EDMOND'S BLOSSOM ALGORITHM

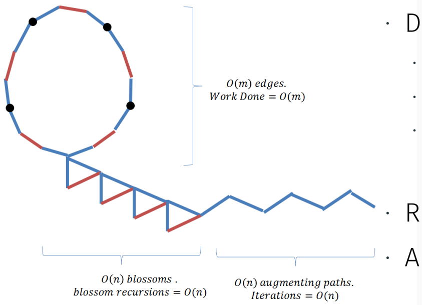
Amy Shoemaker and Sagar Vare

May 31, 2016

CME 323: Distributed Algorithms and Optimization

The Sequential Blossom Algorithm

Blossom Algorithm: to find *maximum matching* in an (undirected, unweighted) graph (as you all already know)



- Data structures:
  - Graph node list, Graph edge list
  - Matching node list, Matching edge list
  - Forest list of trees (tree node list, tree edge list)
- Runtime: $O(n^2m)$
- Analysis is tight for (sparse) kite graph

The Parallel Blossom Algorithm

- Data structures are largely the same as in the sequential case.
- Inherently sequential operations:
  - $O(n)$ from iterations of finding augmenting paths
  - $O(n)$ Blossom recursions
- Wiggle room of $O(m)$ to work with, which can be brought down to $O(n)$ in parallel.
- Intelligent combining scheme to ensure all possible parallelism.
- Runtime: $T_p \leq O(n^3 + n^2 m/p)$

Distributed Blossom Algorithm

- Only edges are distributed; we assume $O(n)$ can be stored locally
- Data structures:

FOREST NODES

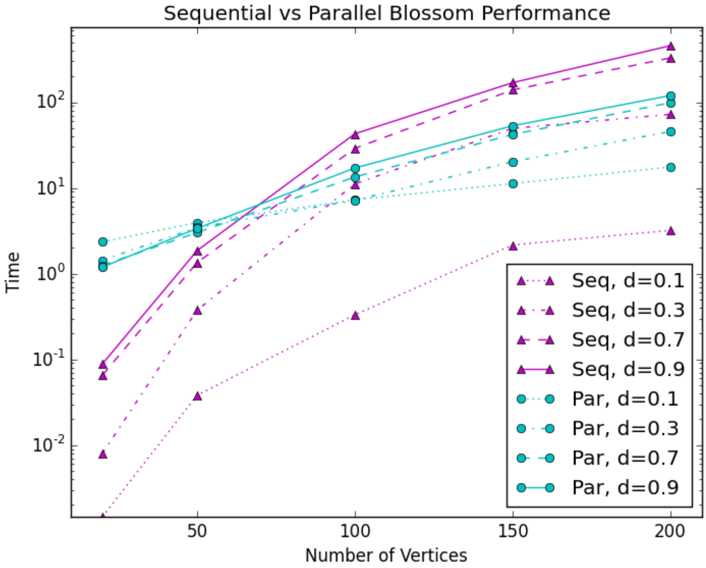
Node	root	parity
1	10	1
2	10	2
3	18	2
4	18	1
5	19	1

FOREST EDGES

Node1	Node2	root
1	2	10
2	1	10
3	4	18
4	3	18
5	8	19

- Updates with broadcast to avoid all-to-all communication of joins
- Analysis:
  - Communication Cost: $O(n^2 m)$
  - No shuffle cost! No joins!
  - Computational Complexity: $O\left(\frac{n^4}{p}\right) =$

$$\underbrace{O(n)}_{\text{Iterations}} * \left[\underbrace{O(n)}_{\text{Blossom Recursion}} * \sum_{v \in \text{Forest Nodes}} \left(\underbrace{O\left(\frac{n}{p}\right)}_{\text{CASE 1}} + \underbrace{O\left(\frac{n}{sp} + \frac{\deg v}{p}\right)}_{\text{CASE 2}}\right) + \underbrace{O\left(\frac{n}{p}\right)}_{\text{CASE 3}}\right]$$


Sequential vs Parallel Blossom Performance
Time
10^2
10^1
10^0
10^{-1}
10^{-2}
50
100
150
200
Number of Vertices
Seq, d=0.1
Seq, d=0.3
Seq, d=0.7
Seq, d=0.9
Par, d=0.1
Par, d=0.3
Par, d=0.7
Par, d=0.9

Questions?