Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks

Huawei Miao

Outline

- Mobile Robot System
- Related Work
  - Communication Faults
  - Fault Tolerance
- Movement Control Algorithms
  - Globalized Algorithms
  - Localized Algorithms
- Simulation
- Conclusion
- Future work
- References

Mobile Robot System (1)

- Mobile robot system
  - Two or more robots establish links between one another
  - Robots communicate directly or through any series of transmission links

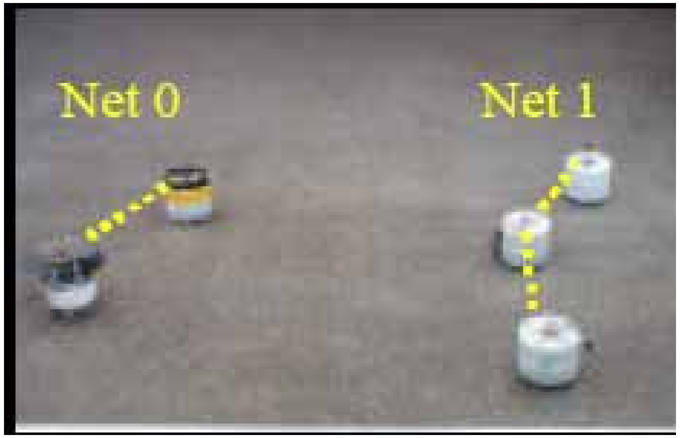
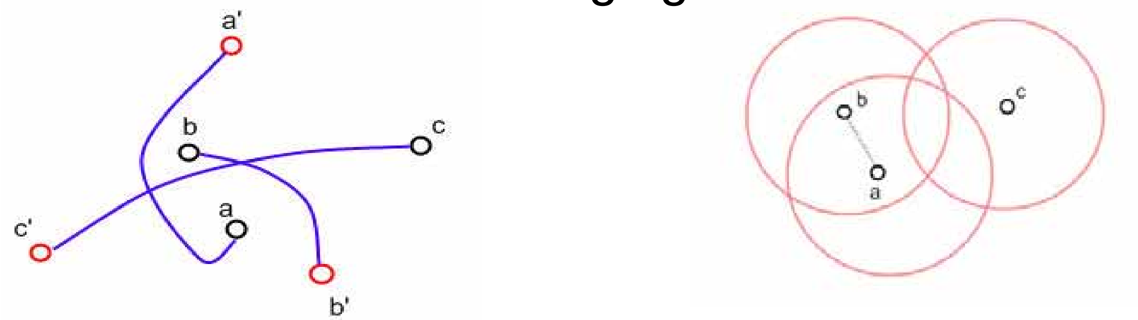


Figure 1: An network involving 5 robot

- Robots move towards goal locations
  - Leave networks
  - Enter different networks
  - Form new network

Mobile Robot System (2)

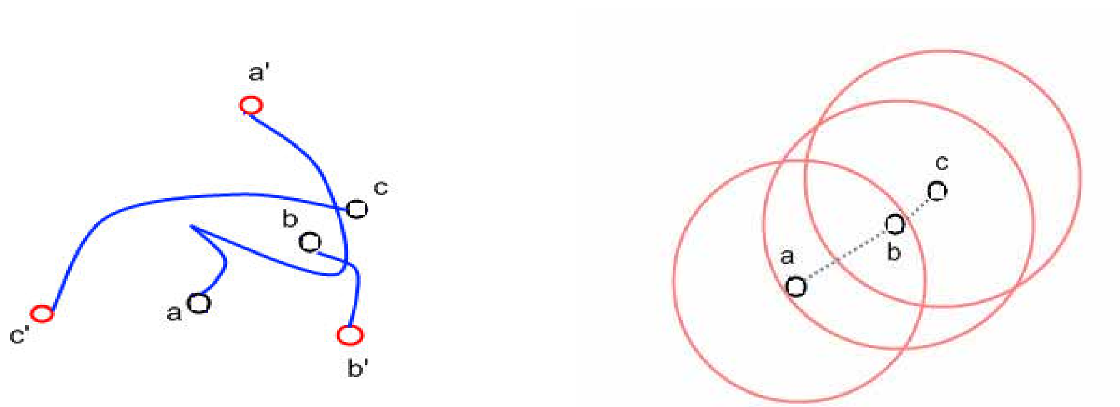
- Process of robot network changing



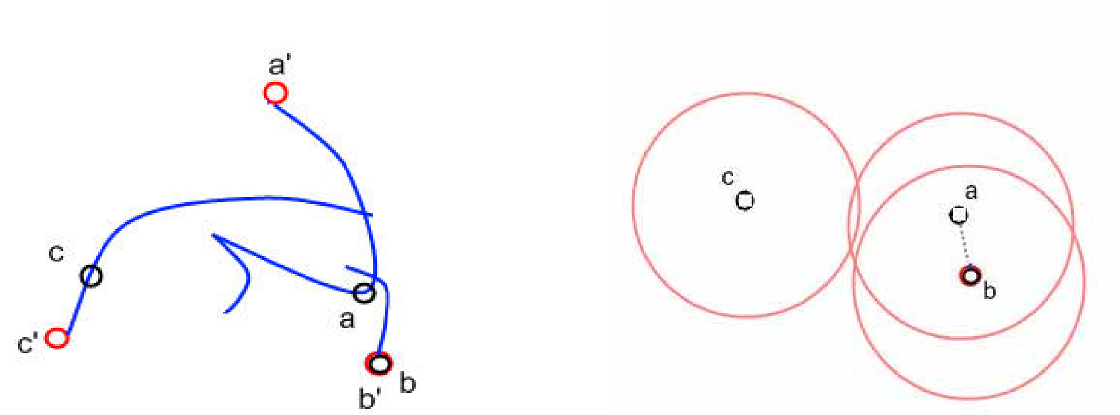
a) Robot a, b and c are following their initial paths respectively. Robot a and b are in one network.

b) Robot a, b and c all are moving to their goal location a', b' and c' with the paths. In this process, c moves into the network.

Mobile Robot System (3)



c) Three robots compute their new paths. A new plan comes out.



d) As robots continue along their new paths, they leave communication range of each other and some network connections are broken.

