CONCORDIA UNIVERSITY

Dept. of Computer Science and Software Engineering

COMP 691 - Winter 2024

Online Algorithms and Competitive Analysis

Denis Pankratov

Report on Paging Problem with Predictions

By: Punamkumar Vekariya

Concordia ID: 40290462

Trend 1: Average Number of Page Faults vs. Cache Size (k)

The following plots depict the average number of page faults against the cache size for different paging algorithms: Optimal (OPT), Blind Oracle, Least Recently Used (LRU), and Combined.

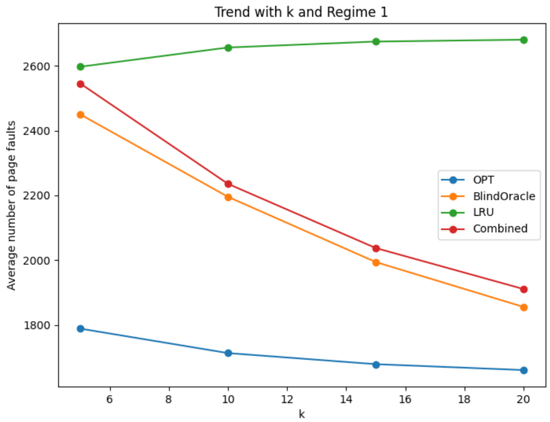

Figure 1

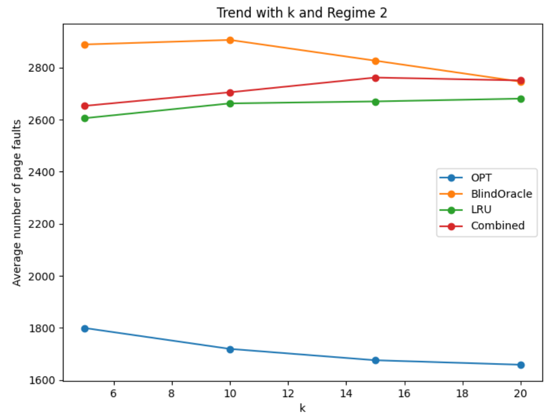

Figure 2

Regime 1: Low Noise (k = [5, 10, 15, 20])

In this regime, as the cache size (k) increases, I observe varying trends in the average number of page faults for different algorithms.

a. OPT: As expected, the average number of page faults decreases with increasing cache size. This is because a larger cache can accommodate more pages, reducing the likelihood of page faults.

b. BlindOracle: Similar to OPT, BlindOracle also exhibits a decreasing trend in page faults with increasing cache size. However, its performance improvement is as significant as compared to OPT. This is because BlindOracle relies on predicting future page accesses, which may benefit as much from increased cache size as OPT does.

c. LRU: Surprisingly, the average number of page faults for LRU does not decrease with increasing cache size. I observe slight increases in page faults with an increase in cache sizes. LRU operates on the principle of removing the least recently used page from the cache when a new page arrives. In a scenario where the cache size (k) increases, LRU might experience a higher number of page faults because it keeps pages in the cache based on their access history. This results in more frequent cache evictions and page faults for LRU.

d. Combined: The Combined algorithm, which switches between BlindOracle and LRU based on a threshold, here it follows the same trend as BlindOracle because it is performing better than LRU. It follows the decreasing trend observed in BlindOracle, suggesting that the switching works correctly.

Regime 2: High Noise (k = [5, 10, 15, 20])

In this regime, characterized by higher noise levels, I observe similar trends in the average number of page faults with varying cache sizes, but with some notable differences.

a. OPT: The trend of decreasing page faults with increasing cache size is consistent with Regime 1.

b. BlindOracle: Unlike Regime 1, where BlindOracle's performance improved with increasing cache size, in Regime 2, BlindOracle's performance deteriorated as cache size increased because of hogh noise.

c. LRU: Similar to Regime 1, LRU's performance in Regime 2 does not consistently improve with increasing cache size.

d. Combined: The Combined algorithm's performance in Regime 2 follows similar trends as in LRU, but with larger fluctuations and variability.

