Simulations for Wildfire Response Optimization

CSC 446 Project Report

Simulating various methods of wildfire response

Arshiya Shams
Computer Science
University of Victoria
Victoria, BC, Canada
arshiyashams0@gmail.com

Colby Musialek
Computer Science
University of Victoria
Victoria, BC, Canada
colbymusialek@uvic.ca

Devin Frioud
Geomatics
University of Victoria
Victoria, BC, Canada
devinfrioud@gmail.com

1 Introduction

Wildfires are a complex natural phenomenon that pose a significant threat to ecosystems, infrastructure, and human lives. Though they can be a beneficial natural occurrence for ecosystems, the increased damage caused by human impacts on the environment as well as the need to maintain infrastructure justifies fire control response. Managing the strategies adopted for responding to fires is essential in order to minimize the total area burned and its associated damage.

There are various strategies that can be used in combating fire. Strategies must account for the prioritization of fires based on size and intensity, as well as the impact of different queuing policies. Simulating these scenarios can provide valuable insights into which strategies are most effective under varying conditions.

The primary goal of this simulation study is to evaluate and compare the effectiveness of different firefighting dispatch strategies. This is all with a priority to minimize the amount of area burned by fires. Specifically, the study seeks to inform better resource allocation policies and improve decision-making for wildfire management, ultimately reducing the destruction caused by wildfires.

2 Simulation Model for Wildfires

Wildfire management can be successfully mapped to a priority queuing model in which the wildfire response teams act as servers and the fires as customers awaiting service. With minor modifications, this approach allows simulations of various resource allocation strategies.

The fires were treated as customers but granted additional attributes representing various conditions in the fires.

- **Size:** In square hectares, the area a fire takes up. This attribute determines the service time and can grow dynamically based on time in the queue. It is provided based on our distribution representing the size at time of being reported.
- **Intensity:** The intensity of the fire encompasses various factors which affect fire growth. This includes fuel material, weather conditions, topography, and a variety of constituents of fire behaviour.

The wildfire response teams are representative of servers and behave very similarly. Between our 3 simulation models we have altered the queueing techniques to mimic alternate wildfire response strategies. For all simulation models we operate under the assumption that we are covering an area equivalent to 1/10 of British Columbia, Canada, which contains 3 wildfire response teams, based on BC's ability to concurrently handle 30 wildfires [2].

In BC, wildfires are handled as full responses or modified responses, depending on threats to public safety, property, the presence of infrastructure or timber, and a variety of other factors. In full response, the fire is controlled until fully extinguished. In modified response, fires are managed to maximize the ecological benefits of the fire and minimize damage to property and infrastructure. For the purpose of this simulation, all fires are treated as full response.

The goal of this simulation was to model wildfires and wildfire response across a subset of BC. Interarrival times of fires and fire sizes were modeled from existing data. Time to extinguish and the growth rate of fires were based on simplified models derived from empirical evidence, as fire behaviour is difficult to predict within the structure of this simulation.

2.1 Data Collection and Use

Wildfire behaviour and prediction are a huge field of study. With this simulation, our goal was to focus on effective response strategies, over perfecting fire modeling. Fire behaviour is inherently highly variable and unpredictable due to countless environmental factors. While the fire behaviour in this simulation has been heavily based on existing data and empirical evidence when available, some elements of behaviour were simplified to fit the broader simulation framework.

Input modeling was performed on data from the Canadian National Fire Database (NFDB), for selected years between 1992 and 2021, for the month of August. August was selected as it typically saw the greatest number of fires and greater area burned, and was therefore optimal for modeling extreme conditions at the peak of the fire season. The years selected were based on the mean behaviour of the past thirty years; outliers with metrics that differed from the expected metrics were not taken into account. Fires considered in the input modeling were limited to those greater than 2 hectares - fires smaller than 2 hectares were limited in accuracy and data availability over the full time period.

Fire sizes over the selected years were fit to the pareto distribution, with scale and shape parameters of 2.10 and 0.42 respectively. Performing a KS-test of the input data against the pareto distribution led to a p-value of 0.15, allowing us to fail to reject the null hypothesis that the input data follows a Pareto distribution, with a confidence level of 0.05. Initial fire sizes were modeled in the simulation according to the pareto distribution, with an upper limit of 250,000 hectares to match empirical values; the largest fire over the input data was approximately 236,000 hectares. Fire growth in the queues was also calculated, using a simplified model built around initial fire size, wait times, and fire intensity. Empirical evidence has shown that large fires exhibit greater potential for growth than small fires, which is reflected in our model [3]. Growth was calculated according to the following formula, and summed with the initial fire size for each individual fire: $growth\ rate\ =\ intensity\ \times\ wait\ time\ \times\ (1\ +\frac{size}{200})$

Figure 1: Pareto distribution fit to fire sizes

Determining the intensity of fires was more complicated. As the system does not account for fuel type, wind speed, moisture, and various other parameters that dictate wildfire behaviour, the intensity system was designed to accommodate a high level of variability. The intensity system is based on Crown Fraction Burned (CFB), a metric employed in the Canadian Forest Fire Behaviour Prediction (FPB) system [5]. CFB gives a number between 0 and 1 corresponding to the fraction of tree crowns consumed by the fire. When fire events are generated, an intensity score between 0 and 1 according to an exponential distribution with lambda 1 is assigned. This intensity score takes into account the rarity and increased severity of crown fires [4]. Wildfires are considered surface fires between 0 and 0.1, intermittent between 0.2 and 0.8, and crown fires at CFB's greater than 0.9. At a value greater than 0.9, the intensity is increased. In the FPB, CFB is based on Buildup Index, Foliar Moisture Content, Surface Fuel Consumption, and Rate of Spread. These indices track fuel available for combustion, moisture in plant material, the amount of fuel consumed by a fire, and the speed of the head of the fire, respectively. By simulating one index taking into account all of these metrics, fire intensity can be modeled without the far more complex landscape simulations required to adequately simulate the various factors of wildfire behaviour.

