated.

A* is currently being used as a machine learning tool to program a robot to find its own least costly path to a destination. The robot detects obstacles and when it is sent on the same path multiple times, it can identify the obstacle locations and avoid it. Results found that the trials using a modified A* allowed the robot to become familiar with its environment faster than the original decision-making algorithm.[8]

IV. Approach

The A* algorithm must be adapted to a 3D, non-uniform grid space in applications of message routing across satellite networks. With satellites, the message can be transmitted across empty space to another satellite if it is in transmission range, so our definition for "adjacency" changes too. Adjacency no longer means next to the node, but rather if the node can transmit to it. We do this by checking the line of sight (LOS) between the two satellites, and if the LOS is unbroken and the receiver is within transmission range, then we can transmit, and this node is considered adjacent. A* utilizes a priority queue, where it analyzes all possible nodes and puts the "best" one at the front of a queue. This is determined based on a heuristic function, which estimates the future cost of other possible nodes and prioritizes the minimum cost node. In our case, the cost is the transmission time, and the distance is used as a heuristic since the messages are assumed to travel at the speed of light. The transit time is minimal, but the message rate, or time for the message to be downloaded, will have the largest effect on the total time.

Simulations of various test cases are run to compare the effectiveness of A* and TOPSIS and ultimately determine which one is the most optimal for this satellite constellation. A constellation of fixed architecture is modeled for the static simulation. Static analysis only considers the constellation in one arrangement and evaluates the number of connections required to deliver the message. This model can be used for dynamic simulations as well, now with the additional factor of a propagating architecture at different points in time. For the static simulation, the time it takes to transmit the message does not impact the analysis since the constellation isn't changing during that time and are assuming every message can be transmitted almost instantaneously.

The process for single analysis is conducted by inputting the constellation architecture, customer satellite position, and ground station definitions. These parameters can be selected or randomized. With these known positions, we can run the simulation. For a static simulation, the constellation is taken to be constant at a fixed point in time. This means that we are assuming that the nodes chosen to transmit the message through maintain its ability to transmit since the constellation is not changing. A static simulation can be helpful in analyzing individual parameters' influence, by varying altitude while keeping a constant architecture. In a dynamic simulation, the constellation is moving and is propagated through time during the message's transit time. This leads to more room for optimizing the path, where the original path taken by the message may no longer be the most effective as the constellation architecture shifts. Dynamic simulations can also be used while varying altitude to create a more realistic model of a working satellite constellation. Both simulations work by having a known starting and ending point and using a predetermined heuristic to choose the least costly path. The message in this model starts off at the customer satellite and all other satellites are evaluated if they are adjacent. If the satellites are within each other's line of sight and transmission ranges, then adjacency holds. The least costly node is chosen and once the message is at the next point, it reevaluates if there was a quicker way for it to get there. A* makes use of a priority queue function which allows for the reevaluation and reprioritization of the current options until the next best possible node is the destination, which in this case is the ground station. Within each simulation, A* is compared with TOPSIS, another decision-making algorithm. The constellation parameters remain constant between each experiment for accurate comparisons between both methods. The results from this experiment can be used to inform future constellation architectures and decisions on satellite communication routing.

---

[7] Cheng et al., *Explaining decision-making algorithms through Ui* 2019

[8] Song & Ma, *Research on mobile robot path planning based on improved A-star algorithm* 2021

V. Results

A. Static Simulation

The static simulation shows the constellation at a single instant in time and is assumed to not change during each trial. Running a static simulation first as opposed to jumping directly to the dynamic simulation allows for the breakdown of the constellation architecture and isolating the effects of each individual parameter. This allows for a more in-depth view of constellation architecture for future planners to consider. Parameter ranges were based on the Center for Strategic and International Studies' article on typical orbit geometry[9]:

- Altitude: 500m ≤ x ≤ 1000m
- Inclination: 30° ≤ x ≤ 150°
- Number of orbits: 5 ≤ x ≤ 50
- Number of satellites per orbit: 1 ≤ x ≤ 20

For our primary analysis, the constellation has properties of altitude = 600m, inclination = 45°, number of orbits = 20, number of satellites per orbit = 15. These values were chosen to be a middle-ground and create a medium-sized constellation, all while having enough data points to analyze. The position of the ground station and customer satellite were randomly initialized and remained constant in our static analysis. These parameters produce results that show the path from the customer satellite to the ground station with varying connecting nodes.

