CZ4052 Cloud Computing

Assignment 1

Submitted by: Lim Zhi Qing

Matriculation Number: U2021897L

NANYANG TECHNOLOGICAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Table of Contents

1. Introduction

This report explores congestion control algorithms through simulation and analysis of the Additive Increase Multiplicative Decrease (AIMD) approach and its derivatives. Section 2 explores the impact of the alpha and beta parameters on efficiency and fairness. Section 3 discusses the AIMD approach while section 4 and 5 explore using exponential increase and logarithmic increases respectively. Figures referenced in section 2 to 5 can be found in Appendix A. Simulation code written using Jupyter Notebook in Python can be found in Appendix B.

2. Changes in AIMD Parameters

2.1. Changes in AIMD Parameters (Alpha)

A simulation was performed for each of the alpha parameter values in the set [1, 2, 4, 8, 16]. As shown in Figure 1 in appendix A, allocation approaches the fairness line at a faster rate with higher alpha values. As shown in Figure 4, the rate of increase in fairness is higher with higher values of alpha. High alpha values cause the congestion window size to exceed the maximum capacity by a greater amount, potentially incurring higher levels of loss, as shown in Figure 2. On the other hand, low alpha values produce a slower increase in window size, reducing efficiency. As shown in Figure 3, overall efficiency is higher with increasing values of alpha. The study suggests that higher alpha values can improve efficiency and fairness but may incur higher rates of loss.

2.2. Changes in AIMD Parameters (Beta)

A simulation was performed for each of the beta parameter values in the set [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. As shown in Figure 5 in appendix A, allocation approaches the fairness line at a faster rate with lower beta values. As shown in Figure 8, the rate of increase in fairness is higher with lower values of beta. Lower beta values, however, produces higher amounts of unused capacity, as shown in Figure 6, reducing efficiency. On the other hand, high beta values reduce the impact on the total allocation, thus retaining high efficiency, as shown in Figure 7. The study suggests that lower beta values improve fairness at the cost of efficiency while higher beta values retain high efficiency at the cost of fairness.

3. Additive Increase Multiplicative Decrease

3.1. Additive Increase Multiplicative Decrease (2 Users)

A simulation was performed for Additive Increase Multiplicative Decrease (AIMD) using alpha = 1 and beta = 0.5 for 2 users. As shown in Figure 9 in appendix A, a sawtooth behaviour can be seen in the allocation for two users. In Figure 10, the congestion windows of both users converge at approximately the 200th iteration. In Figure 11, efficiency starts off high when allocation favours one user, gradually settling to a value of approximately 0.75, while fairness is shown to increase over time.

3.2. Additive Increase Multiplicative Decrease (10 Users)

A second simulation was performed for Additive Increase Multiplicative Decrease (AIMD) using alpha = 1 and beta = 0.5 for 10 users. As shown in Figure 12, the congestion windows of all users converge at approximately the 50th iteration. The conversion is much faster than the previous study with 2 users, as the allocation is distributed more evenly to the 10 users. As shown in Figure 13, fairness increases quickly in relation to the study with 2 users, while efficiency continues to decline at a decreasing rate, settling to a value of approximately 0.76. This study suggests that higher number of users increases the rate of convergence.

4. Exponential Increase

4.1. Exponential Increase (2 Users)

A simulation was performed with exponential increase and multiplicative decrease using alpha = 1, beta = 0.5, and exponent = 0.5 for 2 users. In Figure 14, allocation defers from that of AIMD where allocation favours the user with higher congestion window during the increase phase. However, the rate of increase in allocation is also higher, as shown in Figure 15, where the congestion windows of both users converge at approximately the 80th iteration. As shown in Figure 16, efficiency drops lower initially but stabilizes at a faster rate as compared to AIMD, settling at a value of approximately 0.78, higher than that for AIMD.

4.2. Exponential Increase (10 Users)

A second simulation was performed with exponential increase and multiplicative decrease using alpha = 1, beta = 0.5, and exponent = 0.5 for 10 users. In Figure 17, the congestion windows of all users converge at approximately the 35th iteration, faster than that of AIMD. It can be observed that there are more cycle of increase and decrease phases relative to AIMD in section 3.2. As shown in Figure 18, efficiency decreases much more gradually over time and the overall efficiency is higher at approximately 0.80, higher than that for AIMD. This study suggests that exponential increase improves convergence rate as well as overall efficiency, relative to additive increase.

5. Logarithmic Increase

5.1. Logarithmic Increase (2 Users)

A simulation was performed with logarithmic increase and multiplicative decrease using alpha = 1 and beta = 0.5 for 2 users. In Figure 19, allocation is similar to that of exponential increase in section 4.1 but moves towards the fairness line at a faster rate. As shown in Figure 20, the congestion window of both users converges at approximately the 80th iteration, similar to section 4.1. However, it is observed that there are less cycles of increase and decreases phases.

As shown in Figure 21, efficiency drops lower initially but stabilizes at a faster rate, similar to section 4.1, while settling at a value of approximately 0.76.

