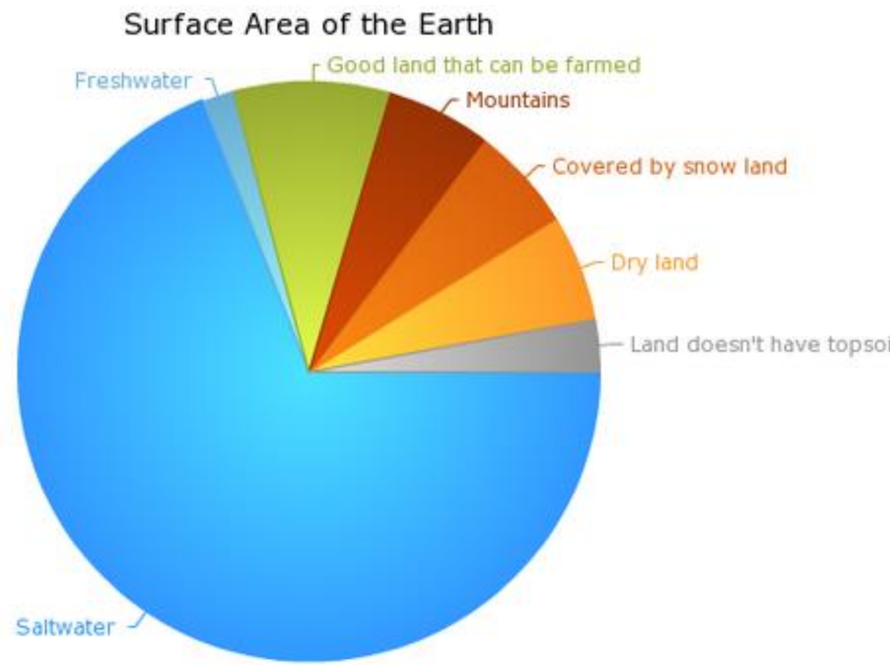
On Efficient Data Collection and Event Detection with Delay Minimization in Deep Sea

Huangyang Zheng and Jie Wu

Computer and Information Sciences
Temple University

1. Introduction

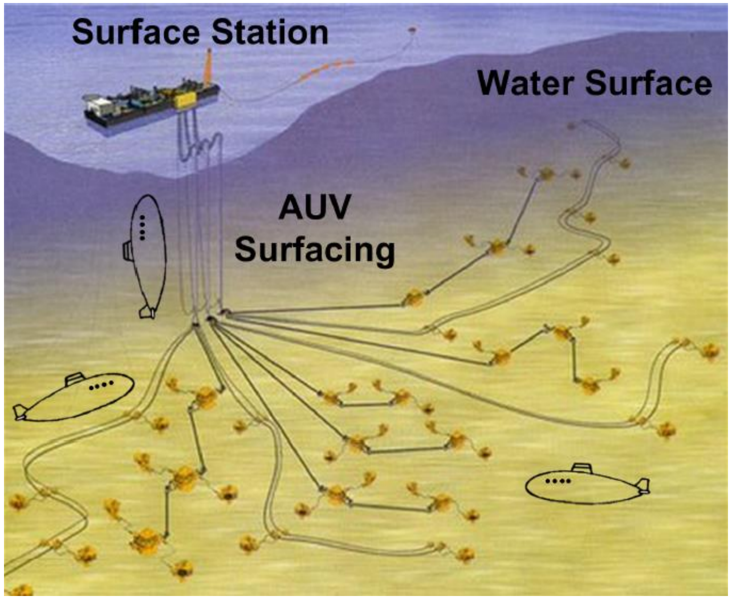
- Efficient searching in deep sea is notoriously difficult due to the vast searching area
  - The search-and-rescue effort of Malaysia flight MH370 in the Pacific ocean
  - The detection of oil pipe leak through robotic submarines in Mexico



- Electromagnetic signal decays quickly in the water, while acoustic signal has a limited bandwidth

