Deterministic Rendezvous Problem

COMP 5703 Seminar

Adam Bennett

Deterministic Rendezvous Problem

- Variant of the more general *Rendezvous Problem*
- Rendezvous Problem:
  - problem in game theory where two robots in a graph must meet at the same node
- In the Deterministic Rendezvous Problem, instructions given to the robots are deterministic

Deterministic Rendezvous Problem

- Both robots are given the same sequence of instructions
  - instruction sequence may refer to the robot's label
- Applications include:
  - Search and Rescue
  - Communications
  - Networks / Operating Systems

Deterministic Rendezvous Problem

- Consider two robots in an unknown, finite, connected, undirected graph
- Each robot knows:
  - *T*, the number of time steps since it has been activated
  - *d*, the degree of the node it is at
  - *L*, the value of a distinct label that it was assigned
- The robots must be provided a set of deterministic instructions that will allow them to find each other

Deterministic Rendezvous Problem

- Many of the parameters of the problem are set by the *adversary*:
  - The size and layout of the graph
  - The initial positions of the two robots
  - When each robot is activated
  - The value of the unique label assigned to each robot

Solutions

- A variety of algorithms exist that solve the problem
- Dessmark et al. [2006] presented an algorithm that solves the problem in time proportional to:

$$O\left(n^5\sqrt{\tau l}+n^{10}l\right)$$

where:

  - *n* is the size of the graph
  - *l* is the length of the shortest labels
  - τ is the difference in activation times

Solutions

- Kowalski and Malinowski [2008] presented an algorithm that solves the problem in time proportional to:

$$O(n^{15}+l^3)$$

  - Doesn't depend on τ, which may be arbitrarily large
  - Uses *backtracking*:
    - the robot remembers the sequence of edges it has traversed

Solutions

- Ta-Shma and Zwick [2014] presented an algorithm that solves the problem in time proportional to:

$$O(n^5 l)$$

  - Doesn't use backtracking
  - Uses *Universal Traversal Sequences*
  - This is the algorithm that will be explored in detail for the remainder of this presentation

Universal Traversal Sequences

- Traversal Sequence:
  - a set of instructions that define a traversal of every node in a particular *d*-regular graph
- Each step in the sequence specifies which neighbour of the current node to visit next
  - Steps are relative to the current node
  - For example, if the current node is v_j, and v_j has d neighbours, then the traversal sequence will specify the next node to visit, v_{j+1}, as the i^{th} neighbour of v_j, where $1 \leq i \leq d$

Universal Traversal Sequences

- Universal Traversal Sequence (UTS):
  - traversal sequence that covers any *n*-vertex graph no matter which node is the starting node
- What if the graph is not *d*-regular?
  - Let *d* be the greatest degree of any node. For any node with degree less than *d*, add self-loops until that node's degree is equal to *d*.

Universal Traversal Sequences

- Aleliunas et al. [1979] presented a proof stating that for any *d*-regular *n*-vertex graph, there exists a UTS for that graph with $\Theta(n^5)$ steps
- The remainder of this presentation assumes that a UTS for a *d*-regular *n*-vertex graph is known

Ta-Shma and Zwick's Solution

- Basic Idea:
  - Robots are guaranteed to meet if one traverses the entire graph while the other remains idle (*rests*)
- Size of the graph is unknown, so the robots use UTSs for increasing values of *n* while periodically resting
- Whether the robot rests before or after completing each traversal sequence depends on its label

Ta-Shma and Zwick's Solution

- For example:

  One robot runs the sequence

  $$U_1 0^{u_1} U_2 0^{u_2} U_4 0^{u_4} U_8 0^{u_8} \ldots U_{2^i} 0^{u_{2^i}} \ldots$$

  while the other robot runs the sequence

  $$0^{u_1} U_1 0^{u_2} U_2 0^{u_4} U_4 0^{u_8} U_8 \ldots 0^{u_{2^i}} U_{2^i} \ldots$$

  where U_i is a UTS for a graph of size *i*, u_i is the number of steps in that UTS, and 0^k represents k steps where the robot rests

