Minimizing Communication Cost in Distributed Multi-query Processing

Jian Li
University of Maryland, College Park
(joint work with Amol Deshpande and Samir Khuller)

Outline

- Motivation & Problem Formulation
- Summary of Our Results
- Multiple Queries
  - NP-Hardness
  - Polynomial time algorithm for tree communication networks
  - Approximation algorithms
- Experiments
- Future Work

Motivation

- Emergence of large-scale distributed query processing
  - Scientific federations like SkyServer, GridDB
  - Publish-subscribe systems and content delivery networks
  - Distributed data streams and web sources
  - Sensor networks
  - Large scale data analytics (MapReduce, Hadoop)

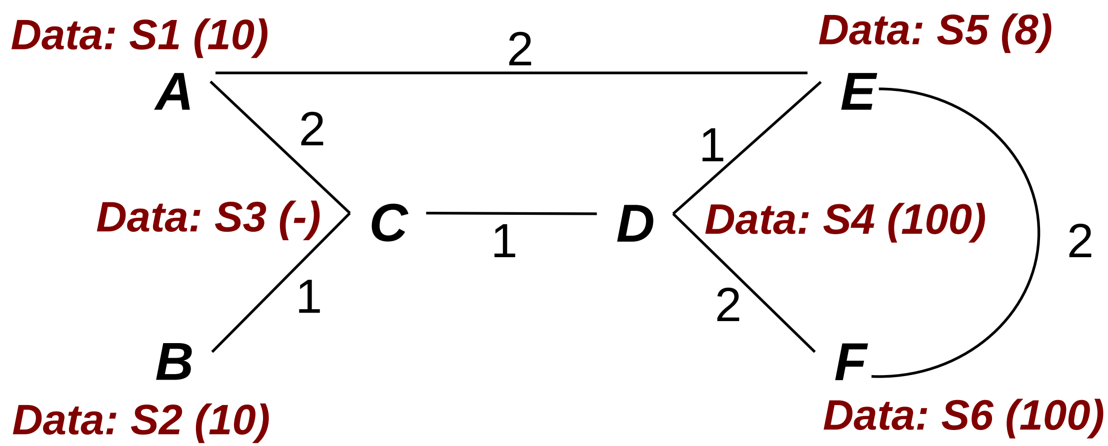
Motivation

- Need to support:
  - *Very large datasets and/or*
  - *Large numbers of users and queries*
- Minimization of communication cost often a key problem
  - *Network utilization in Internet-scale systems*
  - *Energy consumed in sensor networks*

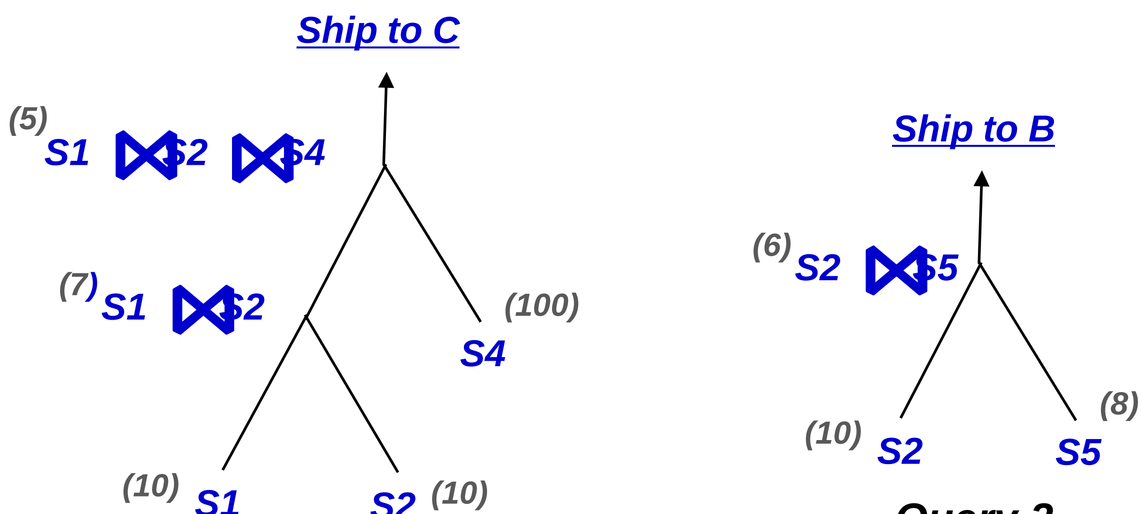
Challenges:

- How to choose query plan
- How to ship data across the network to implement these plans

