Randomized Competitive Algorithms for the List Update Problem

Eric Popelka

About the authors

- **Nick Reingold** - AT&T Bell Labs
- **Jeffery Westbrook** - Yale University
- **Daniel D. Sleator** - Carnegie Mellon
- Published in <u>Algorithmica</u> in 1994

List update problem

- Unsorted linear list
- Cost of accessing an item is equal to its distance from the front
- Can perform transpositions, each with a unit cost of one

Deterministic solution

- Move-to-front (MTF)
  - Move an item to the front each time it is accessed
  - 2-competitive
- MTF is the best that any deterministic online algorithm can do

Can we do better?

- Yes, if we use randomization
- BIT
  - Barely random algorithm
  - “Move-to-front every other access”
  - 1.75-competitive
- COUNTER
  - $\sqrt{3}$-competitive
  - $\sqrt{3} \approx 1.73$

BIT-ACCESS(x)

Require: Each item x has a corresponding bit $b(x)$, initialized uniformly at random

$b(x) = \text{not}(b(x))$
if $b(x) = 1$ **then**
  move x to the front;
end if
process request for item x;

Analysis of BIT

- σ: sequence of m accesses
- **Theorem**: BIT is at most $1.75 \cdot OPT(\sigma) - 3m/4$
- Proof similar to Homework #3

Analysis of BIT

- **Lemma**: After an event, $b(x)\ \forall x$ is equally likely to be 0 or 1, is independent of the bits of other items, and is independent of the positions of items in OPT
- **Proof**:
  - Initial assignment of bit values is chosen uniformly at random
  - Accesses change the values of the bits, but everything is modulo 2
  - Therefore, the bits remain uniformly distributed

Analysis of BIT

- For event i: $\hat{c}_i = c_i + \Phi_i - \Phi_{i-1}$
- An event may be an access or a transposition
- Inversion: (x, y) in OPT, (y, x) in BIT
- Type 1 inversions: $b(x) = 0$
- Type 2 inversions: $b(x) = 1$
- ϕ_1: number of type 1 inversions
- ϕ_2: number of type 2 inversions
- $\Phi = 2\phi_2 + \phi_1$

Analysis of BIT

Case 1: Event i is an access to item x.

- Random variables for the change in potential:
  - **A**: new inversions being created
  - **B**: old inversions being removed
  - **C**: old inversions changing type
- $\Phi_i - \Phi_{i-1} = A + B + C$
- $B + C = -R$, where R is the number of inversions of the form (y, x)

BIT Example

Event i is a request for “Becca”

$\boldsymbol{OPT_{i-1}}$	Mark	Becca	Stephen	Ali
$\boldsymbol{BIT_{i-1}}$	Stephen	Ali	Becca	Mark
$\boldsymbol{b(x)}$	1	0	1	0

Inversions: {(Stephen, Becca), (Stephen, Mark), (Ali, Becca), (Ali, Mark), (Becca, Mark)}

R = # of inversions of the form (y, “Becca”) = 2

$\phi_1 = 3$ $\phi_2 = 2$ $\Phi = 2\phi_2 + \phi_1 = 7$

BIT Example

Event i is a request for “Becca”

OPT_i	Mark	Becca	Stephen	Ali
BIT_i	Stephen	Ali	Becca	Mark
$b(x)$	1	0	0	0

Inversions: {(Stephen, Becca), (Stephen, Mark), (Ali, Becca), (Ali, Mark), (Becca, Mark)}

R = # of inversions of the form (y, “Becca”) = 2

$\phi_1 = 5 \qquad \phi_2 = 0 \qquad \Phi = 2\phi_2 + \phi_1 = 5 \qquad \Delta\Phi = -2$

BIT Example

Event $i+1$ is a request for "Becca"

OPT_{i+1}	Mark	Becca	Stephen	Ali
BIT_{i+1}	Becca	Stephen	Ali	Mark
$b(x)$	1	0	0	0

Inversions: {~~(Stephen, Becca)~~, (Stephen, Mark), ~~(Ali, Becca)~~, (Ali, Mark), (Becca, Mark)}

R = # of inversions of the form (y, "Becca") = 0

$\phi_1 = 3 \qquad \phi_2 = 0 \qquad \Phi = 2\phi_2 + \phi_1 = 3 \qquad \Delta\Phi = -2$

Analysis of BIT

$$\begin{aligned}
\mathsf{E}[\hat{c}_i] &= \mathsf{E}[c_i + \Delta\Phi] \\
&\leq \mathsf{E}[(\mathsf{rank}(x) + R) + (A + B + C)] \\
&= \mathsf{E}[(\mathsf{rank}(x) + R) + (A - R)] \\
&= \mathsf{rank}(x) + \mathsf{E}[A]
\end{aligned}$$

- **A**: new inversions being created
- **B**: old inversions being removed
- **C**: old inversions changing type
- **R**: # of (y, x) inversions

Analysis of BIT

What's the expected value of **A**?