Mobile Robot System (4)

- Characters
  - Exist in unpredictable environments
  - Exist in constantly changing environments

- Mobile robot systems are ideal
  - Self-forming
  - Self-healing
  - Self-organizing

Communication Faults

- Communication faults in robot networks can be caused by
  - Hardware damage
  - Energy consumption
  - Harsh environment conditions
  - Malicious attacks
- A fault in a robot can cause
  - Stopping transmission tasks to others
  - Stopping relaying data to sink
  - Data sent by a robot will be lost if the receiving robot fails

Basic Conception (1)

- Mobile robot network can be represented by graph
  - Vertex represents a network node
  - Edge denotes a communication link between a pair of robots

- Bridge
  - Edge whose removal disconnects the graph

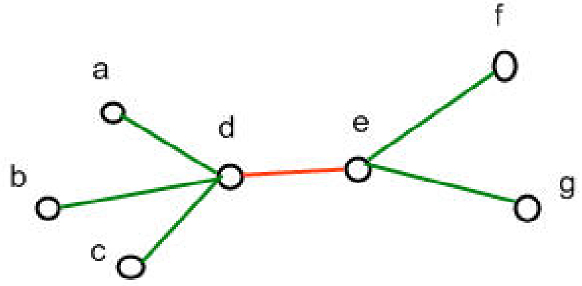


Figure 3: An example of bridge

Basic Conception (2)

- Critical node
  - Graph is disconnected without this node
- Bi-connected graph
  - Remains connected after removing any of its vertices
  - Contain no critical node

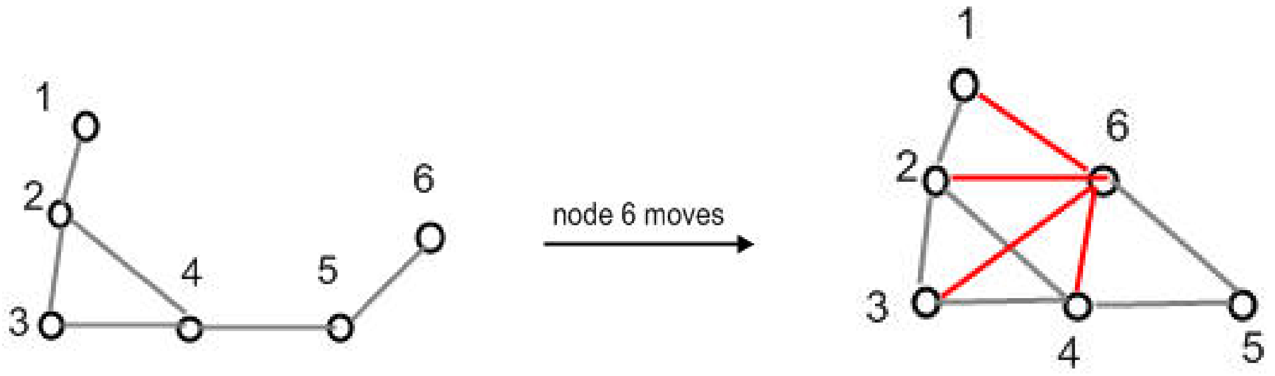


Figure 4: Achieving bi-connectivity by node movement

Fault Tolerance Design

- The robot network is assumed to be
  - Connected
  - Not necessarily bi-connected
  - All nodes maintain canonical network state information
  - All nodes execute the same algorithm
- Fault tolerance
  - Should be at least two node-disjoint paths between each pair of robots
  - Can be achieved by making the communication network bi-connected
- Minimize the total (or average) distance moved by nodes in the network

Movement Control Algorithm

- The goal is to move robot nodes
  - From: an arbitrary initial connected configuration
  - To: bi-connected communication network
  - Use: moving a subset of robot nodes
  - Minimize the total distance travelled by all the robots

- Movement control algorithms
  - Globalized Algorithms
  - Localized Algorithms

Globalized Algorithms

- Globalized movement control
  - Each node shares the knowledge about the rest of the network
  - Each robot is aware of global network topology
  - Robots decide on their new position based on the topological information and then create a fault-tolerant network

- Globalized algorithms
  - One-Dimensional Case
  - Two-Dimensional Case
    - Contraction Algorithm
    - Block Movement Algorithm

One-Dimensional Case (1)

- The one-dimensional case is described as
  - N nodes lie in a straight line
  - Nodes are allowed to move only in two directions along the line

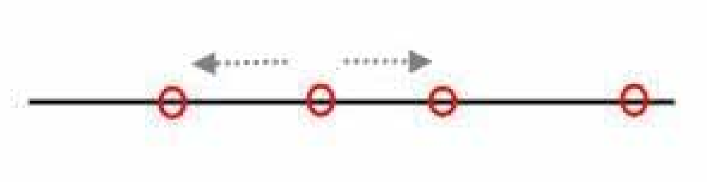


Figure 5: An example of 1D case with 4 nodes in a line

  - Initial positions of the nodes are given by $p_i \in R,\ 1 \le i \le N$
  - Each node's transmission range is 1.0
  - Final configuration given by positions $x_i \in R,\ 1 \le i \le N$ is bi-connected
  - Minimize the total distance moved by nodes

One-Dimensional Case (2)

- Formulate the problem as

Minimize $D_{total} = \sum_{i=1}^{N} |x_i - p_i|$

$$x_1 \geq p_1 \quad (1)$$

$$x_N \leq p_N \quad (2)$$

$$x_i - x_{i-1} \geq 0, 2 \leq i \leq N \quad (3)$$

$$x_i - x_{i-2} \leq 1, 3 \leq i \leq N \quad (4)$$

One-Dimensional Case (3)

- Geometric meaning of the constrains above
  - (1) and (2): network will compress in length after bi-connectivity

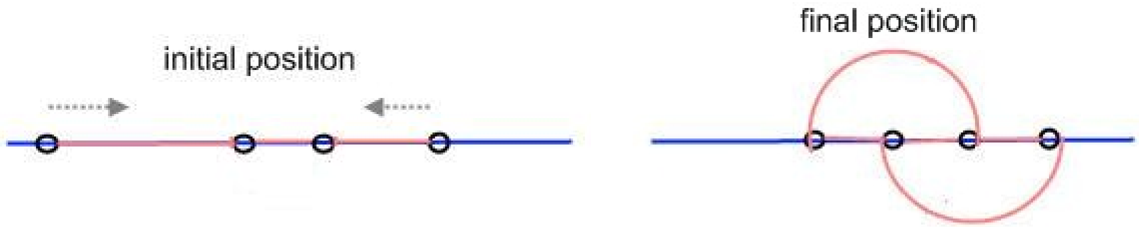


Figure 6: Initial and final positions of 1D network

  - (3): no node needs to move past its neighbors to achieve bi-connectivity

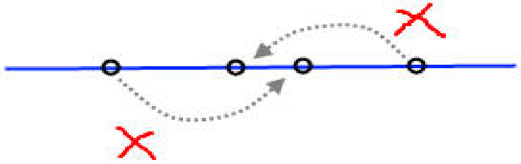


Figure 7: Movement path error

  - (4): every alternate pair of nodes are within transmission range

Contraction Algorithm (1)