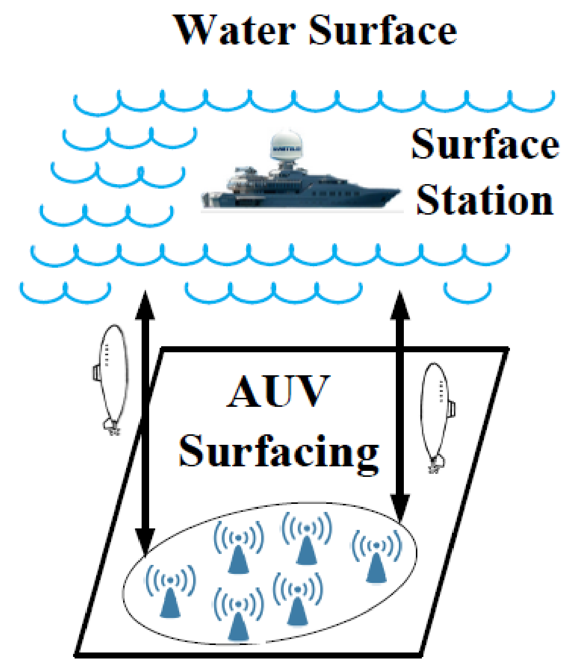
1. Introduction

- Multiple autonomous underwater vehicles (AUVs) are used to surface to transmit collected data (or events)
- The 2-D search space (a set of connected line segments) is parallel to the water surface
- Examples:
  - Sensors on oil pipes
  - Submarine cable
  - Undersea tunnel



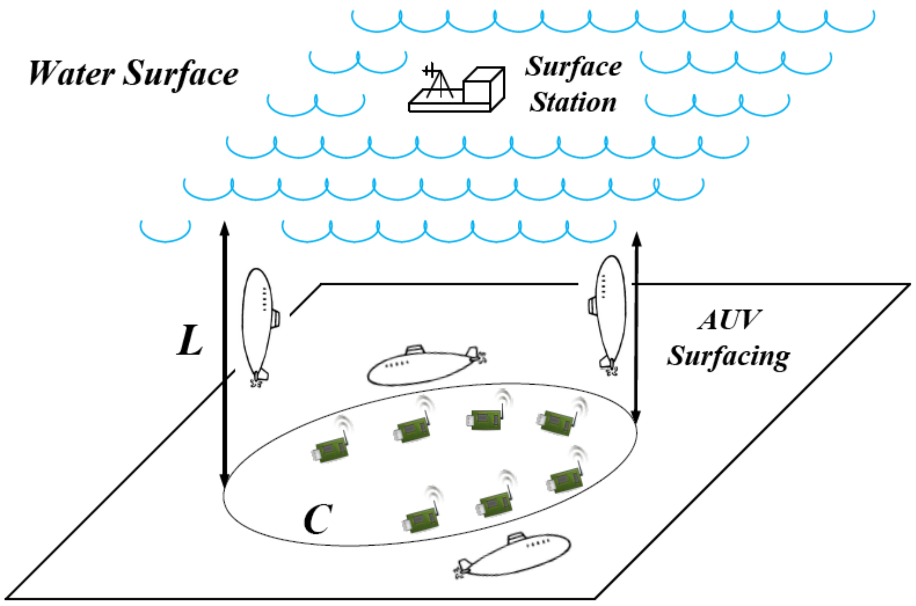
2. Problem Description

- The AUV trajectory planning problem, which aims to minimize the average data delay
  - How can we schedule the AUVs to resurface optimally in a circular search space (Eulerian cycle)?
  - How can we convert a general searching space to a circular search space?
  - Can we shorten the circumference of the converted circular search space?



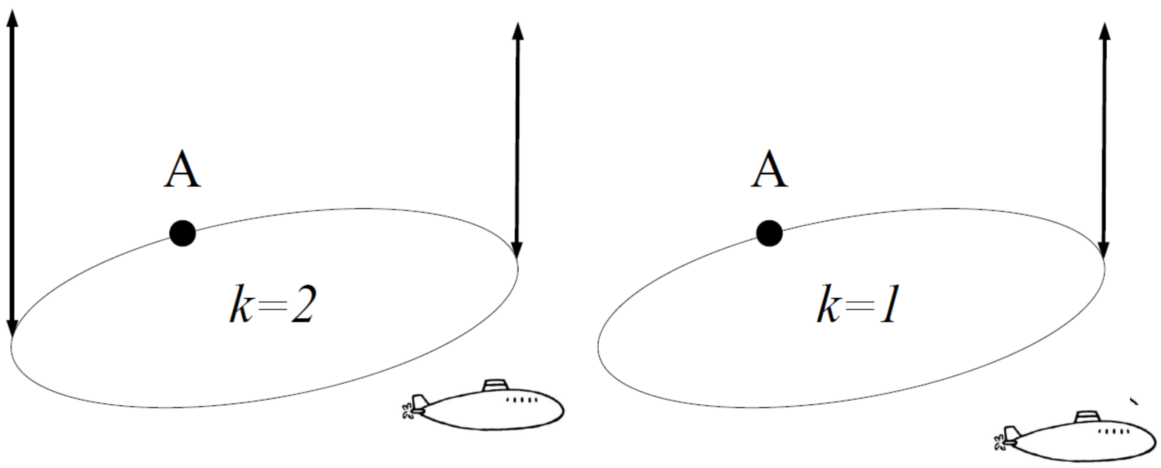
3. Optimal AUV Resurfacing

- Data are uniformly distributed with a fixed generation rate
- Objective: minimize the long-term average data delay (to the water surface)
- The speed of the AUV is unit
  - C : the cycle circumference
  - L : the searching space depth
  - n : the number of AUVs



