PARALLEL ANT COLONY OPTIMIZATION

CSE633 Parallel Algorithm

Matthew Sah

Instructed by Dr. Russ Miller

University at Buffalo The State University of New York

Index

- Problem Statement
- Algorithm
- Parallel Approach
- Results
- Observations

Index

- **Problem Statement**
- Algorithm
- Parallel Approach
- Results
- Observations

Problem Statement

Traveling Salesperson

- Given a list of cities and distances,

  find the shortest distance city A to city B

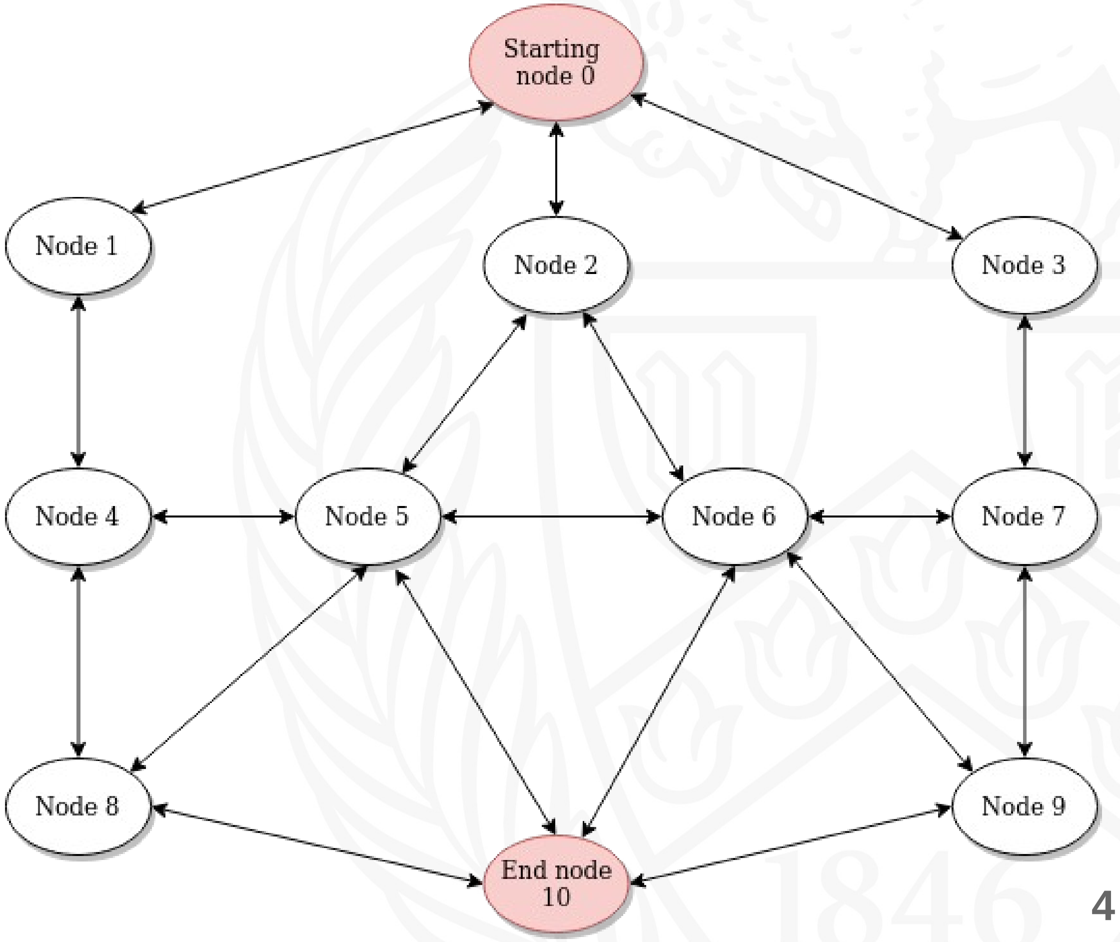
NP-hard Question

- Just try to optimize the solution

Distance has different costs

- Chance of either direction is different

- The order of cities does not matter as long as it has the shortest distance



Index

- Problem Statement
- **Algorithm**
- Parallel Approach
- Results
- Observations

Algorithm - SACO

- The Simple Ant Colony Optimization
- Population based optimization
- Artificial Pheromone System
- Typically used to find shortest path problem
- Computationally Expensive
- Approximate Solution

