Parameterized Distributed Algorithms*

R. Krithika

Indian Institute of Technology Palakkad

Recent Trends in Algorithms

National Institute of Science Education and Research

Based on the manuscript titled Parameterized Distributed Algorithms by Ran Ben-Basat, Ken-ichi Kawarabayashi, Gregory Schwartzman [arXiv:1807.04900]

Parameterized Algorithms

* Multidimensional analysis of the running time
  * Effect of secondary measurements on complexity
  * NP-hard problem: Exponential factor in running time is restricted to a parameter instead of input size

Instance: A graph G on n vertices and integer k
Question: Does G have a solution of size k?

Parameter: k

Design f(k) poly(n) algorithm

$2^{O(k^2)}$ poly(n) $2^{O(k \log k)}$ poly(n)

fixed-parameter tractable (FPT)
or
parameterized algorithm

Parameterized Algorithms

- Multidimensional analysis of the running time
  - Effect of secondary measurements on complexity
  - NP-hard problem: Exponential factor in running time is restricted to a parameter instead of input size

<u>Instance:</u> A graph G on n vertices and integer k
<u>Question:</u> Does G have a solution of size k?

<u>Parameter:</u> k

Design f(k) poly(n) algorithm

$2^{O(k^2)}$ poly(n) $2^{O(k \log k)}$ poly(n)

fixed-parameter tractable (FPT)
or
parameterized algorithm

Model of Computation: Single Processor

Distributed Algorithms

Distributed Algorithms

* A network (graph) of n processors (nodes) perform computation

Distributed Algorithms

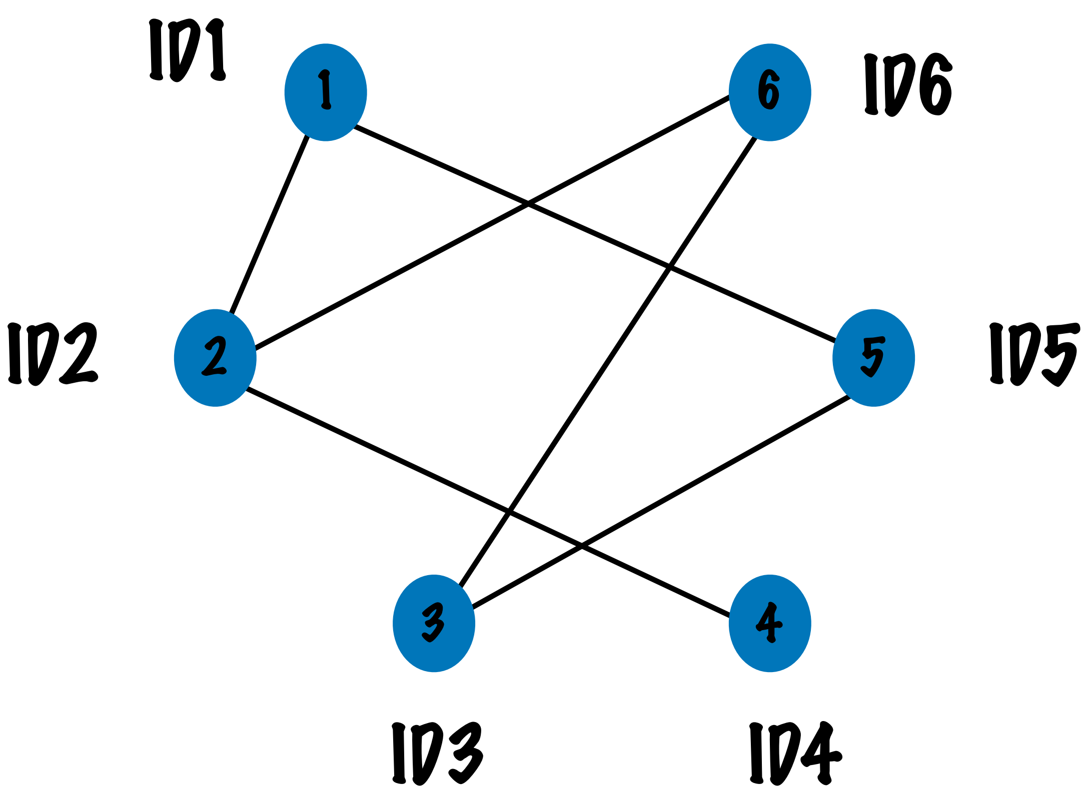
- A network (graph) of n processors (nodes) perform computation
  - Each node has an identifier ($O(\log n)$ bits) and local information (neighbours)

Distributed Algorithms

* A network (graph) of n processors (nodes) perform computation
    * Each node has an identifier (O(log n) bits) and local information (neighbours)
        * Each node knows about its incident edges and may not know anything about the ids of its neighbours