5.2. Logarithmic Increase (10 Users)

A second simulation was performed with logarithmic increase and multiplicative decrease using alpha = 1 and beta = 0.5 for 10 users. In Figure 22, the congestion windows of all users converge at approximately the 45^{th} iteration, faster than that of AIMD. As shown in Figure 23, efficiency decreases initially, but stabilizes at about 0.80, higher than that for AIMD. This study suggests that the performance of logarithmic increase is similar to that of exponential increase.

6. Conclusion

Changes to alpha and beta parameters should be considered in optimizing the performance of the AIMD approach. The results of the studies in section 2 suggest that a higher alpha parameter, or in other words, a higher rate of increase in the increase phase, improves overall efficiency and fairness. However, it is advisable to choose a value that is not too high, to avoid increasing potential loss from a higher than capacity congestion window size. As such, a value of between 4 to 16 could be suitable. The beta parameter is more straightforward, in that lower beta values improve fairness at the cost of efficiency while higher beta values improve efficiency at the cost of fairness. It may be advantageous to pair a high beta value with a higher than 4 alpha value, as the faster increase phase improves fairness while the slower decrease phase improves efficiency. The results of studies in section 3 suggest that a higher number of concurrent users improves the rate of convergence without a noticeable difference in overall efficiency. The results of the studies in section 4 and 5 show that the exponential and logarithmic functions may provide better convergence rate, efficiency and fairness as compared to the additive increase function. However, the results show similarity to the results in section 2, where a faster increase phase resulted in better performance. It can be concluded that a faster increase phase is beneficial in improving efficiency and fairness while at the potential cost of introducing higher levels of loss.

