Self-organizing lists

List L of n elements

- Operation Access(x) costs rank(x) = distance of x from head of L
- L can be reordered by transposing adjacent elements, cost = 1

Ex:

$L \to (12) \to (3) \to (50) \to (14) \to (17) \to (4)$

transpose cost = 1 cost = 4 to access

$\to (12) \to (50) \to (3) \to (14) \to (17) \to (4)$

Def: A sequence [redacted] S of operations is provided one at a time. For each operation, an on-line algorithm must execute the operation immediately (algorithm A).
Off-line algorithm may see all of S in advance.
Tetris: on-line, cannot see in advance.

Goal: minimize total cost $C_A(S)$

Worst-case analysis (online)

Adversary always accesses tail element of L.

$C_A(S) = \Omega(|S| \cdot n)$ worst case

Average-case analysis

Suppose element x is accessed with probability $p(x)$.
→ a priori distribution on elements (input)

$$E[C_A(S)] = \sum_{x \in L} p(x) \cdot \text{rank}_L(x)$$, which is minimized when L is sorted in decreasing order w.r.t. p.

Heuristic: keep count of # times each element is accessed, and maintain list in order of decreasing count.
(assume dist. stays the same ; LLN)

Practice: "Move-to-front" heuristic (MTF)

my comment
actually an approximation! by below analysis

After accessing x, move x to the head of list using transposes. cost $= 2 \cdot \text{rank}(x)$

access + transposes

Responds well to locality in S (not static distribution, access x makes x more likely to be accessed again)

Competitive Analysis

Def. An on-line algorithm A is $\underline{\alpha\text{-competitive}}$ if $\exists$ constant k s.t. for any sequence S of operations

$$C_A(S) \le \alpha\, C_{OPT}(S) + k$$

could be a function or $\underline{\text{const}}$

opt. off line algorithm

no assumption of op distribution [redacted] in S sequence

Theorem: MTF is $\underline{4\text{-competitive}}$ for [redacted] self-organizing lists.

Proof: Let L_i be MTF's list after ith access

L_i^* ... OPT's ....

[redacted]

Let c_i = MTF's cost for ith op

$= 2 \cdot \text{rank}_{L_{i-1}}(x)$ if it accesses x

c_i^* = OPT's cost for ith op

$= \text{rank}_{L^*_{i-1}}(x) + t_i$ if OPT performs t_i transposes

Define the potential function

$\Phi : \{L_i\} \to \mathbb{R}$ by

precedes in the L_i list from head

$$\Phi(L_i) = 2\left|\{(x,y) : x \prec_{L_i} y \text{ and } y \prec_{L_i^*} x\}\right|$$

disagreements between L_i and L_i^*

$= 2 \cdot \#\text{ inversions}$

Ex:

$L_i \quad \to (E) \to (C) \to (A) \to (D) \to (B)$

$L_i^* \quad \to (C) \to (A) \to (B) \to (D) \to (E)$

$\Phi(L_i) = 2 \cdot |\{(E,C), (E,A), (E,D), (E,B), (B,D)\}| = 10$

$= 2 \cdot 5$ inversions

Note: $\Phi(L_i) \geq 0 \quad \forall i$

$\Phi(L_0) = 0$ if MTF and OPT start with same list

the more L^* differs from L as i progresses,

the more work is stored for MTF to use

How much does Φ change from 1 transpose?

$\Delta \Phi = \pm 2$ (creates or destroys 1 inversion)

What happens when op i accesses x?

Define $A = \{y \in L_{i-1} : y \prec_{L_{i-1}} x \text{ and } y \prec_{L^*_{i-1}} x\}$

$B = \{y \in L_{i-1} : y \prec_{L_{i-1}} x \text{ and } y \succ_{L^*_{i-1}} x\}$

$C = \{y \in L_{i-1} : y \succ_{L_{i-1}} x \text{ and } y \prec_{L^*_{i-1}} x\}$

$D = \{y \in L_{i-1} : y \succ_{L_{i-1}} x \text{ and } y \succ_{L^*_{i-1}} x\}$

L_{i-1}: | $A \cup B$ | x | $C \cup D$ |

$r = \text{rank}_{L_{i-1}}(x)$

disjoint

$r = |A| + |B| + 1$

L^*_{i-1}: | $A \cup C$ | x | $B \cup D$ |

$r^* = |A| + |C| + 1$

$r^* = \text{rank}_{L^*_{i-1}}(x)$

Access(x)

When MTF moves x to front, it creates $|A|$ inversions and destroys $|B|$ inversions.

Each transpose by OPT [redacted] creates ≤ 1 inversion.

Thus

$\Phi(L_i) - \Phi(L_{i-1}) \leq 2(|A| - |B| + t_i)$

OPT can go anywhere need to add t_i $\to$ # transposes by OPT

Amortized cost is

$$\hat{c}_i = c_i + \Phi(L_i) - \Phi(L_{i-1})$$
$$\le 2r + 2(|A| - |B| + t_i) \quad \leftarrow r = |A| + |B| + 1$$
$$= 2r + 2[|A| - (r - 1 - |A|) + t_i]$$
$$= \cancel{2r} + 4|A| - \cancel{2r} + 2 + 2t_i$$
$$= 4|A| + 2 + 2t_i$$
$$\le 4(r^* + t_i) \quad \text{since } r^* = |A| + |C| + 1 \ge |A| + 1$$
$$= 4c_i^*$$

my comment
Amortized cost of Access(x) of MTF. The more L^* differs from L, the more work we store for MTF to use.
$\Delta\Phi$ accumulates the difference between L^* and L in terms of inversions.
↓
maps to # of transposes of which MTF did too many! Because MTF is not OPT.
upper bound of transposes

Thus

$$C_{MTF}(s) = \sum_{i=1}^{|s|} c_i = \sum_{i=1}^{|s|} (\hat{c}_i + \Phi(L_{i-1}) - \Phi(L_i))$$
$$\le \left(\sum_{i=1}^{|s|} 4c_i^*\right) + \underbrace{\Phi(L_0)}_{=0} - \underbrace{\Phi(L_{|s|})}_{\ge 0}$$
$$\le 4C_{OPT}(s) \quad ✓$$

4 - competitive

my comment
runtime comparison to OPT
↳ runtime approximation to OPT (offline) without knowing OPT (algorithm)

If we count transposes that move toward the front of L as "free" (models splicing x in and out of L in const. time), then MTF is 2-competitive.

What if $L_0 \ne L_0^*$? $\leftarrow \binom{n-1}{2}$

Then, $\Phi(L_0)$ might be $\Theta(n^2)$

Thus $C_{MTF}(s) \le 4C_{OPT}(s) + \Theta(n^2)$, which is still 4-competitive, since n^2 const as $|s| \to \infty$