

University of Calgary

Assignment 2: Page Replacement

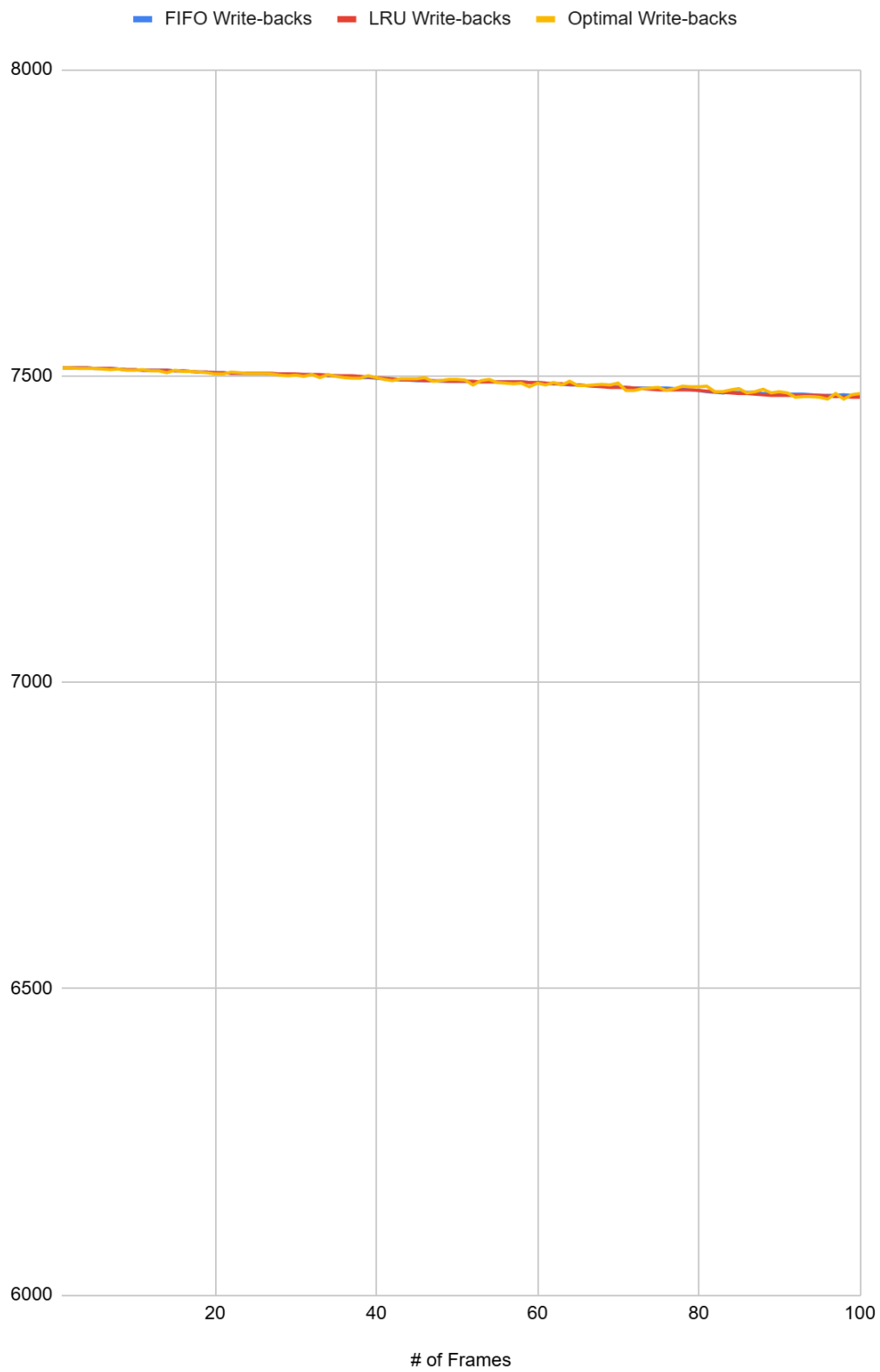
Anmol Ratol (30231177)