Appendix A

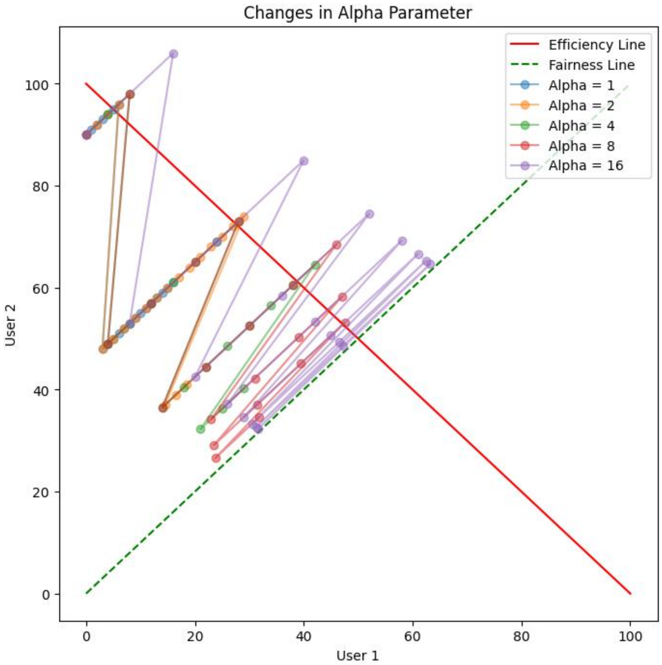

Figure 1. Change in Allocation for different Alpha

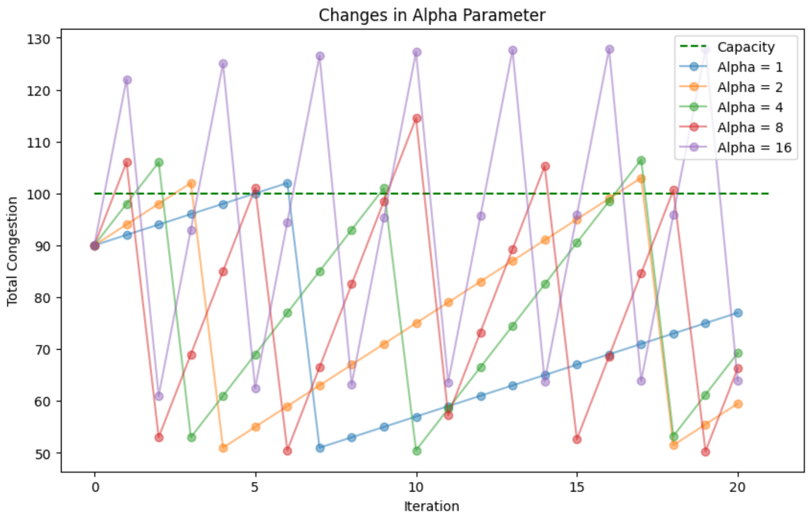

Figure 2. Change in Congestion Window for different Alpha

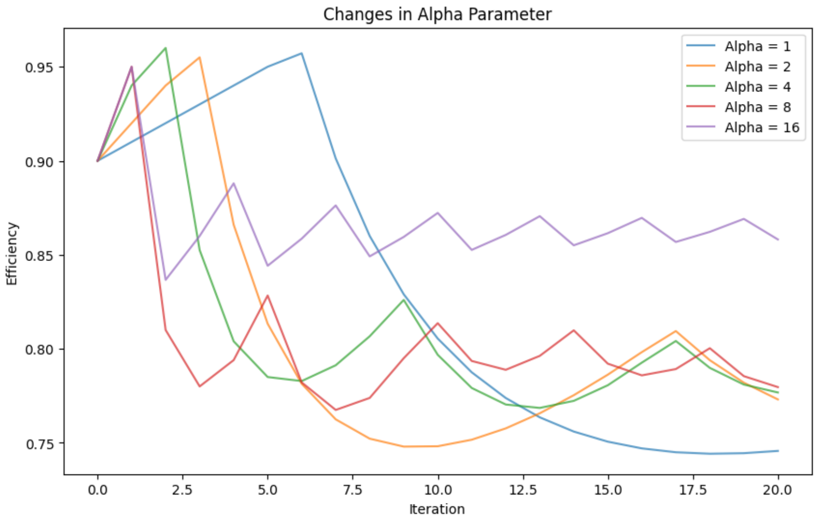

Figure 3. Change in Efficiency for different Alpha

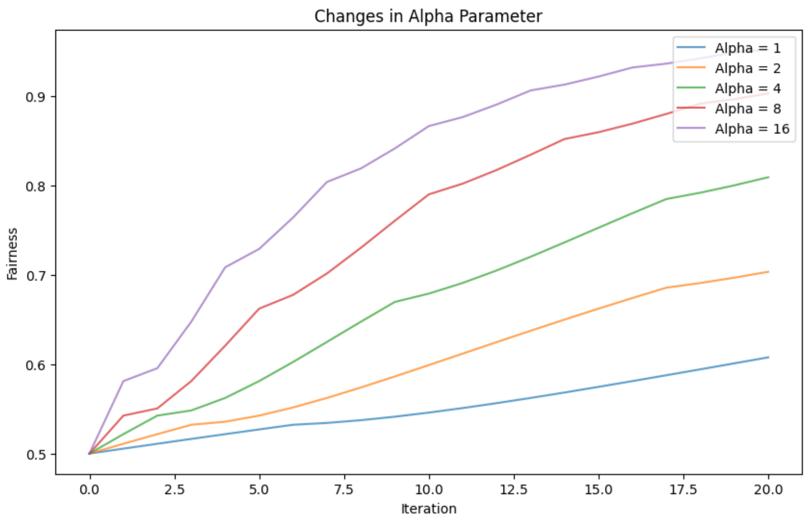

Figure 4. Change in Fairness for different Alpha

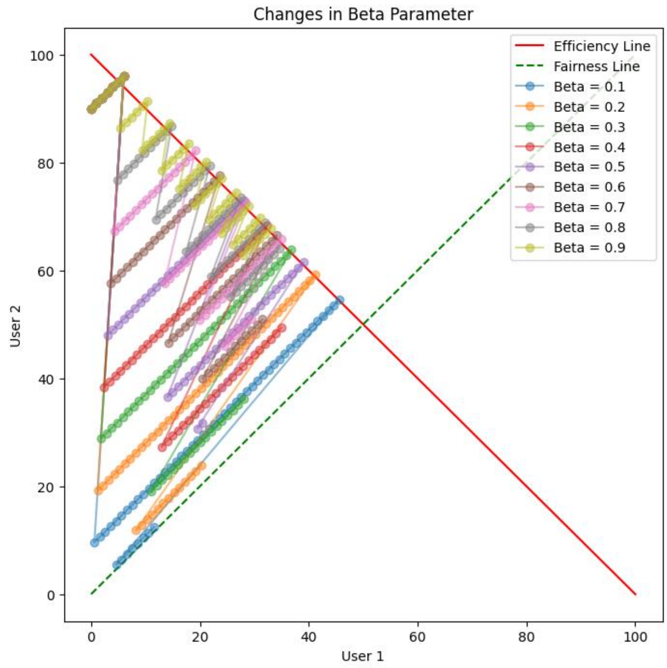

