Low Memory Graph Traversal

Samson Bassett
Master of Science Defense
Simon Fraser University

August 23, 2010

Outline

Introduction

- The graph traversal problem requires an algorithm that starts on an arbitrary vertex, and moves along edges to eventually visit every vertex of the graph.

Introduction

- The graph traversal problem requires an algorithm that starts on an arbitrary vertex, and moves along edges to eventually visit every vertex of the graph.
- Trivial memory lower bound for graph traversal $\Omega(\log n)$. We wish to achieve this lower bound whenever possible.

Introduction

- The graph traversal problem requires an algorithm that starts on an arbitrary vertex, and moves along edges to eventually visit every vertex of the graph.
- Trivial memory lower bound for graph traversal $\Omega(\log n)$. We wish to achieve this lower bound whenever possible.
- Model of Computation is a Finite Automaton that only knows local information of the graph

Introduction

- The graph traversal problem requires an algorithm that starts on an arbitrary vertex, and moves along edges to eventually visit every vertex of the graph.
- Trivial memory lower bound for graph traversal $\Omega(\log n)$. We wish to achieve this lower bound whenever possible.
- Model of Computation is a Finite Automaton that only knows local information of the graph
- There are two main classes of graphs, anonymous graphs and labelled graphs.

Anonymous Graphs

- A graph that does not have a unique label for every vertex is called an anonymous graph.

Anonymous Graphs

- A graph that does not have a unique label for every vertex is called an anonymous graph.
- In anonymous graphs, we allow edges at an incident vertex to be distinguishable from one another, called a *local orientation* of the graph, represented by port numbers

Motivation

- Theorem
  (Fraigniaud et. al., 2005) For any finite automaton with K states, and any $d \geq 3$, there exists an anonymous planar graph G of maximum degree d with at most $K+1$ vertices that the finite automaton cannot traverse.

Motivation

- Theorem
  (Fraigniaud et. al., 2005) For any finite automaton with K states, and any $d \geq 3$, there exists an anonymous planar graph G of maximum degree d with at most $K + 1$ vertices that the finite automaton cannot traverse.
  - What if the local orientation is not arbitrary?

Definition

- Periodic graph traversal requires that an algorithm visits every vertex infinitely many times in a periodic manner

Definition

- Periodic graph traversal requires that an algorithm visits every vertex infinitely many times in a periodic manner
- The period of a traversal on a graph with n vertices is the maximum number of edge traversals performed between two consecutive visits of a generic vertex, denoted by $\pi(n)$

Previous Results

- Theorem
  (Dobrev et. al., 2005) If G is an anonymous graph on n vertices, there exists a local orientation and a corresponding robot (with 1 state) that will periodically traverse G with period $\pi(n) \leq 10n$
- Theorem
  (Ilcinkas, 2008) If G is an anonymous graph on n vertices, there is a local orientation and a robot (with 3 states) that will periodically traverse G with period $\pi(n) \leq 4n - 2$
- Theorem
  (Gąsieniec et. al., 2008) If G is an anonymous graph on n there is a local orientation of G and a robot (with 11 states) that will periodically traverse G, with period $\pi(n) \leq 3.75n - 2$

Lower Bound

- If G is anonymous graph on n vertices, the lower bound of the period of any periodic traversal is $\pi(n) \geq 2n - 2$
- In particular, trees achieve this bound

Lower Bound

- If G is anonymous graph on n vertices, the lower bound of the period of any periodic traversal is $\pi(n) \geq 2n - 2$
- In particular, trees achieve this bound
- We wish to find a class of graphs (other than trees) that achieves the lower bound of $2n - 2$

Classes of Graphs

- Let G be a P_3-free graph with $\delta(G) \geq 2$

Classes of Graphs

- Let G be a P_3-free graph with $\delta(G) \geq 2$

- Lemma
  Let G be a graph with n vertices such that G is P_3-free. Then either G is 2-connected, or $\Delta(G) = n - 1$.