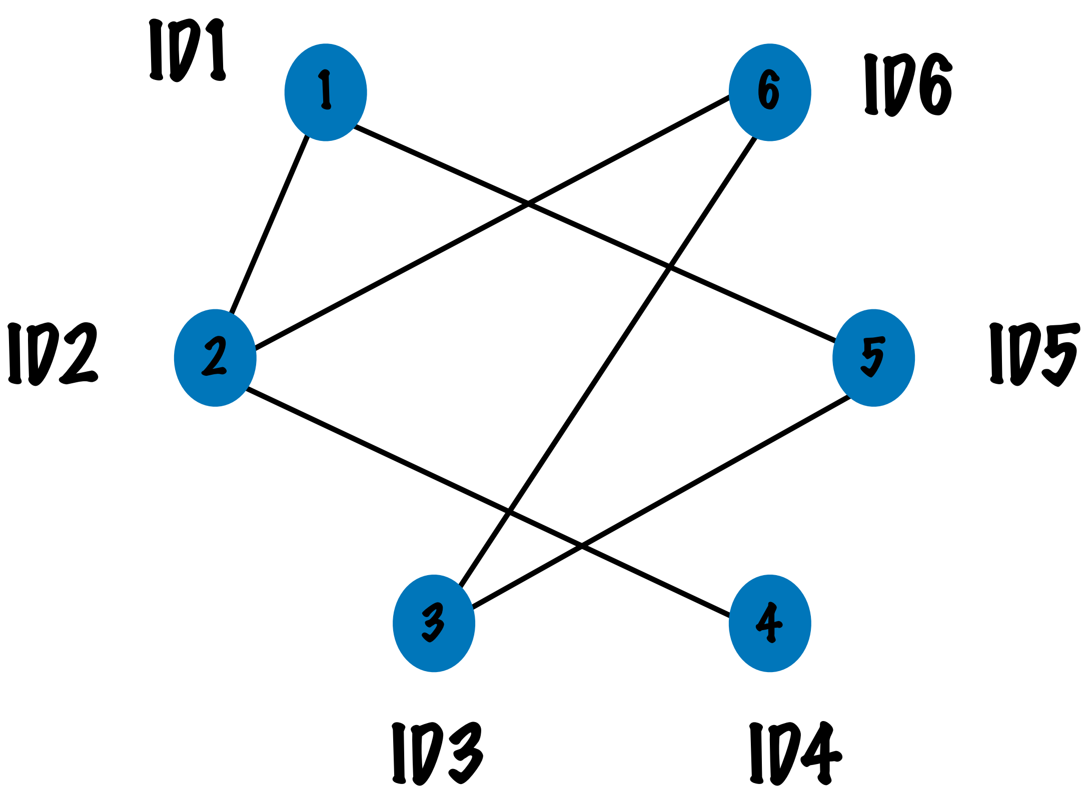
Distributed Algorithms

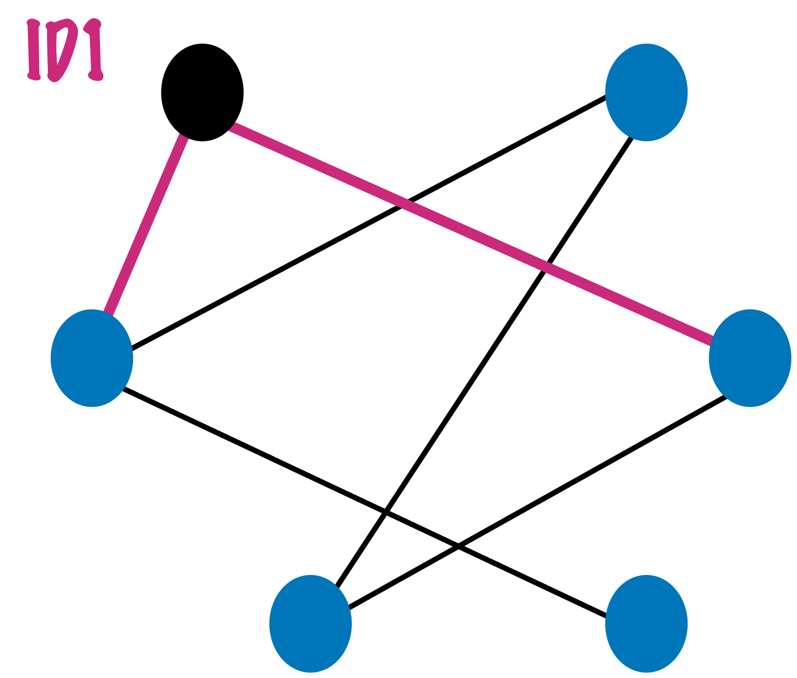
* A network (graph) of n processors (nodes) perform computation
  * Each node has an identifier (O(log n) bits) and local information (neighbours)
    * Each node knows about its incident edges and may not know anything about the ids of its neighbours



Distributed Algorithms

* A network (graph) of n processors (nodes) perform computation
  * Each node has an identifier (O(log n) bits) and local information (neighbours)
    * Each node knows about its incident edges and may not know anything about the ids of its neighbours



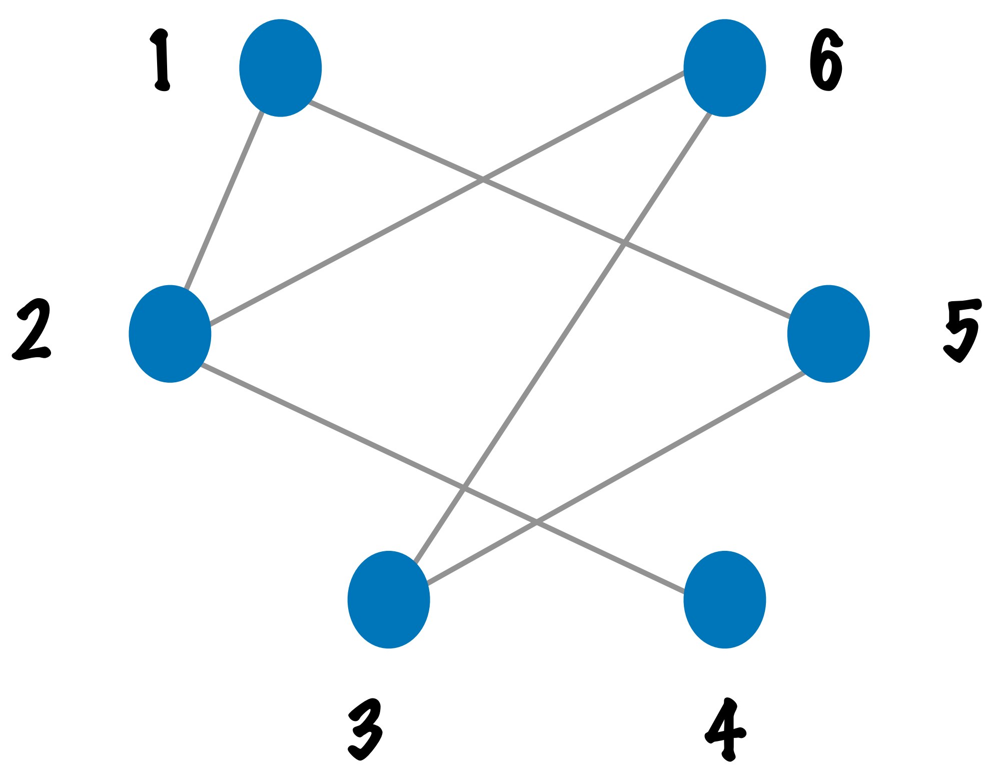


Distributed Algorithms

* Nodes communicate by exchanging messages
* Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
* Running time: no. of communication rounds