CPSC 457

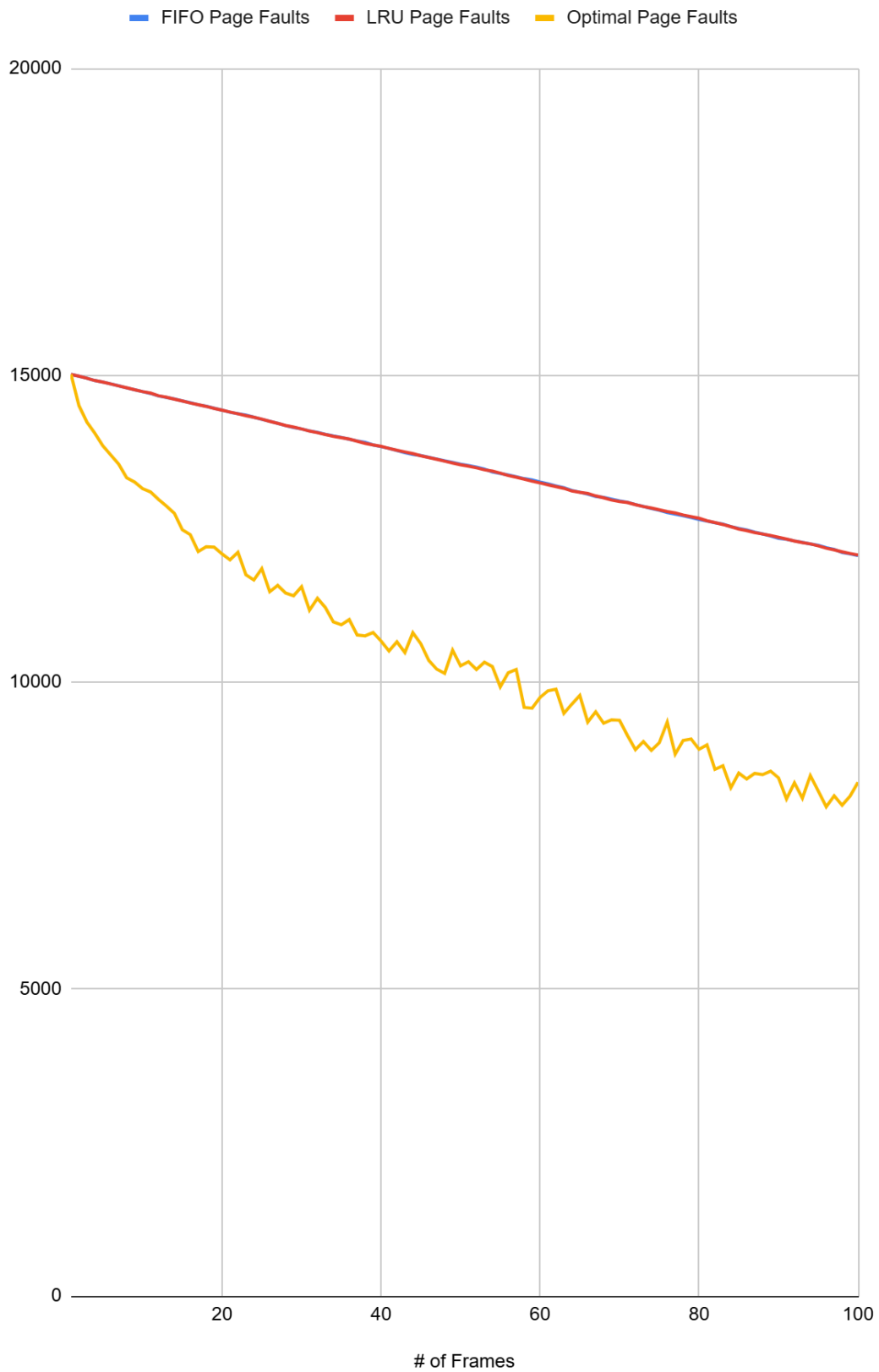
Professor Jalal Kawash

10 Oct. 2024