Algorithm - Process

1. Construct Ant Solutions

2. Pheromones Evaporation

3. Update Pheromones

4. Termination Criteria

Algorithm
1. Constructing Solution

1. Construct Ant Solutions

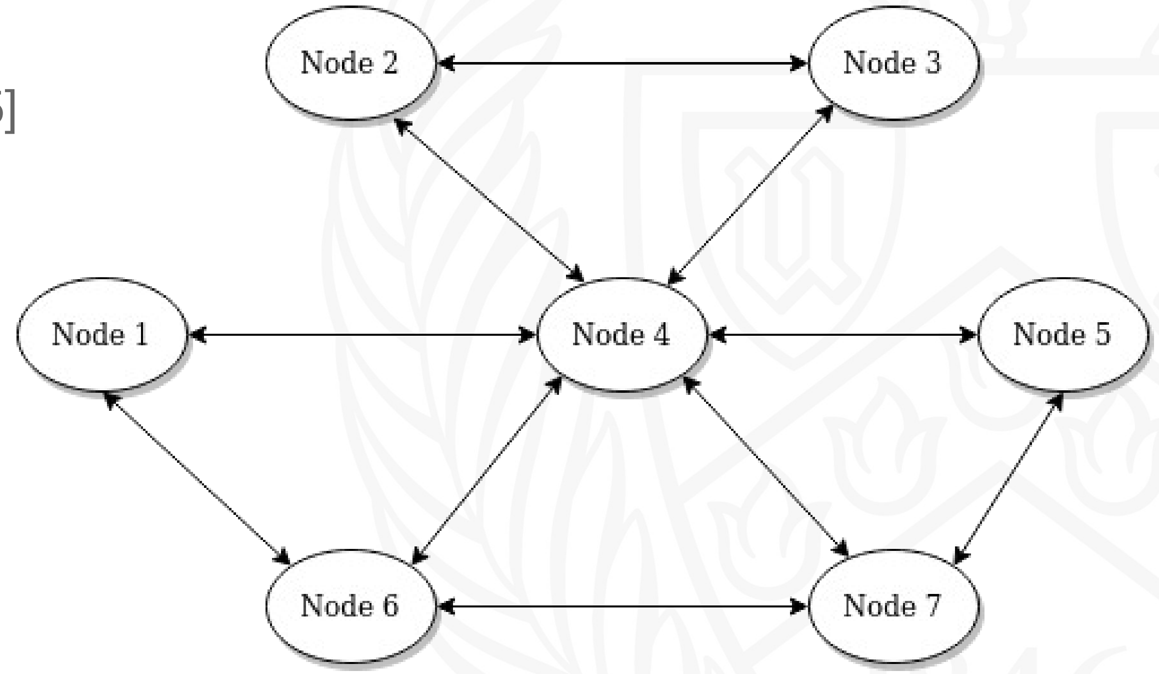
- Each iteration consists of X ants. Each ant constructs it's own solution (each ant walks from start to destination)

Suppose And A's original path[1, 4, 2, 3, 4, 2, 3, 4, 5]

2. Removes loops in solution

- Ant A condensed path [1, 4, 5]

- Ant B Path [1, 6, 7, 5]