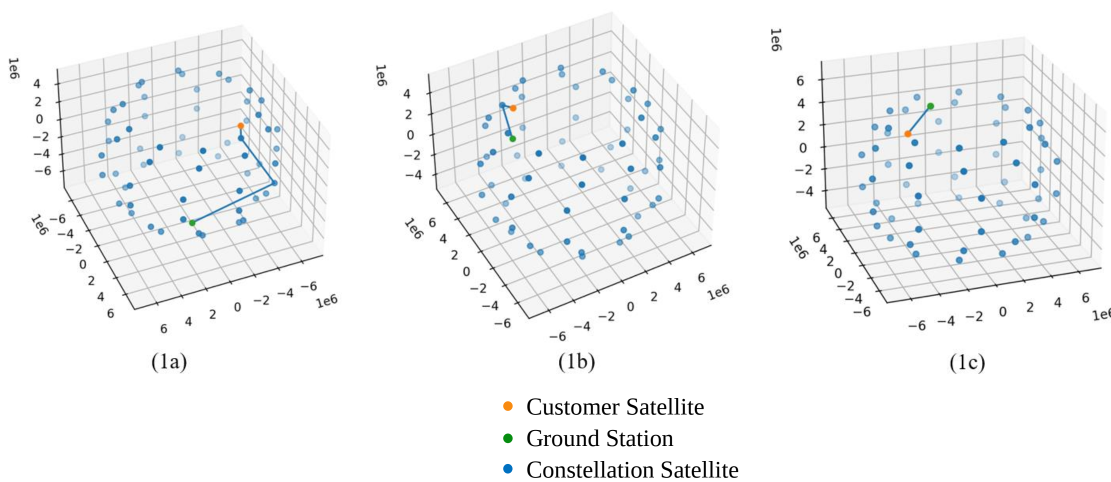


Fig. 1 Sample constellation paths between customer satellite and ground station using A*

Fig. 1 shows randomized simulations of paths between a customer satellite and a ground station. The constellation's architecture is fixed at the specified values, but the customer satellite is propagated at different potential locations. Figures 1a and 1b show successes where the message was successfully routed through the constellation with at least one connecting role. Figure 1c shows a customer link failure where the customer satellite has the ground station in its line of sight and could transmit directly without needing the constellation. In a static simulation, a customer link failure still allows for the transmission of the message, but in a dynamic situation, this could lead to the satellite needing to hold the message until the ground station is within adjacency or reroute the message from the previous node's location. Fig. 2 shows results from 50 trials in 3 separate A* simulations where the propagation value is randomly determined. The values shown are how long it takes to transmit a message from the customer satellite to a ground station. The results for expected transit time reveal to be finite multiples of the message download speed since messages are assumed to be sent at the speed of light. The results range from 0 to 3 intermediate nodes. Results of a transit time around 800 minutes indicate zero intermediate nodes, otherwise referred to as a customer link failure, where the constellation was not needed in the transmission. The other values were able to be transmitted using the constellation and A* and were found to have an expected value around 2000 minutes.

---

[9] Roberts, *Popular Orbits 101* 2022

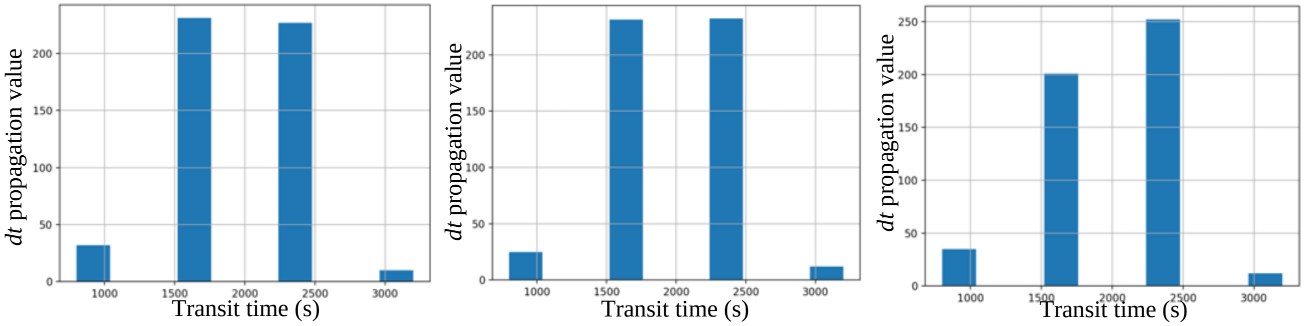


Fig. 2 Sample data visualization of 3 independent simulations calculating possible transit times with varying *dt* values