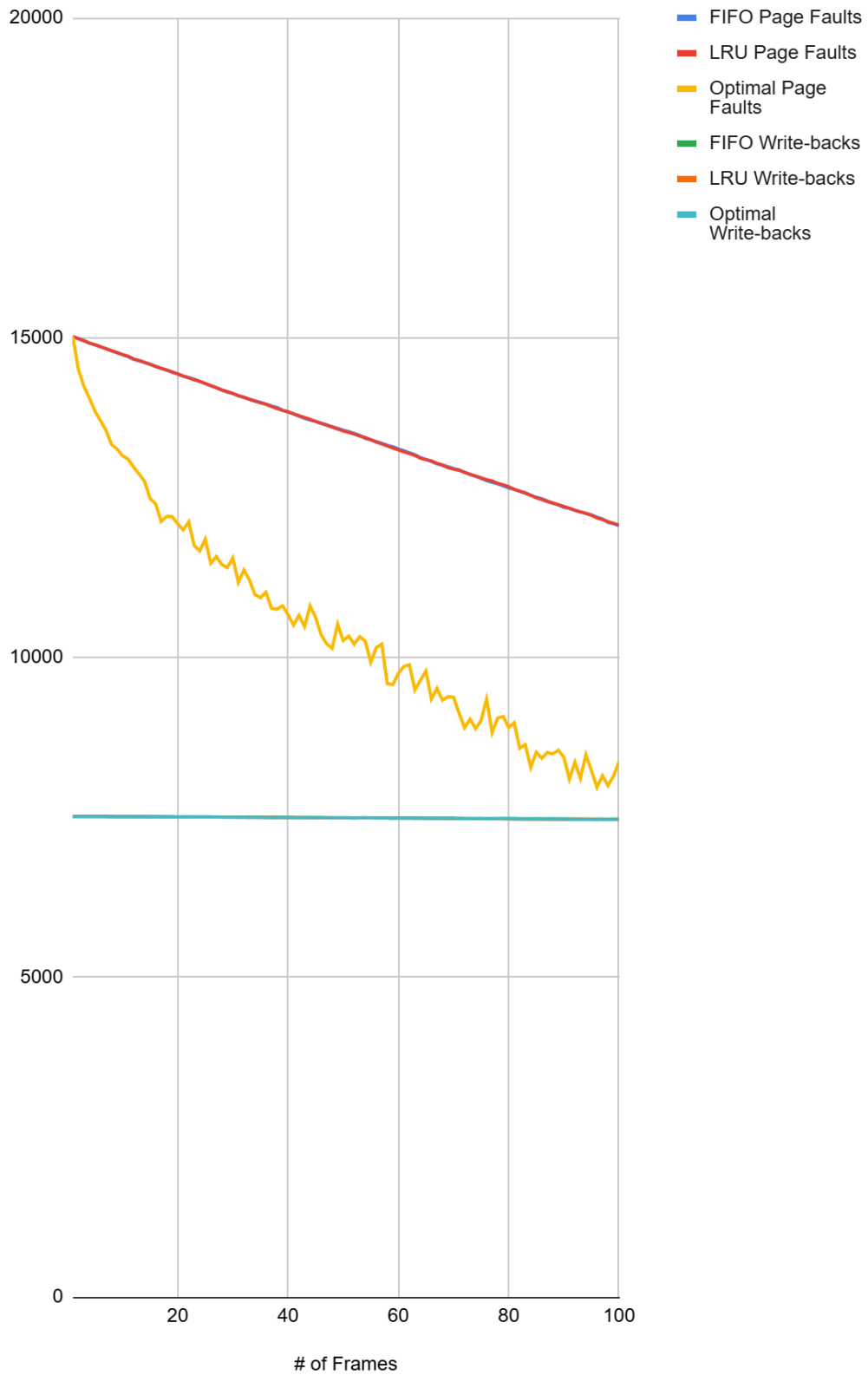
## FIFO, LRU, and Optimal Algorithm Write-backs