Algorithm
3. Update Pheromone

Update Pheromones

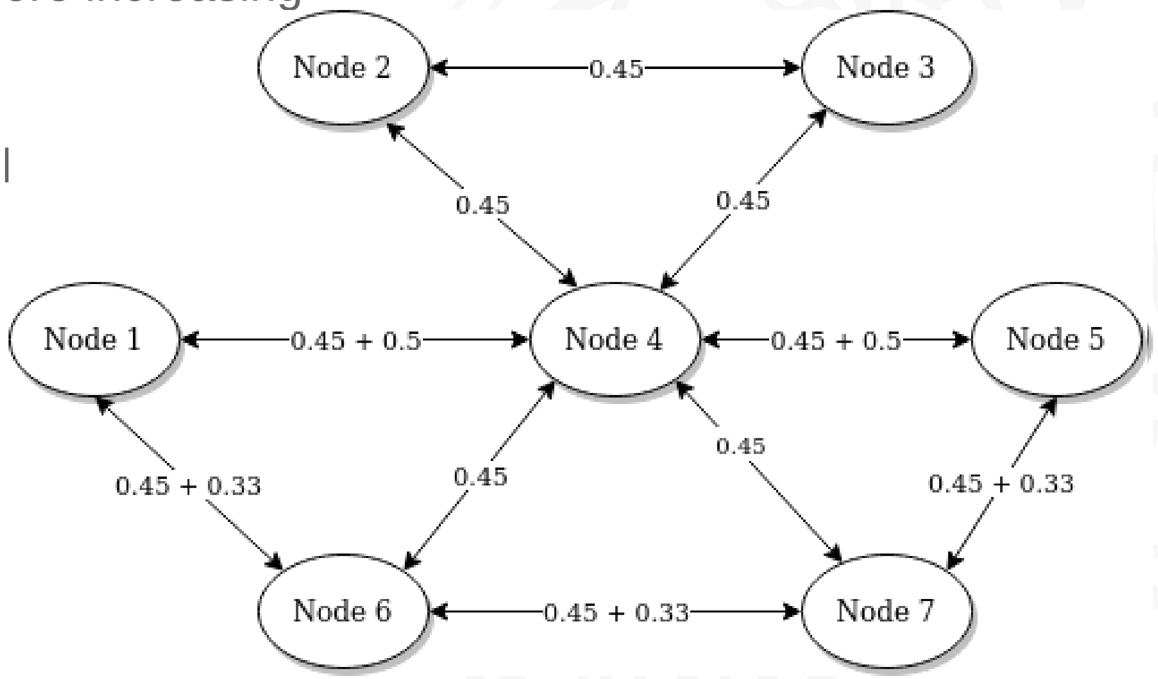
The faster routes have higher pheromones, therefore increasing the chance that the path is chosen by ants

We add the pheromone change to the path taken l

Ant Pheromone = 1 / Edges of Path

- Ant A condensed path [1, 4, 5]
- A's Pheromone Change= 1 / 2

- Ant B Path [1, 6, 7, 5]
- B's Pheromone evaporation= 1 / 3



Algorithm
4. Termination Criteria

1. No major changes in solution

When the solution no longer changes, it may mean that the optimum path has been found