- Only works if the robots are activated at the same time

Ta-Shma and Zwick's Solution

- What if the robots are activated at different times?
  - Add idle periods of length $u_i - 1$ between each step
- Example:
  - One of the robots will run the sequence

$$\sigma_1 0^{u_1-1} \sigma_2 0^{u_1-1} \sigma_3 0^{u_1-1} \ldots \sigma_{u_1} 0^{u_1-1} 0^{2u_1^2} \pi_1 0^{u_2-1} \pi_2 0^{u_2-1} \pi_3 0^{u_2-1} \ldots \pi_{u_2} 0^{u_2} 0^{2u_2^2} \ldots$$

where $\sigma = U_1$ and $\pi = U_2$

Ta-Shma and Zwick's Solution

Some notation:

- Let σ^b =

  $0^{|\sigma|}$ if $b = 0$

  σ if $b = 1$

- Let $\overline{L} = 1 - L$
- Let $\sigma^{m_1 \ldots m_k} = \sigma^{m_1} \sigma^{m_2} \ldots \sigma^{m_k}$
- Let $D_k(\sigma_1 \ldots \sigma_m) = \sigma_1 0^k \sigma_2 0^k \ldots \sigma_m 0^k$

Ta-Shma and Zwick's Solution

- For simplicity's sake, assume that 0 and 1 are the labels chosen for the two robots
- The sequence of instructions that a robot runs is:

$$D_{u_{1-1}}\left(\left(U_1 U_1\right)^{L\bar{L}}\right) D_{u_{2-1}}\left(\left(U_2 U_2\right)^{L\bar{L}}\right) \ldots D_{u_{2^k}-1}\left(\left(U_{2^k} U_{2^k}\right)^{L\bar{L}}\right) \ldots$$

Ta-Shma and Zwick's Solution

$$D_{u_{1-1}}\left(\left(U_1 U_1\right)^{L\bar{L}}\right) D_{u_{2-1}}\left(\left(U_2 U_2\right)^{L\bar{L}}\right) \ldots D_{u_{2^k}-1}\left(\left(U_{2^k} U_{2^k}\right)^{L\bar{L}}\right) \ldots$$

- Consider the following sub-sequence :

$$D_{u_{i-1}}\left(\left(U_i U_i\right)^{L\bar{L}}\right)$$

  - This sub-sequence is known as a *block*
- A block can be rewritten like this:

$$D_{u_{i-1}}\left(\left(U_i U_i\right)^{L}\left(U_i U_i\right)^{\bar{L}}\right)$$

Ta-Shma and Zwick's Solution

$$D_{u_{i-1}}\left(\left(U_i U_i\right)^L \left(U_i U_i\right)^{\bar{L}}\right)$$

- If we let σ=U_i and m=u_i, then the block can be further simplified to:

$$\left(\sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1} \sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1}\right)^L \left(\sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1} \sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1}\right)^{\bar{L}}$$

- A block consists of two parts:

$$\left(\sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1} \sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1}\right)^L$$

$$\left(\sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1} \sigma_1 0^{m-1} \ldots \sigma_m 0^{m-1}\right)^{\bar{L}}$$

- These two parts are known as *chunks*

Ta-Shma and Zwick's Solution

- If the robot's label is 0, then the block is equal to
  $0^{2m^2}\sigma_1 0^{m-1}\ldots\sigma_m 0^{m-1}\sigma_1 0^{m-1}\ldots\sigma_m 0^{m-1}$
- If the robot's label is 1, then the block is equal to
  $\sigma_1 0^{m-1}\ldots\sigma_m 0^{m-1}\sigma_1 0^{m-1}\ldots\sigma_m 0^{m-1} 0^{2m^2}$