- Every robot node
  - Floods a link state update (LSU) to the rest of the network
  - Extracts the location information from all other nodes‘ LSUs
  - Calculates the geographic center C for the entire network

$$\vec{C} = \frac{1}{N}\sum_{i=1}^{N}\vec{p}_i$$

  - Each node j independently moves toward $\vec{C}$ by a weighted distance

$$(1-\alpha)\left\|\vec{C}-\vec{p}_j\right\|$$

  - The total distance is given by $D_{total} = \sum_{i=1}^{N}\left\|\overrightarrow{x_i} - \overrightarrow{p_i}\right\|$

Contraction Algorithm (2)

- Basic idea of contraction algorithm

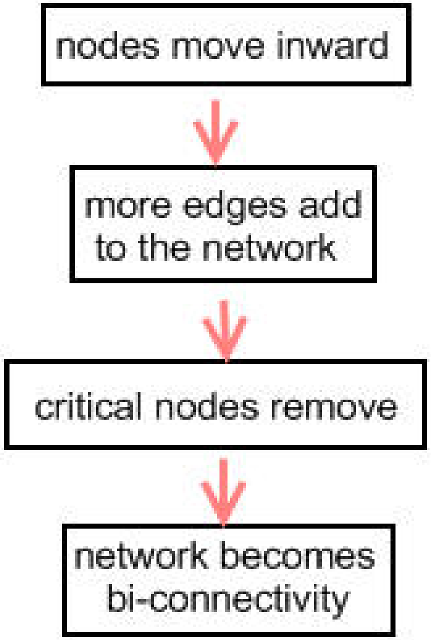


Figure 8: The rationale of contraction algorithm

Block Movement Algorithm (1)

- Bi-connected block tree (BT)
  - Bi-connected components of a graph are identified with critical nodes
  - Corresponding block tree's vertices are bi-connected components (or blocks) and critical nodes
  - Two critical nodes are connected by a bridge, the corresponding block contains no nodes

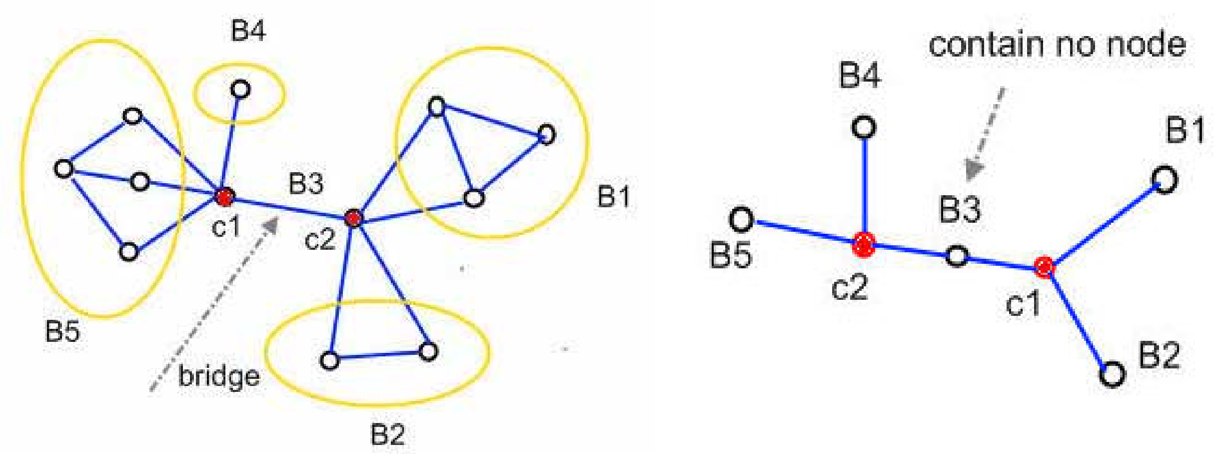


Figure 9: a) Decomposition of a network into bi-connected components
b) The corresponding block tree

Block Movement Algorithm (2)

- Basic idea of block movement algorithm
  - Divide a network into bi-connected blocks
  - The network is a block tree of these blocks
  - Iteratively merging the blocks to form a single bi-connected block

- Which blocks to move
  - Conception
    - Root of BT
    - Leaf block
    - Parent block
    - Empty parent block
    - Parent critical node
    - Parent critical node of parent block

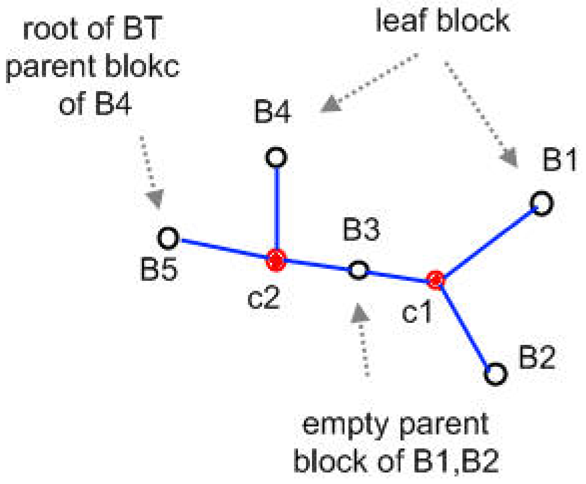


Figure 10: Conception of BT

Block Movement Algorithm (3)

- Which blocks to move (cont.)
  - Block movement strategy
    - Unreasonable movement