2. Max number or iteration reached

We have a set max iteration to keep things in check

Index

- Problem Statement
- Algorithm
- **Parallel Approach**
- Results
- Observations

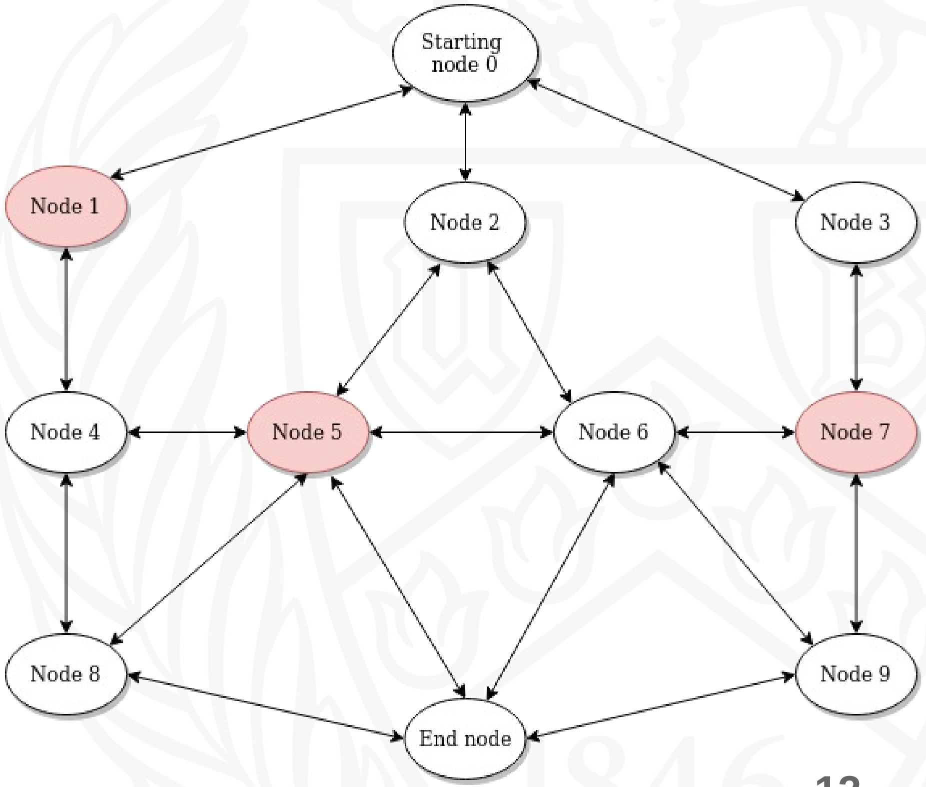
Parallel Approach

1. Each Processor is assigned several random “cities” as center point of random clusters

2. From each “city” attempt to reach the start and end point several times

3. Select the optimal solution solution among the random “cities”

3. Update pheromone according to local optimal



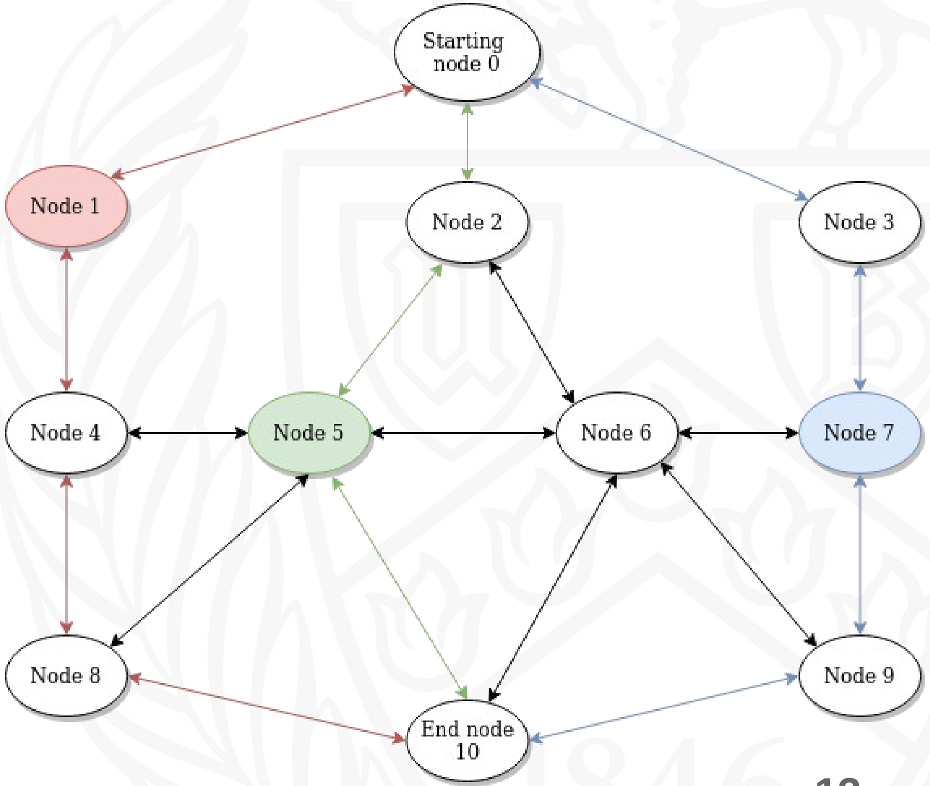
Parallel Approach

