A Scrutiny of Frederickson's Distributed Breadth-First Search Algorithm

Victor van der Veen

Computer Science
Vrije Universiteit Amsterdam

June 26, 2008

Outline

Outline
Introduction
Definitions
Background
A Simple BFS Tree Algorithm
Algorithm
Frederickson's BFS Tree Algorithm
Algorithm
Old-parent problem
Supporting non-FIFO channels
Optimal ℓ value
Conclusions
Drawbacks
Improvements
Questions

Definitions

Definition

A tree is an undirected, connected, acyclic graph.

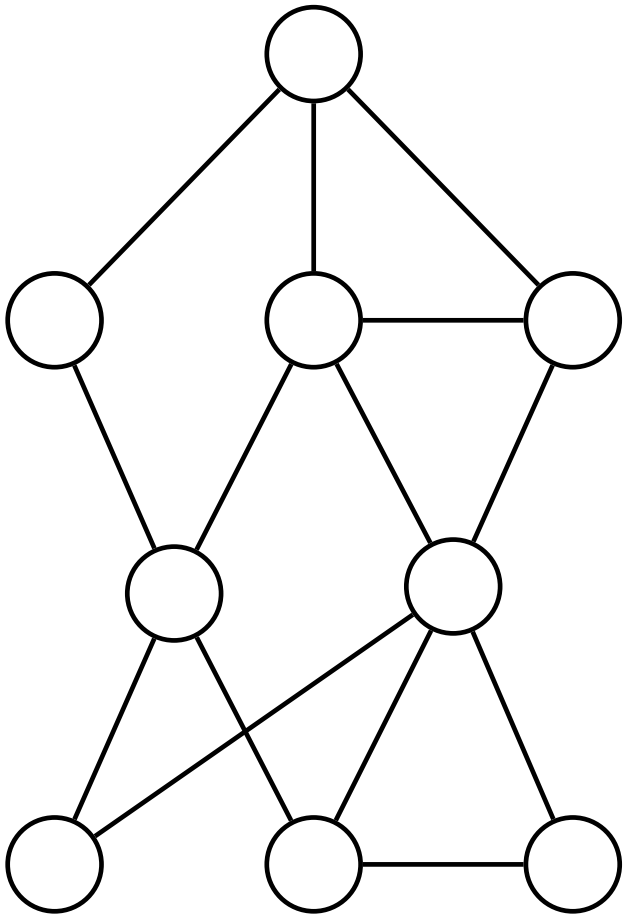
Definition

Every connected graph $G = (V, E)$ contains a spanning tree; that is, a set $E' \subseteq E$ can be chosen, such that (V, E') is a tree.