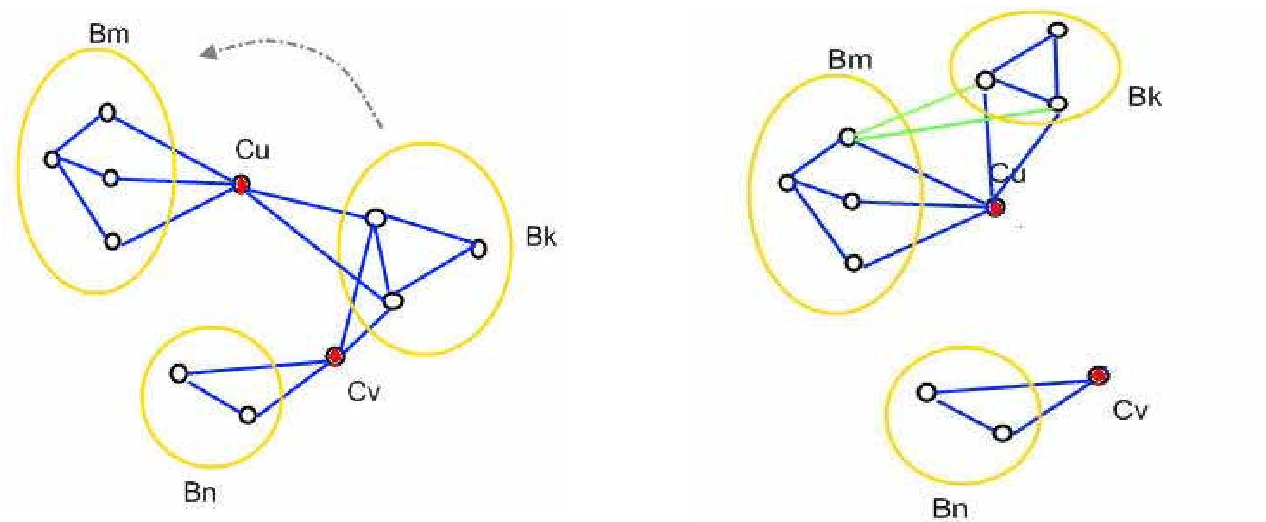


Figure 11: Block movement error

Block Movement Algorithm (4)

- Which blocks to move (cont.)
  - Block movement strategy (cont.)
    - Each leaf block moves toward its parent block
    - Each block moves toward the nearst node in the parent block
    - Parent block is empty, moves toward the parent critical node of parent block

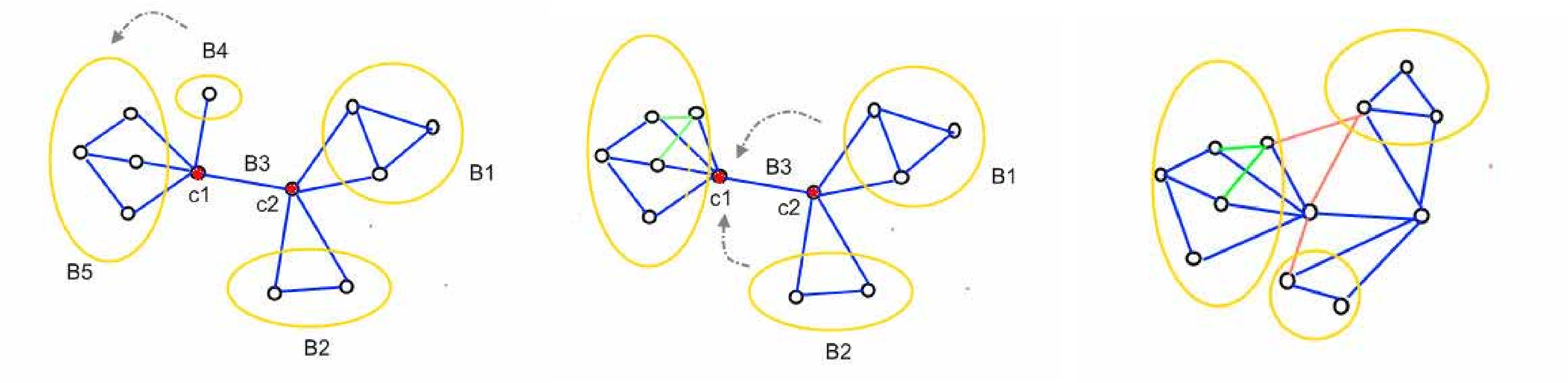


Figure 12: Execution of the block movement algorithm
a) Initial configuration   b) After iteration   c) Final (bi-connected) configuration

Block Movement Algorithm (5)

- Special case to the block movement

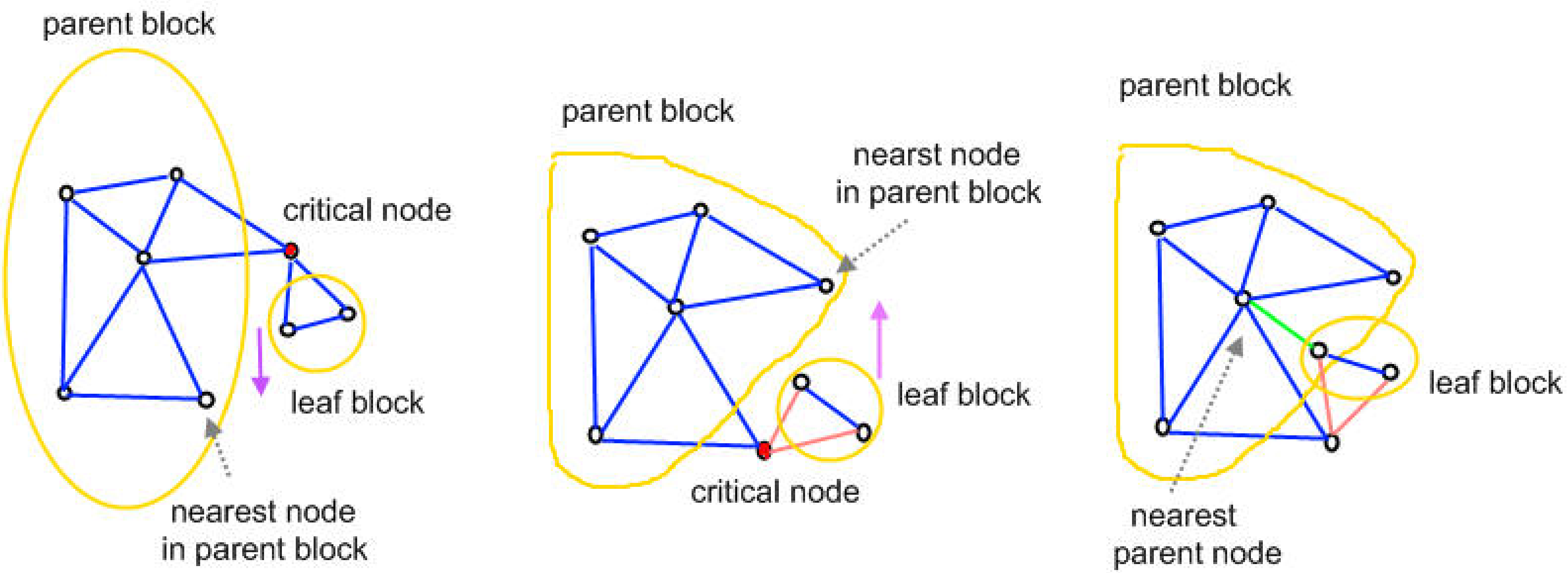


Figure 13: An exception to the block movement scheme and solution

  - Leaf block oscillates between two nodes
  - Translating the block toward the nearest parent node