1. Each Processor is assigned several random “cities” as center point of random clusters

2. From each “city” attempt to reach the start and end point several times

3. Select the optimal solution solution among the random cities

3. Update pheromone according to local optimal



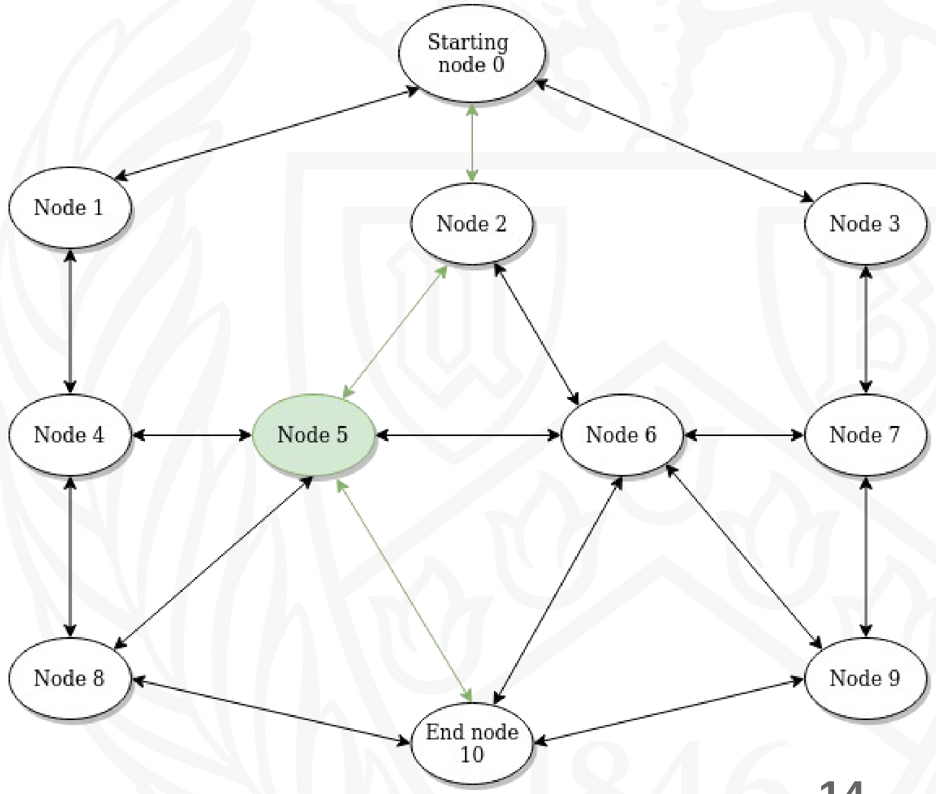
Parallel Approach

1. Each Processor is assigned several random “cities” as center point of random clusters

2. From each city attempt to reach the start and end point several times

3. Select the optimal solution solution among the random cities

4. Update pheromone according to local optimal