Figure 5. Change in Allocation for different Beta

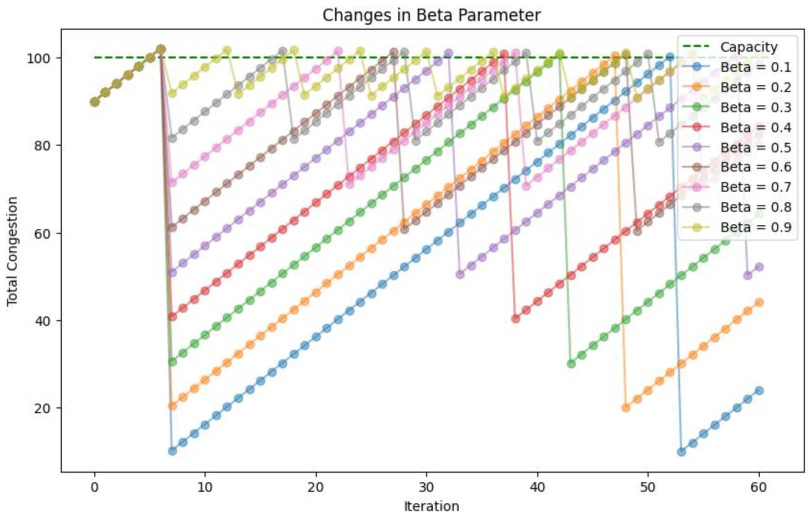

Figure 6. Change in Congestion Window for different Beta

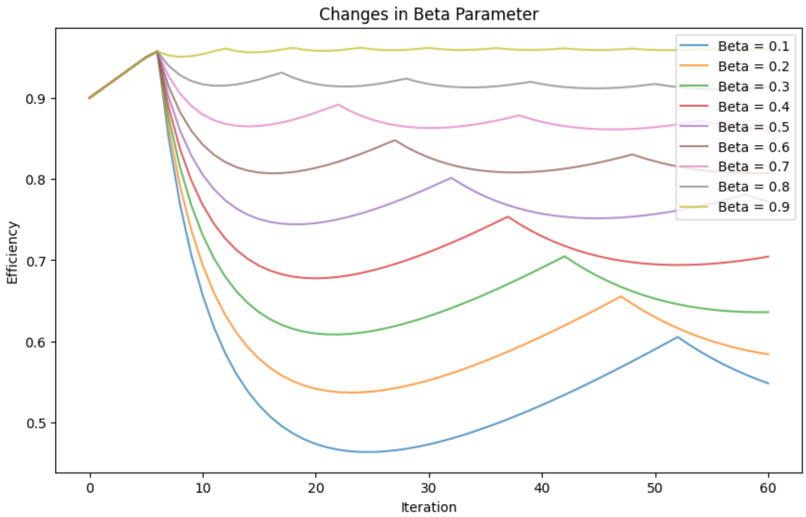

Figure 7. Change in Efficiency for different Beta

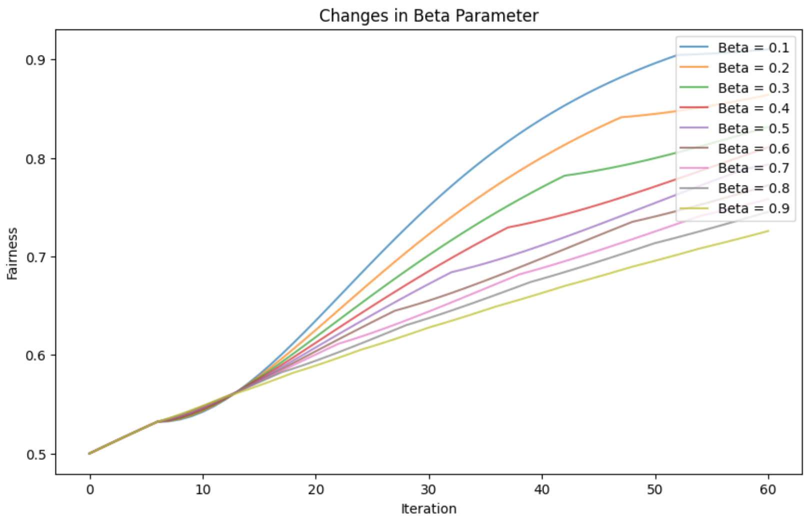

Figure 8. Change in Fairness for different Beta

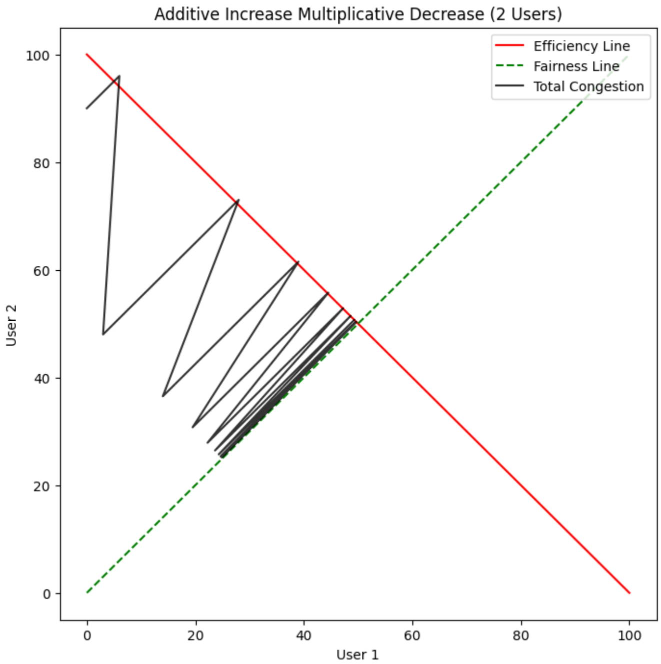

Figure 9. Change in Allocation for AIMD (2 Users)