Block Movement Algorithm (6)

- Characters of algorithm shown in the paper
  - Globalized robot movement control
  - Each mobile robot is assumed to be aware of global network topology
  - Require accurate and global information of entire network
  - Be applicable to only small size network
- For the large scale networks
  - Global network information is hard to obtain and maintain
  - Total distance of movements increase rapidly
  - Total communication overhead on robots increase rapidly

Localized Algorithms

- Localized movement control
  - Use p-hop neighbor information
  - Identify p-hop critical node
  - Control movement of nodes to let the network bi-connectivity
- Localized algorithms
  - Critical node without critical neighbor (Case I)
  - Critical node with one critical neighbor (Case II)
  - Critical node with several critical neighbors (Case III)

Critical Node without Critical Neighbor

- Basic idea
  - Distance between two neighbors is d, communication range is r
  - Select two neighbors from two disjoint sets
  - Move them towards each other until they become neighbors
  - Each node should move (d-r)/2 to reach each other
  - Select two neighbors with the minimum distance d among all possible pairs in the two sets

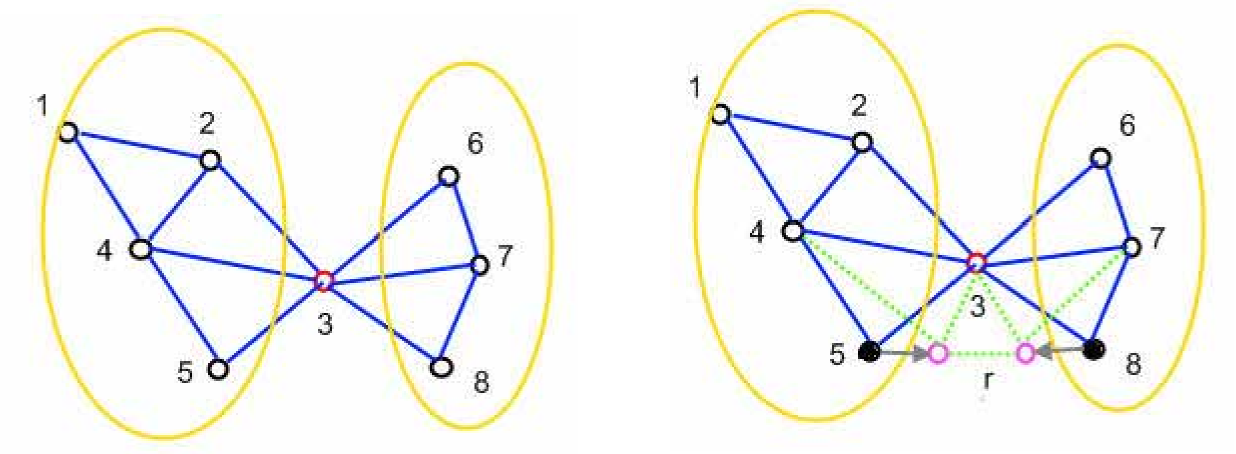


Figure 14: Node 3 is the only critical node and its 2-hop sub-graph

Critical Node with One Critical Neighbor

- Basic idea
  - Node ID : assign priorities to critical nodes
  - Critical node who has larger ID leads movement control
  - Larger ID node select one non-critical neighbor to move toward the other critical node (with smaller ID)
  - Selected neighbor is the nearest to the smaller ID node
  - Problem becomes critical node without critical neighbor (Case I)

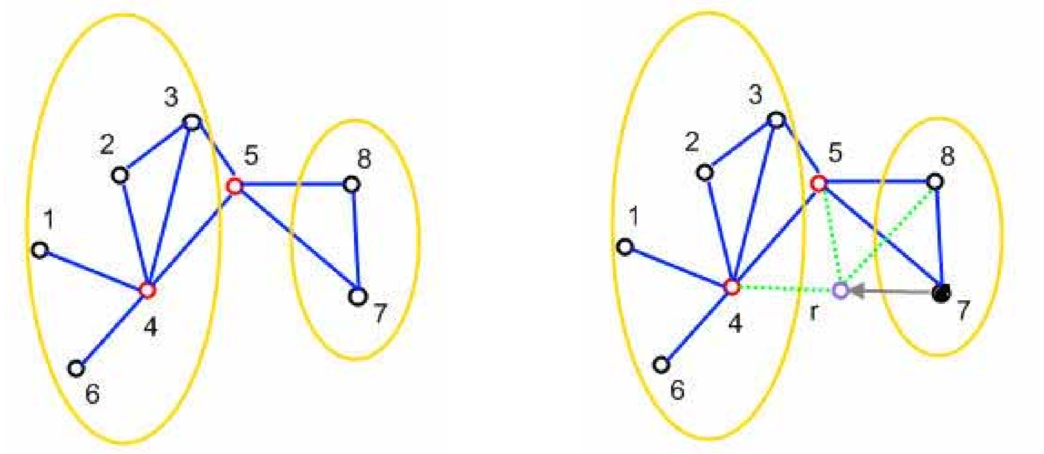


Figure 15: Node 5 with larger node ID and its 2-hop sub-graph

Critical Node with Several Critical Neighbors (1)

- A critical node is
  - Available: it has non-critical neighbors
  - Non-available: otherwise
  - Critical head: available and ID larger than any available critical neighbor or has no available critical neignbors

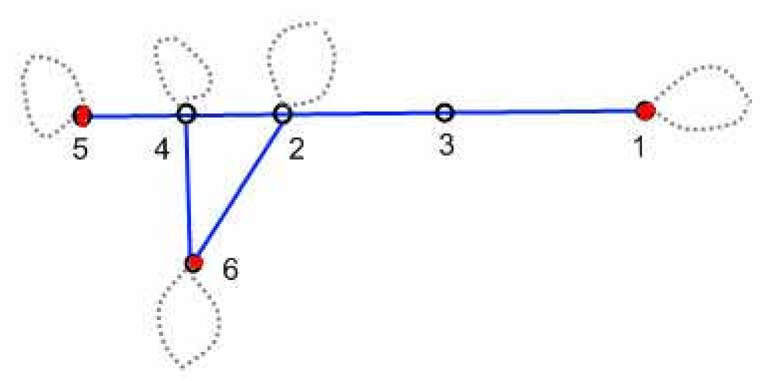


Figure 16: Critical node with several critical neighbors

Critical Node with Several Critical Neighbors (2)