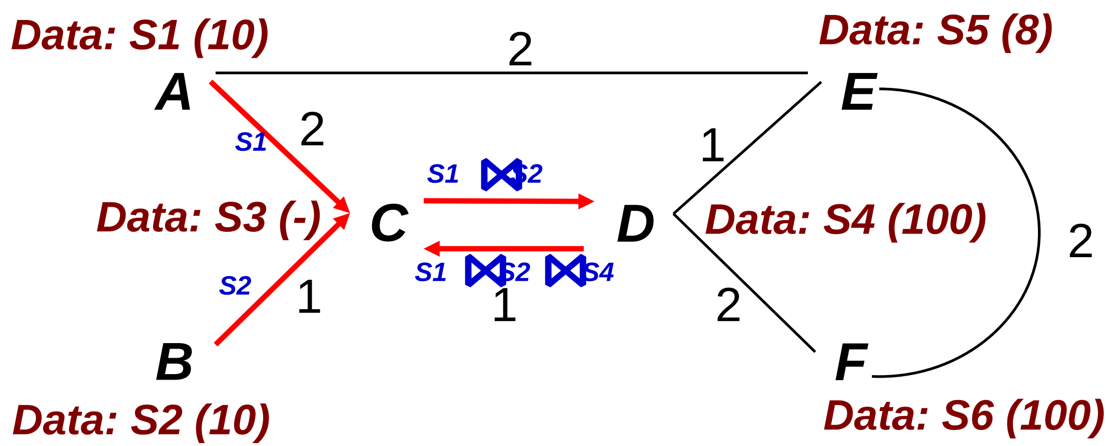
Example: Distributed Databases



Communication Network



Example: Distributed Databases



Cost= 10*2

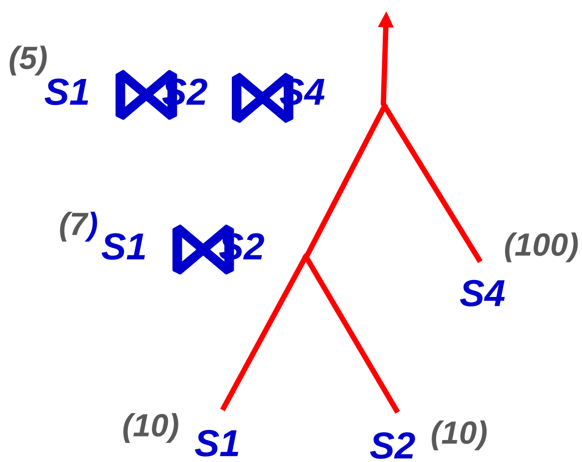
+ 10*1

+ 7*1

+ 5*1

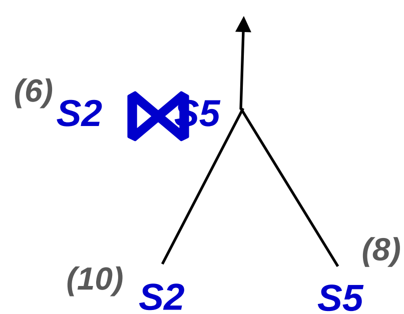
Communication Network

Ship to C



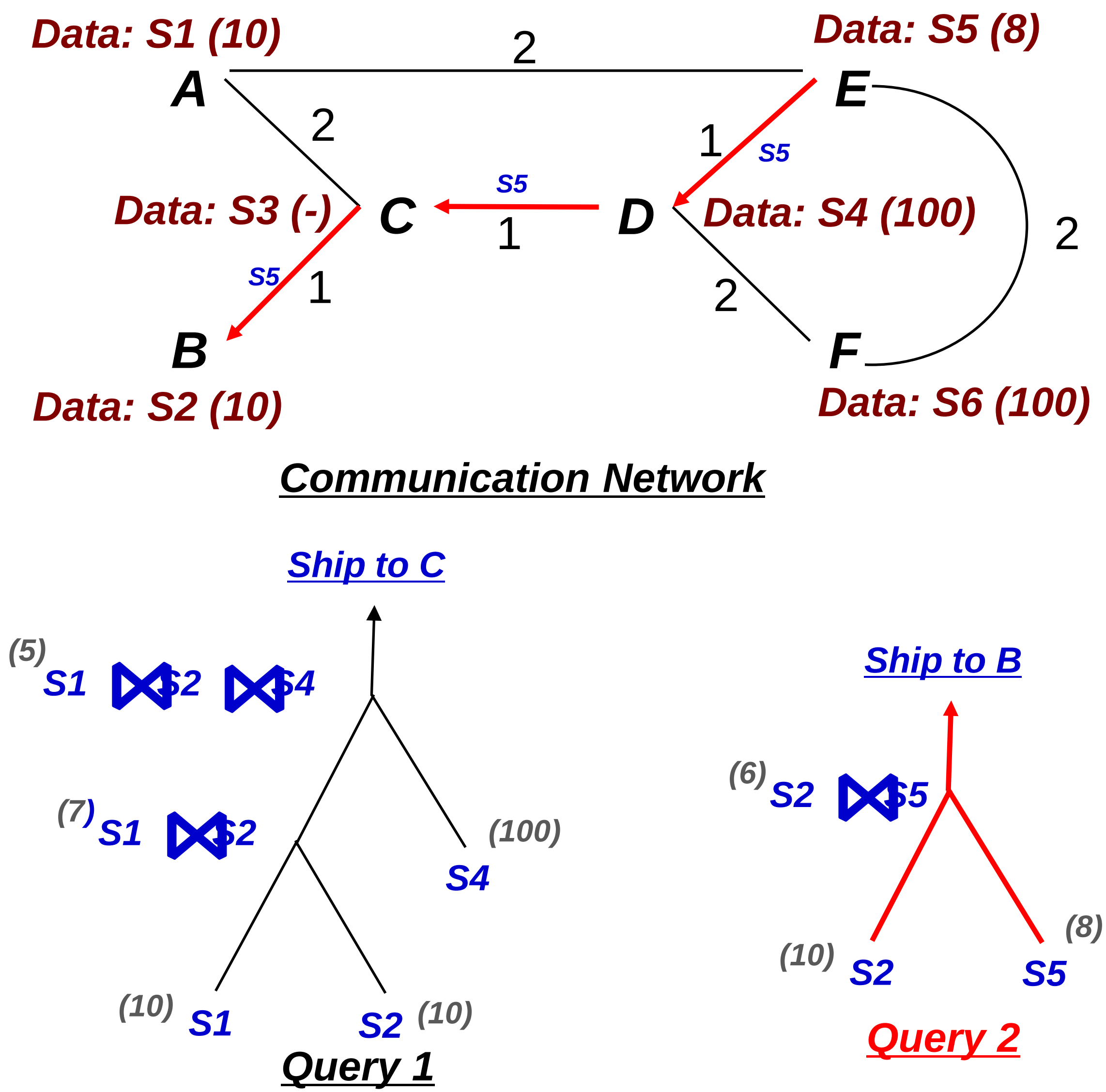
Query 1

Ship to B



Query 2

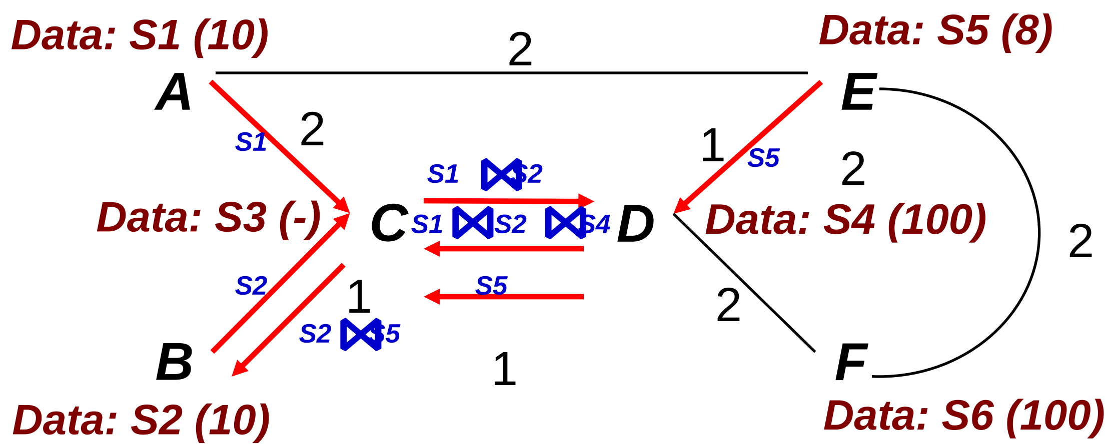
Example: Distributed Databases



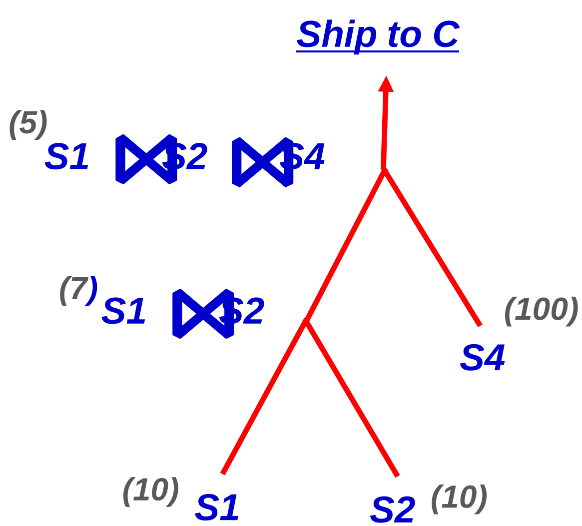
Cost=

8*(1+1+1)