This simulation is run with identical constellation parameters using both A* and TOPSIS. The total transmission time is measured at randomized propagation values. Fig. 3 shows the cumulative probability mass function of the time to deliver the messages for both A* and TOPSIS. When time constraints are placed on the transit time, we can see that using A* has a higher probability of the message reaching its destination. For example, in the case of an extreme weather scenario, researchers need data acquisition from the NOAA-18 customer satellite within 10 minutes. It is shown that there is a 99% chance of the message being received within 10 minutes using A* as opposed to a 78% chance of the message being received within 10 minutes using TOPSIS. If the situation was not an emergency and transit time was not important, both routing algorithms will have the message transmitted by 40 minutes. A* performs better than TOPSIS within shorter times to deliver the message.

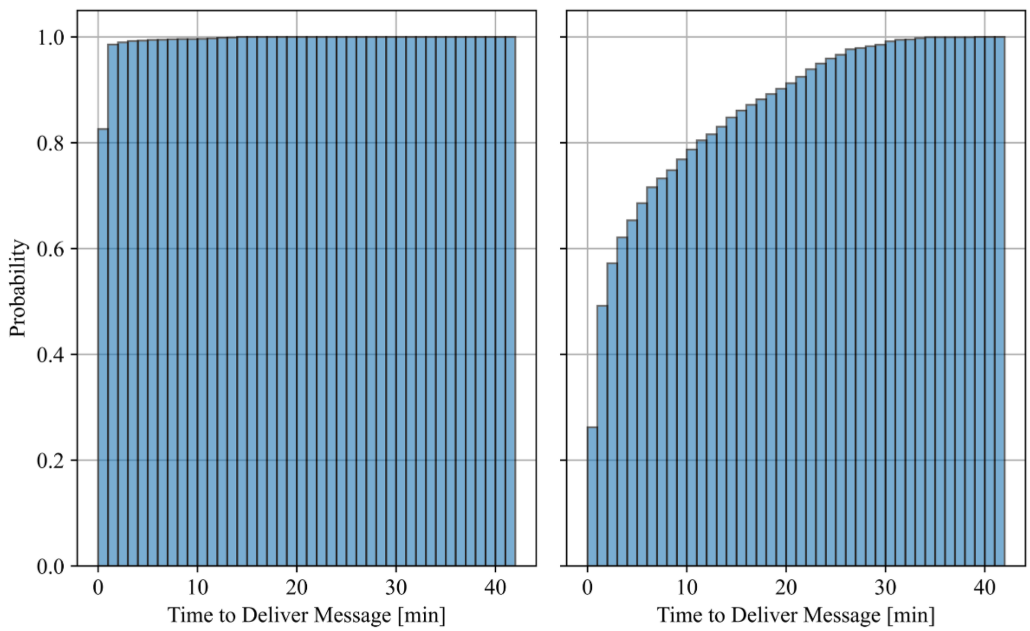


Fig. 3 Probability mass function of expected time to deliver message using A* (left) and TOPSIS (right)

i. Altitude Variation

Up to this point, the analysis has used fixed constellation architecture with parameters specified beforehand. We need to understand the effects of varying these parameters to see its impact on the constellation and then we can compare with TOPSIS. Analyzing each parameter by varying it within its range and holding the others constant will provide insight into how each parameter affects the overall constellation. This also allows us to compare specific cases with TOPSIS to see which is more effective for a range of cases. In these trials, the constellation has the same properties as the primary analysis for 3 of the 4 parameters while one is varying within its range of [500m, 1000m].

The ground station coordinates remain constant in these trials. The number of cases (*n_cases*) for each trial equals 150 while the number of times selected (*nt*) equals 75. Using the numpy.random function in Python, the altitude of the constellation will be varied between 500m and 1000m. The expected transit time between the customer satellite and the ground station indicates how many connecting nodes are needed to successfully transmit the message.