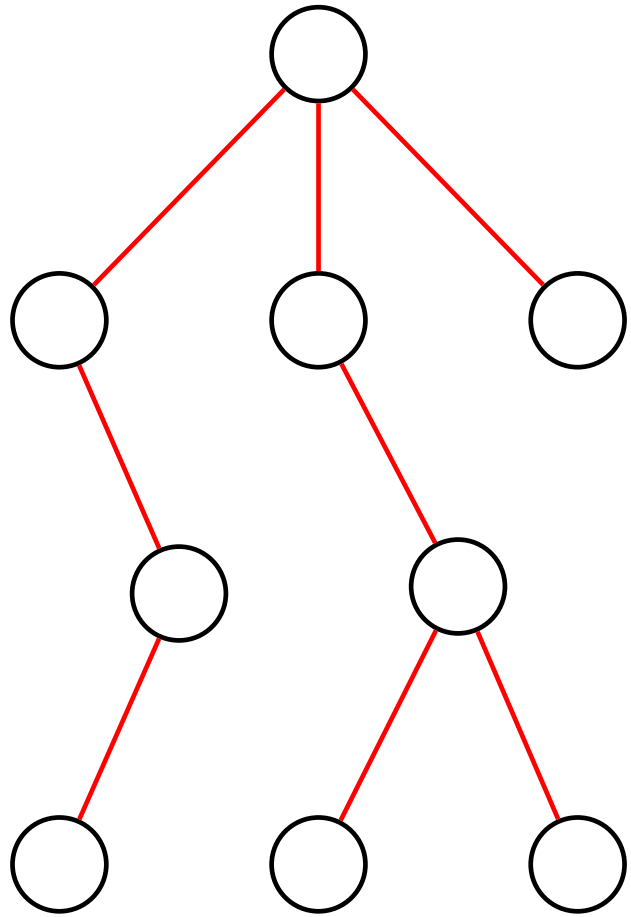
Definition

A spanning tree T, starting from u, of network G is a breadth-first search tree if, for each node, the tree path to u is a minimum-hop path in G.

Definitions



Graph G



Breadth-First Search Tree of G

Background

Literature

- In 1985, Greg N. Frederickson outlined a Breadth-First Search Tree Algorithm in his paper A Single Source Shortest Path Algorithm for a Planar Distributed Network.

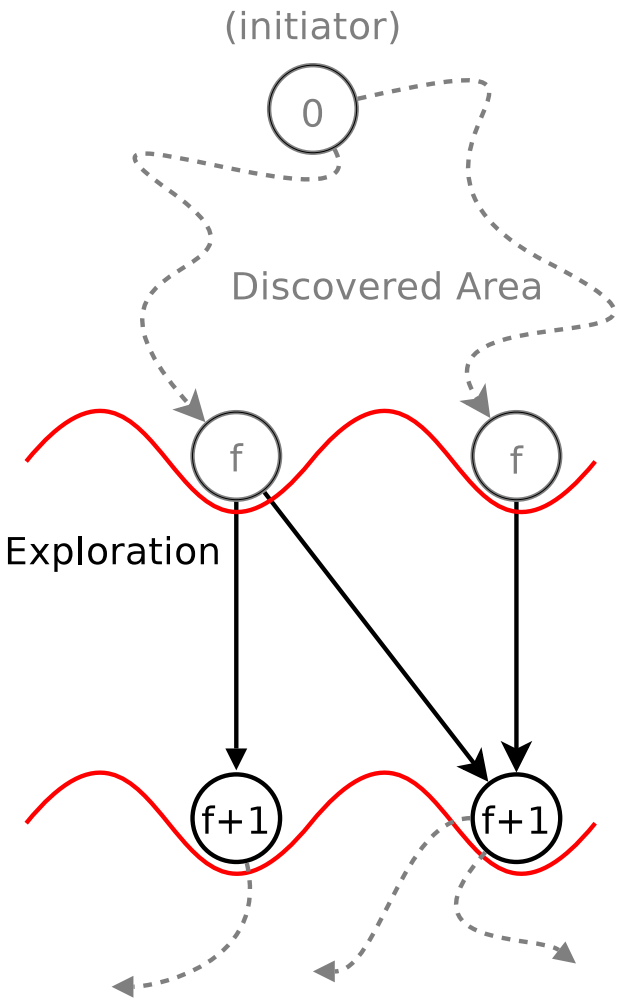
- In 2000, Gerard Tel summarized the algorithm in his book Introduction to Distributed Algorithms.