Service times, defined as the time to extinguish a wildfire, were determined based on a simplified model based on intensity, size, and a base response time. The base response time is intended to simulate the time for wildfire crews to arrive on site - we selected a base response time of 30 minutes based on known response times within BC [2]. Size also plays a role in the service time calculation, as large fires take more time to control. Finally, fire intensity is a key metric for calculating service time. Service times were calculated by adding together the base response time and a factor of the size of the fire, and then multiplying those by the intensity of the fire. This provides us with an estimation of the duration from a reported fire to full control, under a full response approach. Initial efforts to determine a simpler relationship between fire size and duration were made difficult by limited data in the NFDB dataset. No fires in BC tracked the date the fire was put out, and the majority of data from other provinces primarily tracked smaller fires receiving a modified response, and some very large fires. Attempts to predict service times based on fire size through linear regression were ineffective and inconclusive, making generating an accurate and scalable relationship more difficult. Furthermore, specific time to extinguishing of fires depends greatly on the location, terrain, fuel availability, and a variety of other fire metrics that aren't considered. Basing the service times on a simplified model allows these metrics to be accounted for indirectly, and retaining the variability in the time to extinguish wildfires.

Modeling the interarrival times of fire events was complicated by the low temporal resolution of the NFDB dataset, in which fire dates were recorded only in days. Fire counts over time daily time intervals calculated directly from the input data were fit to a poisson distribution, which described the number of fire events over a time interval of days, with $\lambda = 0.44$. Performing a KS-test on the intervals generated a p-value of 0.9307, allowing the null hypothesis that the counts are Poisson distributed to fail to be rejected. From this, an exponentially distributed model of interarrival times was determined, with $\lambda = 2.27$.

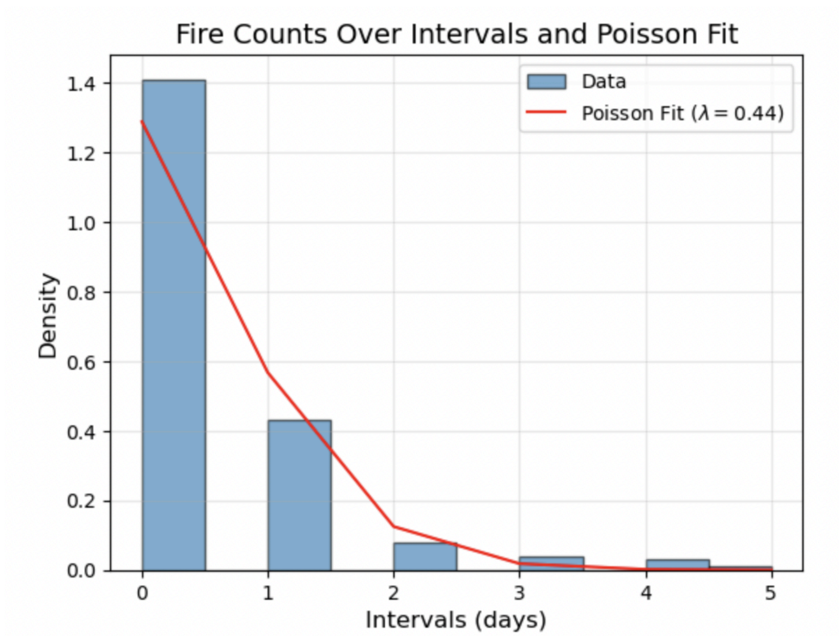

Figure 2: Fire counts over daily intervals

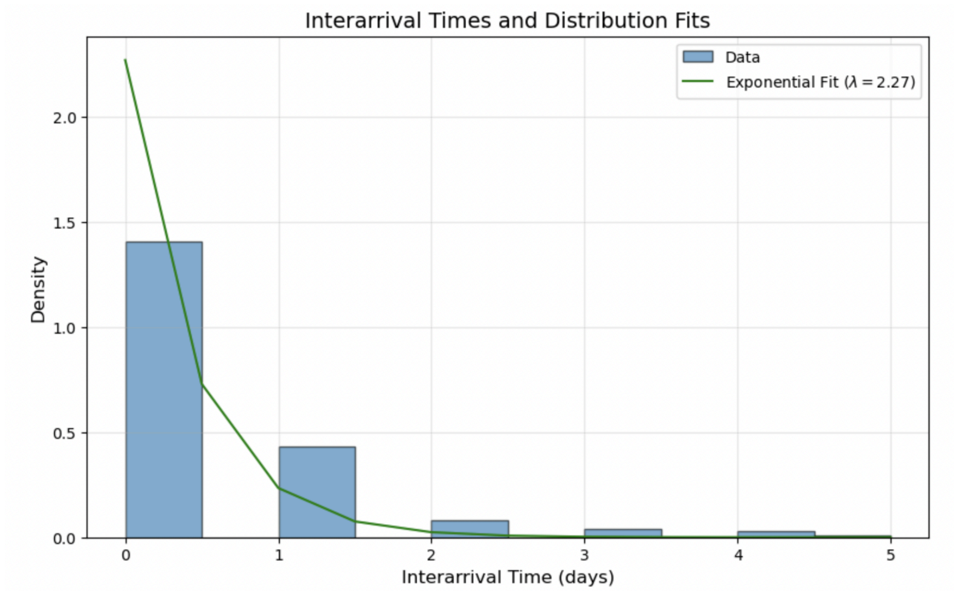

Figure 3: Interarrival time fitting

BC's wildfire fighting capacity is roughly 30 fires at a time. As our input modeling takes into account all the fires in BC, we scaled down the interarrival times by a factor of 10 to model fires occurring in 1/10th of the province, with 3 servers representing a firefighting capacity of 3 fires concurrently. The interarrival time calculation was scaled down by a factor of 10 to represent fires over the smaller region of the province, providing $\lambda = 0.227$ days.