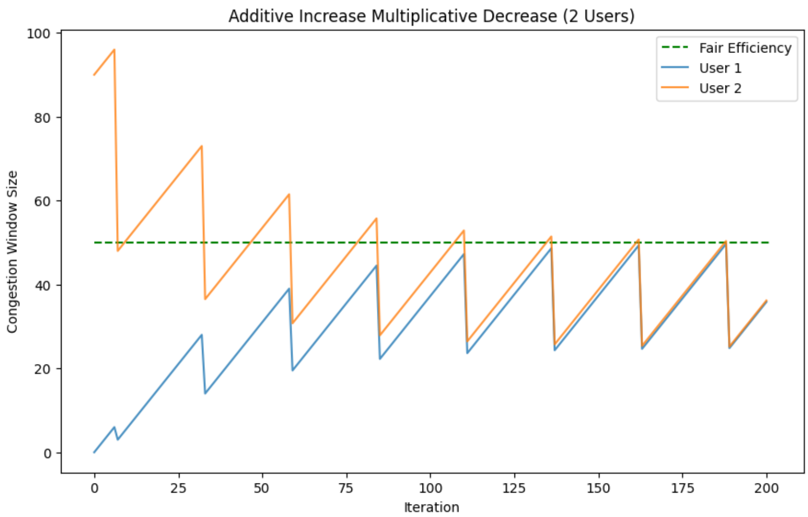

Figure 10. Change in Congestion Window for AIMD (2 Users)

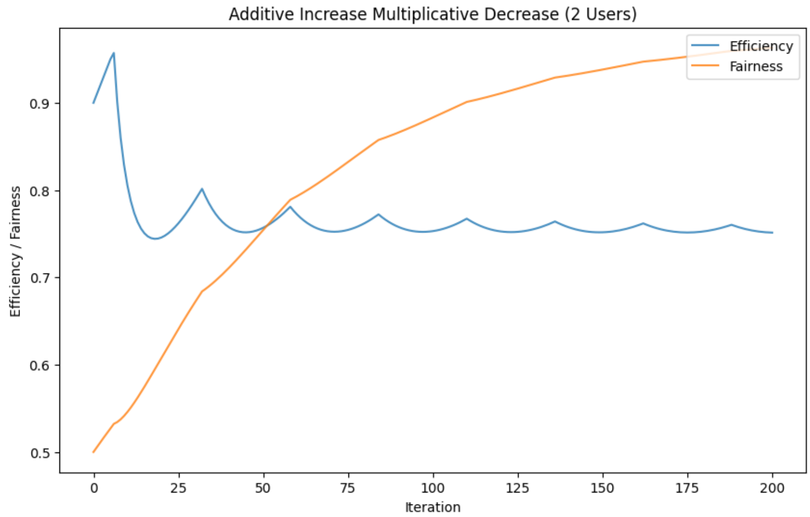

Figure 11. Change in Efficiency and Fairness for AIMD (2 Users)

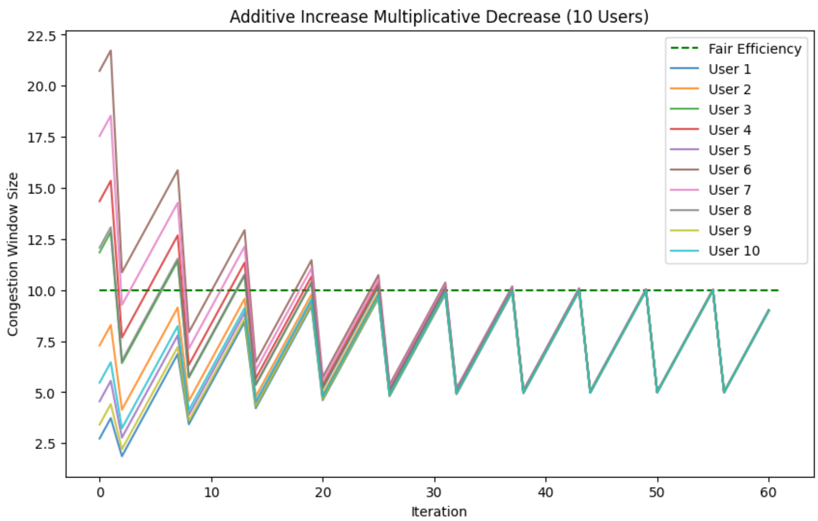

Figure 12. Change in Congestion Window for AIMD (10 Users)

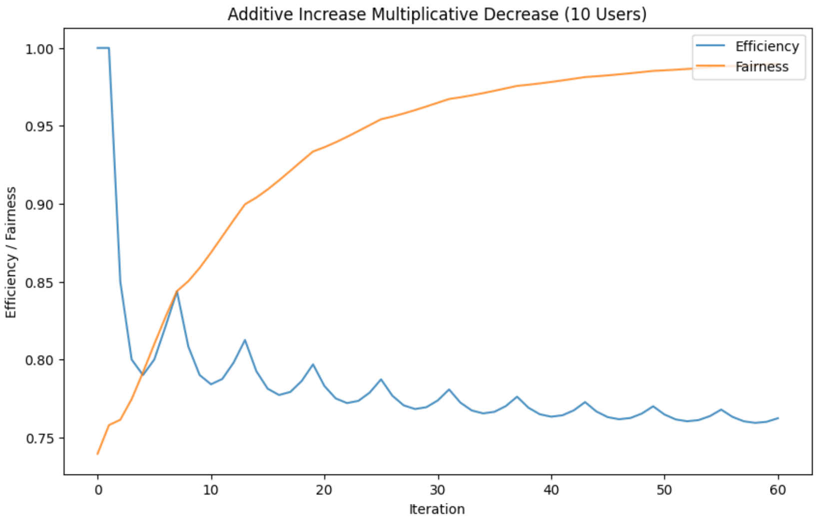

Figure 13. Change in Efficiency and Fairness for AIMD (10 Users)