3. Optimal AUV Resurfacing

- If we have a larger AUV resurfacing frequency, the AUV can bring node A's data to the water surface more quickly
- However, node A's data needs to wait the next AUV for a longer time, since resurfacings take additional time
- AUV should resurface more frequently for shallow search spaces

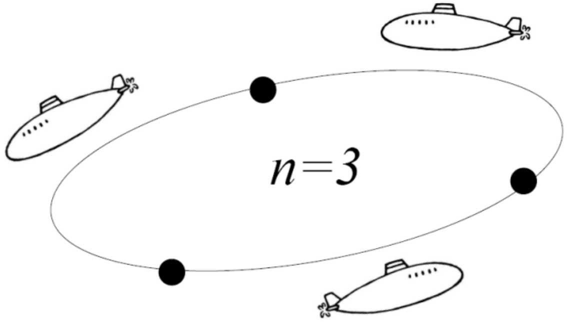


3. Optimal AUV Resurfacing

- Theorem 1. Optimally, the AUV resurfaces after traveling a distance of $\sqrt{2LC}$ on the original cycle (if we only have one AUV)

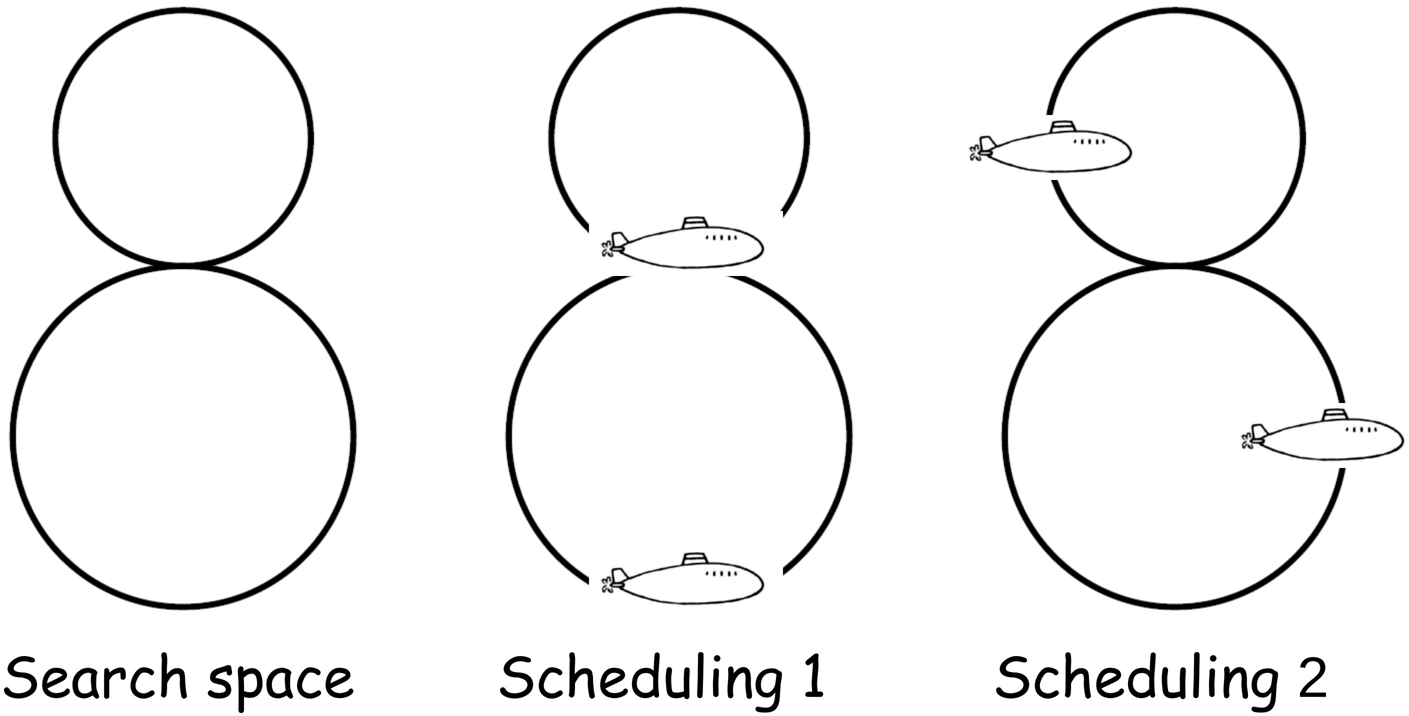
- If we have multiple AUVs (n AUVs), then we can evenly distribute these AUVs on the cycle.