The experiment varied altitude between 500m and 1000m and produced 150 unique and randomized altitudes (*n_cases*). At each altitude point, we ran 75 trial simulations (*nt*) at different randomized altitude values to give a more accurate representation of how each new constellation system would behave. Within each *nt*, there was a randomized number of sub-trials with varying *dt* values between 100 and 10,000. This gives us between 1,125,000 and 112,500,000 data points per simulation.

Based on our data generated from this trial of 150 *n_cases*, the expected value of transit time between NOAA-18 and the ground station is 1959.2 seconds with a standard deviation of 1.78845 seconds. The range of this data set is 265 spanning from 1820 seconds to 2085 seconds (about 30.3 to 34.8 minutes).

$$E(x) = \sum x * p(x) \tag{1}$$

$$E(x) = 1959.2\ sec \tag{2}$$

$$S = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}} \tag{3}$$

$$S = 1.78845\ sec \tag{4}$$

We found that A* is able to transmit the message in all the tested cases with an expected transit time of 1959.2 seconds. A* is then compared to another routing algorithm, which in this case was chosen to be TOPSIS. We can run the same experiments using the TOPSIS algorithm instead of A* and compare the expected transit time. Fig. 1 shows three potential routes for an A* simulation while Fig. 5 shows three potential routes for a TOPSIS simulation.

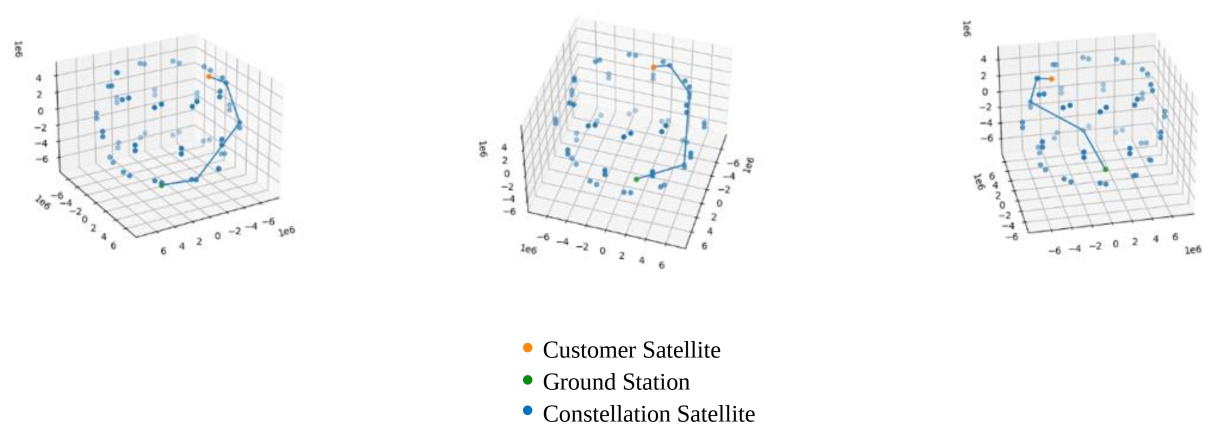


Fig. 5 Example constellation pathways between customer satellite and ground station using TOPSIS

Based on our data generated from our TOPSIS experiment of 150 *n_cases*, our expected value of transit time is 2432.6 seconds with a standard deviation of 3.77298 seconds. The range of this data set is 560 spanning from 2140 seconds to 2700 seconds. Aside from our simulated data, there were points where the TOPSIS simulations failed to find a path from the customer satellite to the ground station. These values were excluded from the calculation and simulations to create a more standardized and consistent comparison of experiments.