Index

- Problem Statement
- Algorithm
- Parallel Approach
- **Results**
- Observations

Results - Test Settings

Max Iterations : 1000

Evaporation Rate : 0.7

Ants : 7

Graph Size : 100, 300, 500, 900, 1000, 2000

Termination Criteria : 20 Consecutive “optimal” solutions

Skylake processors - 1, 16, 32, 64, 128, 256

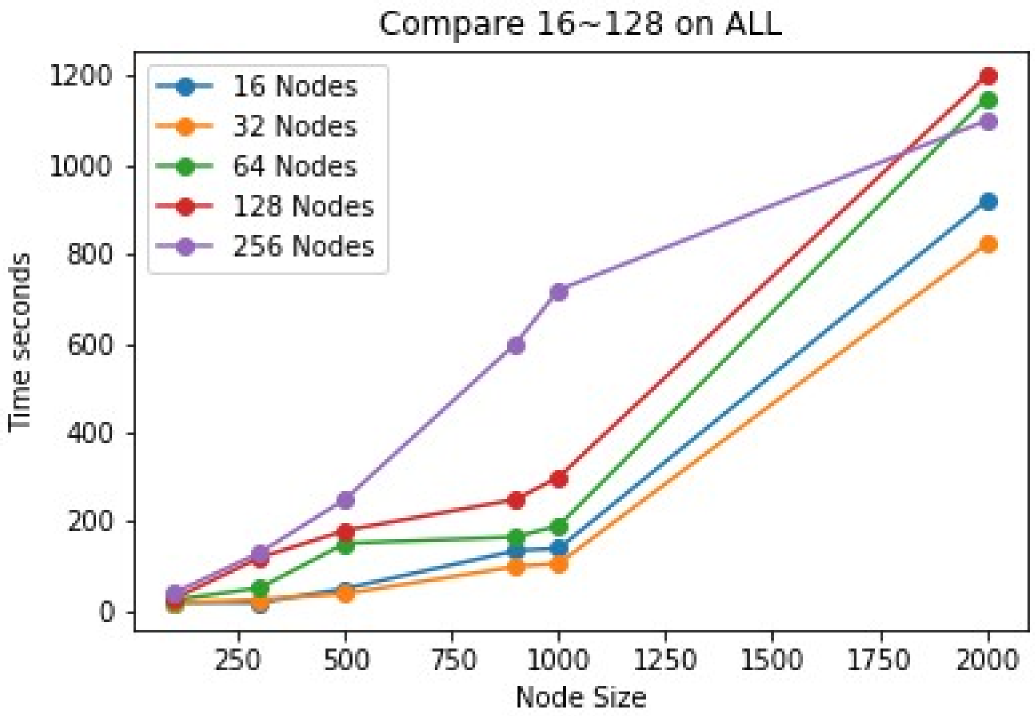
Max runtime : 30 minutes

Weights are initialized with same cost

Results :
16~128 Processors

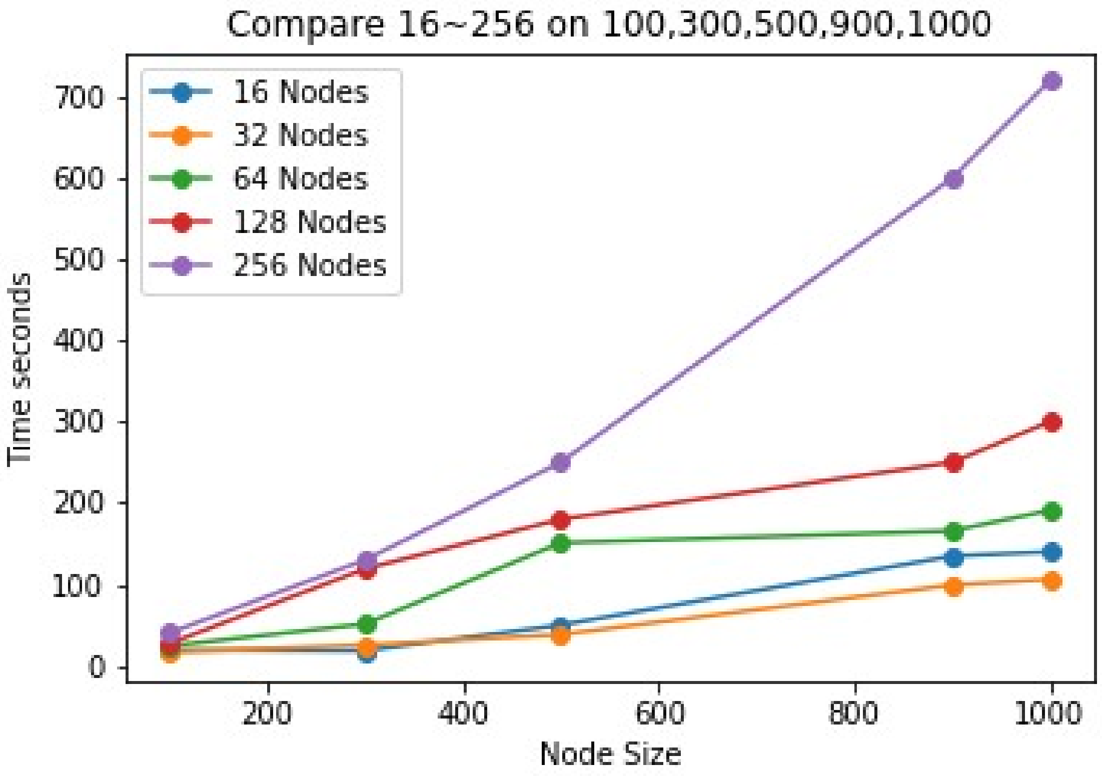
256 processors takes significantly longer on most cases and constantly ends on 1000 iterations.