- Each of these AUVs optimally resurfaces after traveling a distance of $\sqrt{\frac{2LC}{n}}$



4. Constructing of A Cycle

- Why do we use only one large cycle instead of multiple small cycles to cover the search space?



- Theorem 2. Scheduling 2 is always no worse than Scheduling 1, due to more balanced cycling tasks among AUVs

4. Constructing of A Cycle

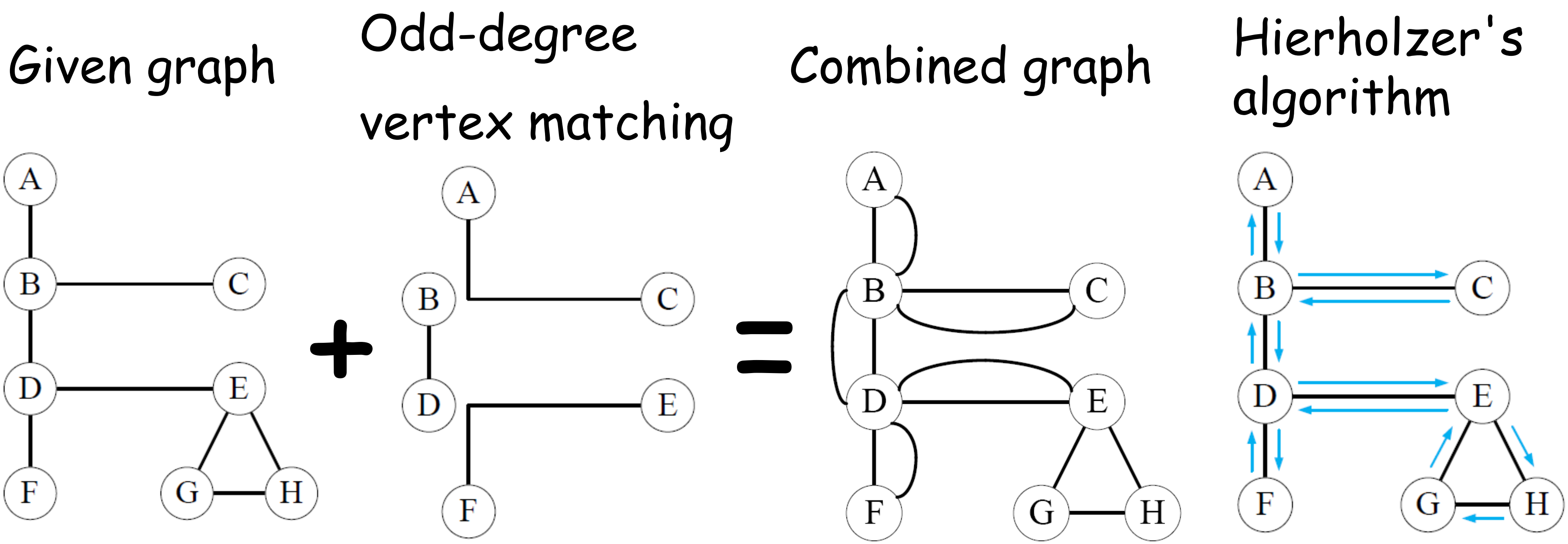
- General search space: a set of connected line segments (called sensing edges in the graph)

- Graph with an even degree for every vertex
  - An Eulerian cycle exists

- Graph has vertices with odd degrees
  - Add redundant edges to make odd degree even
  - We need to minimally pairwise odd degree nodes by adding one link (There are even number of vertices with odd degrees)

4. Constructing of A Cycle

- Construct an Eulerian cycle by adding sensing edges (Algorithm 1)