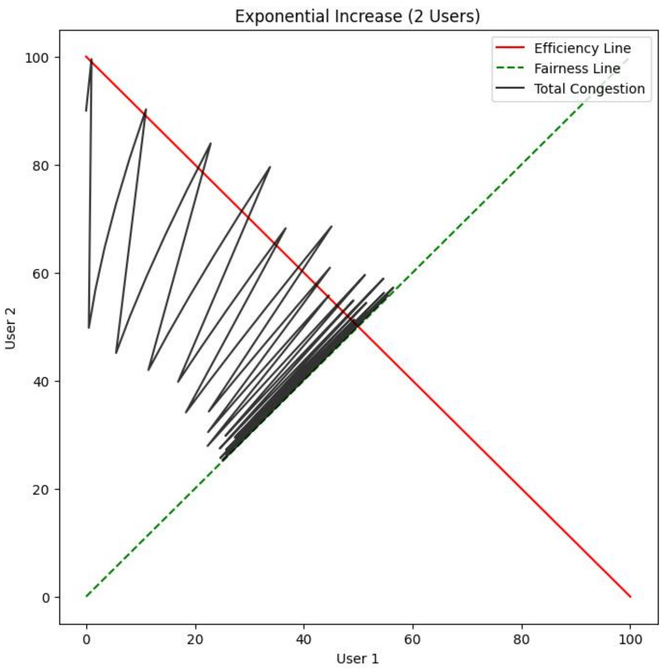

Figure 14. Change in Allocation for Exponential Increase (2 Users)

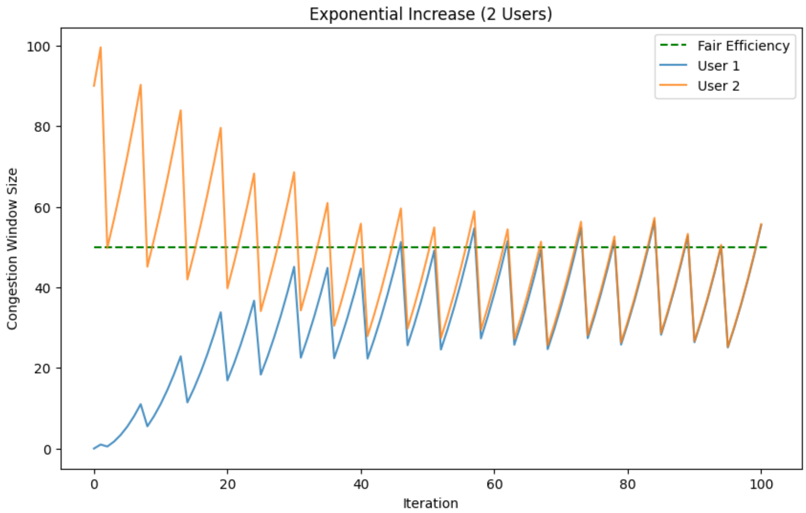

Figure 15. Change in Congestion Window for Exponential Increase (2 Users)

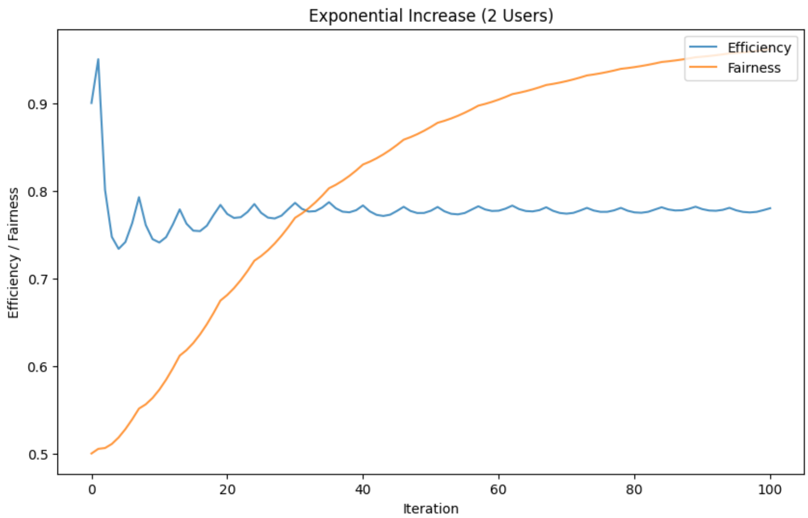

Figure 16. Change in Efficiency and Fairness for Exponential Increase (2 Users)