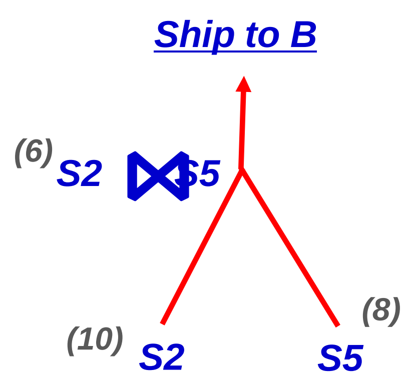
Example: Distributed Databases



Communication Network



Query 1



Query 2

Cost=

10*2

+ 10*1

+ 7*1

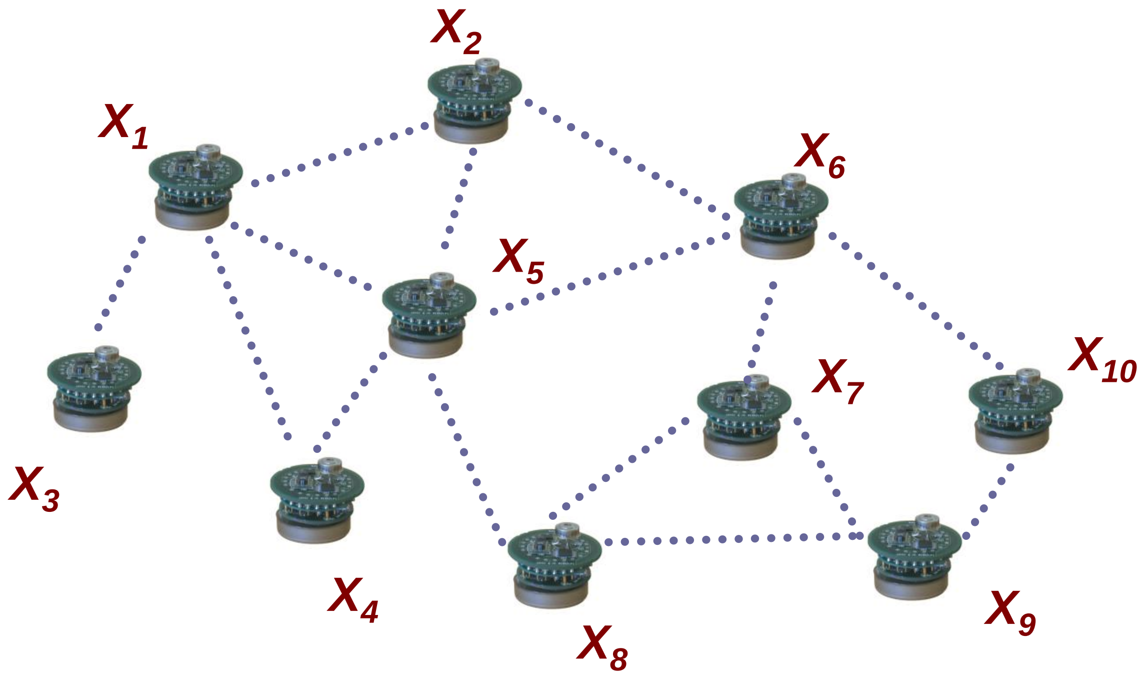
+ 5*1

+ 8*(1+1)

+ 6*1

Example: Sensor Network Aggregates

- [Silberstein and Yang, 2007] Many-to-many Aggregation



Q1 (*issued by* $\mathbf{X_7}$): $2\,X_1 + 3\,X_2 + X_4$
Q2 (*issued by* $\mathbf{X_8}$): $X_1 + X_2 + X_3 + X_4$
Q3 (*issued by* $\mathbf{X_6}$): $2\,X_2 + 3\,X_3 + X_5$

Problem Formulation

- Input:
  - Communication Network $G(V,E)$
    - Edge weights indicate the communication costs
  - Data sources: $S_1, \ldots, S_n$
  - A set of queries: $Q_1, Q_2, \ldots$..
  - For each query Q, a query plan (tree) is given
    - No join order optimization
- Goal:
  - Minimize the communication cost of executing the queries

Our Results

Single Query

- Polynomial time solvable (by standard dynamic programming)

Multiple Queries

- NP-Hard on general communication networks
- Polynomial time solvable on tree communication networks
- O(logn)-approximation for general communication networks
- O(1)-approximation for some special cases

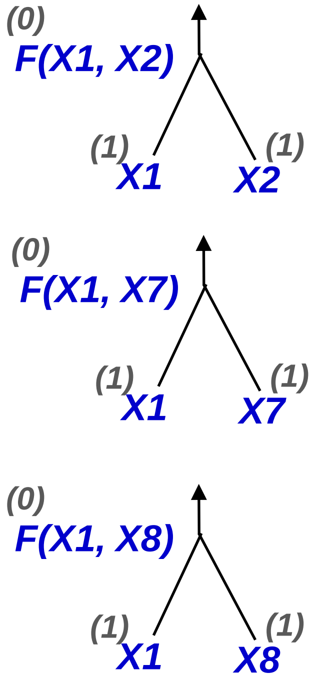
Outline

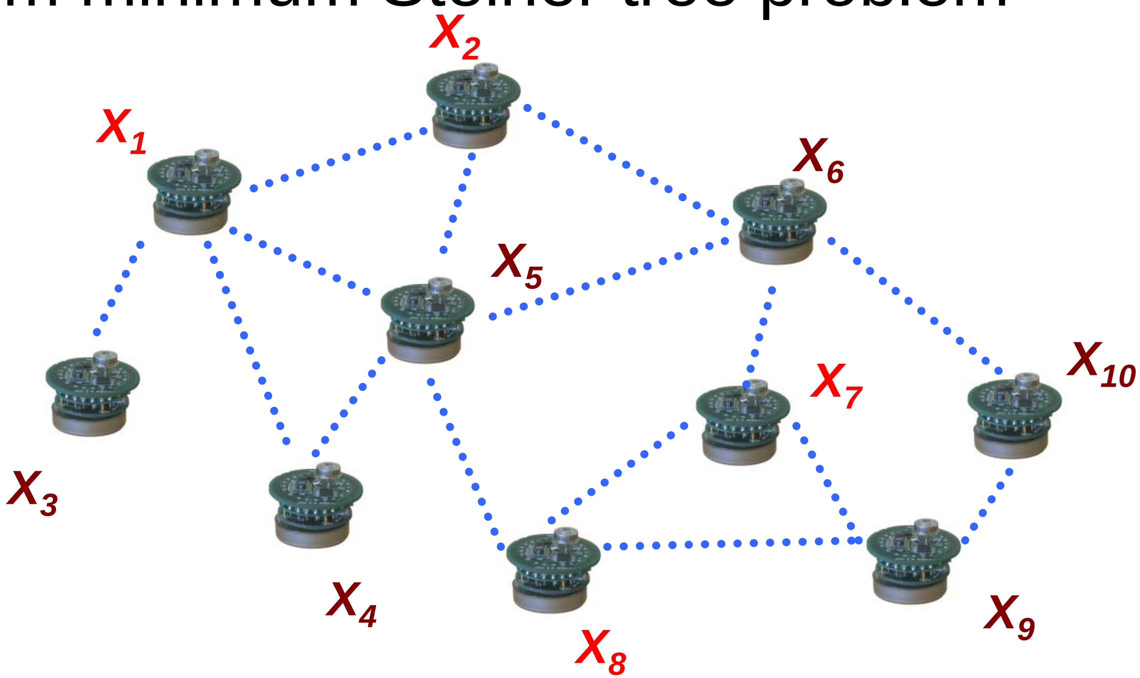
- Motivation & Problem Formulation
- Summary of Our Results
- Multiple Queries
  - NP-Hardness
  - Polynomial time algorithm for tree communication networks
  - Approximation algorithms
- Experiments
- Future Work