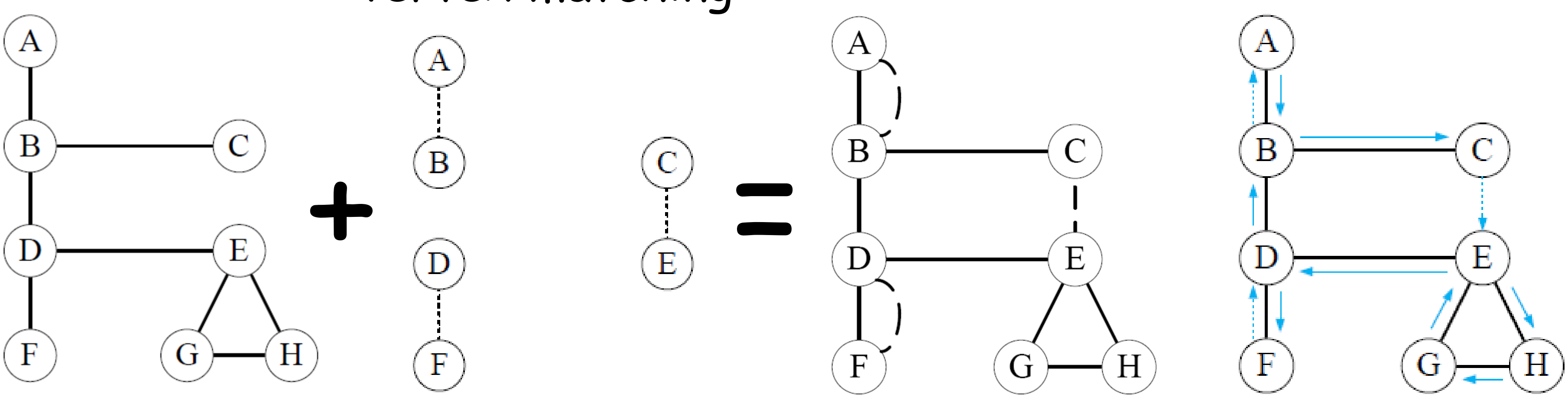


- Some sensing edges are visited for multiple times

5. Cycle Enhancement

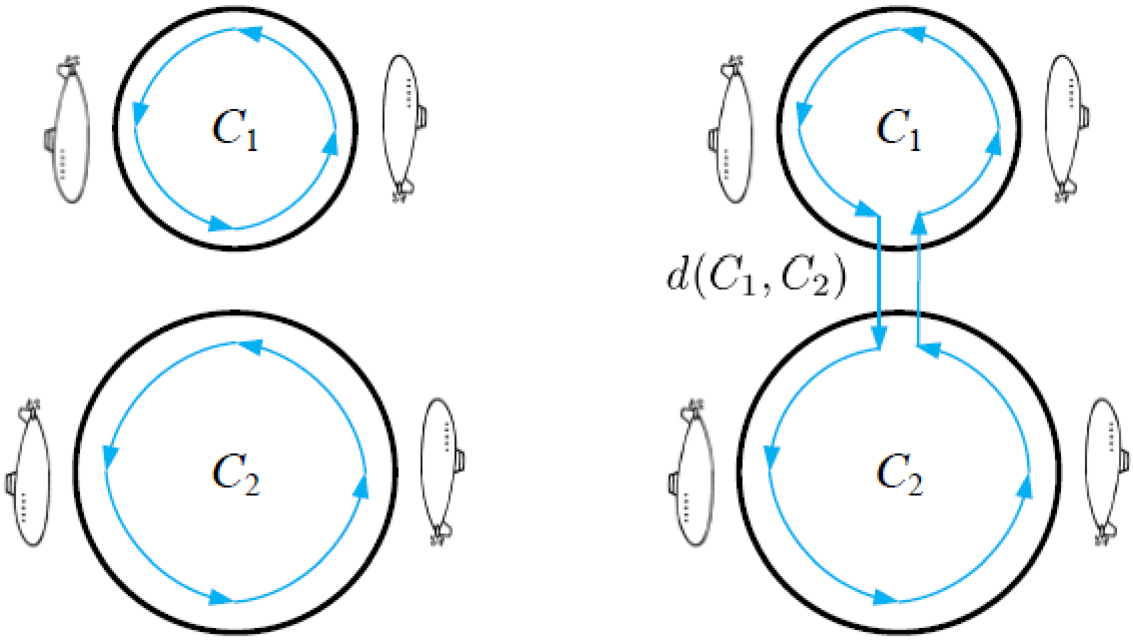
- Geometric shortest non-sensing edges (which may be not in the search space) can shorten the cycle circumference, although no data is collected on them
- Construct the cycle by adding non-sensing edges (Algorithm 2)