3 Methodology

In order to evaluate the effectiveness of wildfire response strategies, 3 models were designed and implemented. Each one represents a strategy attempting to extinguish all fires and minimize area burned. Below we will go over each model including strategies to optimize them for the latter two. All simulations were run with 100 fire events modeled; given our inter arrival rates, 100 fires worked out to cover nearly a month of peak fire season, as modeled from our input data. This number of events also allows for the generation of a number of larger fires through the pareto distribution used.

3.1 System Parameters and Metrics

The system parameters in use in this model are primarily *interarrival times* and *service times*. Interarrival times are based on an exponential distribution derived from data, as previously discussed in section 2.1. Service times are the times to extinguish fires based on a simplified model, also discussed in section 2.1 The performance metrics used to compare between the efficacy of these models are *total area burned* and *average waiting time*. Total area burned represents the final area burned within a replication by all the fires. For each fire, this is recorded as the maximum size it grows to before being extinguished. Total area burned is the summation of the maximum size of each fire. Average waiting time corresponds to the time each fire spends in the queue before being serviced. These numbers will likely be somewhat correlated, as fires grow in the queue while waiting to be serviced. Other metrics we are tracking include *idle time*, or the overall time each server is unutilized.

3.2 Model 1: First in First Out (FIFO)

This model strictly adheres to the principle of servicing fires in the exact order of their arrival. This model allowed for an initial implementation with equal treatment for all fires. We hypothesized that this may be the least effective of all of our methods, as it represents a naive solution to which we would compare subsequent models.

3.3 Model 2: Multi-Queue

This model uses a multi-queue system divided by fire size thresholds, with separate queues for small, medium, and large fires. Each queue had a fire team assigned to it which responded to the fires in the sequence in which they arrived. This categorized fires in such a way that certain sizes of fires would be less delayed in the event they were after a large fire.

The thresholds we used for our different fire sizes needed to represent appropriate proportions of the work and we considered various strategies for this. Initially, we sorted each of the fires into separate queues using the average size of all fires in a given sample. The issue is that this assumes we can deterministically know the sizes of fires, which is not possible in the real-world system. The alternate solution was that we calculated the expected value for a given percentile of the size distribution and those determined the values for the given thresholds. It allowed us to get approximately the distributions of sizes per queue we wanted. This is also a realistic way for fire sizes to be divided in real-world scenarios given the wildfire management administration has determined a distribution to which wildfire sizes can be mapped.

This queuing system has to be carefully balanced to prevent over-prioritization of large fires at the expense of overall system efficiency. A potential consequence is that one wildfire response team is assigned to too many of the largest fires causing it to be working through all of them while the other queues finish their work and remain idle. This led to some work to optimize this system. We constructed a script in order to compare and plot all of the different potential percentile distributions between queues over a specified number of seeds. This allowed us to get a near optimized percentile to set our threshold to in order to get the desired fire distributions.

Below is **Figure 1** which is the result of the comparative script for varying percentiles. As it can be seen, the performance of fire size distributions is best when the large fires contain the 99th and higher percentile of sizes. Additionally, within that group of distributions we were able to narrow down that the approximate ideal distribution was small fires being 0th to 95th percentile, medium fires being 95th to 99th percentile and large fires being the remaining 95th and above percentile. For the comparisons across replications, all future runs of this model will be using these percentile distributions.