A Simple BFS Tree Algorithm

Idea

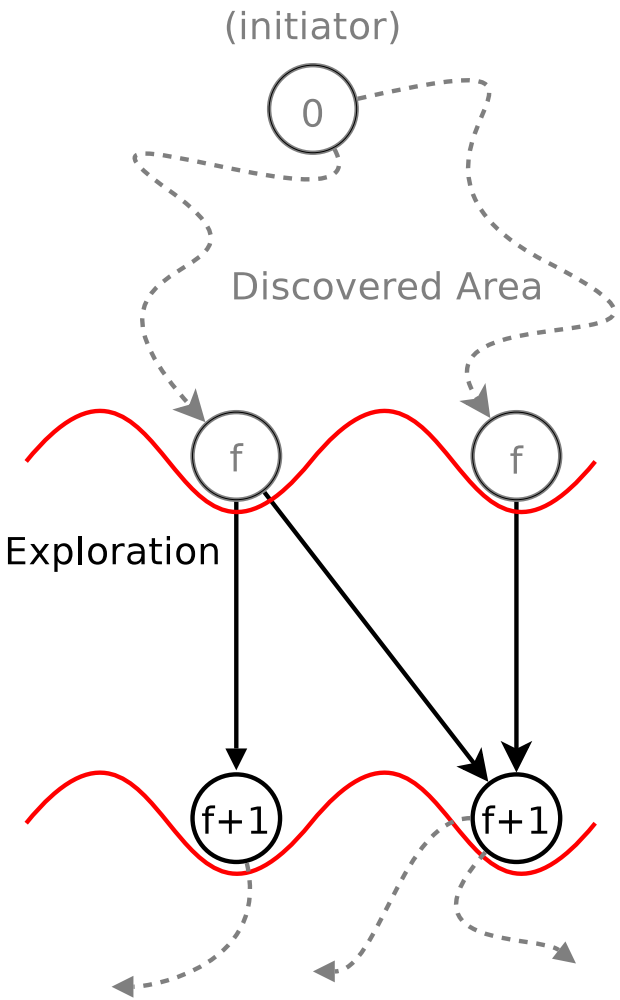
Construct the BFS tree level by level.

A Simple BFS Tree Algorithm



- Initiator sends **forward** to "lowest" nodes.
- Those nodes send **explore** to undiscovered nodes.
- New nodes in tree reply with **reverse**.
- Results go back to initiator.

A Simple BFS Tree Algorithm

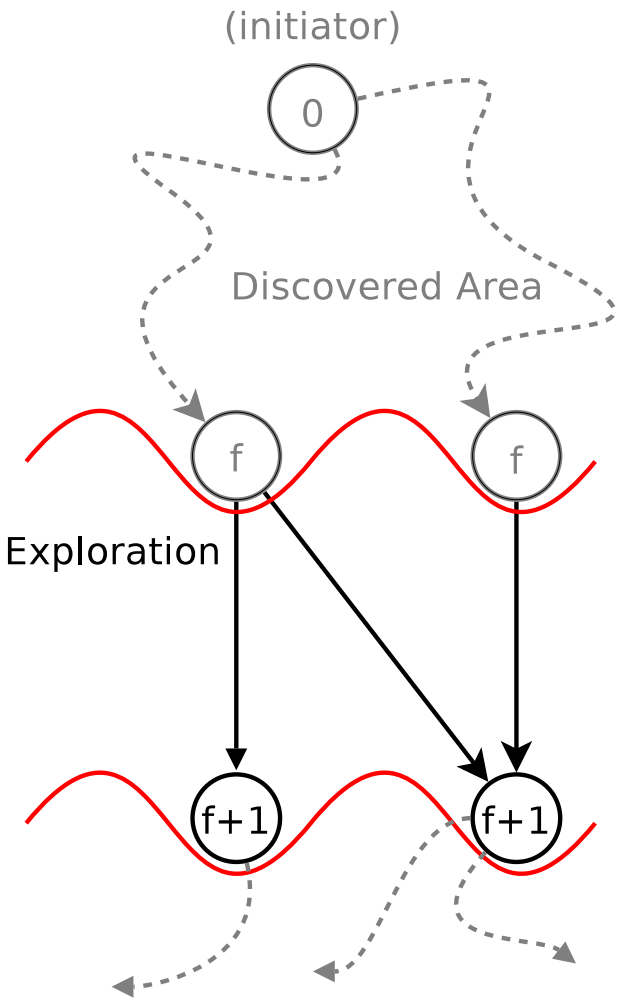


- Initiator sends **forward** to "lowest" nodes.
- Those nodes send **explore** to undiscovered nodes.
- New nodes in tree reply with **reverse**.
- Results go back to initiator.

Difficulties

- What if a node is already discovered?