## FIFO, LRU, and Optimal Algorithm Page Faults



## FIFO, LRU, and Optimal Algorithm Write-backs And Page Faults

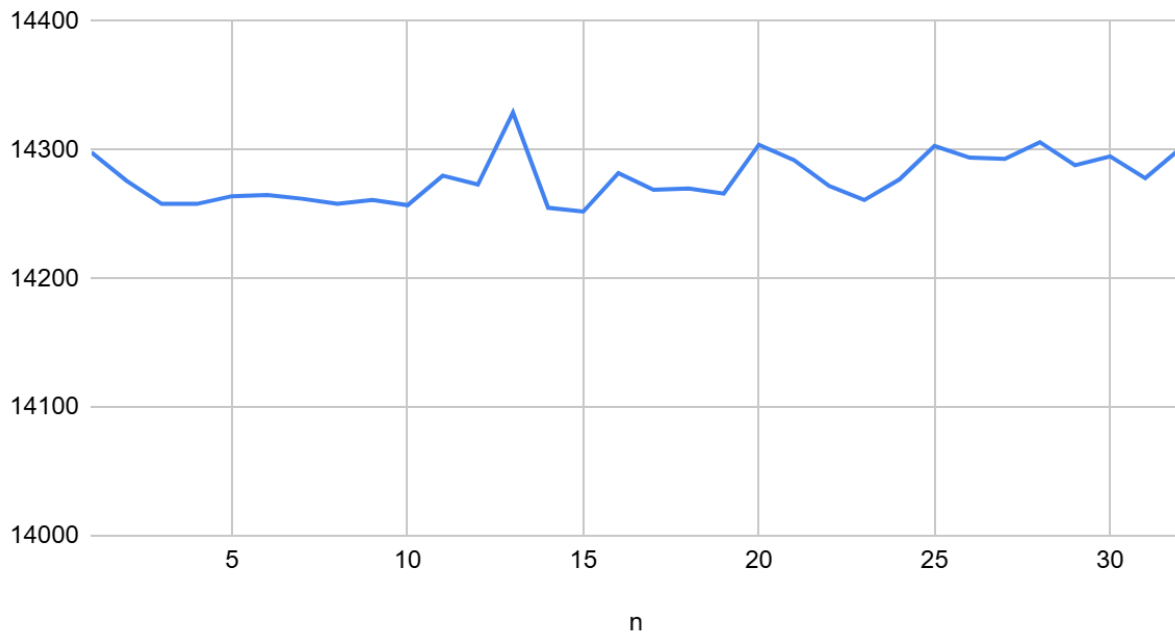


As is to be expected with FIFO, the amount of page faults decreased more or less linearly with an increase in frames. The number of write-backs also decreased slightly, owing to the fact that they only occur when dirty pages are rotated out of memory, and with more frames more dirty pages still loaded in memory at the end of the program execution.

Similarly, the number of page faults also decreased linearly with LRU, and it was generally more efficient than FIFO, although it actually generated more page faults than FIFO once there were around 75 or more frames used. The number of write-backs were virtually identical between the two algorithms, never varying by more than 5 write-backs.

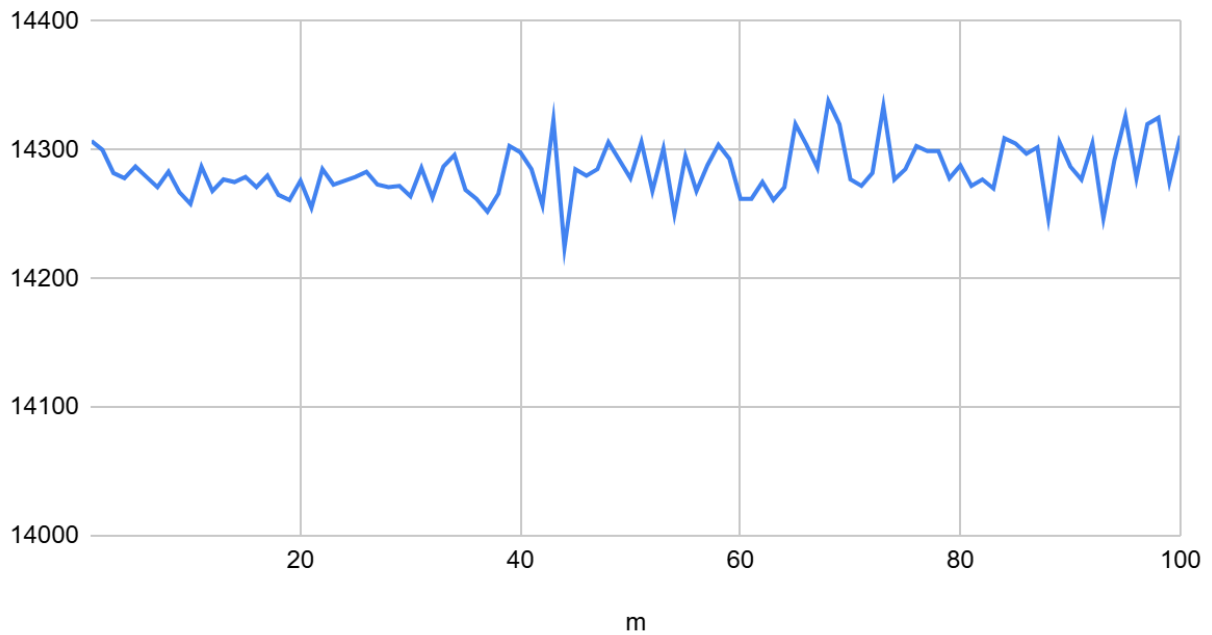
By comparison, the optimal algorithm was far more efficient, decreasing far more sharply per frame compared to the two prior algorithms. It was consistently at only 70 percent of the page faults of either LRU or FIFO. The number of write-backs was consistent with both algorithms, staying within 5 write-backs of either algorithm.

### Clock Algorithm Page Faults For Various N Values (25 frames)



The second chance algorithm seemed to hover around the value of FIFO (14,281) in terms of page faults generated, with some values being slightly better and some being slightly worse. The differences are very slight (the most efficient  $n$  value of 15 generated 14,252 page faults), and they don't seem to obey any significant trend, although the values are slightly higher around  $n = 13$ .

### Clock Algorithm Page Faults For Various M Values (25 frames)



The algorithm oscillated around the amount of page faults generated by FIFO for 25 frames (14,281) regardless of  $m$  values. The most efficient value of  $m$  was 44, generating 14,224 page faults. There doesn't appear to be any significant trend.