Robot

- The robot (finite automaton) has only one state, and it traverses the graph simply by following increasing port numbers

Robot

- The robot (finite automaton) has only one state, and it traverses the graph simply by following increasing port numbers
- The robot will start on any vertex v
- Initially, the robot will leave v on the edge with port $d(v)$
- At each subsequent vertex v, if the robot enters v on port $i < d(v)$, it will leave on the edge with port $i + 1$
- Similarly, if the robot enters v on port $d(v)$, it backtracks along the same edge (with port $d(v)$)

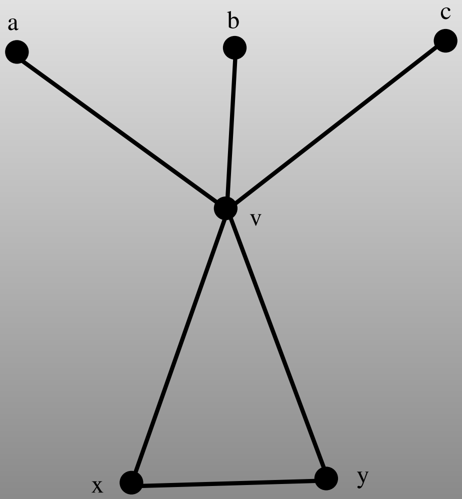
$\Delta(G) = n - 1$

- Let v be a vertex of degree $n - 1$
- Since $\delta(G) \geq 2$, v has two neighbours x and y that are adjacent

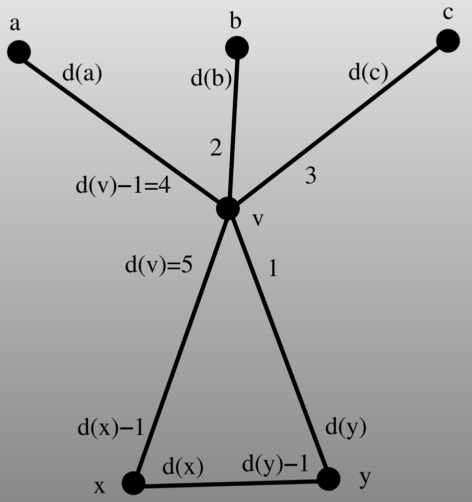
$\Delta(G) = n - 1$

- Let v be a vertex of degree $n - 1$
- Since $\delta(G) \geq 2$, v has two neighbours x and y that are adjacent
- The assignment of port numbers handles all vertices the same except for these three, as shown in the following example

$\Delta(G) = n - 1$



$\Delta(G) = n - 1$



2-connected graphs

- Theorem
  (Fleishner, 1974) If G is 2-connected, then G^2 has a Hamiltonian cycle
  - Let H be a Hamiltonian cycle of G^2

2-connected graphs

- Theorem
  (Fleishner, 1974) If G is 2-connected, then G^2 has a Hamiltonian cycle
  - Let H be a Hamiltonian cycle of G^2
  - The main idea is to create a closed walk H^* based on H that visits every vertex of G using only edges of G
  - Then arrange port numbers so the robot will follow H^*
  - For our purposes, we will work with the *symmetric orientation* of G and G^2, denoted $\vec{G}$ and $\vec{G^2}$, respectively

2-connected graphs

- For every virtual edge of H, we assign a real path of length two, called a relay path

2-connected graphs

- For every virtual edge of H, we assign a real path of length two, called a relay path
- If P is a maximal virtual path in H containing more than one vertex, such that all virtual arcs of P share a common relay vertex w, then the subgraph W of $\vec{G}^2$ consisting of P, and the relay path of every arc of P, is called a *wedge*
- The *size* of W, denoted $s(W)$, is the number of vertices of P

2-connected graphs

- We prove several lemmas related to wedges, the following describes how two wedges can interact

- Lemma
Suppose W_1 and W_2 are distinct wedges. Then

(*i*) *The wedges W_1 and W_2 have no virtual arcs in common.*
(*ii*) *The wedges W_1 and W_2 have no ribs in common.*

2-connected graphs

- Given a vertex v, let (x, v) and (v, y) be the two arcs of H incident with v.

2-connected graphs

- Given a vertex v, let (x, v) and (v, y) be the two arcs of H incident with v.
- The incoming arc of v is either (x, v) if it is real, or the real arc (w, v), where w is the relay vertex of (x, v)

2-connected graphs

- Given a vertex v, let (x, v) and (v, y) be the two arcs of H incident with v.
- The incoming arc of v is either (x, v) if it is real, or the real arc (w, v), where w is the relay vertex of (x, v)
- Similarly, the outgoing arc of v is either (v, y) if it is real, or the real arc (v, w), where w is the relay vertex of (v, y).