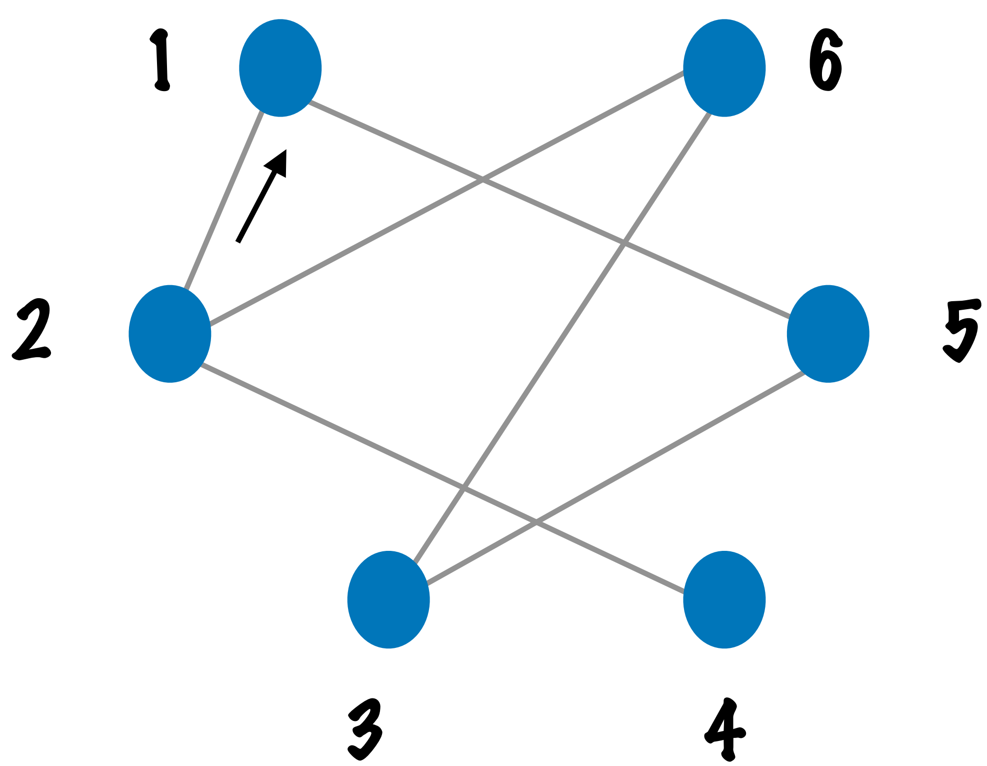
Distributed Algorithms

* Nodes communicate by exchanging messages
* Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
* Running time: no. of communication rounds



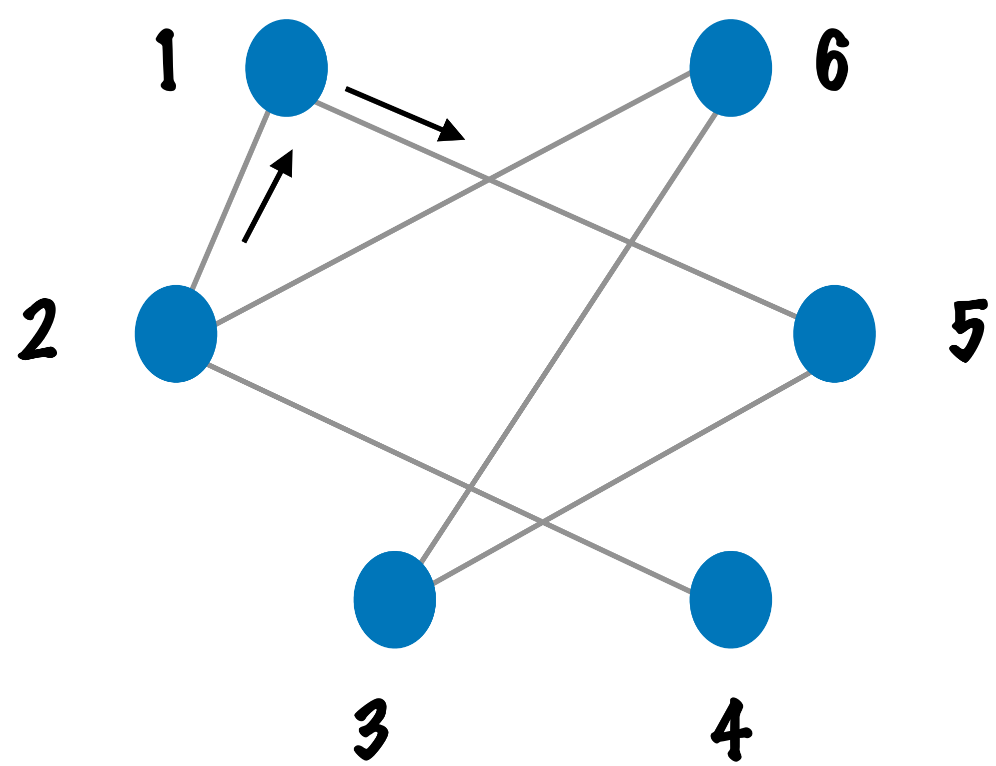
Distributed Algorithms

* Nodes communicate by exchanging messages
* Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
* Running time: no. of communication rounds