Complexity

- NP-Hard for general communication networks
- Reduction from minimum Steiner tree problem

Queries:

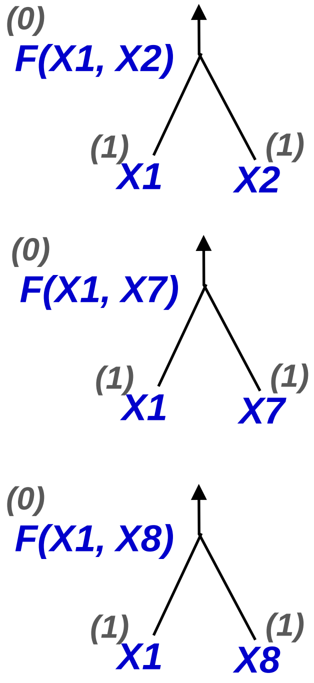


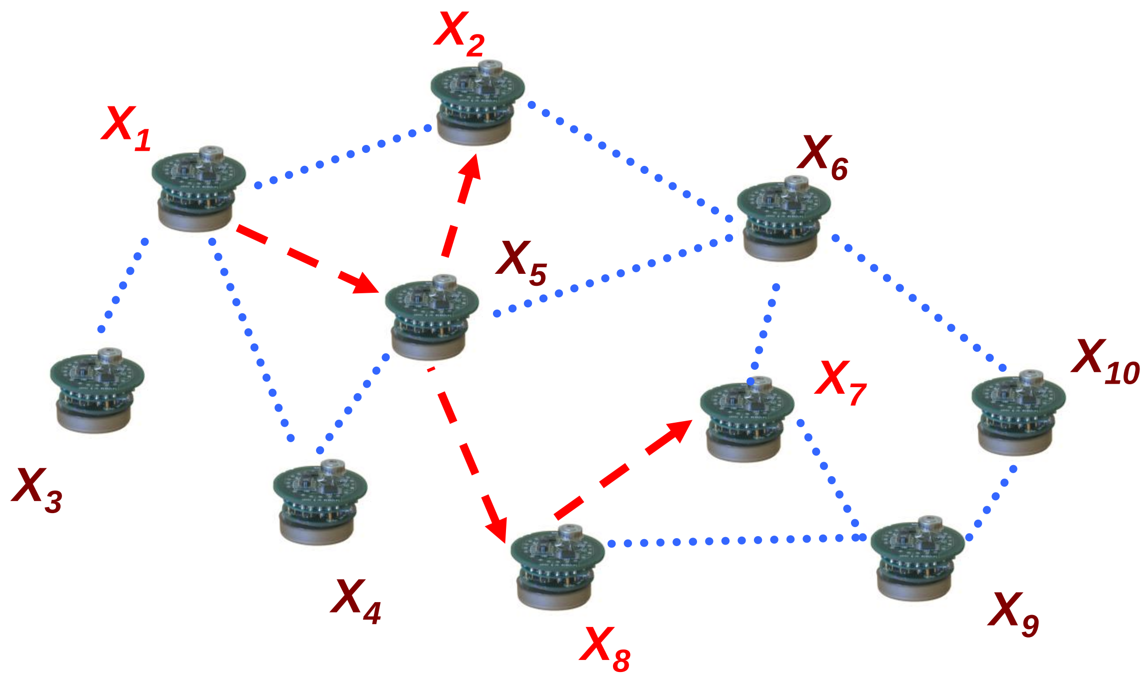


Complexity

- NP-Hard for general communication networks

Queries:





Optimal solution: *Minimum-weight (Steiner) tree connecting X1, X2, X7, X8*

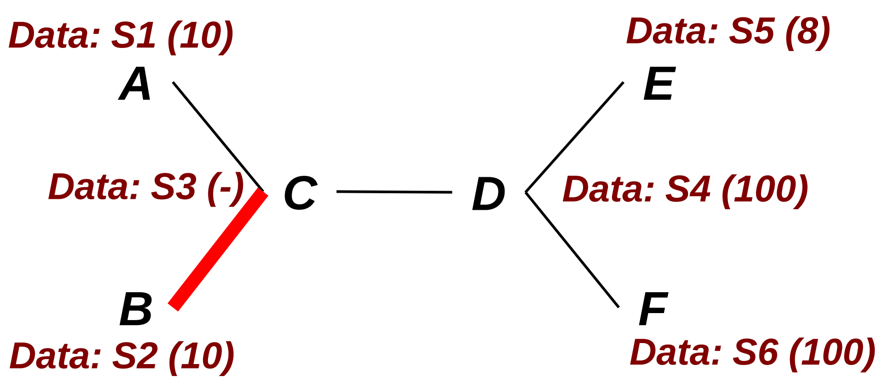
Outline

- Motivation & Problem Formulation
- Summary of Our Results
- Multiple Queries
  - NP-hardness
  - Polynomial time algorithm for tree communication networks
  - Approximation algorithms
- Experiments
- Future Work

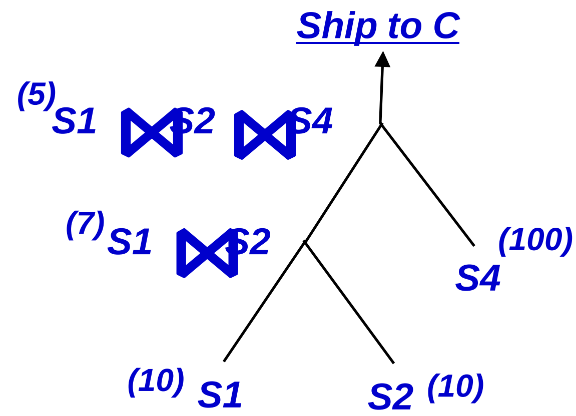
High-level Overview of Our Approach

1. Combine all the query plans into a single *hypergraph*
   - That explicitly captures the data movement sharing opportunities
2. For each edge, decide which data are communicated along that edge
   - By solving a *hypergraph min-cut* problem
3. Combine the local solutions into a single global solution