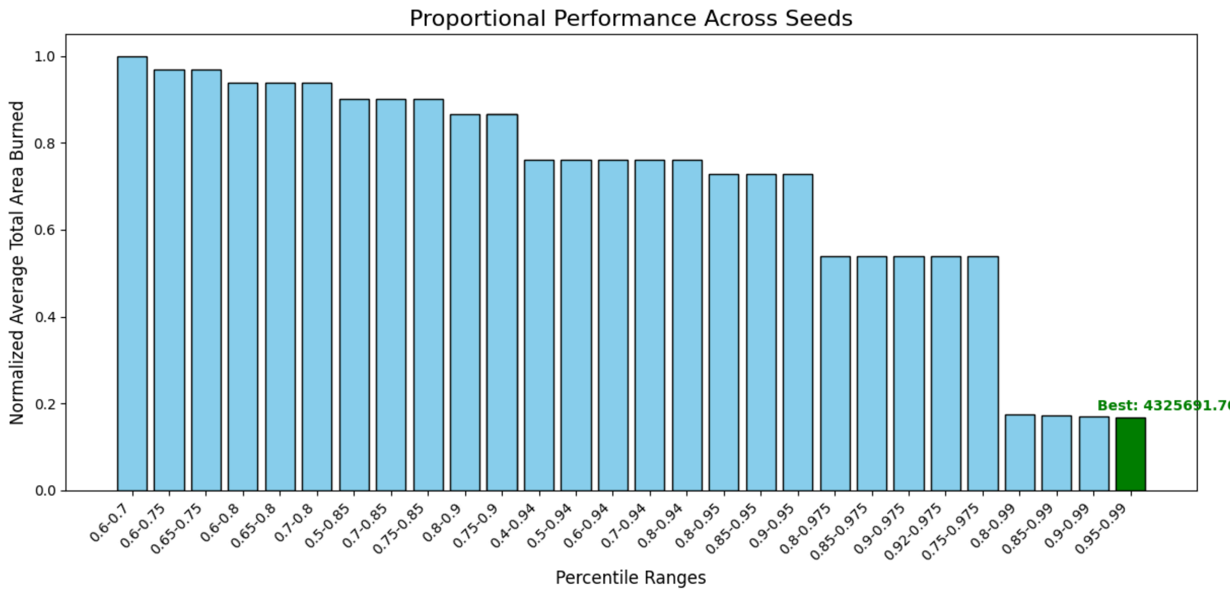

Figure 4: Optimal fire queue distribution configuration for Model 2

This performance trend is likely due to the nature of the Pareto Distribution used in our data. Since when values trend higher, they increase dramatically, partitioning the queue like this is likely isolating a few wildfires or the largest one, which prevents it from sharing a queue with other fires either causing growth or experiencing growth itself. Below is each group of fires and their respective details.

- **Large fires**: Assigned to the large queue when their size exceeds the 99th percentile of the size distribution.
- **Medium fires**: Occupy the medium queue, categorized between the 95th and 99th percentiles.
- **Small fires**: Fires in this queue remain until extinguished, this is the smallest categorization of fires.

Each fire is sorted into its queue and remains there until it is extinguished. Our next queueing system attempts to dynamically shift fires using a similar method to distribute work more effectively.

3.4 Model 3: Multi-Queue with Queue Jumping

This model builds upon the standard multi-queue system but incorporates a dynamic reclassification mechanism. Fires begin in the queue corresponding to their initial size, but as they are managed and their sizes reduce, they are reassigned to smaller size queues once they reach half the size threshold of the lower queue. This reclassification ensures that resources are allocated more effectively as fires shrink, allowing larger fires to move through the system faster while still maintaining appropriate response levels for smaller fires.

In order to assess if the percentile distributions for each fire would perform the same as it did in **Model 2**, we ran the same script. Below in **Figure 2** are the results which demonstrate that it has comparable performance results to **Model 2** though the average area burned is better which we will analyze further in the results section of this report.

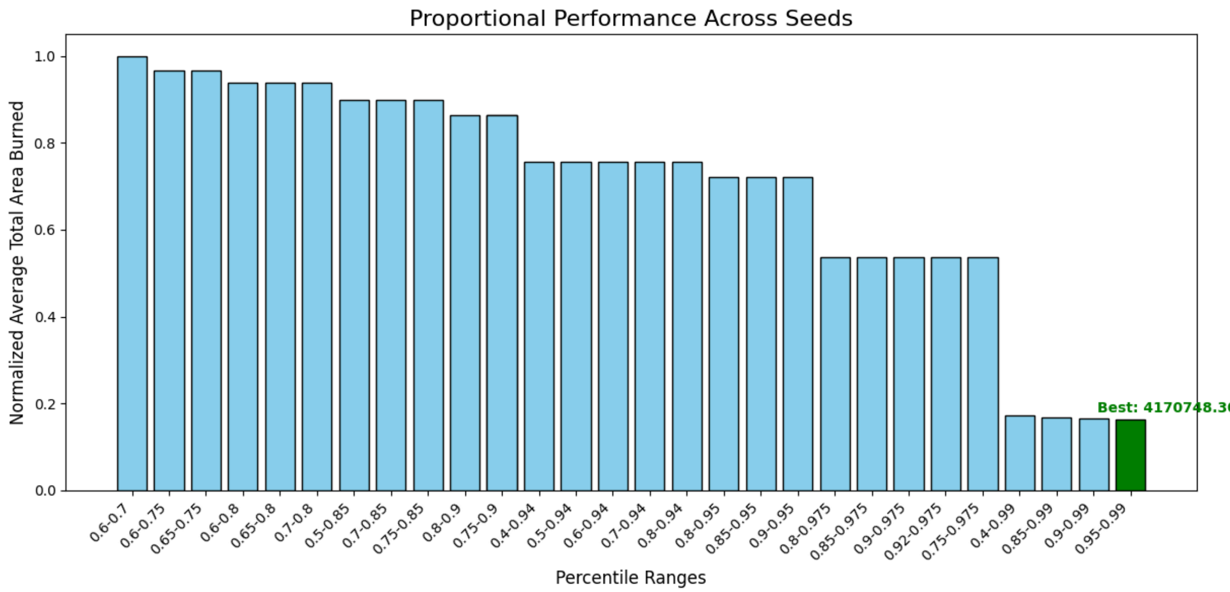

Figure 5: Optimal fire queue distribution configuration for Model 3

Similar to Model 2, this system uses three queues divided by size thresholds: small, medium, and large. However, the reclassification mechanism modifies how fires transition between queues:

- **Large fires**: Initially assigned to the large queue when their size exceeds the 99th percentile of the size distribution. Once a fire reduces below half of the large threshold, it is reclassified into the medium queue.
- **Medium fires**: Initially occupy the medium queue, categorized between the 95th and 99th percentiles. Fires move to the small queue once their size drops below half the medium threshold.
- **Small fires**: Fires in this queue remain until extinguished, as there are no smaller categories.

This model represents a more adaptive approach to fire management, demonstrating how dynamic strategies can enhance efficiency and effectiveness in wildfire simulations.

3.5 Model 4: Multi-Queue with Even Distribution

This model is designed with the intent to distribute the total size of fires between queues as evenly as possible. In order to accomplish this, the simulation uses the same expected value threshold system as both **Model 2** and **Model 3** to establish size-based thresholds for small, medium, and large fires. However, the primary distinction lies in how fires are assigned to queues. Rather than strictly categorizing fires by size thresholds alone, this model ensures an even distribution of fires across all queues.