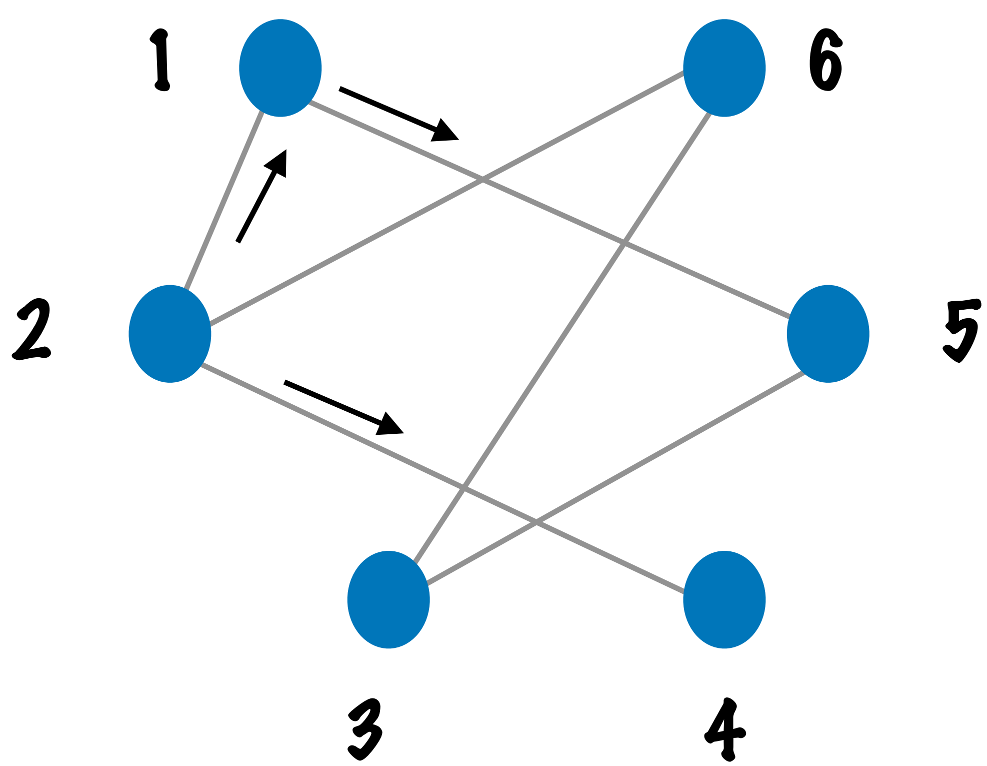
Distributed Algorithms

* Nodes communicate by exchanging messages
* Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
* Running time: no. of communication rounds



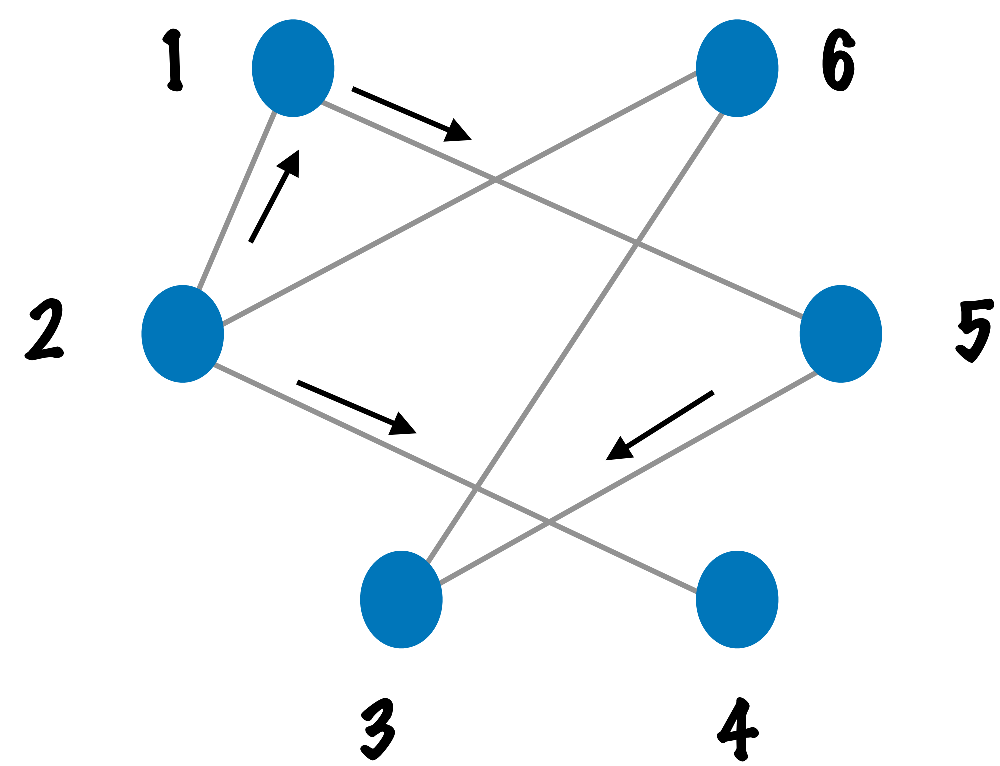
Distributed Algorithms

* Nodes communicate by exchanging messages
* Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
* Running time: no. of communication rounds