Given graph | Odd-degree vertex matching | Combined graph | Hierholzer's algorithm



6. Cycle Merge

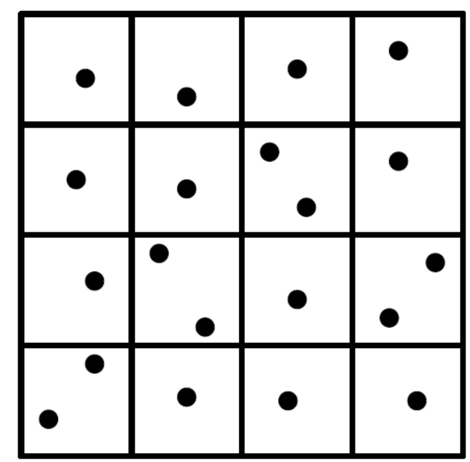
- Cycle merge to further reduce delay
- Greedy cycle merge algorithm

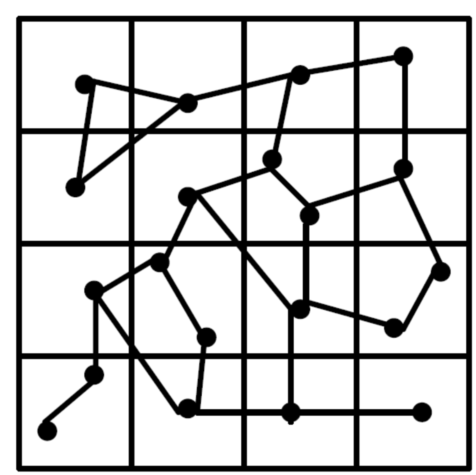


(a) Two cycles. (b) Merge result.

7. Experiments

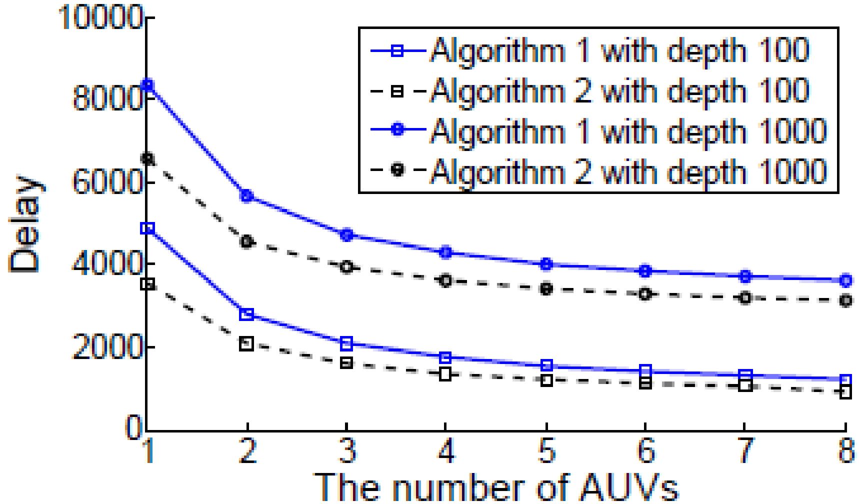
- Settings
  - The test is based on a synthetic trace, which is generated through uniform, random placement of 100 nodes on a 100*100 square unit
  - To guarantee the graph connectivity, a spanning tree is constructed. Additional edges, with given total numbers of 20 and 100, are used to uniform-randomly connect these nodes
  - AUV has unit speed



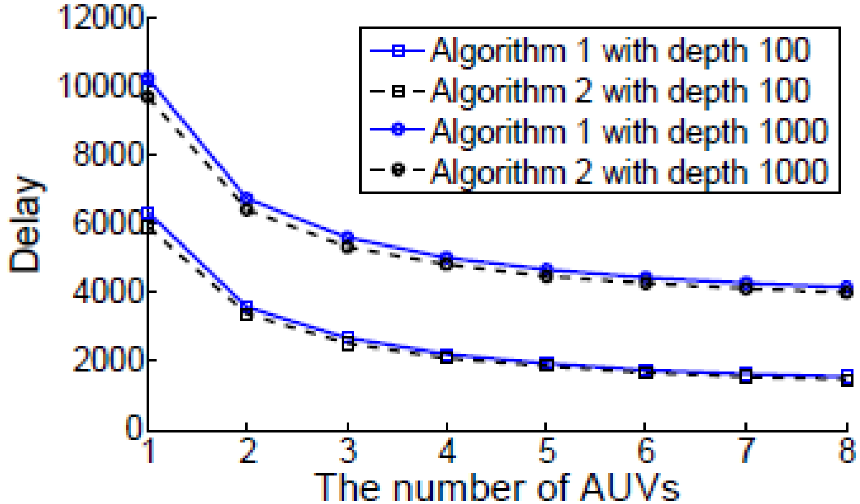


7. Experiments

- Proof-of-concept results:



(a) Given 20 additional edges.



