COMP 691

Online Algorithms and Competitive Analysis

Project Report

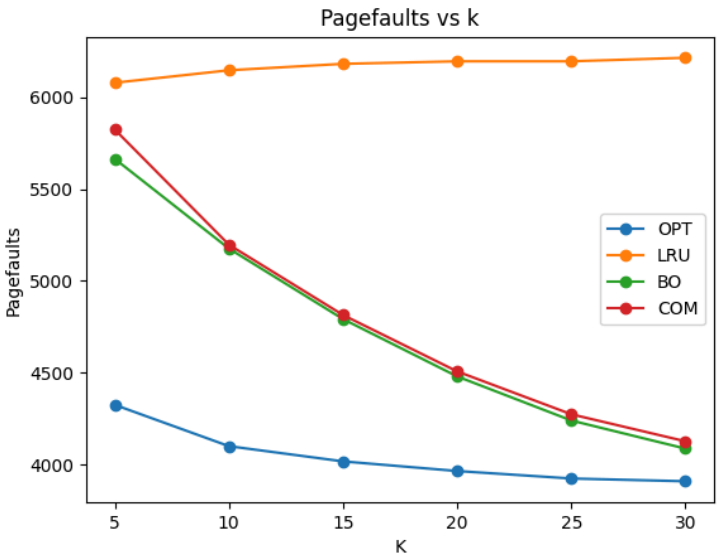
Submitted to: **Professor Denis Pankratov**

Submitted by:

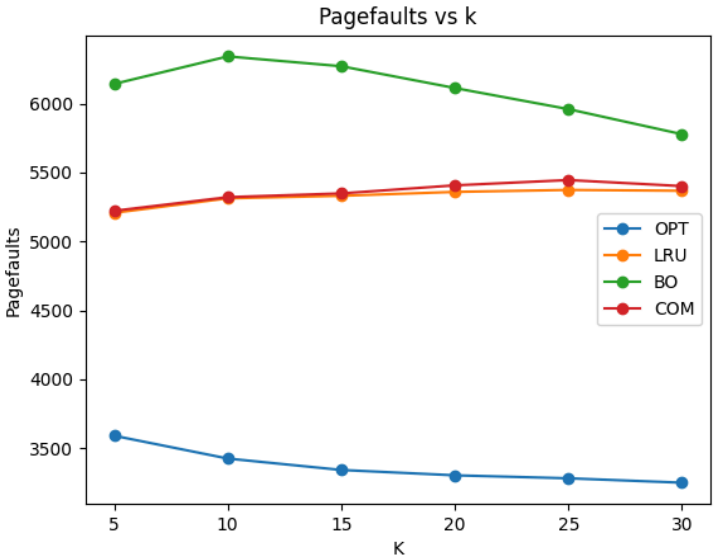
Dhairya Patel

40231545

Trend 1 (dependence on k):



Regime 1



Regime 2

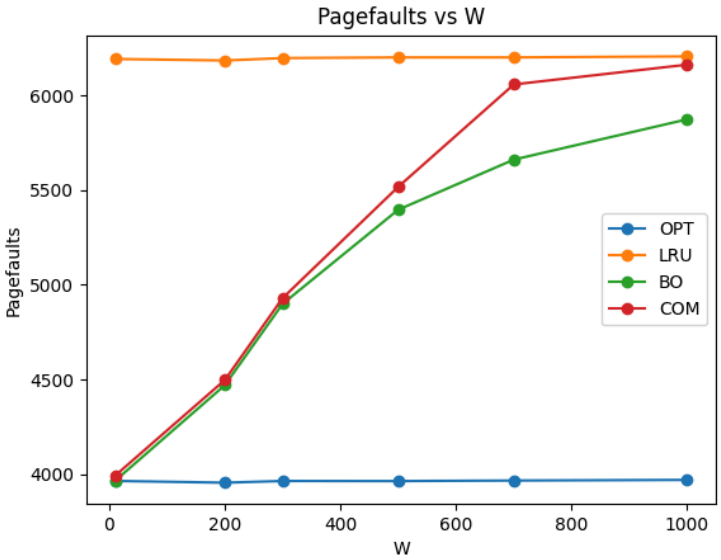
Regime 1 where OPT is significantly better than BlindOracle with $\hat{h}$ values which is significantly better than LRU

In this experiment, we selected certain values for the variables: n = 10000, ε = 0.5, t = 0.5, w = 200, and thr = 0.15. We are now changing the values of k, which are set to [5, 10, 15, 20, 25, 30]. To preserve a relationship between *k* and N, where N = 10k, we are using the values of N as [50, 100, 150, 200, 250, 300]. We can see that for this set of parameters, we get OPT which is better than Blind Oracle and which in turn is better than LRU. Here as we increase the value of *k* and N we can see that page faults for OPT and Blind Oracle decrease because we have kept w (window size) of 200 so as *k* increases the noise in $\hat{h}$ becomes less and less as it goes near the size of noise window. page faults for LRU on the other side are growing as we know that as we increase *k* page faults will increase. We can see that Combined Alg follows the Blind Oracle which is expected because the pagefaults in LRU are more than BlindOracle.So Combined Alg switched from LRU to BlindOracle when it saw the page faults in LRU increased.