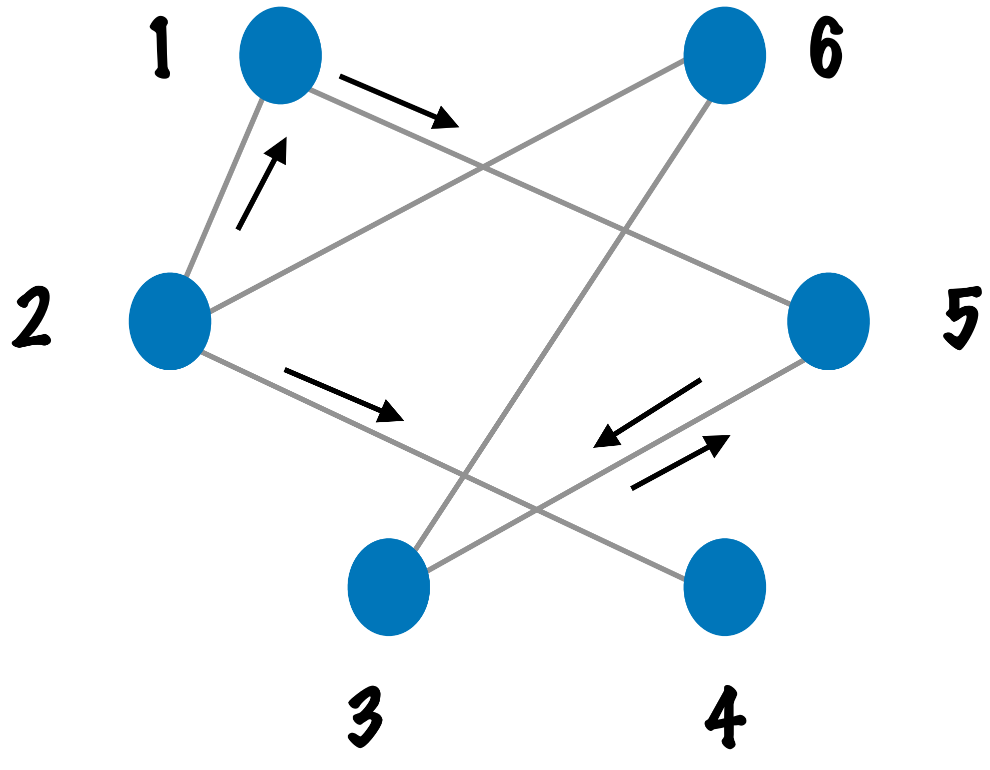
Distributed Algorithms

* Nodes communicate by exchanging messages
* Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
* Running time: no. of communication rounds

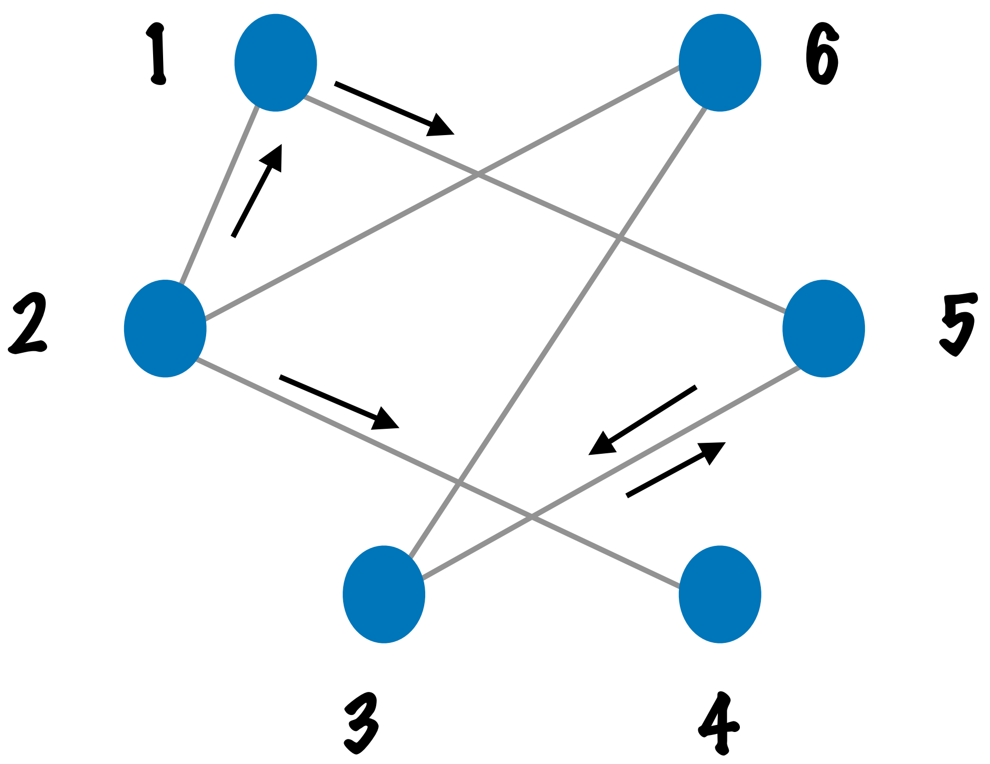


Distributed Algorithms

- Nodes communicate by exchanging messages
- Computation proceeds in synchronous rounds - time steps partitioned into discrete rounds
- Running time: no. of communication rounds