A Simple BFS Tree Algorithm



- Initiator sends **forward** to "lowest" nodes.
- Those nodes send **explore** to undiscovered nodes.
- New nodes in tree reply with **reverse**.
- Results go back to initiator.

Difficulties

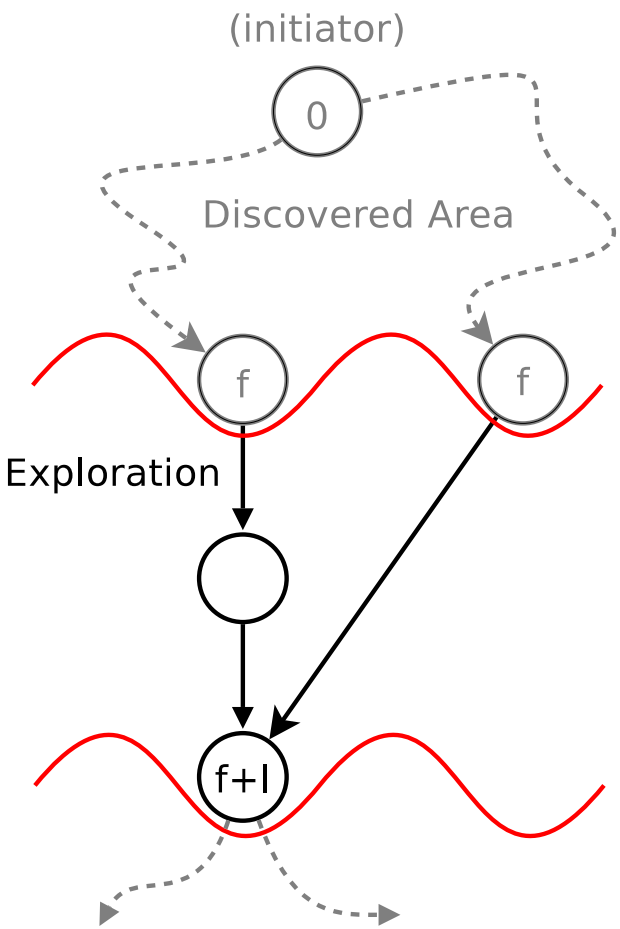
- What if a node is already discovered?
- Simply send "I will not be your child" as **reverse**.

Frederickson's BFS Tree Algorithm

Idea

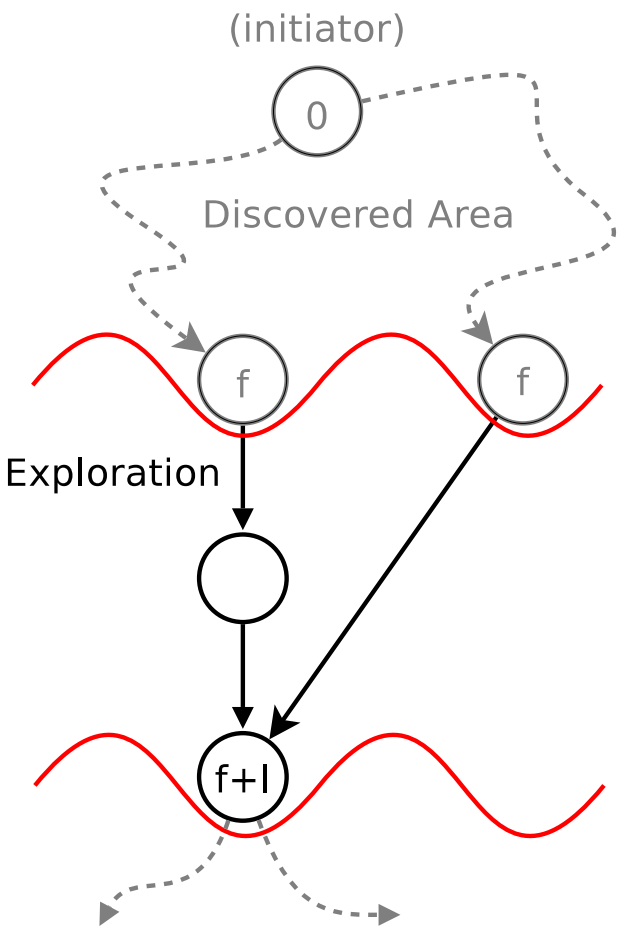
Construct ℓ levels in one round ($\ell > 1$).

