List Access Problem (TIMESTAMP and COMB)

Vamsi Addanki

University of Vienna

vamsi.addanki@univie.ac.at

November 8, 2021

Overview

1. Preliminaries
   - Full Cost Model
   - Partial Cost Model
   - Pairwise Property
2. Deterministic TIMESTAMP
   - Pairwise Property of TIMESTAMP(0)
3. COMB

Preliminaries (Full Cost Model)

Accessing i^{th} element in the list costs i. For example, accessing the first element costs 1.

Preliminaries (Partial Cost Model)

Accessing i^{th} element in the list costs $i-1$. For example, accessing the first element costs 0.

Preliminaries (Pairwise Property)

Lemma

If ALG is c-competitive on a request sequence in the partial cost model, then ALG is c-competitive in the full cost model.

$$ALG_f(\sigma) = ALG_p(\sigma) + m$$

$$ALG_p(\sigma) \leq c \cdot OPT_p(\sigma) + \alpha$$

$$ALG_f(\sigma) \leq c \cdot OPT_f(\sigma) + \alpha$$

Preliminaries (Pairwise Property)

- Let $ALG^*(x,j) = 1$ if x precedes the element σ_j in the list on the j^{th} request and 0 otherwise.
- $ALG^*(x,j)$ could be thought of as the charge on element x for impeding access to element σ_j in the partial cost model.

If the request sequence σ has length m and L is the set of elements in the list, then using ALG^* we can write the total cost of ALG,x

$$ALG(\sigma) = \sum_{j=1}^{m} \overbrace{\left(\sum_{x \in L} ALG^*(x,j)\right)}^{Access\ \ cost\ \ for\ \ j^{th} request}$$

$$ALG(\sigma) = \sum_{x \in L} \sum_{j=1}^{m} ALG^*(x,j)$$

Preliminaries (Pairwise Property)

$$ALG(\sigma) = \sum_{x \in L} \sum_{y \in L} \sum_{j | \sigma_j = y} ALG^*(x, j)$$

$$ALG(\sigma) = \sum_{\{x,y\} \subseteq L} \sum_{j | \sigma_j \in \{x,y\}} (ALG^*(x, j) + ALG^*(y, j))$$

For each pair of elements x and y, we compute the cost due to x impeding access to y, and the cost due to y impeding access to x. Because this is in the partial cost model, one of $ALG^*(x, j)$, $ALG^*(y, j)$ is always zero!

Preliminaries (Pairwise Property)

Let $ALG_{xy}(\sigma) = \sum_{j|\sigma_j \in \{x,y\}} (ALG^*(x,j) + ALG^*(y,j))$. Then the cost of ALG simplifies to,

$$ALG(\sigma) = \sum_{\{x,y\} \subseteq L} ALG_{xy}(\sigma)$$

Let $\widetilde{OPT}(\sigma_{xy})$ be the cost of an optimal offline algorithm that serves the request sequence σ_{xy} on two item list.

$$OPT_{xy}(\sigma) \geq \widetilde{OPT}(\sigma_{xy})$$

Preliminaries (Pairwise Property)

Here $ALG(\sigma_{xy})$ be the cost of ALG on the two element list of x and y over the arbitrarily long sequence of requests to x and y from σ. In other words, if we project the list and the request sequence onto the items x and y (i.e. remove everything else from the list and σ), then $ALG(\sigma_{xy})$ is the cost of ALG on the projected list and request sequence.

Pairwise Property:

$$ALG_{xy}(\sigma) = ALG(\sigma_{xy})$$

Preliminaries (Pairwise Property)

Lemma

An algorithm satisfies the pairwise property if and only if for every request sequence σ, when ALG serves σ, the relative order of every two elements x and y in the list is the same as their relative order when ALG serves σ_{xy}.

Preliminaries (Competitiveness)

For an ALG to be c-competitive, it suffice to show that,

$$ALG_{xy}(\sigma) \leq c \cdot \widetilde{OPT}_{xy}(\sigma_{xy})$$

Similarly, for randomized algorithms,

$$E[ALG_{xy}(\sigma)] \leq c \cdot E[\widetilde{OPT}_{xy}(\sigma_{xy})]$$

TIMESTAMP(0)

On a request for item x in the list, TIMESTAMP(0) moves x directly in front of the first item in the list that was accessed at most once since the last request for x. If there is no such item, or x has not been requested before, do nothing.

Pairwise Property of TIMESTAMP(0)

Lemma

After the TIMESTAMP(0) algorithm has served a request sequence σ, element x is before element y if and only if the sequence σ_{xy} terminates in the subsequence xx, xyx or xxy, or if x was before y initially and y was requested at most once.

Pairwise Property of TIMESTAMP(0)

$(\implies)$

- Notice that in the cases where σ_{xy} ends in xx or xyx, y is requested at most once between the final two x's. Therefore, x must be moved in front of y at the end.
- If σ_{xy} ends in xxy, then x is requested twice in a row, which moves it in front of y, and at least twice between the final two y's in the sequence, if there are two. Therefore, the final request to y does not move it in front of x, and so x is before y.
- If y is requested at most once, then it will not be moved in front of x ever, and so if x starts before y, it will also end before y.

Pairwise Property of TIMESTAMP(0)

$(\Longleftarrow)$

- If element x is before element y in the list after TIMESTAMP(0) services σ_{xy}, then one of two things must have happened: either , or
  - y was requested at most once between the final two requests for x, or
  - if there were fewer than 2 requests for x, then x started before y and there were no more than 1 request for y in the sequence.

So we see that x ending before y implies that σ_{xy} ends in xx, xyx, or xxy, or contains at most one y, with x starting before y in the list. This concludes the proof.

COMB

With probability $\frac{4}{5}$ serve the request sequence with BIT, with probability $\frac{1}{5}$ serve the request sequence with TIMESTAMP(0).

COMB

Theorem

COMB is $\frac{8}{5}$-competitive against oblivious adversaries.

COMB

Lemma 3. *Suppose that BIT has served the request sequence xyx, or the sequence yx on a list where initially x preceded y. Then x is in front of y with probability 3/4.*

Proof. We show that after *BIT* has served either sequence, item y is in front of x if and only if the bit of x is 0 and the bit of y is 1: Namely, if the bit of x was set to 1 at the last request to x, then x was moved to the front. Otherwise, x's bit is 0, so the bit was set to 1 at the preceding request to x (in the sequence xyx) and x is front of y at the time of the request to y (which holds by assumption for the sequence yx). Thus, y's bit must have been set to 1 after the request to y to move y in front. The bits of both items are independent, so y is in front of x with probability 1/4. □

COMB

Lemma 4. *In the initial list of two items, let x be in front of y. The following table describes the expected cost for serving the indicated request sequences, where $l \geq 0$ and $k \geq 1$, by the algorithms BIT, TS, and $\overline{OPT}$.*

request sequence	BIT	TS	$\overline{OPT}$
$x^l yy$	$\frac{3}{2}$	2	1
$x^l (yx)^k yy$	$\frac{3}{2}k + 1$	$2k$	$k + 1$
$x^l (yx)^k x$	$\frac{3}{2}k + \frac{1}{4}$	$2k - 1$	k

References

S. Albers. Improved randomized on-line algorithms for the list update problem. In Proc. of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 412-419, 1995.

S. Albers, B. von Stengel and R. Werchner. A combined BIT and TIMESTAMP algorithm for the list update problem. Information Processing Letters, 56:135-139, 1995.

Thank You