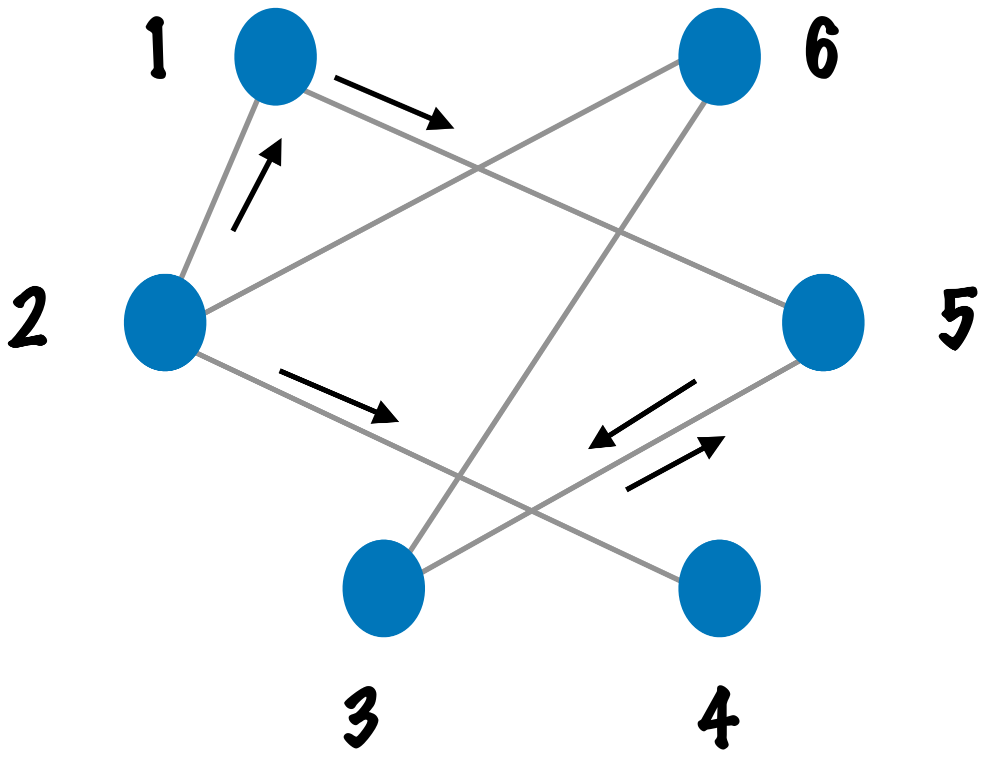


Distributed Algorithms



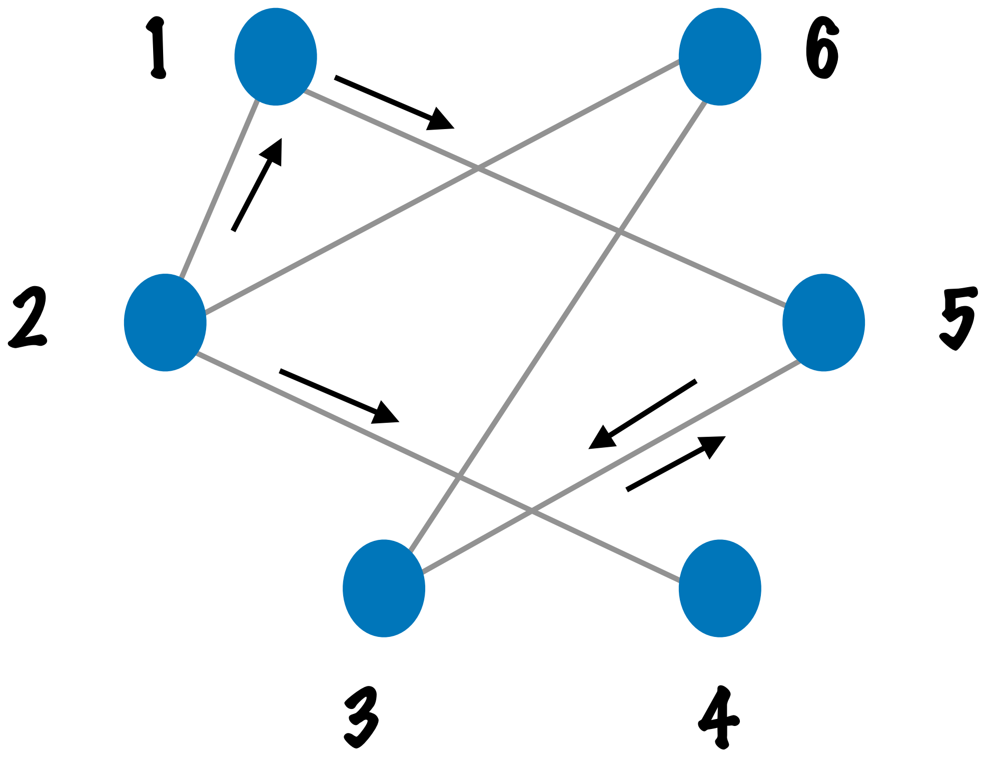
Distributed Algorithms

* In one round, each vertex



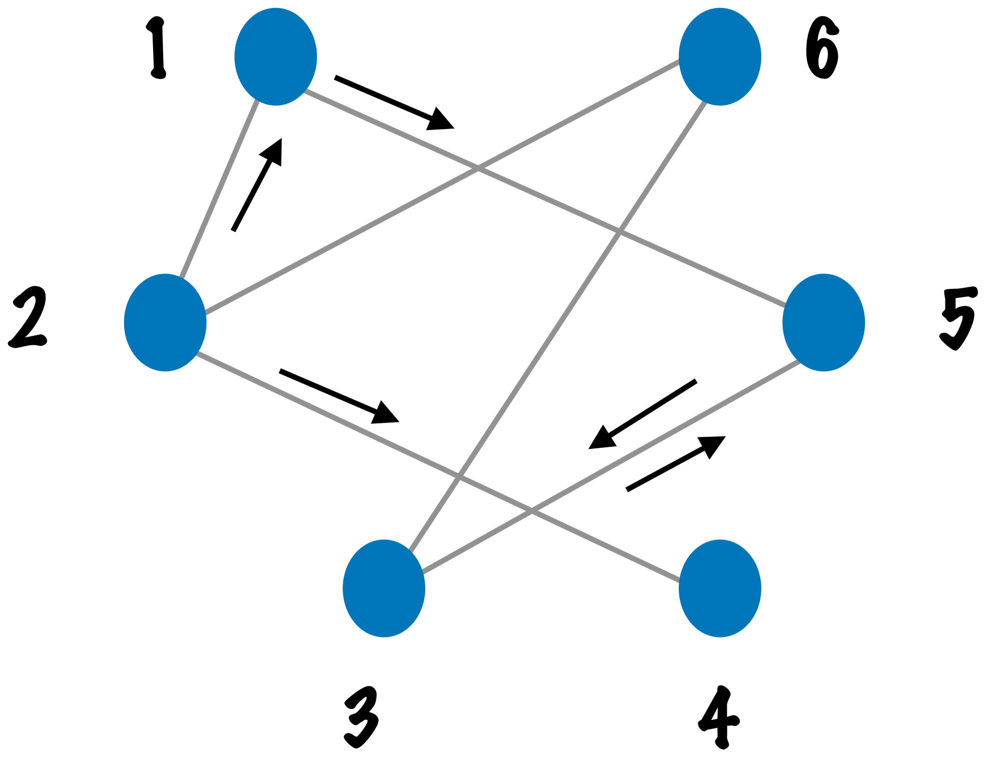
Distributed Algorithms

- In one round, each vertex
  - Computes



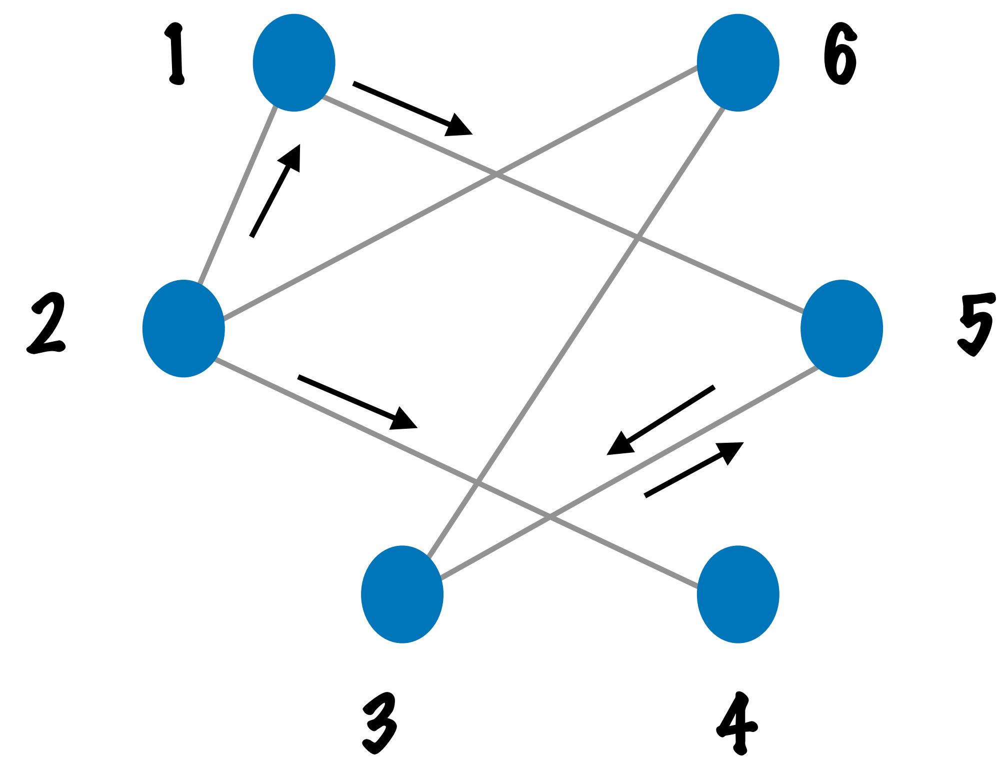