Trend 2: Average Number of Page Faults vs. Noise Parameter (*w*)

The plots below illustrate the relationship between the average number of page faults and the noise parameter *w*w for different page replacement algorithms.

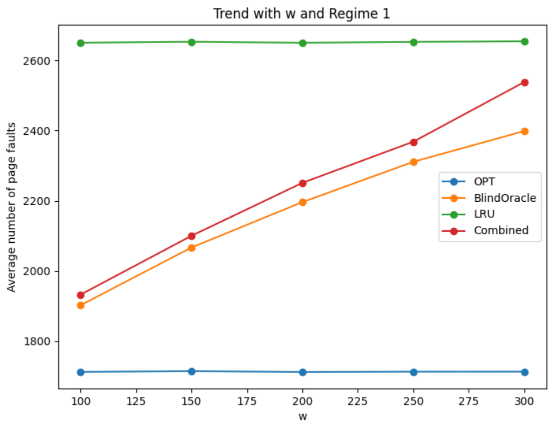

Figure 3

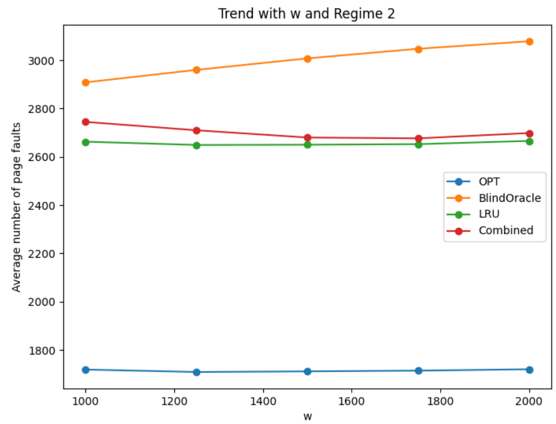

Figure 4

Regime 1: Low Noise (w = 100 - 300)

In this regime, characterized by lower noise levels, I observe how varying the noise parameter (w) affects the average number of page faults for different algorithms.

a. OPT: The average number of page faults for OPT remains stable across different values of (w), indicating that changes in noise levels have no impact on OPT's performance
b. BlindOracle: As (w) increases, the average number of page faults for BlindOracle also increases. This suggests that higher noise levels make it more challenging for BlindOracle to accurately predict future page accesses, resulting in more page faults. The increase in page faults is gradual, indicating a proportional increase in performance degradation with higher noise levels.
c. LRU: Similar to OPT, LRU's performance remains stable across different values of (w).
d. Combined: The Combined algorithm's performance exhibits trends similar to BlindOracle, with increasing average page faults as (w) increases. However, the magnitude of the increase is slightly higher compared to BlindOracle.

Regime 2: High Noise (w = 1000 - 2000)

In this regime, characterized by higher noise levels, I observe how the performance of different algorithms is affected by increasing noise parameters (w).

a. OPT: Similar to Regime 1, OPT's performance remains stable across different values of (w).
b. BlindOracle: As expected, BlindOracle's performance deteriorates significantly with increasing noise levels. The average number of page faults increases substantially as (w) increases, indicating that BlindOracle's predictive capabilities are severely affected by noise. The increase in page faults is more drastic compared to Regime 1, highlighting the sensitivity of BlindOracle to noise in high-noise environments.
c. LRU: LRU's performance remains stable across different values of (w), similar to its behavior in Regime 1.
d. Combined: The Combined algorithm's performance in Regime 2 mirrors the trends observed in BlindOracle, with a significant increase in average page faults as (w) increases. However, the increase in page faults is slightly higher compared to BlindOracle.

Trend 3: Average Number of Page Faults vs. Locality Parameter (ϵ)

The plots below display the impact of the locality parameter ϵ on the average number of page faults for different page replacement algorithms.