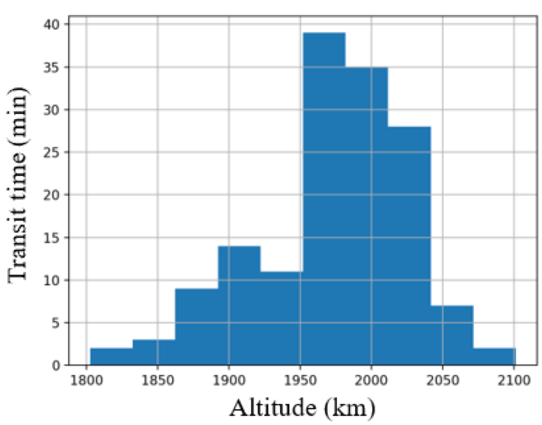
$$E(x) = \sum x * p(x) \tag{5}$$

$$E(x) = 2432.6\ sec \tag{6}$$

$$S = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}} \tag{7}$$

$$S = 3.77298\ sec \tag{8}$$

Fig. 6 shows a histogram comparison of A* simulations and TOPSIS simulations. In these trials, A* is shown to consistently outperform the transit time of TOPSIS in terms of transit time, with a significantly higher frequency of shorter times. These trials were run with identical inputs and respective heuristic functions prioritizing the least costly node. The results show that A* is more efficient than TOPSIS at finding the shortest and most efficient path in a given network, resulting in a lower transit time. The histograms show that the transit time for A* is more concentrated around the mean and has a lower standard deviation compared to TOPSIS. This shows that A* is more consistent in its performance, with less variability in the transit time. TOPSIS is shown to have a broader distribution of transit time, with a higher standard deviation, indicating that its performance is less predictable and more variable.



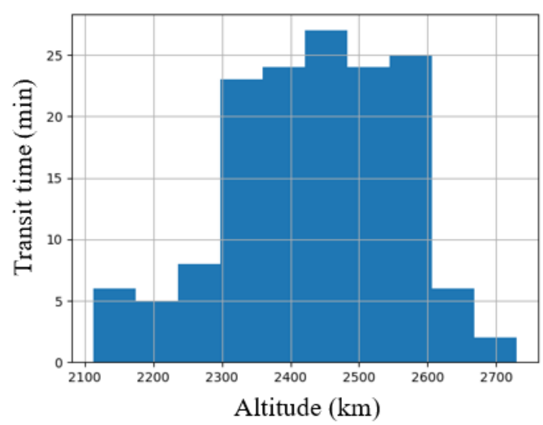


Fig. 6 Histogram of expected transit time vs. altitude of constellation system using A* (left) and TOPSIS (right)

Fig. 7 compares statistical properties of the A* and TOPSIS simulations. The average transit time of A* is shown to be significantly less than the average transit time of TOPSIS. Additionally, the A* transit times were more consistent than the TOPSIS times, with A* having a much lower standard deviation in each trial. A* mostly had a lower fail rate, with TOPSIS showing an increase in failures as the altitude increased. This is significant because it highlights the A* algorithm's robustness and ability to handle complex routing problems, even in difficult satellite configurations. The ability to minimize failures is crucial in ensuring the efficient delivery of messages, especially in critical applications such as emergency response, military communication, and satellite navigation. With these chosen metrics, A* outperformed TOPSIS in the static simulation for minimizing transit time as well as predictability. This observation suggests that the A* algorithm can produce consistent results and minimize variability, which is crucial in ensuring reliable and efficient message delivery.

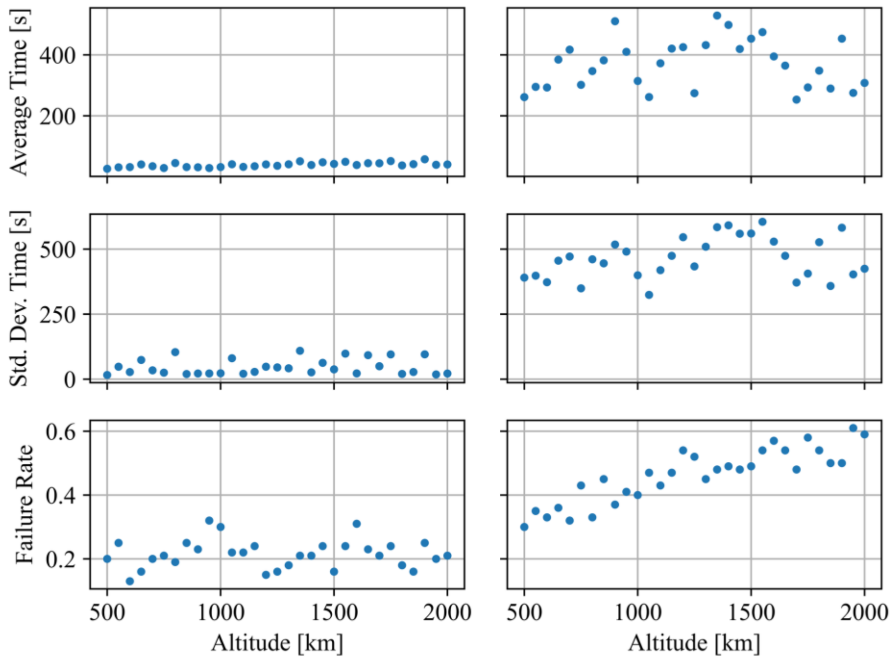


Fig. 7 Comparison of scatter plots showing transit time, standard deviation, and failure rate vs. altitude of constellation system using A* (left) and TOPSIS (right)