When a fire arrives, it is assigned to a queue not solely based on its size but also on the current load (number of fires) in each queue. The goal is to achieve an even distribution of workload across all queues. This ensures that no single queue becomes overwhelmed with too many fires of any category while others remain underutilized. For example:

- If a queue handling "large fires" has fewer fires than the others, a new large fire will be assigned to it regardless of when it arrived relative to others.
- Similarly, "medium" and "small" fires are distributed to queues to balance the total number of fires and, consequently, the workload.

The individual queues themselves operate as standard First-In-First-Out systems, processing fires in the order they arrive. Fires assigned to a queue are managed independently, with each fire team handling the fires assigned to its queue.

Just as we did for **Model 2** and **Model 3**, we ran a script to determine an effective percentile distribution of the data. As is shown in the figure below, it can clearly be seen that a high threshold for large fires is beneficial. The clear performer as shown by the green bar is with small being 0th to 80th percentile, medium being 80th to 97.5th percentile and large being 97.5th to 100th percentile of fire sizes. For the comparative runs in subsequent sections of this report it is assumed we will be using this distribution at all times.

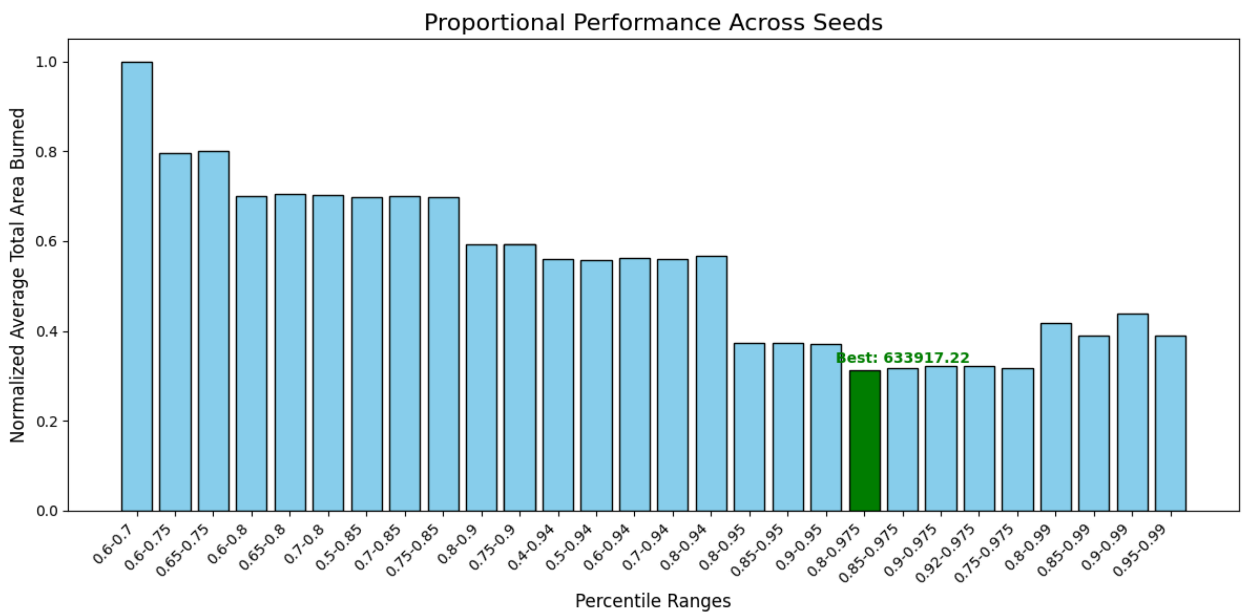

Figure 6: Optimal fire queue distribution configuration for Model 4

3.6 Comparative Metrics

The most important comparative metric to our study has to be total area burned, as we aim to minimize loss of trees during a fire. When examining the total area burned averaged over multiple runs of the simulation over different seeds, the FIFO queue with 3 servers had by far the least loss of forested area in comparison to our size based queue, losing approximately 94% less forested area than both.

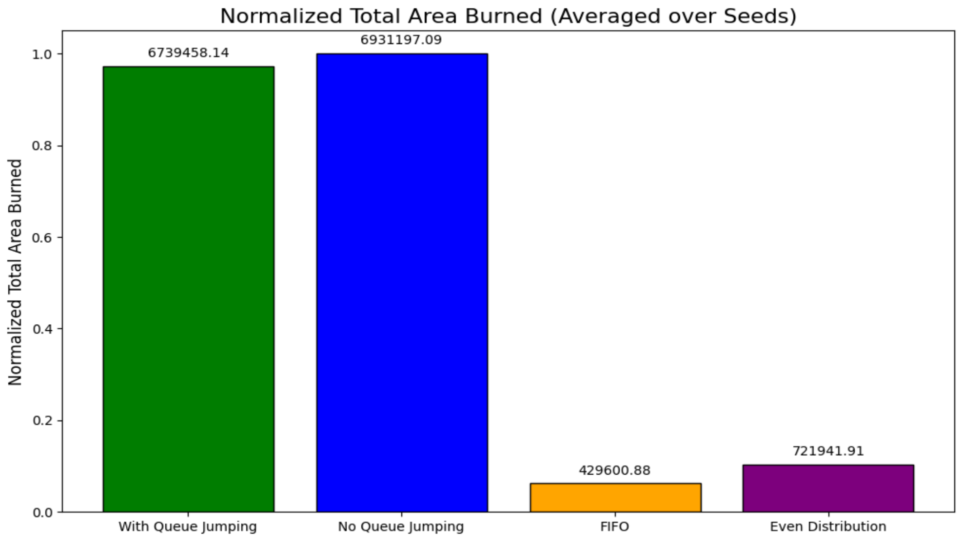

Figure 7: A graph that illustrates the total area burned throughout 1000 different seeds for each model.