Distributed Algorithms

* In one round, each vertex
    * Computes
        * Local computation is unlimited

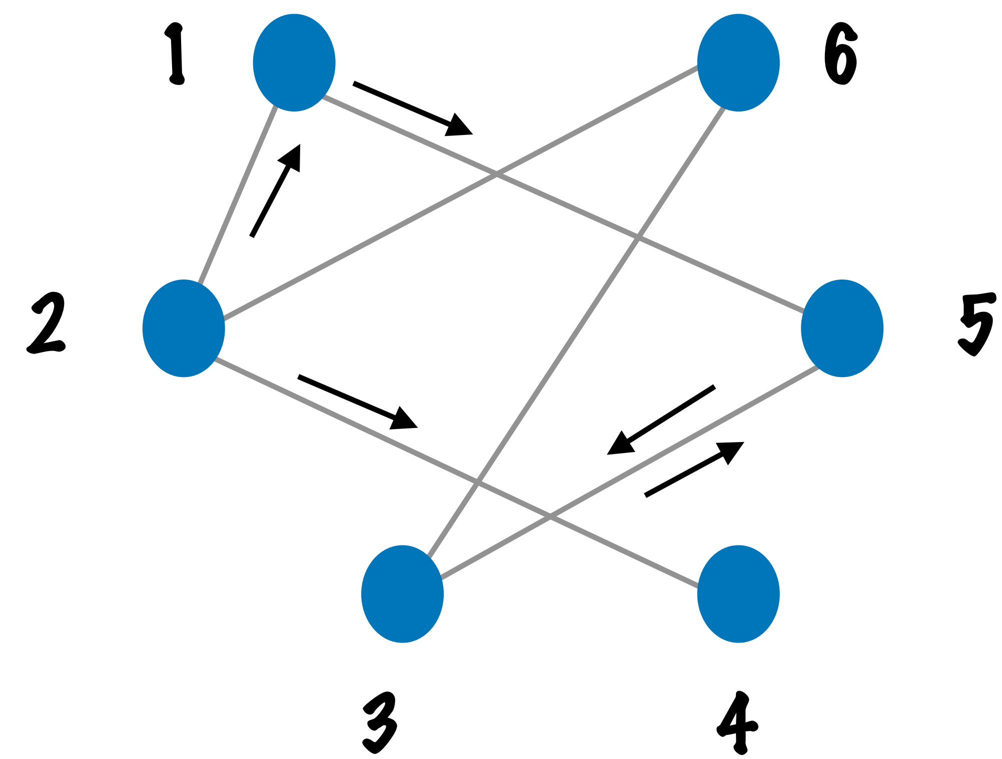


Distributed Algorithms

* In one round, each vertex
  * Computes
    * Local computation is unlimited
  * Sends/receives messages