Samples with less processors end within the iteration range but typically have sub-par results



Results :
16~256 Processors

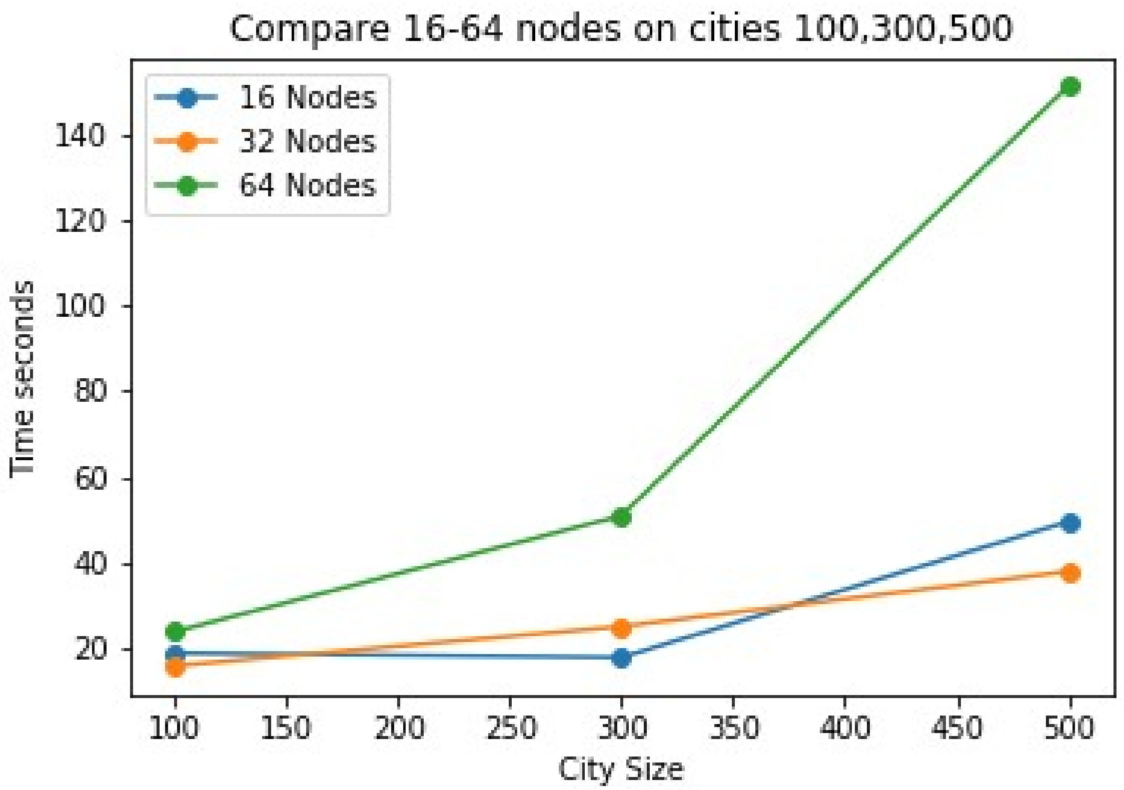
16 and 32 processors takes the least time