Correctness Proof

- Let b_i = the size of the block $D_{u_{i-1}}\left(\left(U_i U_i\right)^{L\bar{L}}\right)$
- Let w_i = the size of $D_{u_{i-1}}\left(\left(U_i\right)^{L}\right)$
- A chunk has size $2m^2$ (where $m=u_n$), and w_n is equal in size to half a chunk, therefore $w_n = u_n^2$

Correctness Proof

- From Aleliunas et al. [1979], it is known that $u_n = \Theta(n^c)$, so $w_n = \Theta(n^{2c})$
- $b_n = 4w_n = \Theta(n^{2c})$
- Assume $4u_n \leq u_{2n}$, for every $n = 2^i$

  (since $4u_n = \Theta(n^c)$ and $u_{2n} = u_n^2 = \Theta(n^{2c})$, for $n = 2^i$)
- $16b_n \leq b_{2n}$, for every $n = 2^i$

Correctness Proof

- From $16b_n \leq b_{2n}$, we get for all $j \geq 1$:

$$\sum_{i=0}^{j} b_{2i} < \frac{1}{15} b_{2j}$$

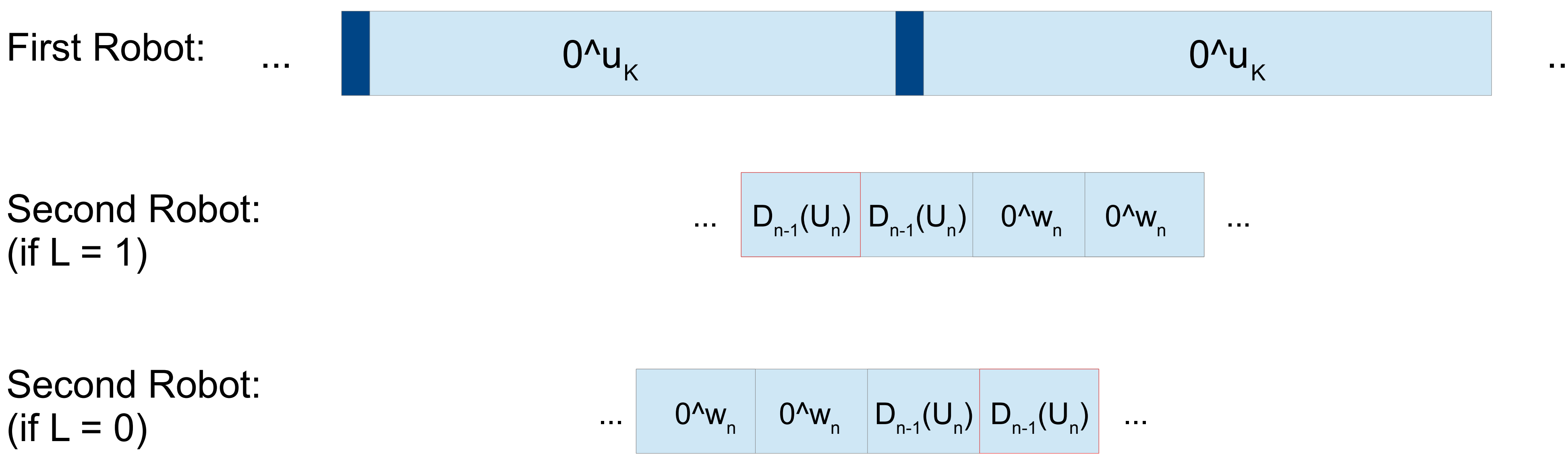
- If one robot is in block *i* when the other is activated, the former robot will be less than 1/4 through the block after *i* when the latter robot begins the block after *i*

Correctness Proof

- Let *K* be the index of the block that the first robot to be activated is in when the second robot reaches the block with index *n*
  - By index, we mean the value of i in the current block $D_{u_{i-1}}\left(\left(U_i U_i\right)^{L\bar{L}}\right)$, which will always be a power of two
- There are two cases to consider: the case where $u_K \geq b_n$, and the case where $u_K < b_n$