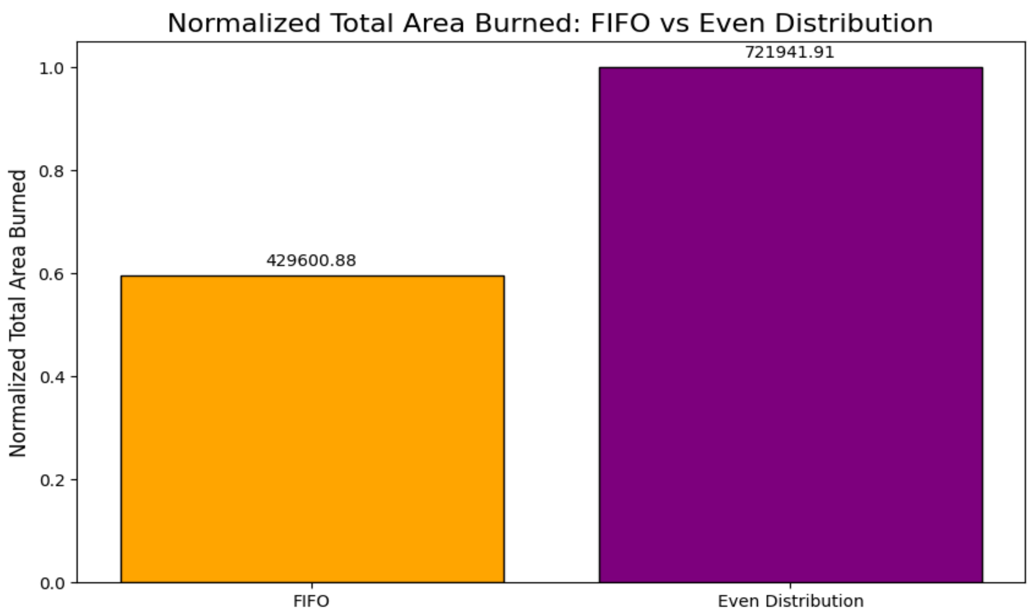

Figure 8: A comparison up close of the higher performing models, FIFO and Even Distribution.

Furthermore, when compared to a 3-queue system with evenly distributed fires the single queue 3 server system out performs it by quite a bit as well, with the latter having 40.5% less area lost in comparison.

This is not to say however that the FIFO system was always the best system, out of one thousand runs of the system 82 times, the queue jumping model performed the best, 38 the evenly distributed queues model, and staggeringly 879 times the FIFO model. So while it has not performed the best in all cases, 80% of the time it was the best running system; Furthermore, while some of the models occasionally had the better end result, their worst cases never got close to the worst case of the FIFO queue.