2-connected graphs

- Given a vertex v, let (x, v) and (v, y) be the two arcs of H incident with v.
- The incoming arc of v is either (x, v) if it is real, or the real arc (w, v), where w is the relay vertex of (x, v)
- Similarly, the outgoing arc of v is either (v, y) if it is real, or the real arc (v, w), where w is the relay vertex of (v, y).
- If e is the incoming arc of v and e^{-1} is the outgoing arc of v, we say that e is a backtrack arc of v, (in which case e^{-1} is also a backtrack arc of v

2-connected graphs

- The *list of ordered wedges* $W_1, W_2, \ldots, W_k$ *at* v is the ordered list of all wedges with the common relay vertex v such that for W_i and W_j with $i < j$, the virtual path of W_i occurs before the virtual path of W_j when following H starting from v

2-connected graphs

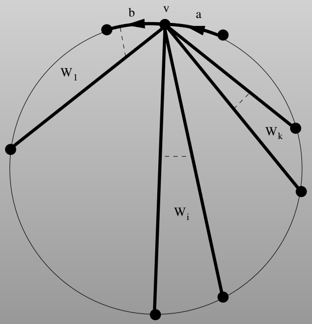
- The *list of ordered wedges* $W_1, W_2, \ldots, W_k$ *at* v is the ordered list of all wedges with the common relay vertex v such that for W_i and W_j with $i < j$, the virtual path of W_i occurs before the virtual path of W_j when following H starting from v
- There are a few technical lemmas that determine how this list of wedges at v interact with the incoming and outgoing arcs of v
- Using these we have five cases (subroutines) for the Port-Numbering procedure

2-connected graphs

- The *list of ordered wedges* $W_1, W_2, \ldots, W_k$ *at* v is the ordered list of all wedges with the common relay vertex v such that for W_i and W_j with $i < j$, the virtual path of W_i occurs before the virtual path of W_j when following H starting from v
- There are a few technical lemmas that determine how this list of wedges at v interact with the incoming and outgoing arcs of v
- Using these we have five cases (subroutines) for the Port-Numbering procedure
- We will fix v, and say a is the incoming arc of v, and b is the outgoing arc of v

2-connected graphs

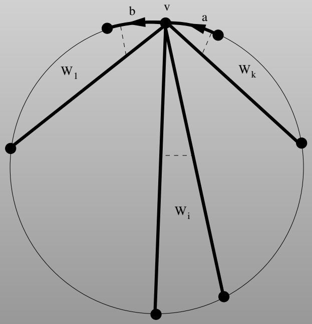
Assign $d(v) - s(W_1)$ to a^{-1}
Assign $[d(v) - s(W_1) + 1, d(v)]$ to W_1



The arc a is an isolated arc of v, and b is not.

2-connected graphs

Assign $[d(v) - s(W_1) + 1, d(v)]$ to W_1
Assign $[d(v) - s(W_1) - s(W_k) + 1, d(v) - s(W_1)]$ to W_k



Neither a nor b are isolated arcs of v

Periodic Traversal Result

- Theorem

  Let G be a P_3-free graph, with $\delta(G) \geq 2$. Then there exists a local orientation and a corresponding robot that will perform a periodic traversal of G with period $\pi(n) \leq 2n - 2$

Future Work

- Drop the condition $\delta(G) \geq 2$ by adding a constant number of states to the robot
- Create a procedure to assign ports for P_3-free graphs with $\delta(G) = 1$

Future Work

- Drop the condition $\delta(G) \geq 2$ by adding a constant number of states to the robot
- Create a procedure to assign ports for P_3-free graphs with $\delta(G) = 1$
- We conjecture that, using similar techniques as shown in this thesis, it is possible to create a local orientation (given some robot with a constant number of states) for 2-connected graphs that are not P_3-free

References

- R. Fleischer. The square of every two-connected graph is Hamiltonian. *Journal of Combinatorial Theory, Series B*, **16**(1) 29–34 (1974)
- P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc, D. Peleg. Graph exploration by a finite automaton. *Theoretical Computer Science*, **345** 331–344 (2005)
- L. Gąsieniec, R. Klasing, R. Martin, A. Navarra, X. Zhang. Fast periodic graph exploration with constant memory. *J Computer and System Science*, **74**(5) 808–822 (2008)