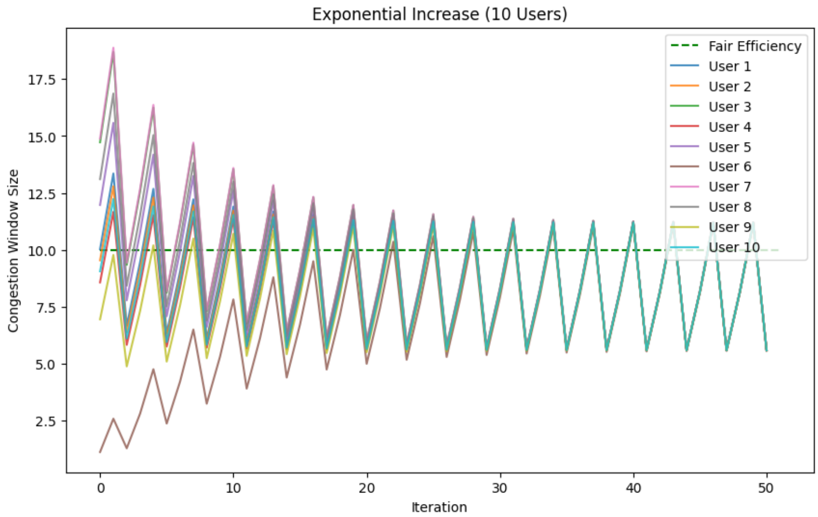

Figure 17. Change in Congestion Window for Exponential Increase (10 Users)

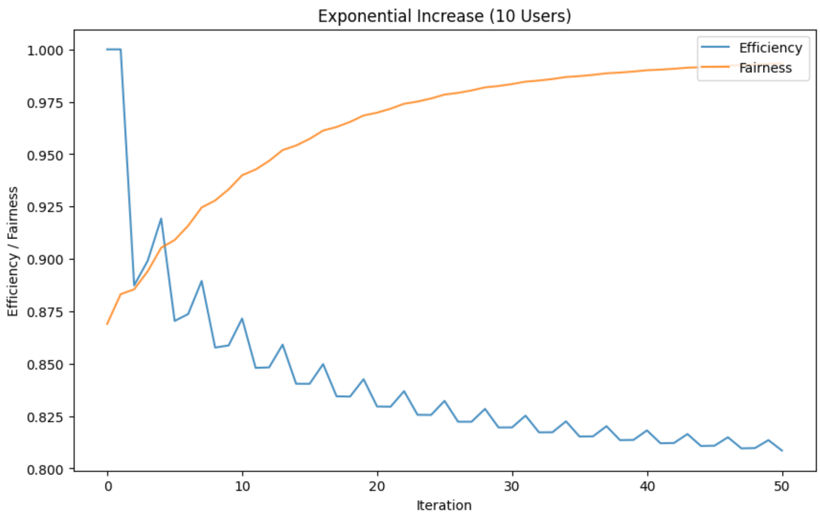

Figure 18. Change in Efficiency and Fairness for Exponential Increase (10 Users)

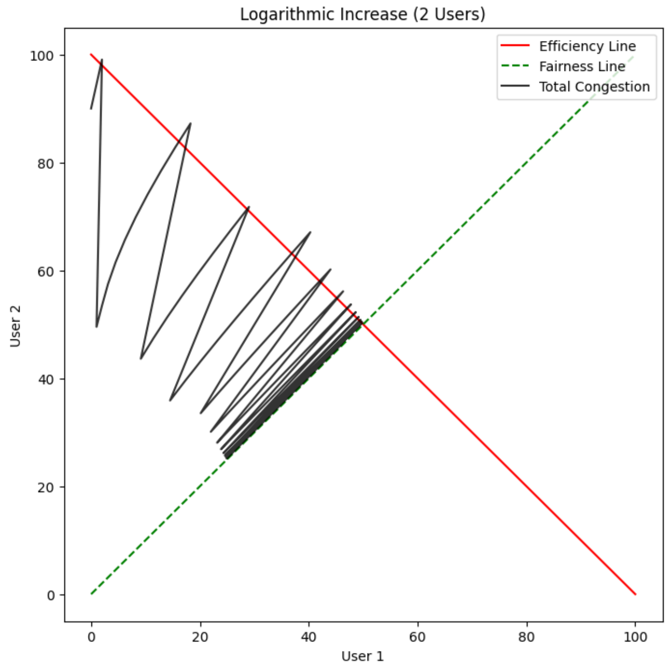

Figure 19. Change in Allocation for Logarithmic Increase (2 Users)

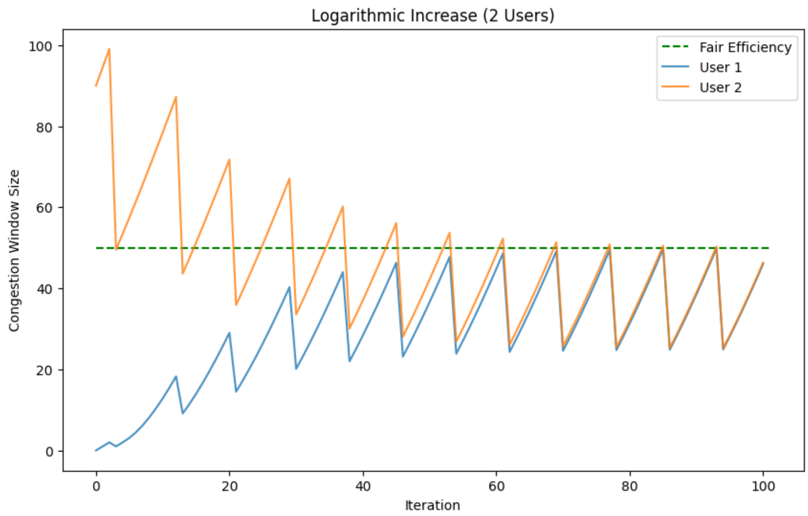

Figure 20. Change in Congestion Window for Logarithmic Increase (2 Users)