- Both BIT and OPT may move x forward
- Let $z_1, z_2, \ldots, z_{k-1}$ be the items preceding x in OPT
- Inversion created if OPT or BIT (but not both) move x forward past some z_i

Analysis of BIT

New random variable: Z_i

- Z_i measures the change in potential due to each pair (x, z_i)
- If $b(x) = 0$, x moves to the front of BIT
  - Worst case: New inversions (x, z_i) of type $1 + b(z_i)$ created for $1 \leq i \leq \text{rank}'(x) - 1$
- If $b(x) = 1$, x does not move
  - Now $b(x) = 0$
  - Worst case: New inversions (z_i, x) of type 1 created for $\text{rank}'(x) \leq i \leq \text{rank}(x) - 1$

BIT Example

$rank(x)$	1	2	3	4	5	6	7
OPT_{i-1}	Mark	Ali	Kim	Stephen	Becca	Will	David
BIT_{i-1}	Stephen	Will	Kim	David	Becca	Ali	Mark
$b(x)$	0	0	1	1	1	1	1

15 Inversions:

- (Stephen, Kim), (Stephen, Ali), (Stephen, Mark)
- (Will, Kim), (Will, Becca), (Will, Ali), (Will, Mark)
- (Kim, Ali), (Kim, Mark)
- (David, Becca), (David, Ali), (David, Mark)
- (Becca, Ali), (Becca, Mark)
- (Ali, Mark)

BIT Example

Event i is a request for “Becca”

$rank(x)$	1	2	3	4	5	6	7
OPT_i	Mark	Ali	Kim	Becca	Stephen	Will	David
BIT_i	Stephen	Will	Kim	David	Becca	Ali	Mark
$b(x)$	0	0	1	1	0	1	1

16 Inversions:

- (Stephen, Kim), (Stephen, Becca), (Stephen, Ali), (Stephen, Mark)
- (Will, Kim), (Will, Becca), (Will, Ali), (Will, Mark)
- (Kim, Ali), (Kim, Mark)
- (David, Becca), (David, Ali), (David, Mark)
- (Becca, Ali), (Becca, Mark)
- (Ali, Mark)

Analysis of BIT

$$\mathsf{E}[b(x)] = \tfrac{1}{2} \ \forall x$$

$$\begin{aligned}
\mathsf{E}[A] &= \sum_{i=1}^{\mathsf{rank}(x)-1} \mathsf{E}[Z_i] \\
&\le \sum_{i=1}^{\mathsf{rank}'(x)-1} \frac{1}{2}\left(\frac{1}{2}\cdot 2 + \frac{1}{2}\cdot 1\right) + \sum_{i=\mathsf{rank}'(x)}^{\mathsf{rank}(x)-1} \frac{1}{2}\cdot 1 \\
&\le \tfrac{3}{4}(\mathsf{rank}(x) - 1)
\end{aligned}$$

$$\therefore \mathsf{E}[\hat{c}_i] \le 1.75 \cdot OPT_i - \tfrac{3}{4}$$

Analysis of BIT

Case 2: OPT performs a transposition at event i

- OPT will pay a cost of one
- We might have an inversion now
- It might be type 1 or type 2, each with a probability of $\frac{1}{2}$
- A type 1 inversion increases Φ by 1
- A type 2 inversion increases Φ by 2
- $$\begin{aligned} \mathrm{E}[\hat{c}_i] &= \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2 \\ &\leq 1.5 \cdot OPT_i \end{aligned}$$

COUNTER(s, S)

- "Move-to-front on steroids"
- $s \in \mathbb{Z}^+$
- $S \subset \{0, 1, \ldots, s-1\}, S \neq \emptyset$
- Keeps a mod s counter for each item
- Each counter is randomly set to some number $\{0, 1, \ldots, s-1\}$
- BIT is COUNTER(2, {1})

COUNTER-ACCESS(s, S, x)

decrement x's counter mod s;
if $x \in S$ **then**
    move x to the front;
end if
process the request for item x;

Analysis of COUNTER

- $c(x)$ = # of accesses to x before x moves to the front
- p_j = probability that an item will next move to the front after j accesses for $j = 1, 2, \ldots, s$ = $\frac{1}{s}$
- After initialization, Pr$[c(x) = j]$ is $p_j\ \forall x$
- **Claim**: COUNTER(s, S) is $\max\{\sum_{j=1}^{s-1} jp_j, 1 + p_1 \sum_{j=1}^{s-1} jp_j\}$-competitive

Analysis of COUNTER

- Inversion (y, x) is type j if $c(x) = j$
- ϕ_j = # of inversions of type j
- $\Phi = \sum_{j=1}^{s} j \cdot \phi_j$

Analysis of COUNTER

Case 1: Event i is an access to item x.

- x does not move to the front
  - $c(x)$ decreases by one
  - $\Delta\Phi$ = # of inversions of the form (y, x)
- x moves to the front
  - $\Delta\Phi$ = # of inversions of the form (y, x)

Analysis of COUNTER

Let **A** be a random variable giving the number of new inversions created

$$\begin{aligned}\mathsf{E}[\hat{c}_i] &= \mathsf{E}[c_i + \Delta\Phi] \\ &= \mathsf{rank}(x) + \mathsf{E}[A] \\ &\leq \mathsf{rank}(x) + (\mathsf{rank}'(x) - 1)p_1 \sum_{j=1}^{s} jp_j\end{aligned}$$

$$\therefore COUNTER_i \leq (1 + p_1 \sum_{j=1}^{s} jp_j) \cdot OPT_i$$

Analysis of COUNTER

Case 2: OPT performs a transposition at event i

- OPT will pay a cost of one
- We might have an inversion now
- E$[\Delta\Phi] = \sum_{j=1}^{s} j p_j$
- $\therefore COUNTER_i \leq (\sum_{j=1}^{s} j p_j) \cdot OPT_i$

Competitive Ratio of COUNTER

- Pick good values for s and S
  - COUNTER(7, {0, 2, 4}) $\approx$ 1.735-competitive
- Use the RANDOM-RESET algorithm
  - Keep a counter from 1 to s for each item
  - Move to front when an item's counter gets to 1, and reset it to j with some probability π_j
  - Simple Markov chain
  - Can get the best competitive ratio, $\sqrt{3}$

References

References

[1] Fei Li. Online algorithms - introduction, list update, 2010.

[2] Nick Reingold, Jeffery Westbrook, and Daniel D. Sleator. Randomized competitive algorithms for the list update problem. Algorithmica, 11:15–32, 1994. 10.1007/BF01294261.