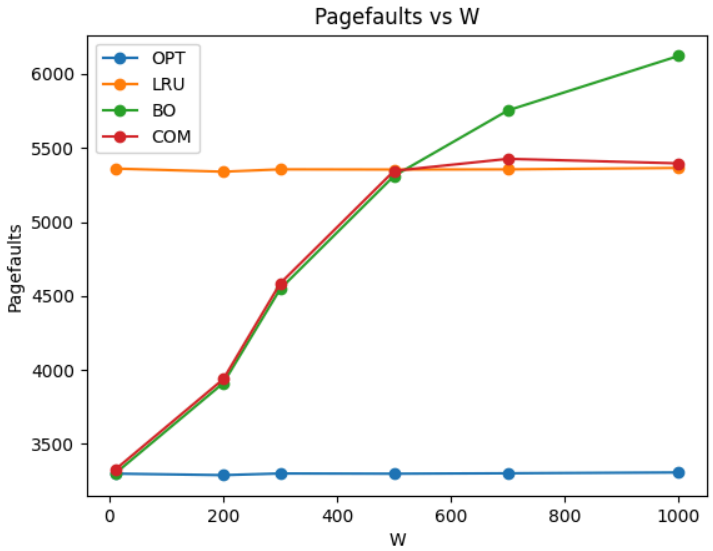
Regime 2 where OPT is significantly better than LRU which is significantly better than BlindOracle with $\hat{h}$ values.

Here, we are adjusting the values of *k* = [5, 10, 15, 20, 25, 30], and to keep the link between *k* and N (N = 10k), we use values of N = [50, 100, 150, 200, 250, 300]. We also took values of n = 10000, ε = 0.6, t = 0.7, w = 1000, and thr = 0.15 .It is evident that with these particular sets of settings, OPT outperforms LRU and BlindOracle itself. Since we are increasing the value of *k* as predicted, we can observe that OPT and Blind Oracle are following the same trajectory as in regime 1. In this scenario, it's anticipated that page faults in LRU increase, as expected. The Combined Algorithm tracks LRU because it generally incurs fewer page faults. However, it's worth noting that the Combined Algorithm's page faults are slightly higher than LRU. This is expected because the algorithm incurs additional *k* page faults when transitioning between LRU and Blind Oracle, or vice versa.

Trend 2 (dependence on w):



Regime 1



Regime 2

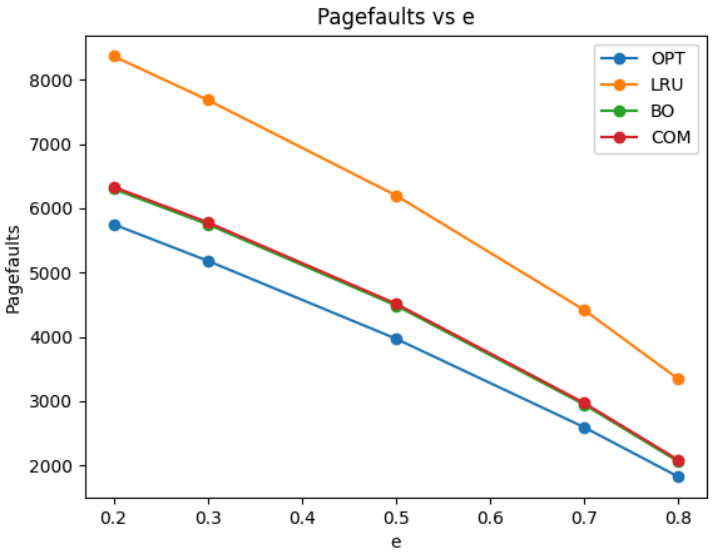
Regime 1 where OPT is significantly better than BlindOracle with $\hat{h}$ values which is significantly better than LRU.