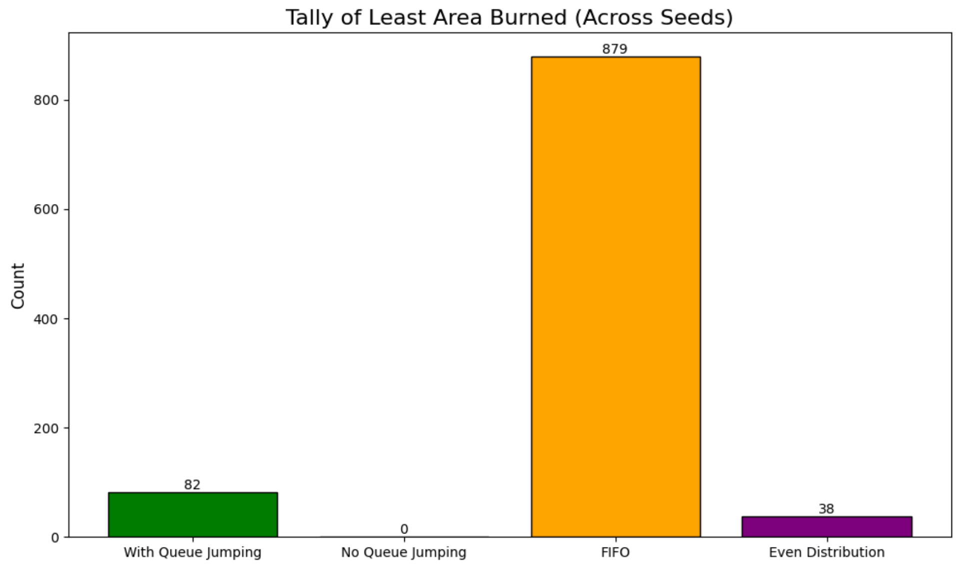

Figure 9: Number of times each model performed the best over 1000 seeds

4 Analysis

In order to capture a sample for this report we wrote a script to run each of the four models over 5 different seeds. Below are the results for 5 replications, with CRNs used for all queuing models within each replication.

```
=== Simulation Run 1 ===

--- With Queue Jumping ---

Average Waiting Time: 268.16

Average Service Time: 119.29

Average Time Between Arrivals: 0.23

Average Time Customer Spent in System: 388.59

Total Area Burned: 862421.20

--- No Queue Jumping ---

Average Waiting Time: 270.40

Average Service Time: 121.40

Average Time Between Arrivals: 0.23

Average Time Customer Spent in System: 391.81

Total Area Burned: 868762.99

--- FIFO ---

Average Waiting Time: 4.70

Average Service Time: 29.71

Average Time Between Arrivals: 0.23

Average Time Customer Spent in System: 34.40

Total Area Burned: 223479.67

--- Even Distribution ---

Average Waiting Time: 249.11

Average Service Time: 38.92

Average Time Between Arrivals: 0.23

Average Time Customer Spent in System: 288.03

Total Area Burned: 290758.65

=== End of Run ===
```

```
=== Simulation Run 2 ===

--- With Queue Jumping ---

Average Waiting Time: 128.50

Average Service Time: 589.33

Average Time Between Arrivals: 0.26

Average Time Customer Spent in System: 760.25

Total Area Burned: 3934971.79

--- No Queue Jumping ---

Average Waiting Time: 127.69

Average Service Time: 633.36

Average Time Between Arrivals: 0.26

Average Time Customer Spent in System: 761.05

Total Area Burned: 3945227.60

--- FIFO ---

Average Waiting Time: 91.60

Average Service Time: 110.23

Average Time Between Arrivals: 0.26

Average Time Customer Spent in System: 201.83

Total Area Burned: 679138.31

--- Even Distribution ---

Average Waiting Time: 5016.94

Average Service Time: 346.74

Average Time Between Arrivals: 0.26

Average Time Customer Spent in System: 5363.68

Total Area Burned: 2182106.37

=== End of Run ===
```

```
=== Simulation Run 3 ===

--- With Queue Jumping ---

Average Waiting Time: 174.14

Average Service Time: 5.70

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 206.16

Total Area Burned: 293166.33

--- No Queue Jumping ---

Average Waiting Time: 168.54

Average Service Time: 32.07

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 200.60

Total Area Burned: 293343.71

--- FIFO ---

Average Waiting Time: 3.26

Average Service Time: 29.94

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 33.20

Total Area Burned: 279777.72

--- Even Distribution ---

Average Waiting Time: 740.60

Average Service Time: 36.11

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 776.70

Total Area Burned: 319777.65

=== End of Run ===
```

```
=== Simulation Run 4 ===

--- With Queue Jumping ---

Average Waiting Time: 100.78

Average Service Time: 10.50

Average Time Between Arrivals: 0.21

Average Time Customer Spent in System: 112.17

Total Area Burned: 75212.43

--- No Queue Jumping ---

Average Waiting Time: 100.97

Average Service Time: 11.42

Average Time Between Arrivals: 0.21

Average Time Customer Spent in System: 112.38

Total Area Burned: 75406.96

--- FIFO ---

Average Waiting Time: 2.47

Average Service Time: 9.04

Average Time Between Arrivals: 0.21

Average Time Customer Spent in System: 11.51

Total Area Burned: 59767.79

--- Even Distribution ---

Average Waiting Time: 244.63

Average Service Time: 12.80

Average Time Between Arrivals: 0.21

Average Time Customer Spent in System: 257.42

Total Area Burned: 85270.58

=== End of Run ===
```

```
=== Simulation Run 5 ===

--- With Queue Jumping ---

Average Waiting Time: 3027.31

Average Service Time: 9541.93

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 12614.54

Total Area Burned: 75175516.18

--- No Queue Jumping ---

Average Waiting Time: 3068.34

Average Service Time: 9713.63

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 12781.98

Total Area Burned: 76200422.80

--- FIFO ---

Average Waiting Time: 3.45

Average Service Time: 156.15

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 159.59

Total Area Burned: 1058968.72

--- Even Distribution ---

Average Waiting Time: 5232.32

Average Service Time: 550.79

Average Time Between Arrivals: 0.24

Average Time Customer Spent in System: 5783.12

Total Area Burned: 4101156.76

=== End of Run ===
```

From our multiple runs of the simulations, we can see that in every metric the FIFO queue outperforms its competitor queues. We can see that even in its worst case scenario the time waiting in queue for the fires is far lower than the rest of lower than every other queue, in the case of the queue-jumping this would be expected as that aims to minimize purely the area lost, however even in that department it does not perform up to par with the FIFO queue. Removing the FIFO queue from the equation, if we were to compare the other 3 queue models, we can see that the performance of the even distribution is heavily reliant on the size of the fires, as in situations where the fires are more on the heavy end, such as run 5 and run 2, it will outperform the other two. However, when the fire sizes are on the smaller end on average, its performance in terms of area burned takes a hit in comparison with the other two. In general, FIFO is the best performing queue, in terms of every metric, and if we were to not include it in our results, the even distribution performs best for larger fire sizes while the queue-jumping and no-queue-jumping models perform best for smaller average fire size simulations, given that in a real life situation, you are not able to know what the fire size averages will actually be, it makes it infeasible to pick which queue would work better in the given situation.

5 Conclusion

This study demonstrated the value of simulation-based analysis in assessing and optimizing wildfire response strategies. Using various queueing systems, we evaluated how different dispatch policies influenced key performance metrics such as total area burned, average waiting time, and system idle times. The comparative results revealed meaningful insights into the efficacy of each model, while highlighting areas for further refinement.

5.1 Key Findings

1. **FIFO Model:**
    - The FIFO model exhibited the least total area burned across the majority of simulation runs. It was the best-performing model in approximately 80% of the replications.
    - While simple, the FIFO system's prioritization of response purely by arrival time ensures that fires are attended to quickly, minimizing growth caused by delays.
2. **Multi-Queue Size-Based Model:**
    - Dividing fires by size thresholds provided a balanced approach, but its performance was sensitive to the choice of percentile thresholds for fire categorization.
    - While having a better best case than FIFO, this model suffered from inefficient queue balancing, leaving some response teams idle
    - In the event that there were multiple many large fires, it would put them all in the large queue causing the fires that grew the most to get large amounts of time to grow leading to extreme worst case scenarios.
3. **Multi-Queue with Queue Jumping:**
    - Incorporating queue jumping allowed for dynamic reassignment of fires as they reduced in size, improving resource allocation efficiency.
    - This model felt the same performance woes as its non-jumping counterpart and did not surpass FIFO on an average case basis
    - This model did improve upon the performance of the previous but not to such a degree where it bested FIFO outside of best case scenario
4. **Multi-Queue with Even Distribution:**
    - Ensuring an even distribution of fires across all queues regardless of size improved workload balance and reduced idle times.