- Basic idea
  - Each critical head selects one of its critical neighbors to pair with
  - To be deterministic
    - Available critical neighbor (if any) with largest ID is selected
    - Otherwise non-available critical neighbor with the largest ID
  - Call case II of the movement control
  - Pair-wise merging continues until all critical nodes become non-critical
  - No action will be taken if there are no critical heads in the network

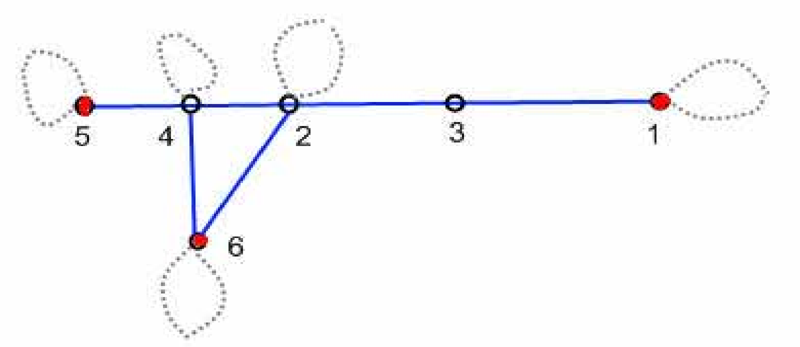


Figure 16: Critical node with several critical neighbors

Critical Node with Several Critical Neighbors (3)

- An illustrative example

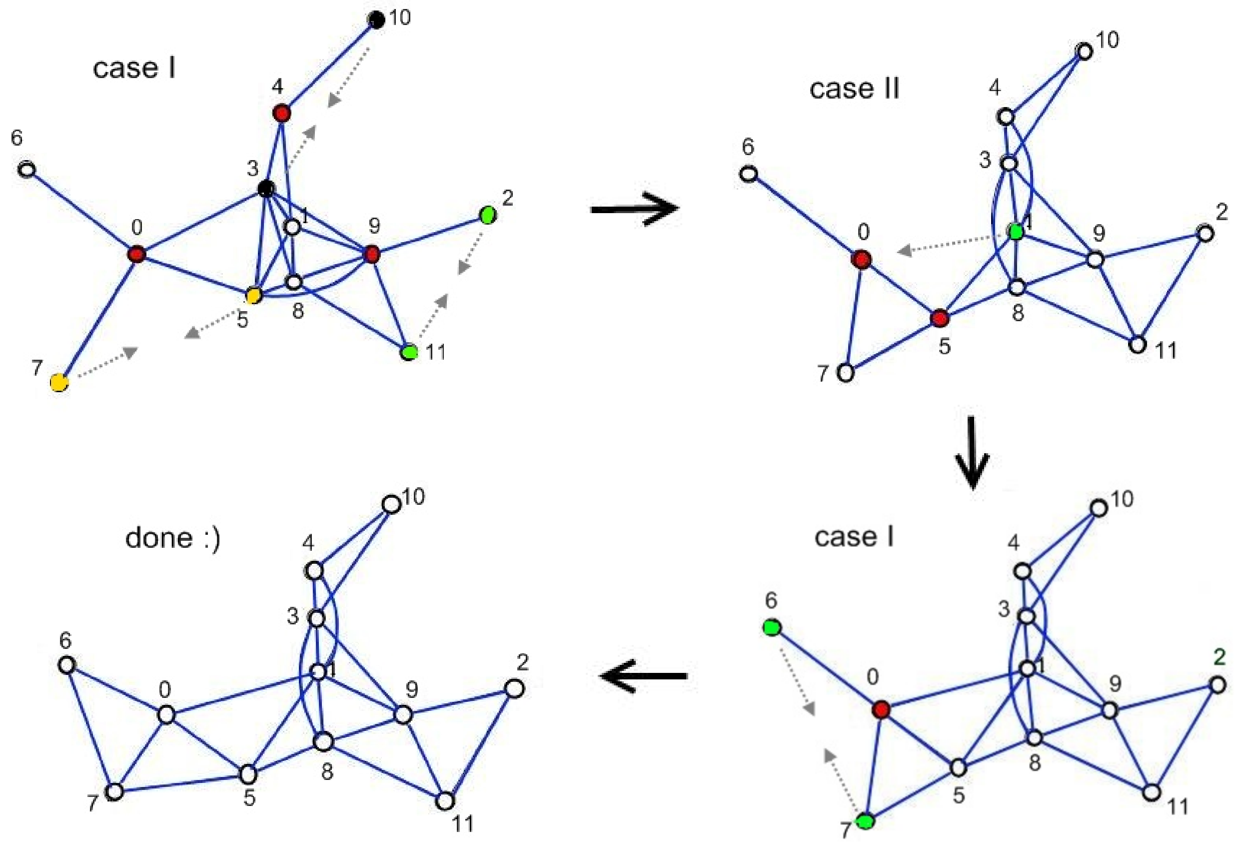


Figure 17: An example of case III

Simulation for Globalized Algorithms (1)

- Performance evaluation of 1D network
  - Simulation environment
    - up to 200 collinear nodes
    - With transmission range 1.0
    - Starting positions randomly
  - Use total distance traveled metric (D_{total})

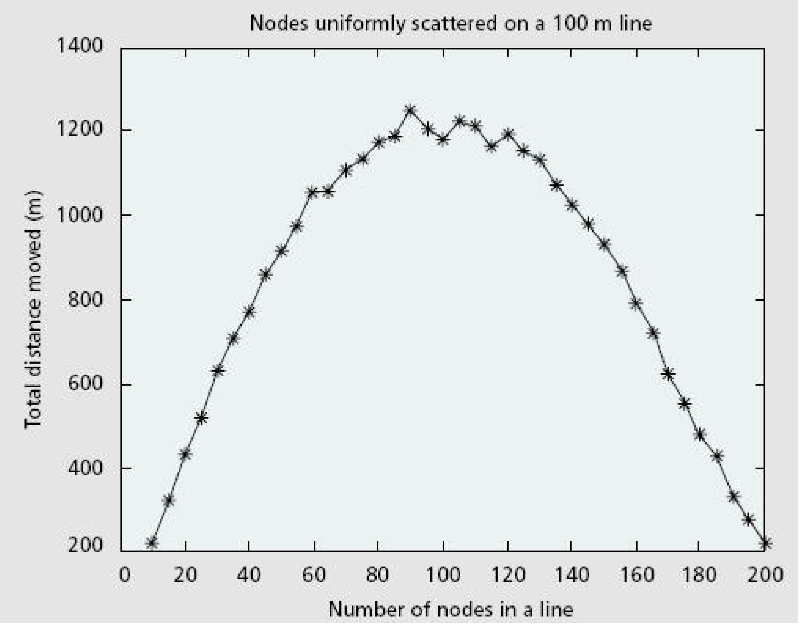


Figure 18: Total distance moved by nodes (1D networks)