Fig. 8 shows a comparison of the outcome probabilities of A* and TOPSIS simulations. The consistent occurrence of a 20% no connection outcome between both A* and TOPSIS can indicate that the constellation parameters are not optimized for efficient message delivery and there needs to be more satellites for the constellation to become more effective. This is due in part to the arbitrary values picked for the satellite constellation, and future researchers can work to optimize this area. This percentage of failures highlights the importance of having adequate coverage to ensure connectivity and minimize message failures. Time expired refers to an arbitrary one-hour limit where if the message was not received within an hour, it was treated as a failure. In this scenario, there were no time expired outcomes, however it is worth noting that in sparser constellations this outcome may become more prevalent. A* is shown to have a greater percentage of successes compared to TOPSIS, with about 80% and 60% respectively. As altitude increases, A* is still able to transmit the messages consistently, while the number of customer link failures increases with TOPSIS as altitude increases.

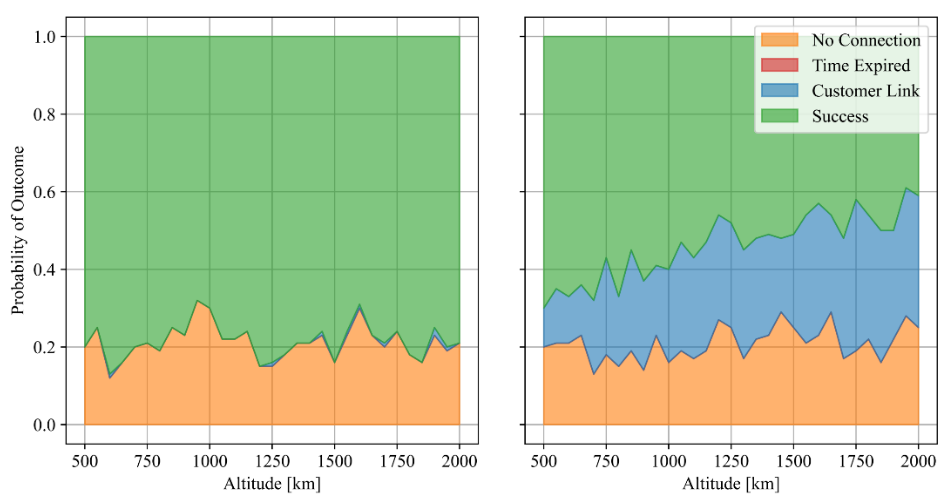


Fig. 8 Outcome probabilities of A* (left) and TOPSIS (right) routing at varying altitudes

In these static simulations, A* has consistently performed better than TOPSIS at various propagation values and varying altitudes. While both experiments have a significant percentage of no connection failures, this could be due to errors with the constellation's design and not the algorithms themselves. The results of this experiment highlight the need for an effective constellation architecture to achieve an optimal network, but it also shows the increased rate of successes of the A* algorithm compared to TOPSIS.

B. Dynamic Simulation

The A* algorithm's success could come from its ability to constantly reevaluate if the message is heading down the correct path. This phenomenon can be exemplified more in dynamic situations, where the constellation is propagated while the message is being transmitted. A dynamic simulation was run using the same input parameters as the static simulation, with the addition of propagating the constellation, ground station, and customer satellite. These inputs were propagated at randomized but consistent values. 100 trials at 11 different altitudes were run using both A* and TOPSIS to compare the routing algorithms in a dynamic situation as well. Fig. 9 shows the results from the dynamic simulation of varying altitude and measuring transit time using both A* and TOPSIS. Most of the A* times were below 100 minutes while the TOPSIS results were more varied. In dynamic simulations as well as static, A* appears to outperform TOPSIS in terms of minimizing transit speed.