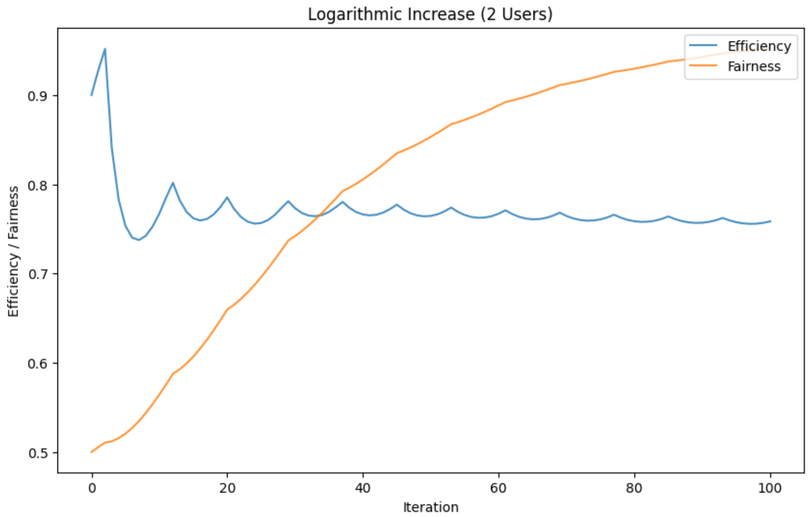

Figure 21. Change in Efficiency and Fairness for Logarithmic Increase (2 Users)

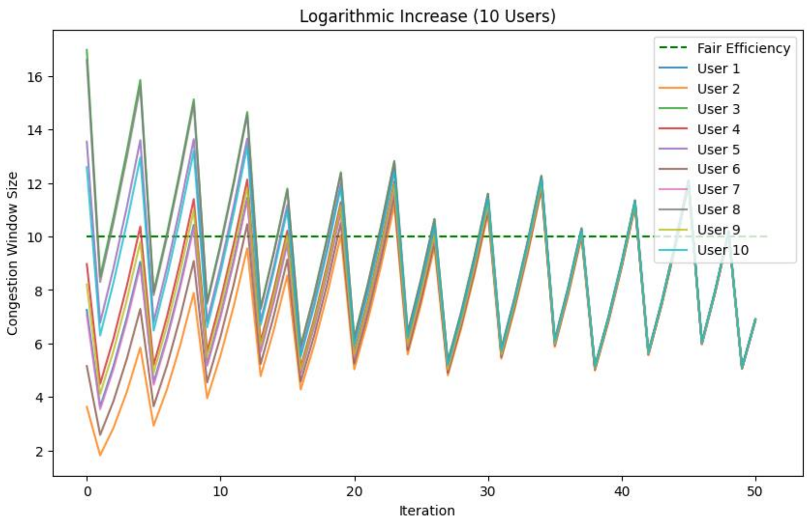

Figure 22. Change in Congestion Window for Logarithmic Increase (10 Users)

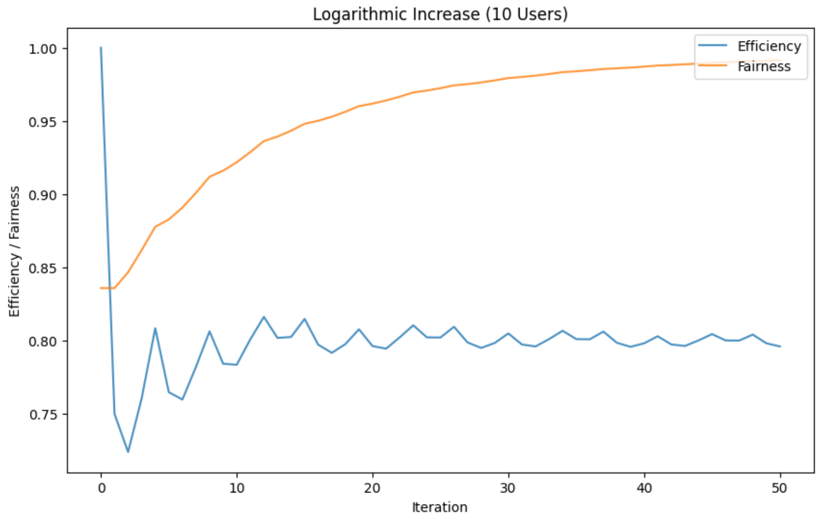

Figure 23. Change in Efficiency and Fairness for Logarithmic Increase (10 Users)

Appendix B