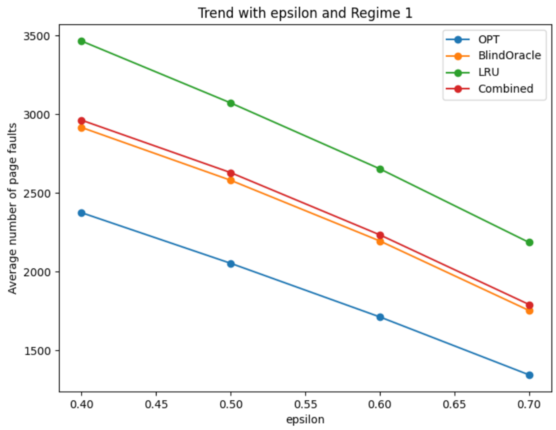

Figure 5

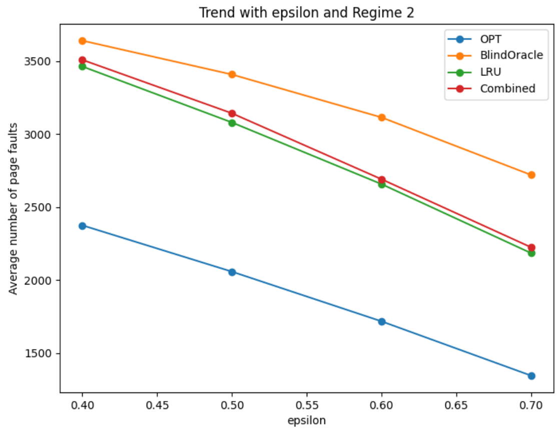

Figure 6

Regime 1: Low Locality epsilon = 0.4 - 0.7

In this regime, characterized by lower values of the locality parameter ϵ, I observe how varying ϵ affects the average number of page faults for different algorithms.

a. OPT: The average number of page faults for OPT decreases as ϵ increases. This is expected since higher values of ϵ indicate higher locality, making random generated sequence more consistent.
b. BlindOracle: Similar to OPT, the average number of page faults for BlindOracle decreases with increasing ϵ.
c. LRU: LRU exhibits a similar trend to OPT and BlindOracle, with the average number of page faults decreasing as ϵ increases.
d. Combined: The Combined algorithm's performance mirrors the trends observed in OPT, BlindOracle, and LRU, with a decrease in the average number of page faults as ϵ increases.

Regime 2: High Locality epsilon = 0.4 - 0.7

In this regime, characterized by higher values of the locality parameter ϵ, I observe how the performance of different algorithms is affected.

a. OPT: Similar to Regime 1 The average number of page faults for OPT decreases as ϵ increases.
b. BlindOracle: Similar to OPT, BlindOracle's performance also keeps decreasing across different values of ϵ. This suggests that BlindOracle's predictive capabilities may not benefit significantly from higher levels of locality in this regime.
c. LRU: LRU's performance exhibits a slight decrease in the average number of page faults as ϵ increases, albeit less pronounced compared to Regime 1.
d. Combined: The Combined algorithm's performance shows trends similar to LRU, with a decrease in the average number of page faults as ϵ increases.

Trend 4: Average Number of Page Faults vs. Noise Parameter (τ)

The plots below illustrate how the noise parameter τ affects the average number of page faults for different page replacement algorithms.