Steps 1 and 2: Single Query



Communication Network



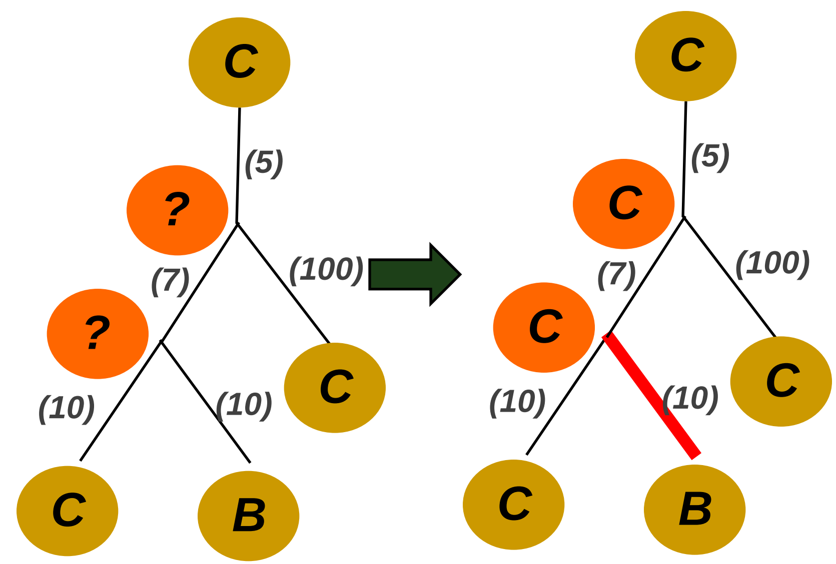
Query 1

Solving for edge (B, C)

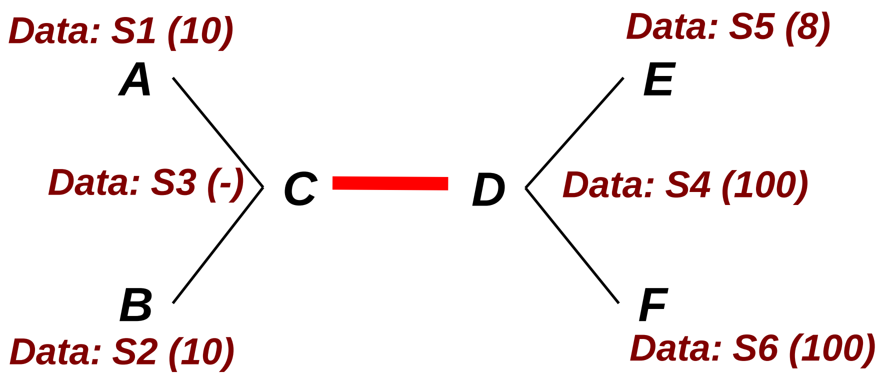
1. Label the nodes as B or C if possible, ? Otherwise
2. Solve a partition problem to resolve ?’s
3. “Cut” edges indicate the data movement

Solution

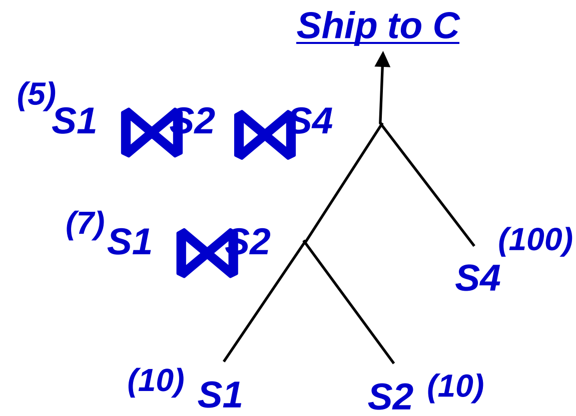
S2 moves across edge (B, C)



Steps 1 and 2: Single Query



Communication Network



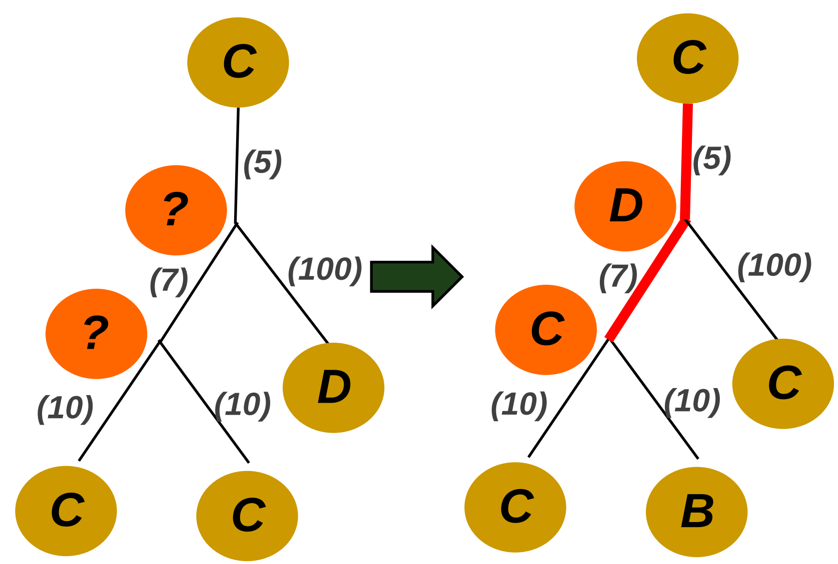
Query 1

Solving for edge (C, D)

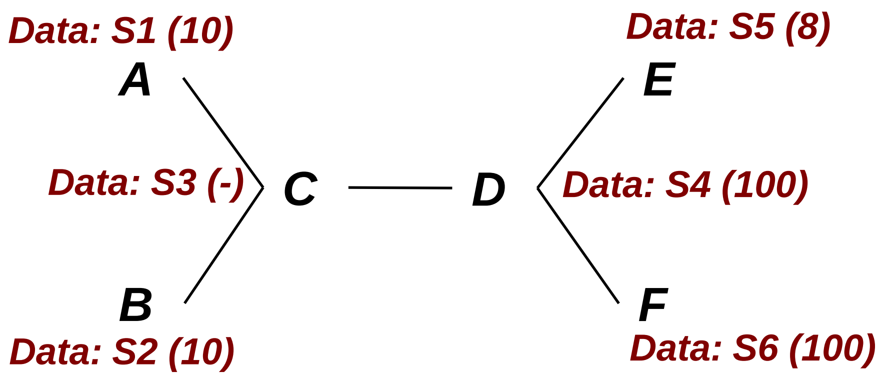
Solution

S1S2 moves from C to D

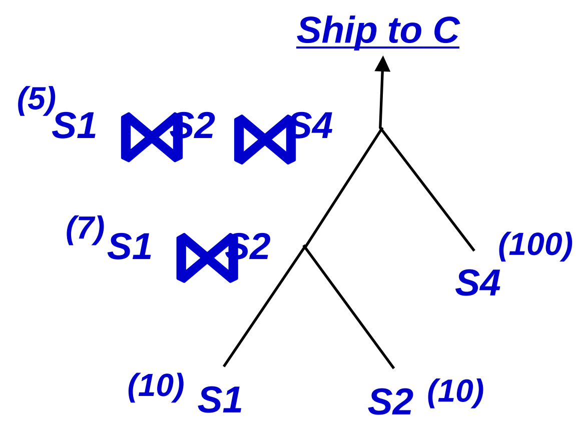
S1S2S4 moves from D to C



Step 3: Single Query



Communication Network



Query 1

Solution for (B, C)