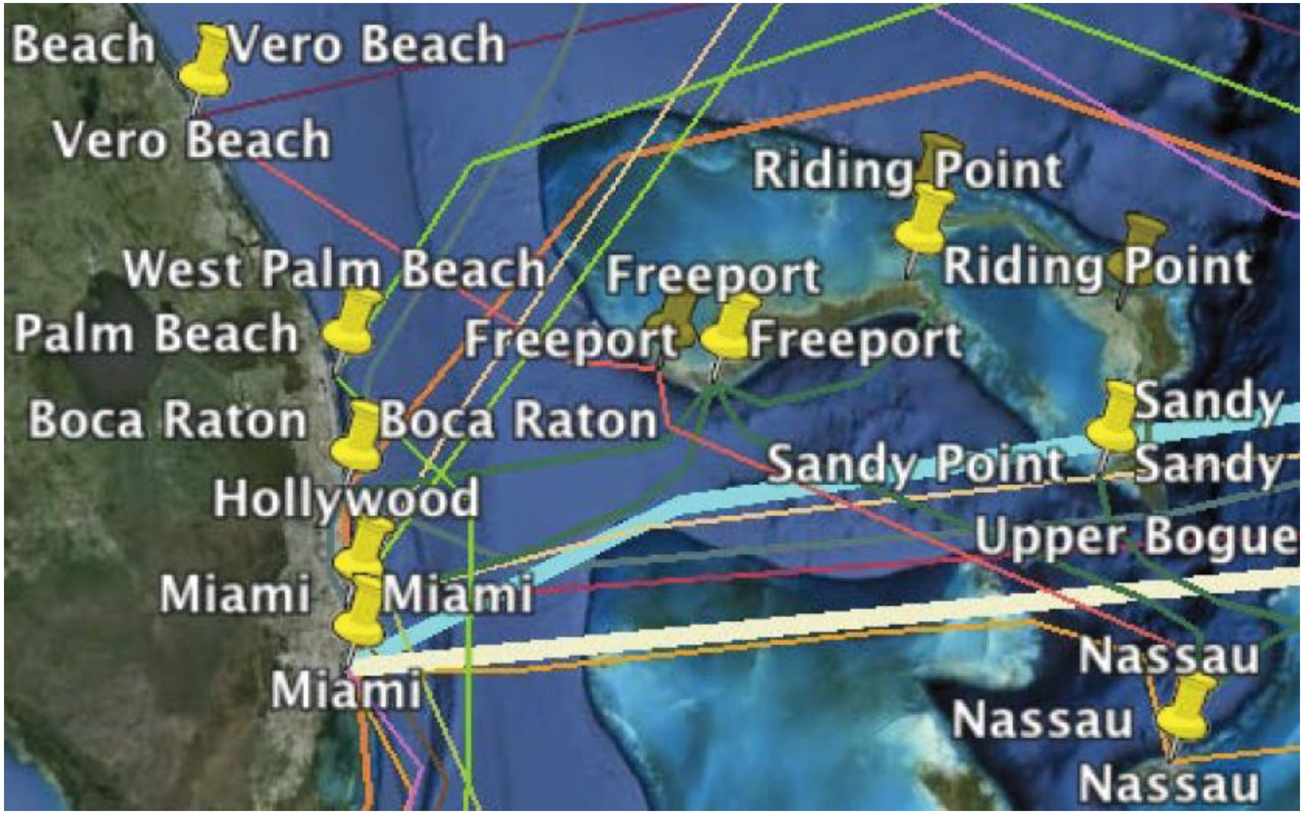
(b) Given 100 additional edges.

