Domain Wall Memory Management

*Kirk Pruhs*
*University of Pittsburgh*

Coauthors: Neil Olver, Kevin Schewior, Rene Sitters and Leen Stougie.
Memory Technologies

Science 2008
DWM = Domain Wall Memory
= Racetrack Memory

- Nickel-iron alloy wires 1-10 microns (millionth of a meter) in length
- Data held in domain walls between regions of different polarization
- 10 microns length could hold 100 domain walls
- Data is written or read by read/write head on silicon base
- Relevant domain wall shunted to read/write head by applying charge
- Reversing charge moves domain walls back
Theoretical View:
DWM = tape with read/write head(s)
Offline Static Track Management Problem

• Input: sequence of items (virtual memory addresses)
  • e.g. A, B, A, C, A, B, D, A

• Feasible solution: permutation of the n items representing the mapping from virtual memory to physical memory

• Objective: Minimize the total (or average) distance the tape head has to move to access these items in this order
Example:

- Input: A, B, C, A, B, D

- Feasible solution: B A D C

- A ⇒ B cost 1
- B ⇒ C cost 3
- C ⇒ A cost 2
- A ⇒ B cost 1
- B ⇒ D cost 2
- Total cost of this assignment = 1 + 3 + 2 + 1 = 7
Static Track Management
aka Minimum Linear Arrangement (MLA)

• Track management input: A, B, C, A, B, D

• Minimum linear arrangement input = access graph
Known Results for MLA /Static Track Management

- NP-hard
- Poly-time $O(\log^2 n)$ approximation [Hansen 1989]
- Poly-time $O(\log n \log \log n)$ approximation [CHKR 2006, FL 2007]
- No PTAS under complexity theoretic assumptions [AMS2011]
Rest of the Talk

- Hansen’s algorithm and analysis for static track management
- Introduction to dynamic track management
- Issues with extending Hansen’s algorithm to dynamic track management
- How we overcome these issues
Hansen’s Algorithm Design

- **Root Problem:** Determine which items $L$ are on the left and which items $R$ should go on the right with the objective of minimizing the number of consecutive accesses that cross the boundary.

- Recurse on $L$
- Recurse on $R$
Dynamic Track Management

- Root Problem: Determine which items L are on the left and which items R should go on the right with the objective of minimize number of access that cross the boundary

- Question: A $c$-approximation for Root Problem yields $\frac{C}{c}$-approximation algorithm for MLA?
Hansen’s Algorithm Analysis

• Question: A c-approximation for Root Problem yields ?-approximation algorithm for MLA?

• Issue: Optimal may put very different items on the left and right.
Hansen’s Algorithm Analysis

• Theorem: A c-approximation for the Root Problem yields a $O(c \times \log n)$-approximation for MLA.

• Proof:

  – Lemma: Each instance I of MLA has the following
    • **Subadditivity Property:** for all partitions L and R of I, $\text{Opt}(I) \geq \text{Opt}(L) + \text{Opt}(R)$

  – Lemma: Subadditivity and c-approximation for the Root Problem imply approximation $c^*$ (height of recursion)
Dynamic Track Management

- Everything the same as static track management except that the possible operations are:
  - Move head one position left or right
  - Swap current items with the item to the left or the right

- Comment: Potentially beneficial if working set changes
Issues with Applying Hansen’s Algorithm for Dynamic Track Management

1. Straight-forward recursion can’t work
   - Items can enter and exit in recursive subproblems

2. Determining a solvable base problem
   - Analogous to finding balanced cut of the access graph in MLA

3. Finding something analogous to the subadditivity property in the MLA analysis
   - Not clear what subadditivity should mean
   - For most obvious candidates, not clear it holds
Dealing With Lack of Subadditivity

• Identify subcollection of cost that are subadditive

• Base algorithm is good approximation for subadditive costs

• Subadditive costs are a large portion of the total cost
• Base problem = per time balanced cuts on time indexed graph with
  
  – Vertices for each (item, time) pair
  
  – Consistency Edges: An edge between two vertices with the same item with consecutive times
  
  – Request edges: An edge of the form ((A, t), (B, t)) if item A as accessed at time t-1 and item B was accessed at time t
  
  – Example of time indexed graph for input sequence C, B, A, C, B shown
Illustration of Laminar Family of Balanced Cuts

• Parent in orange

• Here parent has 6 children in laminar family

• Yellow edges = not counted in subadditive costs
Dynamic Min Balanced Cut

- Obvious open problem: Find a poly-log competitive online algorithm for dynamic track management

- We have an $\Omega(\log n)$ general lower bound on competitiveness of any online algorithm
Core Online Problem

• Setting
  – You have a collection of items
  – You must maintain a partition of the items into two halves

• Input that arrives over time: Pair \((x_t, y_t)\) arrive at time \(t\)

• Costs: 1 if items \(x_t\) and \(y_t\) are in different halves

• Algorithmic decision: Can swap items between halves at a cost of 1 per swap

• Objective: Minimize total cost
Thanks for Listening!
Any Questions?

"That's all Folks!"