S2 moves from B to C

Solution for (C, D)

S1S2 moves from C to D

S1S2S4 moves from D to C

Solution for (A, C)

…

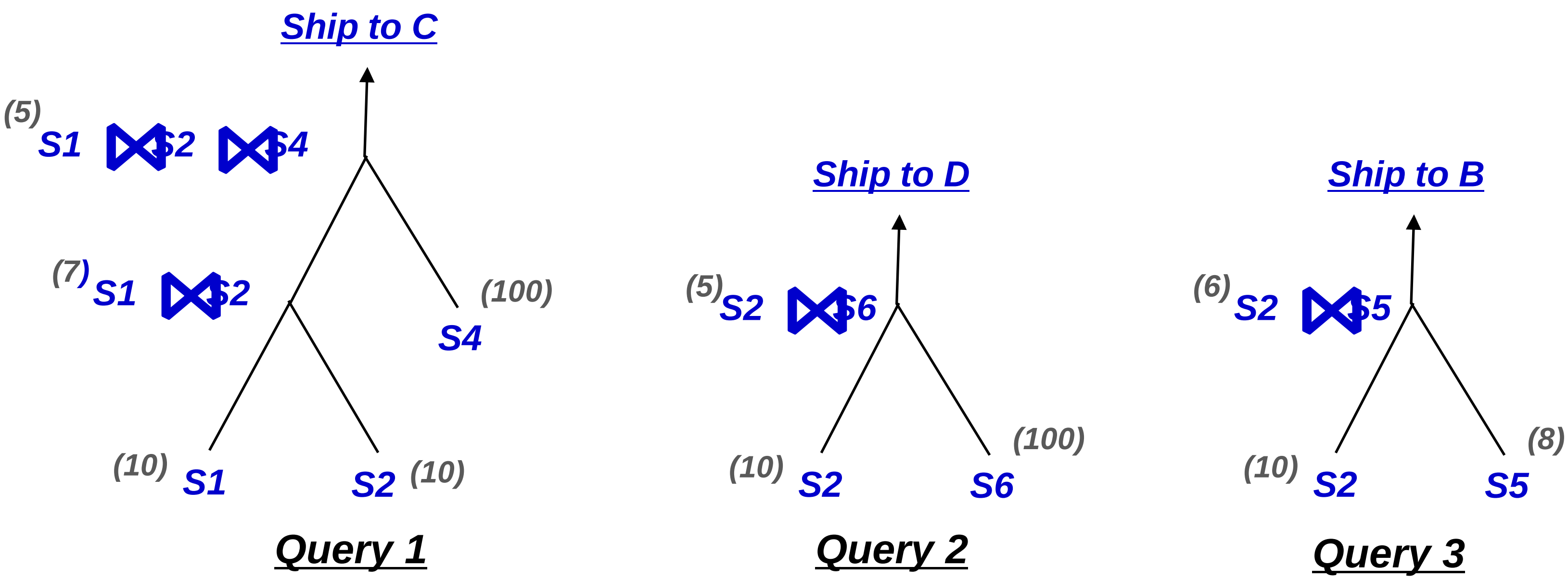
Key Question:

Are these movements consistent with each other ?

Answer:

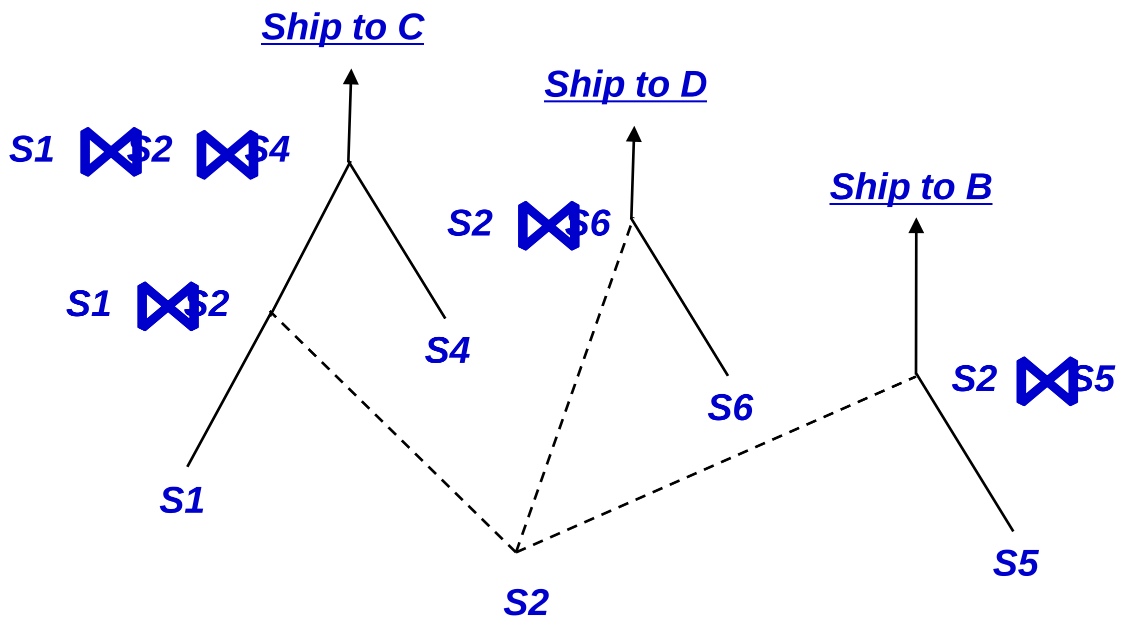
Yes, given unique min-cut solutions.