Frederickson's BFS Tree Algorithm



- New nodes also **explore**
- When ℓ levels are explored, **reverse** is started

Frederickson's BFS Tree Algorithm



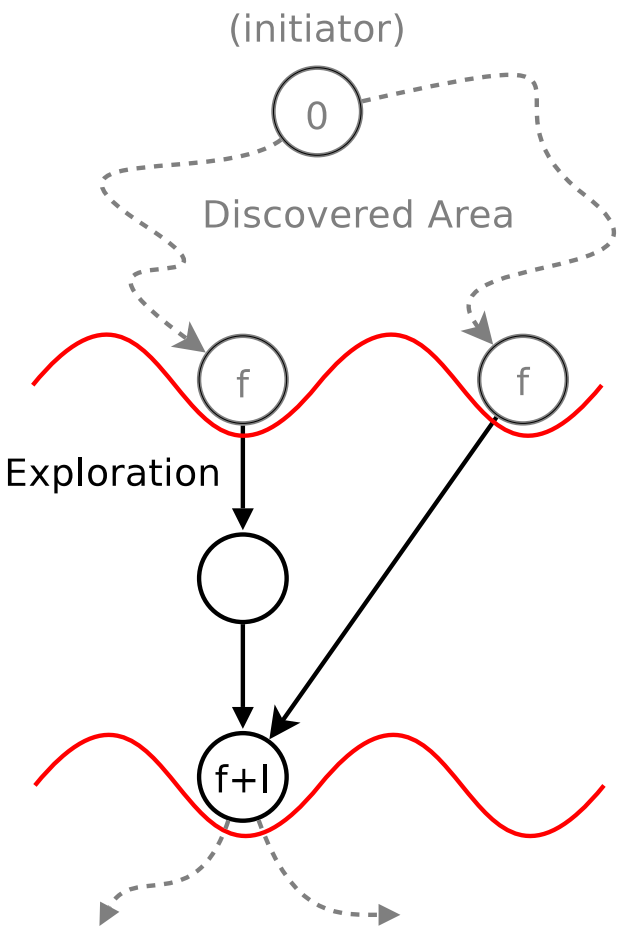
- New nodes also **explore**
- When ℓ levels are explored, **reverse** is started

Difficulties

Old-parent problem

- What if an already discovered node found a shorter path?

Frederickson's BFS Tree Algorithm



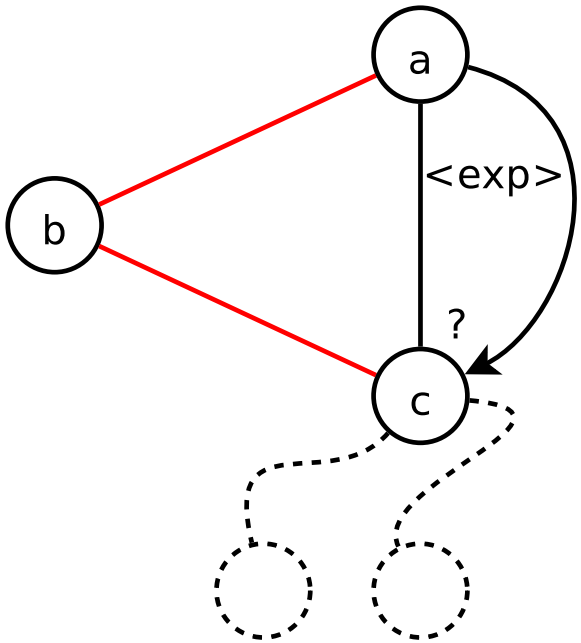
- New nodes also **explore**
- When ℓ levels are explored, **reverse** is started

Difficulties

Old-parent problem

- What if an already discovered node found a shorter path?
- Somehow tell former parent things changed...