Results : 16~64 Processors

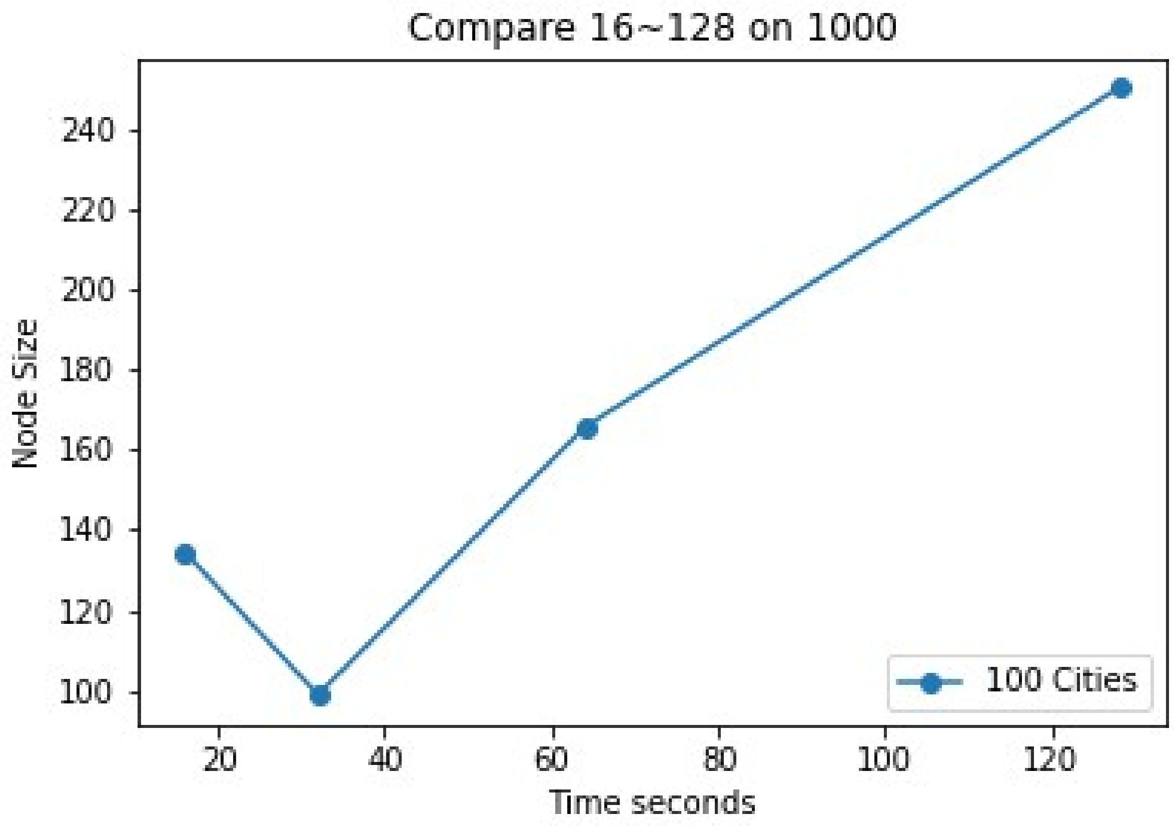
32 processors performs slightly better on the long term

Time	16 Processors	32 Processors
100 * 100	19	16
300 * 300	18	25
500 * 500	49	38



Results : 1000*1000

Processor #	Time (Second)
16	134
32	99
64	165
128	259



Index

- Problem Statement
- Algorithm
- Parallel Approach
- Results
- **Observations**

Observation

1. More processors isn’t necessary better

a. A single ant’s perception of the weights won’t change until it has reached the end point. By which many iterations, or none, may have passed

2. Communication Cost is really heavy

a. Updating the weight matrix requires heavy communication between the processors

Observation

3. On larger maps, finding a path alone is very expensive
    a. less processors converges on less optimal solutions
    b. more processors will run a lot faster, but will almost never converge

Using more processors...

Can sometimes reach optimal solution the first time an ant has finished it’s path, but during the first round for all the ants the solution has a high variance, therefore affecting how well the solution converges.

Simply put, the optimal solution may not have enough influence over the other non-optimal solutions

Ants round 1 for 256 processors

-> 2 -> 12 -> _bunch or random not optimal solutions_ -> ...

Ants round N for 256 processors

-> _bunch or random not optimal solutions_ -> ... -> 2 -> ...

Using less processors...

The solution will converge too fast, and very often not on the optimal solution since the influence is too strong

for example...

Ants round 1 for 4 processors

-> 10 -> 20 -> 15…

Ants round N for 4 processors

-> 10 -> 10 -> 10…

Observation

To find Best Time Given Solution:

- how long does it take to reach optimal solution?

- Looking for path with route length 2, how long will it take

- Try more processors

To find Best Solution Given Time:

- In a time limit, how good will be your optimal result?

- Give me your best result in 3 minutes

- Try less processors

QUESTIONS