Simulation for Globalized Algorithms (2)

- Performance evaluation of 2D network
  - Simulation environment
    - 1km*1km square area
    - Up to 50 robots randomly distributed
    - With transmission range 250m
    - Ground assumed to be flat without obstacles and trenches
  - Use total distance traveled metric (D_{total})

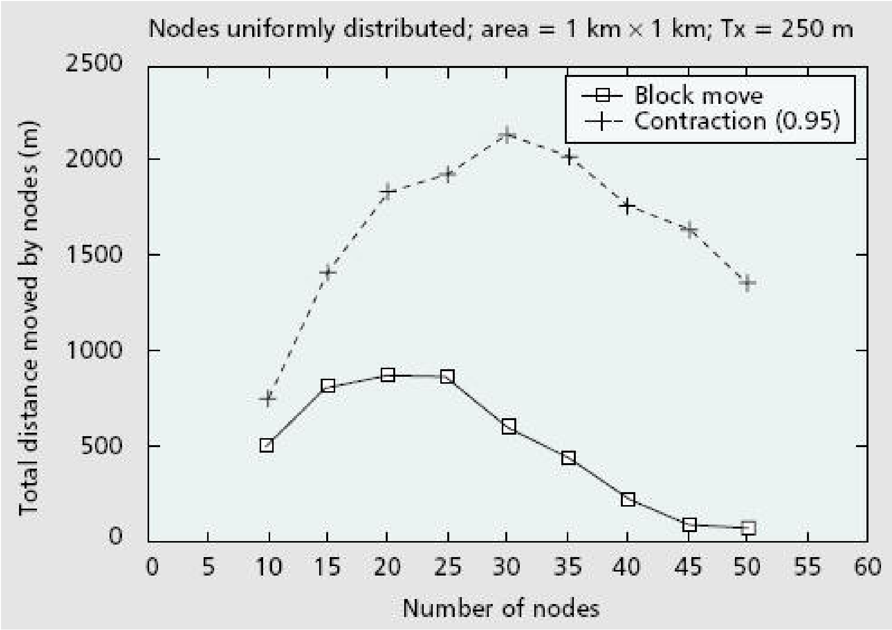


Figure 19: Total distance moved by nodes (2D networks)

Simulation for Localized Algorithms (1)

- Performance evaluation of 2D network
  - Simulation environment
    - Area from 300m^2 to 3000m^2
    - With communication range 10m
    - Network density of d=10 (an average of 10 neighbors per node)

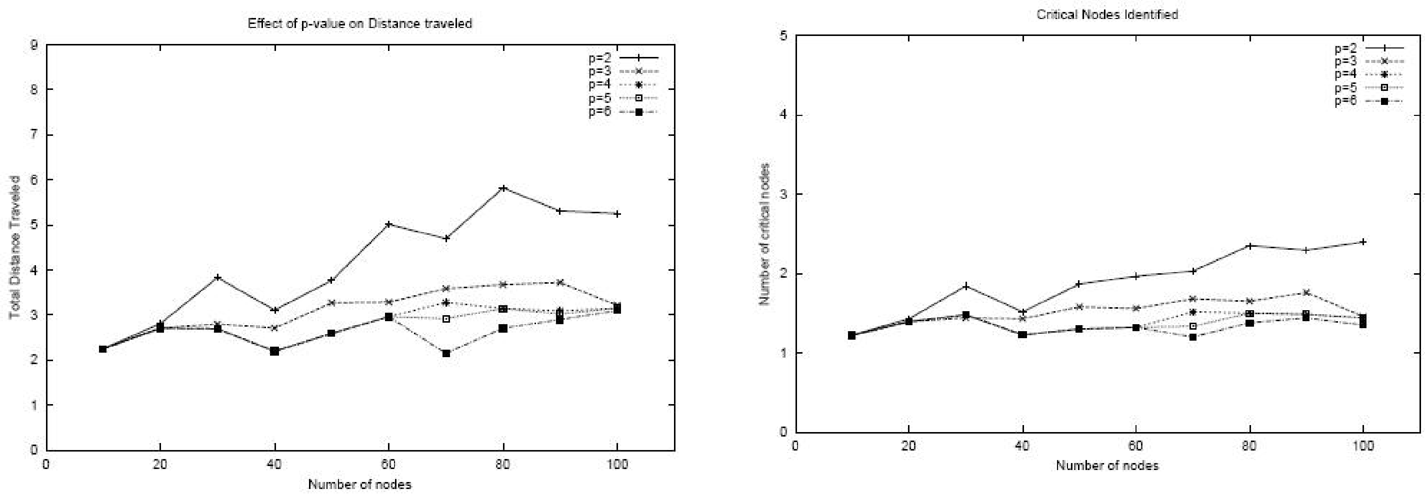


Figure 20: Total distance moved and number of critical nodes for various of p and n, for d=10

Simulation for Localized Algorithms (2)