Correctness Proof

- Case 1: $u_K \geq b_n$
  - The first robot rests for the amount of time it takes the second robot to complete an entire traversal



Correctness Proof

- Case 2: $u_K < b_n$
  - $u_K = \Theta(K^c)$ and $b_n = \Theta(n^{2c})$, so $K < O(n^2)$
  - The second robot finishes block K and begins block $2K$ after $O(K^{c2}) = O(n^{4c})$ steps
    - The first robot must still be on the first quarter of block $2K$ at this time.

First Robot (if first robot's L = 0)

...	0^w_{2K}	0^w_{2K}	$D_{2K-1}(U_{2K})$	$D_{2K-1}(U_{2K})$	...

First Robot (if first robot's L = 1)

...	$D_{2K-1}(U_{2K})$	$D_{2K-1}(U_{2K})$	0^w_{2K}	0^w_{2K}	...

Second Robot (if second robot's L = 1)

...	$D_{2K-1}(U_{2K})$	$D_{2K-1}(U_{2K})$	0^w_{2K}	0^w_{2K}	...

Second Robot (if second robot's L = 0)

...	0^w_{2K}	0^w_{2K}	$D_{2K-1}(U_{2K})$	$D_{2K-1}(U_{2K})$	...

Time Complexity

- Case 1: $u_K \geq b_n$
  - The robots meet by the time the second robot finishes block *n*
  - Block *n* is the ln(n)th block run by the second robot
  - Each block has length $b_n = \Theta(n^{2c})$
  - Robots meet $\ln(n)\Theta(n^{2c}) \leq O(n*n^{2c}) \leq O(n^{4c})$ steps after the second robot is activated

Time Complexity

- Case 2: $u_K < b_n$
  - Recall from earlier:
    - $u_K = \Theta(K^c)$ and $b_n = \Theta(n^{2c})$, so $K < O(n^2)$
    - The second robot finishes block K and begins block $2K$ after $O(K^{2c}) = O(n^{4c})$ steps
  - Robots meet in first half of block $2K$
  - So they will meet after $O(n^{4c}) + b_{2K} / 2$ steps
    - $b_{2K} = \Theta(2K^{2c}) = O(n^{4c})$
    - $O(n^{4c}) + b_K / 2 = 2O(n^{4c}) = O(n^{4c})$
      - The robots meet $O(n^{4c})$ steps after the second robot is activated

Ta-Shma and Zwick's Solution

- Both robots have been proven to meet at most O(n^{4c}) steps after the second robot has been activated
- Ta-Shma and Zwick also show:
  - how to reduce the time complexity to O(n^c) steps after the second robot has been activated
  - how to deal with arbitrary labels (ie: labels where the robots don't have 0 and 1 as labels)
    - The time complexity when dealing with arbitrary labels is O(ln^c) steps after the second robot has been activated, where l is the length of the shortest label
  - these are beyond the scope of this presentation

References

- R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. 1979. Random walks, universal traversal sequences, and the complexity of maze problems. In FOCS. 218-223
- Alpern, Steve, Shmuel Gal and MyiLibrary, The Theory of Search Games and Rendezvous (Kluwer Academic Publishers, 2003) vol 55
- A. Dessmark, P. Fraingnaud, D. Kowalski, and A. Pelc. 2006. Deterministic rendezvous in graphs. *Algorithmica* 46, 1 (2006), 69-96
- D. R. Kowalski and A. Malinowski. 2008. How to meet in anonymous network. *Theoretical Computer Science* 399, 1-2 (2008), 141-156
- Amnon Ta-Shma and Uri Zwick. 2014. Deterministic rendezvous, treasure hunts, and strongly universal traversal sequences, universal exploration sequences. ACN Trans. Algor. 10, 3, Article 12 (April 2014), 15 pages.