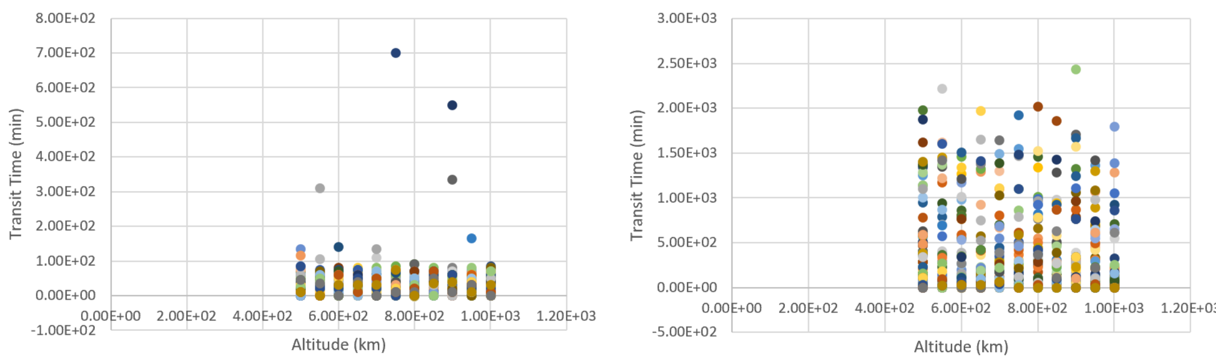


Fig. 9 Dynamic simulation of altitude vs. expected transit time for A* (left) and TOPSIS (right)

A dynamic comparison of A* and TOPSIS shows that A* has routed the messages faster and more reliably than TOPSIS had. This could be due in part to the heuristic function between the two algorithms. In this experiment, both the A* heuristic function and the TOPSIS weightings were treated as equal, but different algorithms prioritize different aspects and have different ways of ordering. For it to truly be equal, another sub-study should be conducted to compare TOPSIS and A* with differing heuristic functions to balance their weighting schemes.

VI. Conclusion

Our analysis of altitude variation using A* indicates that higher altitude reduces expected time. When the constellation altitude increases, the visibility of each satellite increases, with the Earth acting as less of an obstruction to the line of sight. Increasing the constellation altitude between 500m and 1000m shows a lower transit time between a customer satellite and the ground station. Too high of a constellation altitude would mean the satellites aren't within transmission range of each other, but between 500m and 1000m the constellations are still able to transmit and gain effectiveness as altitude is increased. The higher the constellation, the easier the satellites can transmit due to a better line of sight. This leads to less nodes being used to transit the message. Instead of needing to use multiple nodes to transmit around the Earth, there can be fewer to accomplish the same task at a higher altitude. The findings of this study can be valuable in improving the efficiency and reliability of message delivery in various applications, including satellite communication.

Our comparison between A* and TOPSIS shows that A* consistently outperforms TOPSIS. After our experiment using A* and TOPSIS as routing methods, we found that the expected transit time for A* is 1959.2 seconds and the expected transit time for TOPSIS is 2434.6 seconds. Based on our simulation, the A* expected value is 473.4 seconds faster than the TOPSIS expected value. The standard deviation for A* is 1.78845 seconds while the standard deviation for TOPSIS is 3.77296 seconds. The standard deviation for A* is smaller, which indicates that each value falls within a smaller range and is closer to the expected value than for TOPSIS. 95% of all transit times would fall within two standard deviations of the expected value. According to these calculations, 95% of all data points for A* would fall in the range [1955.6231, 1962.7769] while 95% of all data points for TOPSIS would fall in the range [2427.0508, 2442.1459]. Throughout all *n_cases*, A* has consistently performed better against TOPSIS, with a lower expected value for transit time and a smaller standard deviation and range. The expected value for A* is 475.4 seconds lower than expected value for TOPSIS, and based on our sample data, the maximum transit time for A* is still lower than the minimum transit time for TOPSIS. When operating weather satellites like NOAA-18, it's important to optimize all routing to ensure information gets transmitted as soon as possible. A smaller standard deviation indicates that the values for transit time in practice would also lie around the expected value. Each transit time in practice would be more predictable to follow, and in the case of NOAA-18, predictability is good because it can standardize the message transit time without having to rely on varying configuration at various *dt* values. Overall, our comparison between A* and TOPSIS shows that A* is more effective in most areas and should be implemented as the future satellite routing method for the proposed constellation.