Multiple Queries



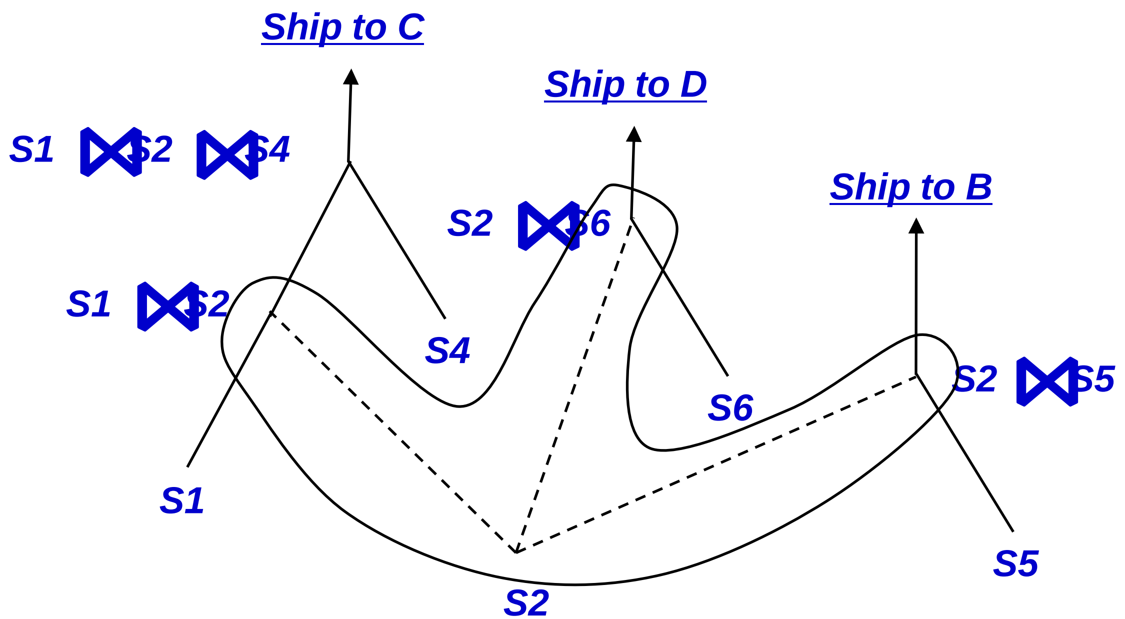
Multiple Queries

We create hyperedges



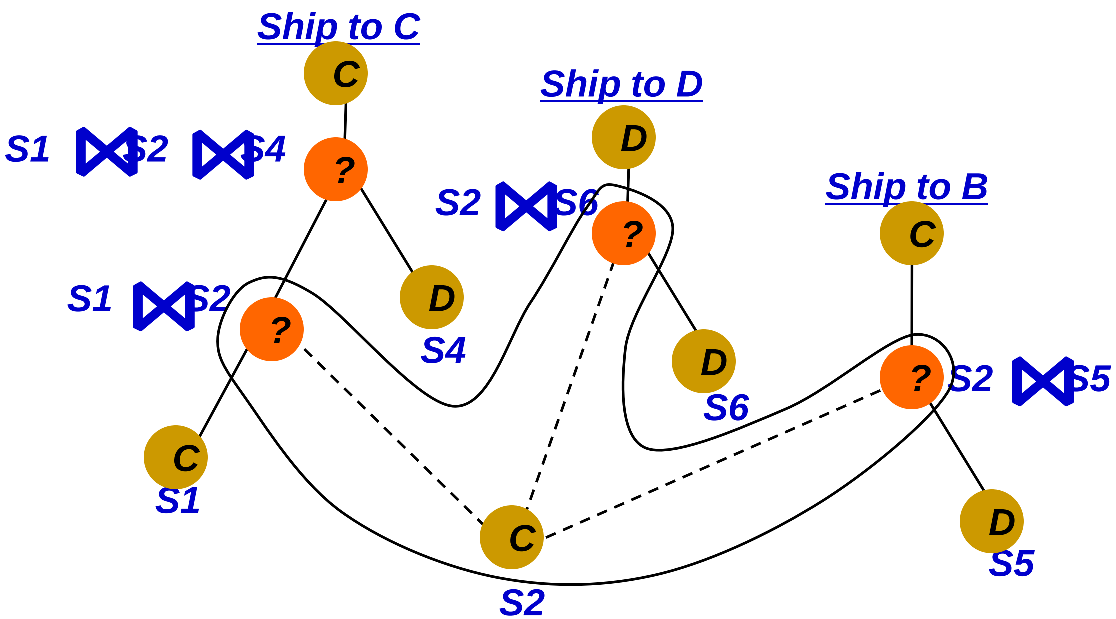
Multiple Queries

We create hyperedges



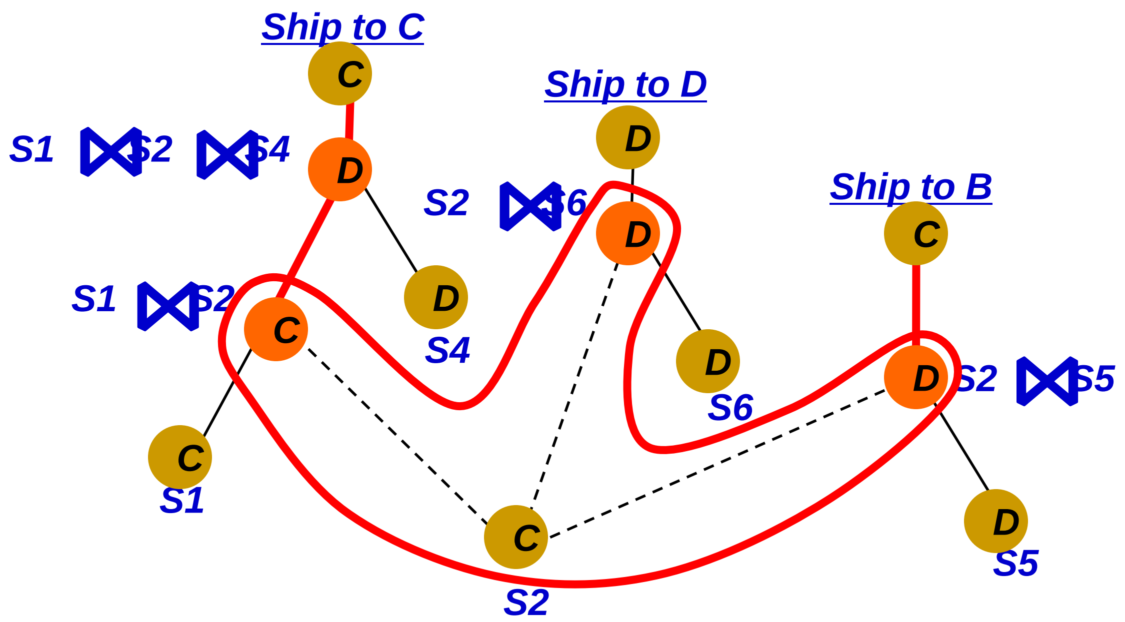
Multiple Queries

Solve for edge (C,D)



Multiple Queries

Solution for edge (C,D) : Hypergraph Partition



Why *hyperedges ?*

So we don’t count data movements multiple times

(e.g. Data item S2 above)

Multiple Queries

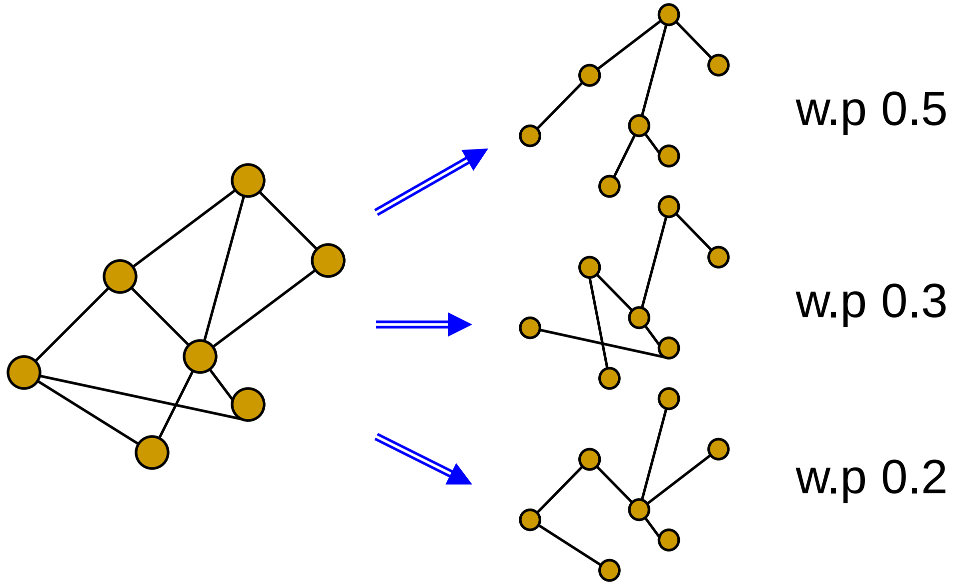
- Add hyperedges corresponding to shared data sources
- For each edge, solve a hypergraph partition problem, (which can be solved by min-cut algorithm)
- Again we can prove the consistency of these local movements
- Complexity: *m* max-flow min-cut computations where *m* is #edges in the tree

