Comparison of Move-to-Front (MTF) On-Line Strategy to Optimal (OPT) Off-Line Strategy

Key Differences:

MTF is given requests one-at-a-time and thus applies the obvious strategy

1) Go down the list to the requested item
2) Bring it to the front of the list

OPT is given the entire sequence of accesses and may perform anticipatory actions to improve the lists. (There is no charge for the computation to determine the actions. This problem is NP-hard [1], so it is more than just a convenience . . .) When processing a request, the following processing occurs

1) Go down the list to the requested item
2) Optionally, bring requested item closer to the front of the list
3) Optionally, perform transpositions on adjacent elements

Potential Function:

To simplify the analysis, assume both strategies start with same list. The lists may differ when processing the sequences.

The potential (Φ) for MTF is the number of inversions in the MTF list with respect to the OPT list. (In other words, count the number of pairs whose order is different in the two lists. 2, 3, 4, 5, 1 and 3, 5, 4, 1, 2 have 10 inversions.

Defining Φ this way gives $\Phi(0) = 0$ and $\Phi(i) \geq 0$, which are critical later in the analysis.

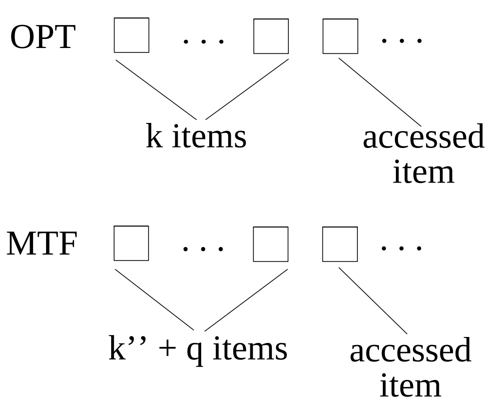
Actual Cost:

MTF: Count number of items touched during search. No charge to bring requested item to front. (Searching 2, 3, 4, 5, 1 for 5 costs 4 units)

OPT: Cost is the sum of the number of items touched during search and the number of transpositions. No charge for bringing requested item closer to the front.

Claim that $\hat{C}_{MTF}(i) = C_{MTF}(i) + \Phi(i) - \Phi(i - 1) \leq 2C_{OPT}(i) - 1$

The following diagram describes the worst-case situation:



When the item is accessed by OPT, it is also moved $k' \le k$ positions toward the beginning of the list and then p transpositions occur. This gives $C_{OPT}(i) = k + p + 1$. The movement of the accessed item increases Φ by no more than k'. Likewise, the transposes increase Φ by no more than p.

For MTF, the $k'' \le k - k'$ items and the q items consider that a subset of the k - k' items now at the beginning of OPT, along with q items that are now after the accessed item in OPT, precede the accessed item in MTF.
This gives $C_{MTF}(i) = k'' + q + 1$. With respect to Φ after the OPT processing, there is an increase of k'' and a decrease of q.

Overall, this gives:

$$
\begin{aligned}
\hat{}C_{MTF}(i) &\le C_{MTF}(i) + k' + p + k'' - q \\
&= k'' + q + 1 + k' + p + k'' - q \\
&= 2k'' + k' + p + 1 \\
&\le 2(k - k') + k' + p + 1 \\
&= 2k - k' + p + 1 \\
&= (2k + 2p + 2 - 1) - k' - p \\
&\le 2C_{OPT}(i) - 1
\end{aligned}
$$

Summing over sequences:

$$\sum_{i=1}^{m} \hat{}C_{MTF}(i) =$$

$$\sum_{i=1}^{m} \left(C_{MTF}(i) + \Phi(i) - \Phi(i-1) \right) = \sum_{i=1}^{m} C_{MTF}(i) + \Phi(m) - \Phi(0) \le 2 \sum_{i=1}^{m} C_{OPT}(i) - m$$

$$\sum_{i=1}^{m} C_{MTF}(i) \le 2 \sum_{i=1}^{m} C_{OPT}(i) - m - \Phi(m)$$

$$\le 2 \sum_{i=1}^{m} C_{OPT}(i)$$

[1] C. Ambühl, "Offline List Update is NP-Hard", *Lecture Notes in Computer Science 1879*, Springer-Verlag, 2000.