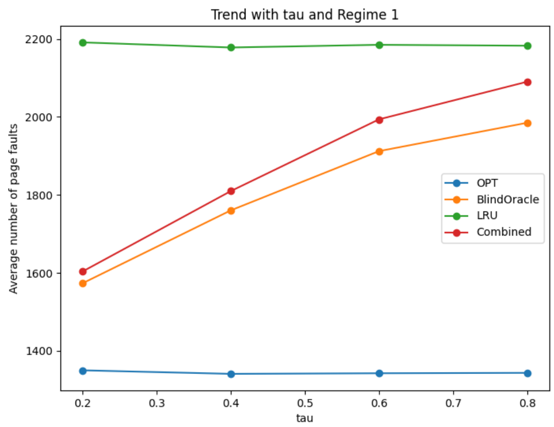

Figure 7

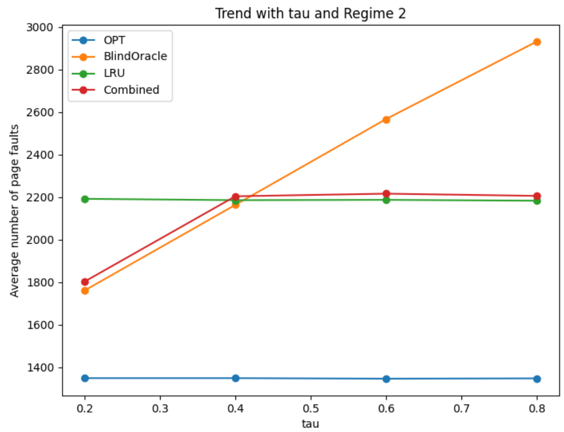

Figure 8

Regime 1: Low Noise tau = 0.2 - 0.8

In this regime, characterized by low noise levels, I observe how varying τ affects the average number of page faults for different algorithms.

a. OPT: The average number of page faults for OPT remains stable across different values of τ. This is expected since OPT does not depend on tau.
b. BlindOracle: The average number of page faults for BlindOracle increases with higher values of τ. This is because BlindOracle relies on predicted values $\hat{h}$ which are subject to noise. Higher noise levels lead to less accurate predictions, resulting in more page faults.
c. LRU: LRU's performance remains stable across different values of τ, similar to OPT.
d. Combined: The Combined algorithm's performance mirrors the trends observed in BlindOracle, with an increase in the average number of page faults as τ increases.

Regime 2: High Noise tau = 0.2 - 0.8

In this regime, characterized by high noise levels, I observe how the performance of different algorithms is affected.

a. OPT: The average number of page faults for OPT remains stable across different values of τ, similar to Regime 1.
b. BlindOracle: BlindOracle's performance deteriorates significantly with higher values of τ, as evidenced by the substantial increase in the average number of page faults. This highlights the sensitivity of BlindOracle to noise in the predicted values leads to poorer performance.
c. LRU: LRU's performance remains stable across different values of τ, similar to Regime 1.
d. Combined: Initially, the Combined algorithm follows the behavior of BlindOracle, as BlindOracle performs better than LRU. However, at $\tau = 0.4$, there is a shift where LRU starts to perform better than BlindOracle. Therefore, after that point, the Combined algorithm starts to follow LRU instead of BlindOracle.

Conclusion

In this analysis, I investigated four trends in the performance of paging algorithms under varying parameters. All four trends also contain two different regimes, I analyzed how changes in the value of $k, w, esilon, and\ tau$ affect the average number of page faults for four different algorithms: *OPT, LRU, BlindOracle, and Combined.*

In trend one, I observed the dependence on k, where k represents the number of Cache size. In both regimes, I found that as k increases, the average number of page faults decreases for all algorithms except LRU. This is consistent with the intuition that a larger cache size allows for better caching, resulting in fewer page faults.

In trend two, I examined the influence of w, the noise parameter, on the predicted sequence generator. My findings revealed that as w increases, the average number of page faults increases as well for BlindOracle and Combined algorithm, which indicates an increase in distribution in sequence leads to more page faults. This suggests that a larger w introduces more noise.

In trend three, I explored the impact of epsilon, which controls the amount of locality in the generated sequence. I observed that higher values of epsilon lead to a decrease in the average number of page faults for all algorithms. This says that increased locality in sequence improves algorithm performance by enhancing cache efficiency.

In trend four, I investigated the dependence on tau, the noise parameter. My results demonstrated that higher levels of noise tau lead to an increase in the average number of page faults for BlindOracle and the Combined algorithm, while OPT and LRU remain stable.