In this section, we selected certain values for the variables:k = 20; N = 200; n = 10000; ε = 0.5; t = 0.5, and thr = 0.15. Now varying the value of w which is a noise parameter;w = [10,200,300,500,700,1000]. Here we can see that the opt remains somewhat constant it does not change much as it is not dependent on the value of w. We can see that the page faults of Blind Oracle increase as we increase the value of w because it increases noise in the $\hat{h}$ a sequence that is input to BlindOracle. Here the LRU also stays constant over all the values of the w as LRU also is not dependent on the value of w. Here we can see that the Combined algorithm follows BlindOracle at the start and then as we increase the value of the w it goes far from the BlindOracle because BlindOracle starts making more pagefaults so Combined Alg makes more switches so there are more PageFaults in Combined.

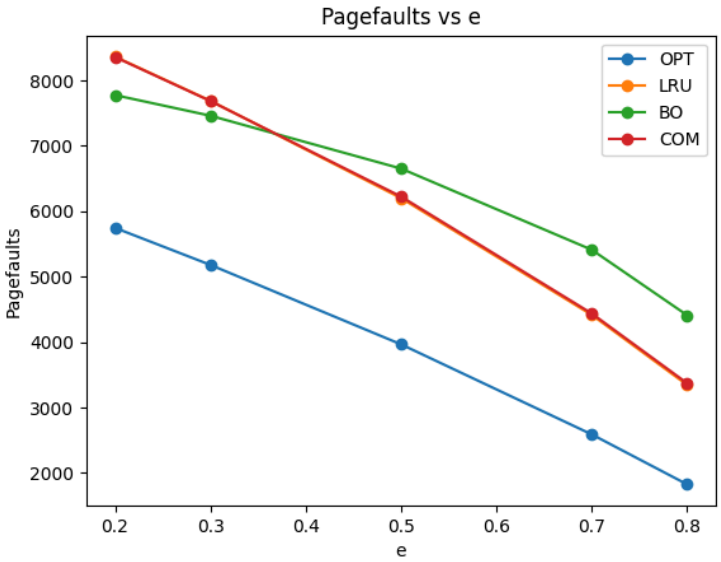
Regime 2 where OPT is significantly better than LRU which is significantly better than BlindOracle with $\hat{h}$ values.

Here the selected values for the parameters are k = 20; N = 200 ; n = 10000; ε = 0.6; t = 0.7 and the = 0.15. We vary the values of w = [10,200,300,500,700,1000]. Here the OPT, LRU, and BlindOracle follow the same trend as in Regime 1. Here we make an interesting observation that till the value of w was 500 the BlindOracle was performing better than the LRU but as w increased we saw that LRU started performing better than LRU. Due to this Combined ALG was first following BlindOracle and then when LRU started performing better the Combined Alg switched to LRU. Another thing to observe is as we Combined switched to LRU we got PageFaults slightly more than the LRU because at the time of switching, we added k page faults.

Trend 3 (dependence on ε):



Regime 1



Regime 2

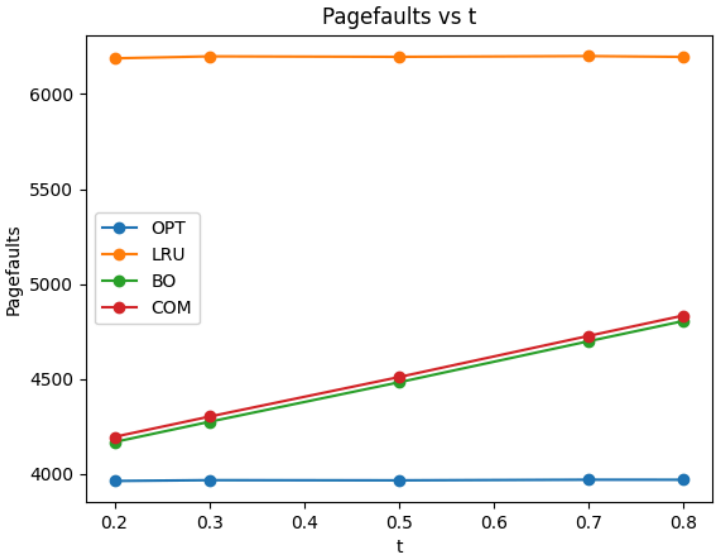
Regime 1 where OPT is significantly better than BlindOracle with $\hat{h}$ values which is significantly better than LRU

The parameters in this case have been set at k = 20, N = 200, n = 10000, w = 200, t = 0.5, and the = 0.15. We change ε = [0.2,0.3,0.5,0.7,0.8] to different numbers. Here ε belongs to (0, 1) is an extra parameter that controls the amount of locality in a sequence to be generated. We can note that as we increase the ε the page faults decrease this is the case for all the algorithms. This is because as we increase the ε the probability that the next page will be from the cache increases so every time we get a page that is already available in the current cache. Here Combined Alg follows the BlindOracle because it has fewer pagefaults than LRU.