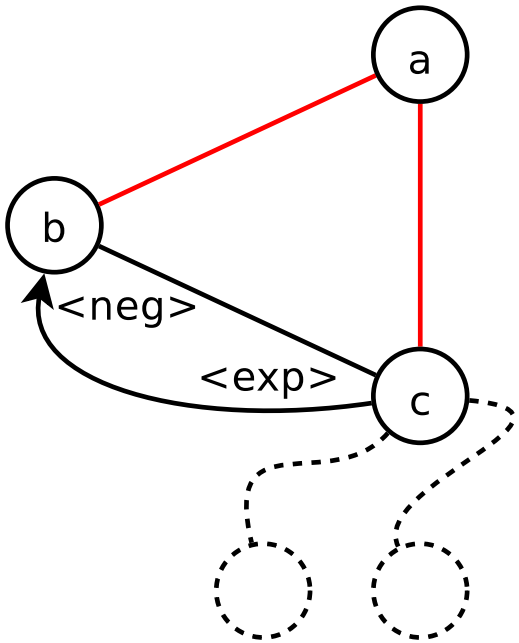
Old-parent problem



Problem

- *a* sends **explore** to *b* and ... *c*
- *b* sends **explore** to *c*
- *c* sends **explore** to neighbors
- *c* ... receives **explore** from *b*

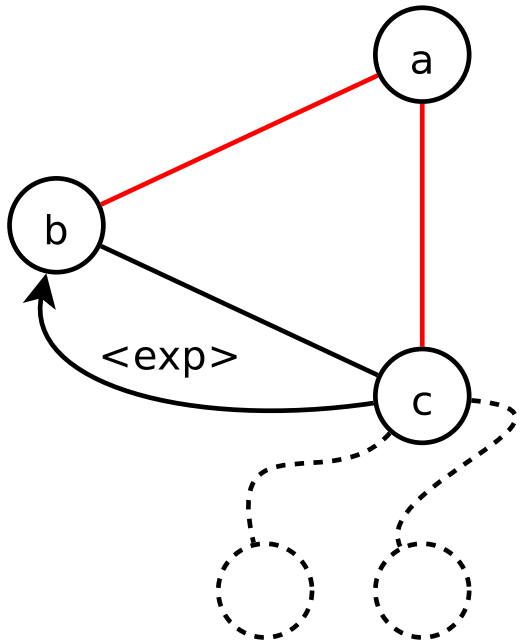
Old-parent problem



Frederickson's solution

- c sends **negative** to b
- b notices that c apparantly found another father
- c sends **explore** to neighbors
- ...

Old-parent problem



New solution

- c sends **explore** to neighbors
- b notices that c apparantly found another father
- ...

Supporting non-FIFO channels

Frederickson and Tel stated: only FIFO channel support, but:

- No problems with **forward**/**reverse**
- Delayed **explore** messages can be recognized. (**explore**'s with high level are discarded)
- Delayed **reverse** messages can be recognized. (**reverse**'s from parent are discarded)

My version of the algorithm works for non-FIFO channels.

Optimal ℓ value

- (N is number of nodes, E is number of edges)
- Message Complexity $O\left(\frac{N^2}{\ell} + \ell E\right)$
- $\frac{N^2}{\ell} = \ell E$
- $\ell = \frac{N}{\sqrt{E}}$

Conclusions

Frederickson's Description

- Complex solution for old-parent problem.
- Not well documented.
- No notion about non-FIFO channels.

Tel's Description

- No solution for old-parent problem.
- Not well documented.
- No notion about non-FIFO channels.

Conclusions

Improvements

- Correct and detailled description.
- Elegant solution for old-parent problem.
- Showed that algorithm works with non-FIFO channels.
- Pseudocode provided.
- Implementation provided (in ANSI C).

Questions?