7. Experiment

- A sparser graph leads to a larger gap between Algorithms 1 and 2 (an improvement of more than 20% for the sparse graph)
- The gap of pairwising odd vertices through the shortest path and that through the geometry link is becoming smaller, when the trace gets denser
- The delay reduction brought by one more AUV decreases, with respect to the current number of AUVs (i.e., the effect of diminishing return)

7. Experiment

- Real trace-driven experiments
  - Oil pipe layout near Florida (sea depth: 3790m)



7. Experiment

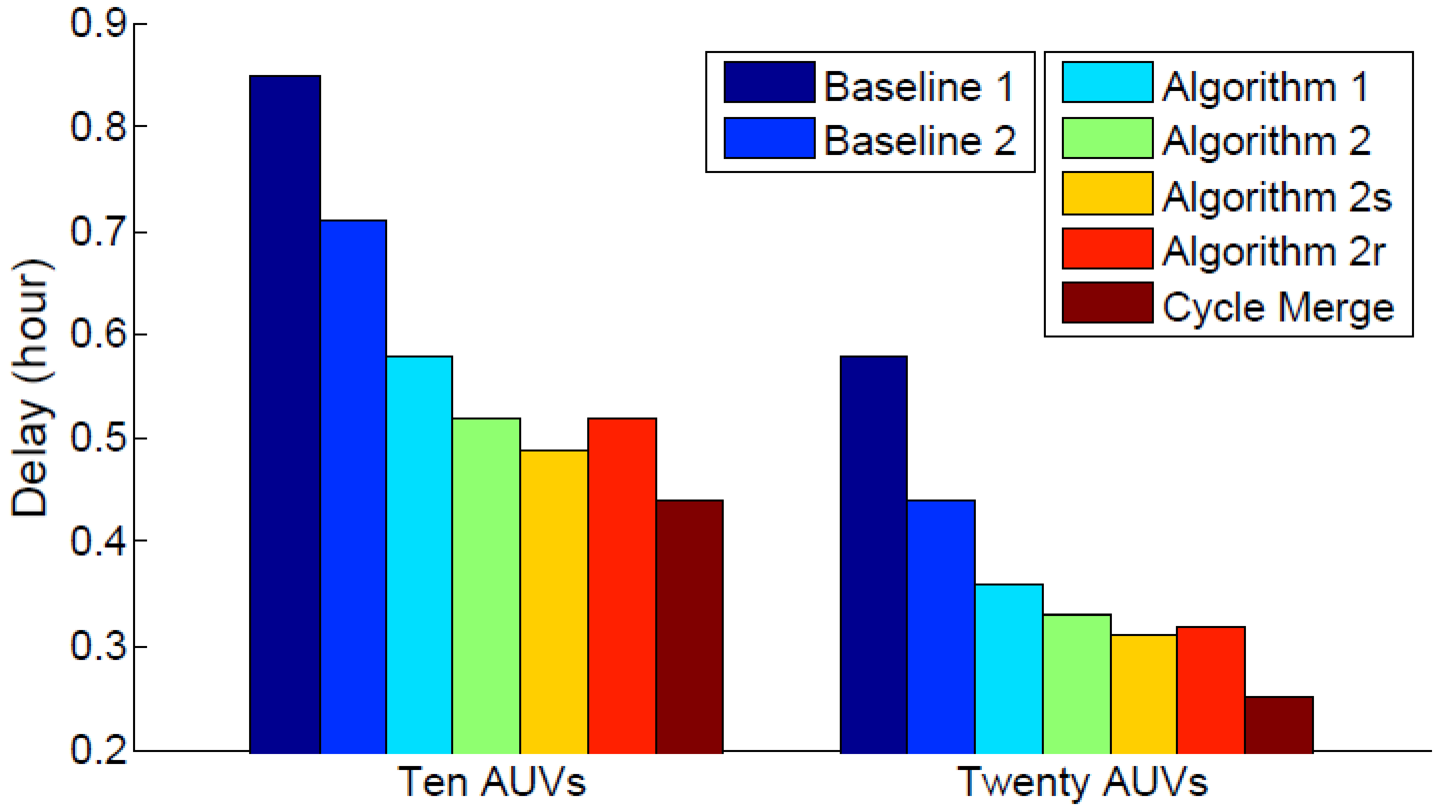
- Real trace-driven experiments (20 AUVs)
  - AUV cruising speed is 37km/h
  - AUV diving/surfacing speed is 26km/h
  - Sensors are uniformly placed along each pipe

- Baseline 1: distribute AUVs uniformly to each pipe. AUVs go back and forth independently

- Baseline 2: distribute AUVs according to the pipe length. AUVs go back and forth independently

7. Experiment

– Real trace-driven experiment result:



8. Conclusions and Future work

- The AUV trajectory planning determines the AUV resurfacing frequencies and their locations
- The deep sea trajectory planning is simplified to an extended Euler cycle problem
- Future Work
  - A more general approach for the search space that is not a set of connected segments

End

Q & A