- Localized algorithm is not always successful for sparse networks

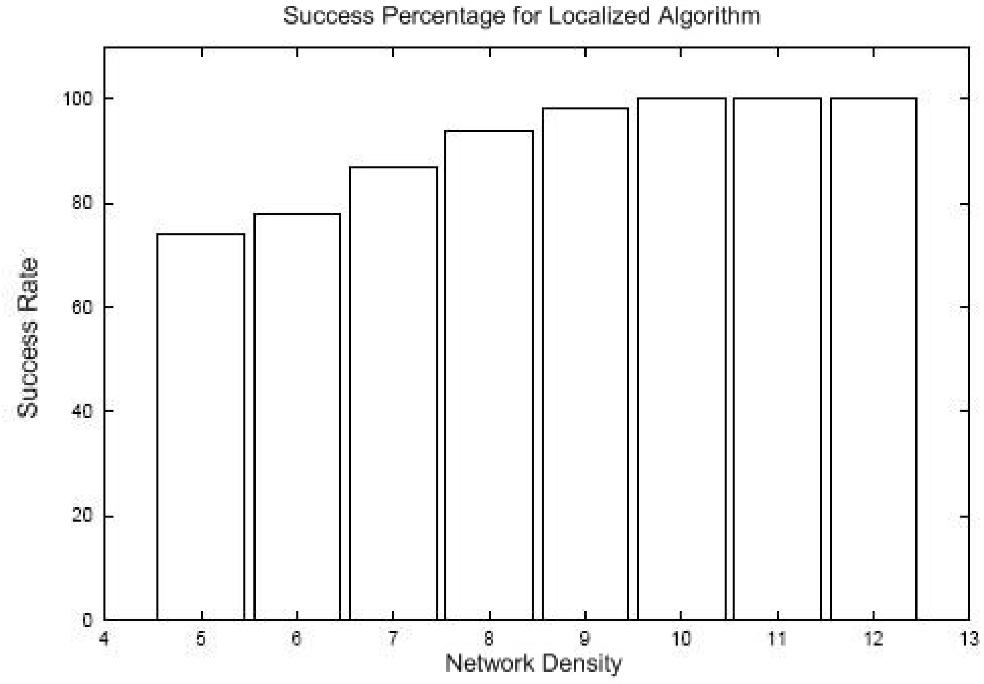


Figure 21: Percentage rate of success of localized algorithm on networks of various densities (fixed size=100)

Comparison

- Critical node identification by two algorithms
- Total distance moved by two algorithms (average case)

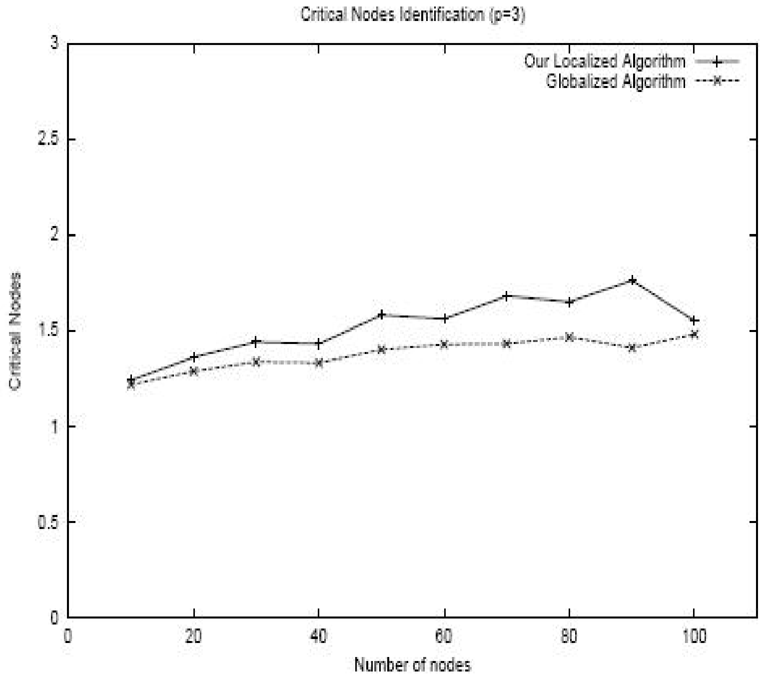


Figure 22: Critical nodes comparison

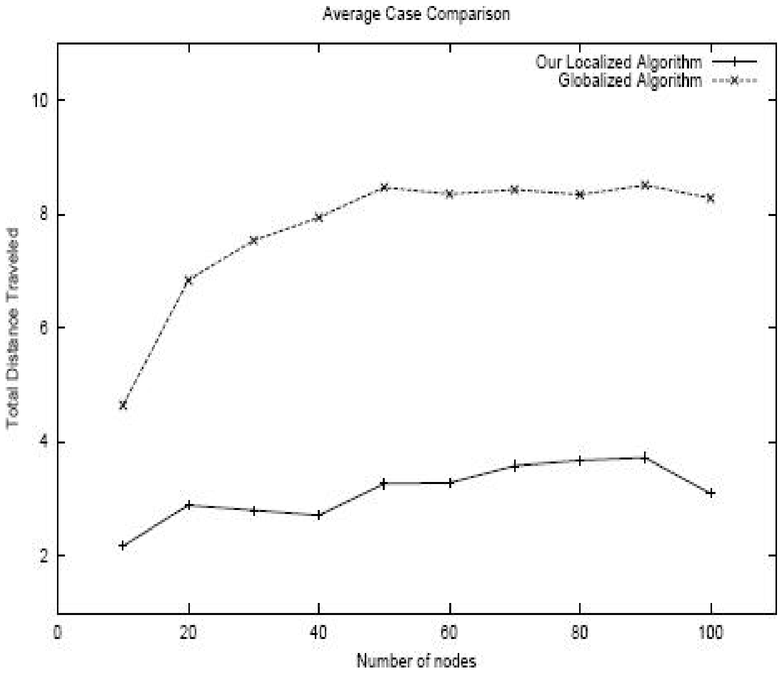


Figure 23: Total distance moved comparison

Conclusion (1)

- 1D network problem can be solved by globalized algorithm in polynomial time by applying LP techniques
- 2D network problem can be solved both by globalized and localized algorithms
- Using globalized algorithms, iterative block movement algorithm significantly outperforms the contraction heuristic in the total distance traveled metric

Conclusion (2)

- Localized movement control algorithm significantly outperforms its globalized counterpart

- In most cases, information about 3-hop neighbors only is sufficient to convert the network to a bi-connected one in an efficient manner

- Global information about the network is not necessary to achieve bi-connectivity

Future work

- If there exists any localized algorithm that guarantees bi-connectivity starting from any connected network

- Constructing a connected and fault-tolerant network starting from a disconnected network

- Preserve area coverage and certain functionalities, while attempting to bi-connect

References

- Main paper

P. Basu and J. Redi
Movement control algorithms for realization of fault-tolerant ad hoc robot networks
IEEE Network, 18(4):36-44, 2004

- Additional paper

Shantanu Das · Hai Liu · Amiya Nayak · Ivan Stojmenovi´c
A Localized Algorithm for Bi-Connectivity of Connected Mobile Robots
Telecommunication Systems

Thanks :)

Questions??