- This model performed well in scenarios with high variability in fire sizes and inter-arrival times, offering a reliable balance between minimizing area burned and ensuring resource efficiency.
- While not as effective as FIFO in minimizing total area burned, the even distribution model exhibited consistent and predictable performance, avoiding extreme worst-case outcomes.

5.2 Implications

The results of this study suggest that the optimal wildfire response strategy depends heavily on the underlying conditions. In scenarios where fire arrival rates and sizes are highly variable, the FIFO model provides the most consistent and effective results for minimizing total area burned. However, in cases where resource allocation efficiency is prioritized, models such as queue jumping or even distribution offer a robust alternative.

The models outside of FIFO we adopted provided a complex solution to the problem of wildfire control but unfortunately did not pose a superior solution. Since these models did have a significantly better best case we are led to believe that with further adjustments any one could potentially outperform FIFO.

5.3 Improvements to the Response Strategies

In this section we will take an opportunity to reflect on the performance of our wildfire response strategies by suggesting improvements for **Model 2, Model 3,** and **Model 4**.

5.3.1 Size-Based Queue Improvement Suggestions

The size-based queue system, while effective in categorizing fires based on their size thresholds, faced challenges in balancing workloads between queues and ensuring optimal resource allocation. Below are the suggested improvements:

1. **Dynamic Threshold Adjustments:** Adjust size thresholds dynamically based on real-time fire arrival patterns and queue workloads.
2. **Resource Redistribution:** Allow for flexible resource allocation between queues. For instance, if the small fire queue is idle while the large fire queue is overwhelmed, reassign idle teams to assist with large fires.
3. **Queue Consolidation for Sparse Scenarios:** During periods of low fire activity, reduce the number of queues by merging medium and large fire queues to prevent idle resources.

5.3.2 Size-Based Queue with Queue Jumping Improvement Suggestions

The queue jumping model introduced dynamic fire reclassification, which enhanced adaptability but added complexity. To refine this system:

1. **Optimizing Jump Criteria:** Experiment with alternate thresholds for when fires jump between queues. For example, instead of waiting for fires to reduce to half the lower size threshold, consider a proportion of remaining service time as the jumping criterion.

2. **Predictive Reassignment:** Utilize predictive modeling to anticipate when fires are likely to reach the jump threshold and preemptively adjust queue assignments. This could smooth transitions and avoid bottlenecks.
3. **Load Balancing During Jumping:** Ensure that when fires jump to smaller queues, they do not overload the receiving queue. Introduce a secondary balancing mechanism to assign the fire to the least-loaded queue.
4. **Cost-Benefit Analysis of Jumping:** Evaluate whether jumping smaller fires to lower queues significantly impacts total area burned. If the benefit is marginal, adjust the jumping logic to reduce unnecessary transitions.

5.3.3 Evenly Distributed Queue Improvement Suggestions

The even distribution model aimed to balance workload but occasionally struggled with maintaining optimal prioritization and efficiency. Improvements include:

1. **Weighted Distribution:** Incorporate a weighted distribution approach that considers the severity of fires (e.g., size and intensity) alongside workload balancing. Larger or more intense fires could be given slightly higher priority in assignment.
2. **Avoiding Oversimplification:** Ensure that the balancing logic doesn't oversimplify fire assignment. For instance, a focus solely on fire count may ignore the fact that a queue with fewer but larger fires might still be more overloaded than another.
3. **Queue Rebalancing:** Incorporate a weighted distribution approach that considers the severity of fires (e.g., size and intensity) alongside workload balancing. Larger or more intense fires could be given slightly higher priority in assignment.

5.4 Alternate Response Strategies

Throughout the implementation various alternate queuing strategies that could yield effective responses came to our attention. Below is each idea listen and explained

1. **Queue Sorting by Intensity:** By dividing up wildfires by intensity rather than size, we could use various queuing methods to address wildfires with this important factor considered.
2. **Priority Queue Sorted by Weighted Intensity and Size:** A hybrid approach to the size sorted queue and intensity sorted queue, this system would use weighted values for the intensity and size to establish a score to assign it to a priority queue.
3. **Cumulative Size Based Queues**: This queue in theory could be effective in balancing out all of the sizes reducing overloaded queues. As each fire arrives, this one would sort it into the queue with the least amount of fire size currently assigned.

6 References

[1] Canadian Interagency Forest Fire Centre, 2023 Canada Wildfire Annual Report, 2023. [Online]. Available: https://ciffc.ca/sites/default/files/2024-03/03.07.24_CIFFC_2023CanadaReport%20%281%29.pdf

[2] Government of British Columbia, Wildfire Crews and Response Tools. [Online]. Available: https://www2.gov.bc.ca/gov/content/safety/wildfire-status/wildfire-response/wildfire-personnel-and-response-tools/wildfire-crews

[3] A. Author et al., "The connection between exponential growth of fires and their size," PMC, 2022. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9286820/

[4] U.S. Department of Agriculture, "Fuel Models for Predicting Fire Behavior," Research Paper INT-438, 1988. [Online]. Available: https://www.fs.usda.gov/rm/pubs_int/int_rp438.pdf

[5] Canadian Wildland Fire Information System, Canadian Forest Fire Behavior Prediction System. [Online]. Available: https://cwfis.cfs.nrcan.gc.ca/background/summary/fbp