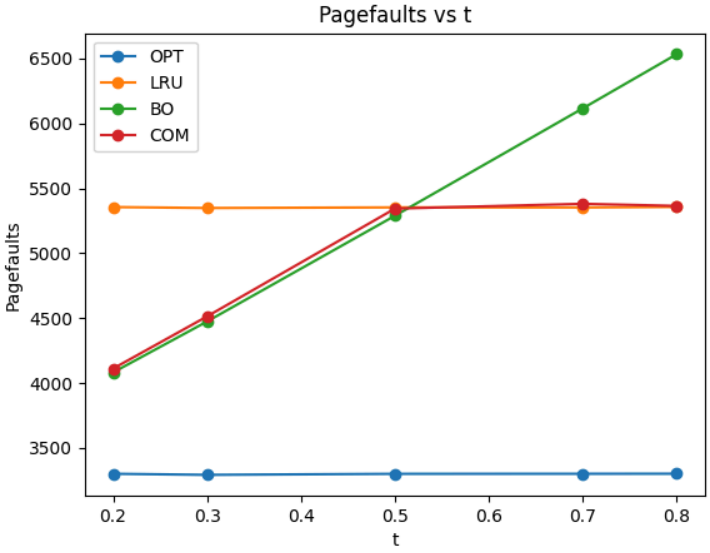
Regime 2 where OPT is significantly better than LRU which is significantly better than BlindOracle with $\hat{h}$ values.

The parameters in this case have been set at k = 20, N = 200, n = 10000, w = 1000, t = 0.7, and the = 0.15. We change ε = [0.2,0.3,0.5,0.7,0.8] to different numbers. Here also the trend for all algorithms is the same as of regime 1. Here Blind Oracle starts better than LRU but as we increase the value of ε , LRU performs better. The Combined Alg follows the LRU from the start this can be explained as there is not much difference between page faults of Blind Oracle and LRU.

Trend 4 (dependence on t):



Regime 1



Regime 2

Regime 1 where OPT is significantly better than BlindOracle with $\hat{h}$ values which is significantly better than LRU

In this section, we have fixed the values of k = 20, N = 200, n = 10000 , ε = 0.5, w = 200, and thr = 0.15. We vary the t = [0.2,0.3,0.5,0.7,0.8] . here t is a noise parameter. So with probability 1- t , we don't change the $\hat{h}$ and with probability t we change $\hat{h}$. Here we can observe that LRU and OPT stays almost constant because they do not depend on t . Here we can also see that page faults of Blind Oracle increase as we increase t which is expected as we know that it is dependent on t. Here as BlindOracle is performing better than LRU so Combined Alg follows BlindOracle.

Regime 2 where OPT is significantly better than LRU which is significantly better than BlindOracle with $\hat{h}$ values.

In this section, we have fixed the values of k = 20, N = 200, n = 10000, ε = 0.6, w = 1000, and thr = 0.15. We vary the t = [0.2,0.3,0.5,0.7,0.8]. Here the OPT and LRU have the same trend as in regime 1. BlindOracle as expected worsens in performance as t increases. Here it is fascinating to note that at first BlindOracle performs better in the start then it worsens than LRU. Combined Alg as expected follows the Algorithm which has less number of page faults. So here we can see that before t = 0.5, Combined Alg follows BlindOracle, and then after that follows LRU.

Conclusion:

Based on the outlined trends in the system's behavior across different parameter variations, we can draw several conclusions:

Dependence on *k*: The increase in *k* and *N* generally leads to a decrease in page faults for OPT and Blind Oracle, while LRU tends to increase. The Combined Algorithm dynamically adapts between LRU and Blind Oracle based on observed page faults.

Dependence on *w*: Changes in *w* affect the performance of Blind Oracle more significantly than OPT and LRU. As *w* increases, Blind Oracle's page faults rise, leading to Combined Algorithm switching behavior, resulting in slightly higher page faults due to transitions.

Dependence on *e*: Increasing *e* decreases page faults across all algorithms due to increased locality in page references. However, the performance of Blind Oracle relative to LRU shifts with different *e* values, influencing the Combined Algorithm's choice between the two.

Dependence on *t*: Blind Oracle's performance is inversely proportional to *t*, as it introduces noise in the sequence. Combined Algorithm's behavior switches between Blind Oracle and LRU based on their relative performance, particularly evident when *t* surpasses a threshold around 0.5.

Overall, the behavior of the system is intricately linked to the interplay of these parameters, with the Combined Algorithm dynamically adjusting its strategy between LRU and Blind Oracle to optimize page fault performance.