This experiment was inspired by the proposed Phoenix FOSM-1 constellation and NOAA-18 satellites. While this research aimed to determine if A* or TOPSIS should be used to route the messages around the satellite, future research can be conducted to test the limitations of A*. In this analysis, the only parameter we tested A* against was altitude variation, but our constellation takes four inputs as part of its architecture: altitude, inclination, number of orbits, and number of satellites per orbit. Testing A* against the other 3 parameters can help future researchers shape the constellation architecture and determine whether A* should be used depending on the chosen architecture. The experiments involving the other parameters can work backwards to discovering the most optimal satellite configuration. The values in this experiment were arbitrarily selected based on a suggested range, but more research in this area could improve this range and create the most optimal satellite constellation for message transmission.

Additionally, the simulations created for this experiment were only done statically. To create a more accurate, dynamic simulation, the model produced would need to be time dependent. For any given constellation, we can run A* as normal, but simulate the intermediate constellation propagations while the message is being transmitted. While this occurs, the message location should be tracked along with the rate of download and transmission to monitor the message's status. At each intermediate node, the transmitter and receiver satellites must maintain connectivity and the line of sight must be maintained. During transmission, the connectivity should be monitored and in case of any interruption, the next best satellite to transmit to at that specific instant should be switched to. The A* algorithm should be expanded to account for this such scenario, as well as if a line of sight is interrupted, if it's more effective to hold onto the message and transmit at a later point or find an alternative, longer path and continue to transmit. This would involve propagation of the constellation at every 1-5 seconds for a more accurate representation and a way to monitor the percentage of the message downloaded and transmitted instead of propagating once it reaches a new node. A fully dynamic simulation that would resemble the actual constellation would require A* to be performed at set intervals during the messages' transmission instead of waiting until it reaches a node.

This research only compared A* with TOPSIS, but there is a multitude of other routing algorithms to choose from, each with their own benefits. In the case of a dynamic, 3D satellite constellation, A* could still potentially not be the best possible option, and future researchers should explore other possibilities.

References

Periodicals

[1] AlShawi, I. S., Yan, L., Pan, W., and Luo, B., “Lifetime enhancement in wireless sensor networks using Fuzzy Approach and A-star algorithm,” IET Conference on Wireless Sensor Systems (WSS 2012), 2012.
doi: 10.1049/cp.2012.0611

[2] Cheng, H.-F., Wang, R., Zhang, Z., O'Connell, F., Gray, T., Harper, F. M., and Zhu, H., “Explaining decision-making algorithms through Ui,” Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019.
doi: 10.1145/3290605.3300789

[3] Rahim, R., Supiyandi, S., Siahaan, A. P., Listyorini, T., Utomo, A. P., Triyanto, W. A., Irawan, Y., Aisyah, S., Khairani, M., Sundari, S., and Khairunnisa, K., “TOPSIS method application for decision support system in Internal Control for selecting best employees,” Journal of Physics: Conference Series, vol. 1028, 2018, p. 012052.
doi: 10.1088/1742-6596/1028/1/012052

[4] Rao, R. V., “Comparison of different MADM methods for different decision making Situationsof the manufacturing environment,” Springer Series in Advanced Manufacturing, 2012, pp. 205–242.
doi: 10.1007/978-1-4471-4375-8_7

[5] Song, Y., and Ma, P., “Research on mobile robot path planning based on improved A-star algorithm,” 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), 2021.
doi: 10.1109/eiecs53707.2021.9588002

[6] Triantaphyllou, E., “Multi-criteria decision making methods,” Applied Optimization, 2000, pp. 5–21.
doi: 10.1007/978-1-4757-3157-6_2

[7] Wiryadinata, R., Khoirussolih, M., Rohanah, N., Muttakin, I., and Firmansyah, T., “Image Data Acquisition for NOAA 18 and NOAA 19 weather satellites using QFH antenna and RTL-SDR,” MATEC Web of Conferences, vol. 218, 2018, p. 02002.
doi: 10.1051/matecconf/201821802002

Websites

[8] Roberts, T. G., “Popular orbits 101,” Aerospace Security Available: https://aerospace.csis.org/aerospace101/earth-orbit-101/#:~:text=The%20majority%20of%20satellites%20orbiting,full%20orbit%20around%20the%20Earth.