Outline

- Motivation & Problem Formulation
- Summary of Our Results
- Multiple Queries
  - NP-Hardness
  - Polynomial time algorithm for tree communication networks
  - Approximation algorithms
- Experiments
- Future Work

O(logn)-approximation for General Networks

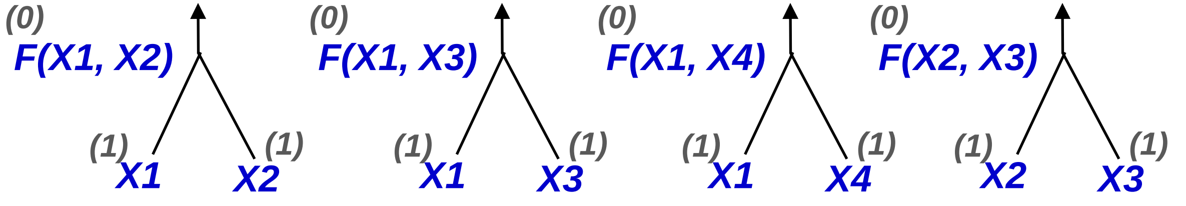
1. Construct a distribution of trees base on the communication network by using metric embedding [Fakcharoenphol/Rao/Talwar 06]
2. Randomly pick a tree and solve the problem on the tree optimally
3. Map the solution back to the original network



[FRT 06] Any metric can be embedded into a distribution of tree metrics with an O(log n)-distortion.

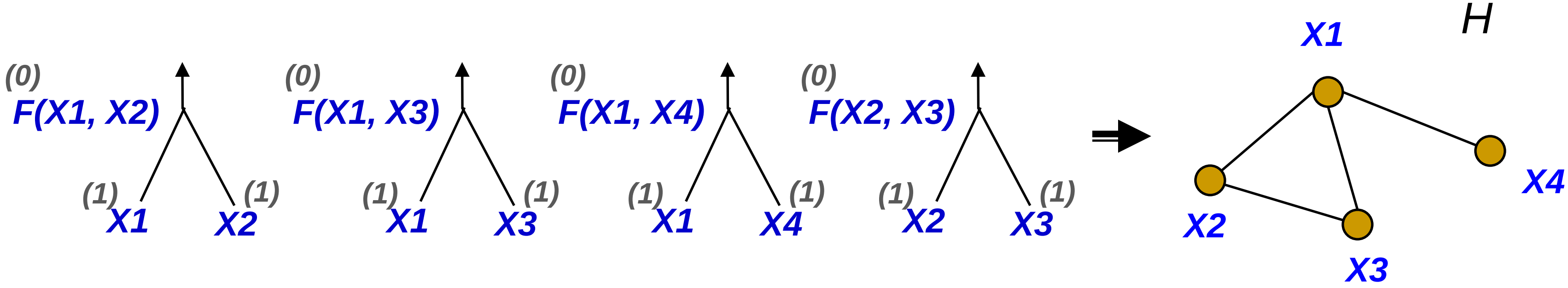
O(1)-approximations for some special cases

- “Pairs Problem”: Each query has only two data sources. The size of the result is zero.



O(1)-approximations for some special cases

- "Pairs Problem": Each query has only two data sources. The size of the result is zero.



- We can capture the queries by a graph *H*
  - *H* is a tree : *2p*
  - *H* is planar : *6p*
  - *Deg(H)*<= *D* : *D*

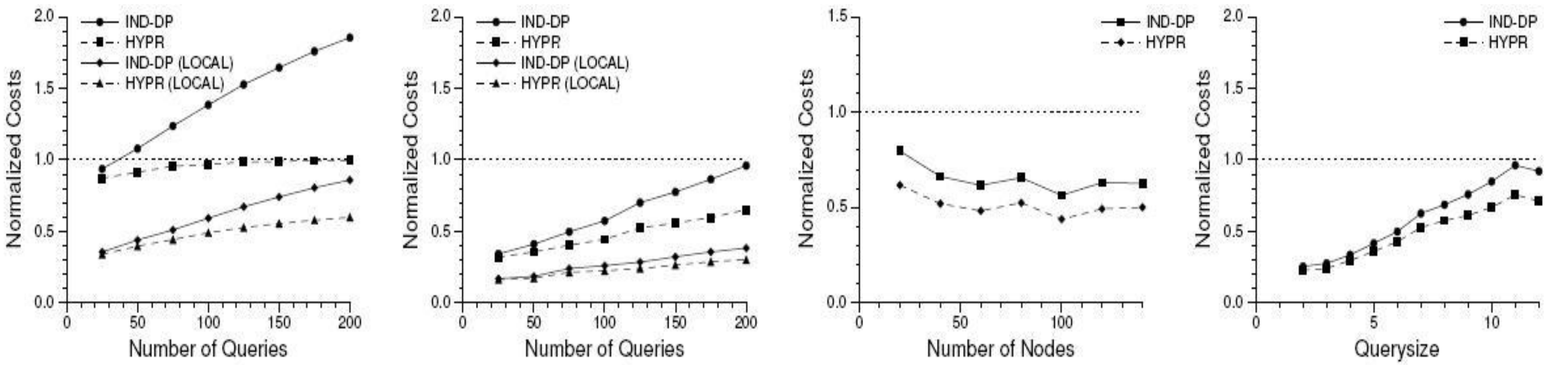
  Where *p* is the approroixmation ratio for minimum Steiner tree problem

Outline

- Motivation & Problem Formulation
- Our Results
- Multiple Queries
  - NP-hardness
  - On trees, Polynomial time Algorithm
  - Approximations
- Experiments
- Future work

Experiments

- IND-DP: optimize each query separately
- HYPR: the hypergraph min-cut approach



(i) Varying No. of Queries - Dataset 1 (ii) Varying No. of Queries - Dataset 2 (iii) Varying No. of Nodes - Dataset 2 (iv) Varying Max Query-Size - Dataset 2

Communication network: a spanning tree over a set of point randomly distributed in a 2-d plane

Datasets1: the sizes of sources are identical.
Datesets2: the sizes of sources are randomly chosen from a skewed distribution.
Workload: Each query is over a randomly chosen subset of sources.
LOCAL: all queries are chosen to be geometrically co-located sources

Future Directions

- Constant approximations for general communication networks
- Sharing intermediate results generated during query execution
- Online algorithms for handling new queries

Thanks