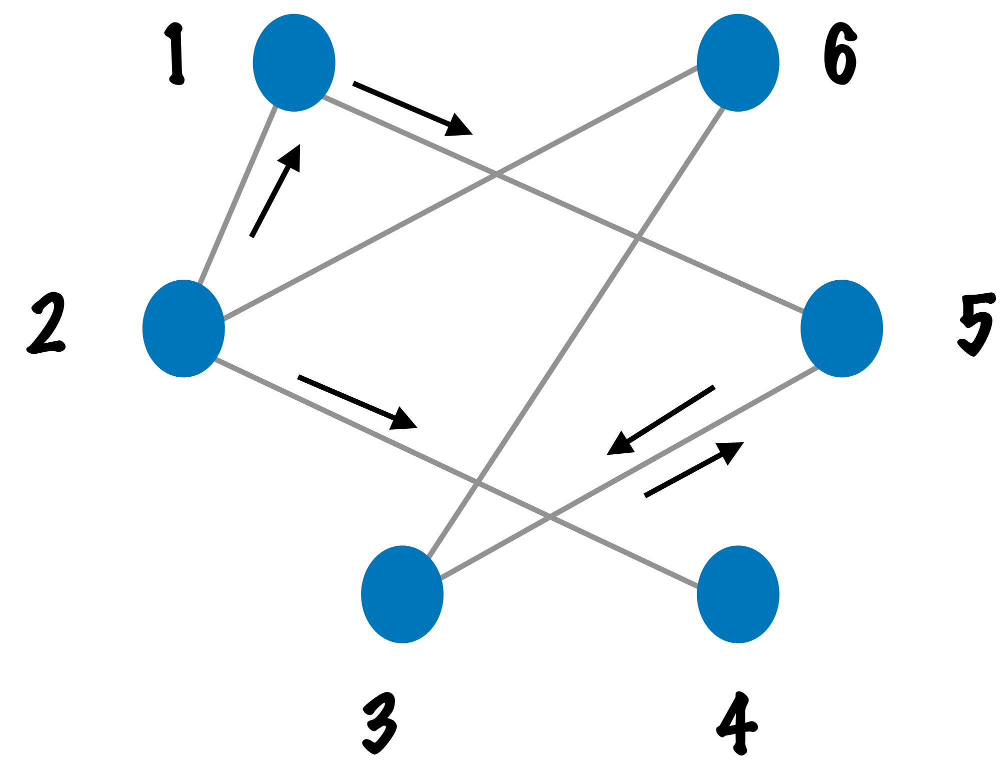


Distributed Algorithms



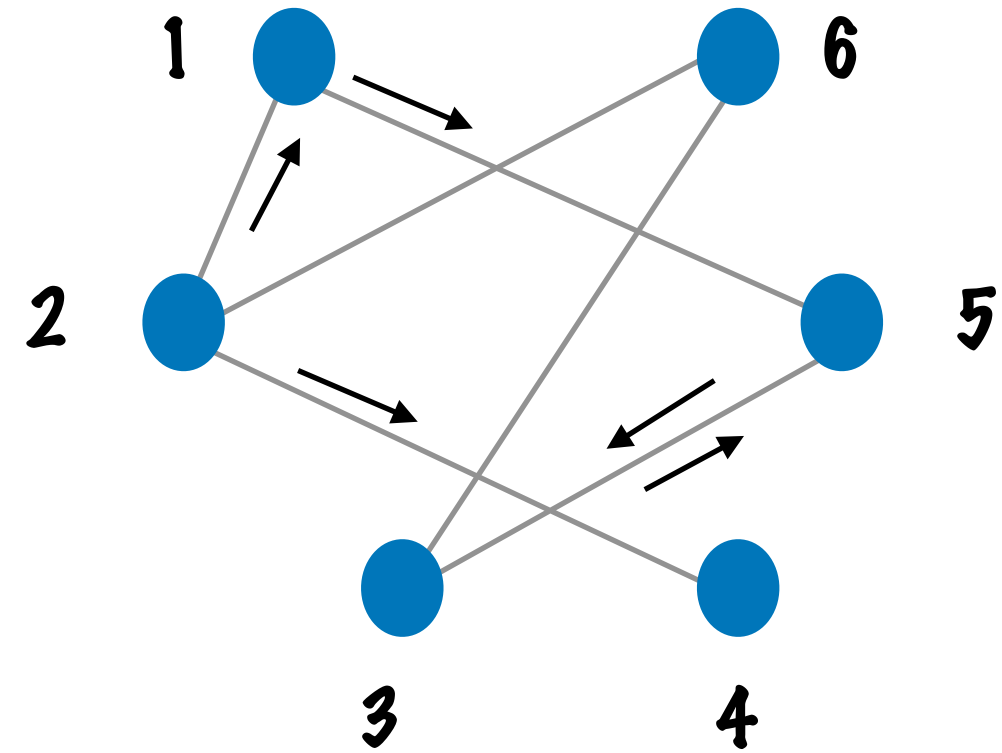
* In one round, each vertex
  * Computes
    * Local computation is unlimited
  * Sends/receives messages
    * Msg sent to a nbr is assumed to be at the beginning of the round and is received by the nbr at the end of the same round with no delay

Distributed Algorithms



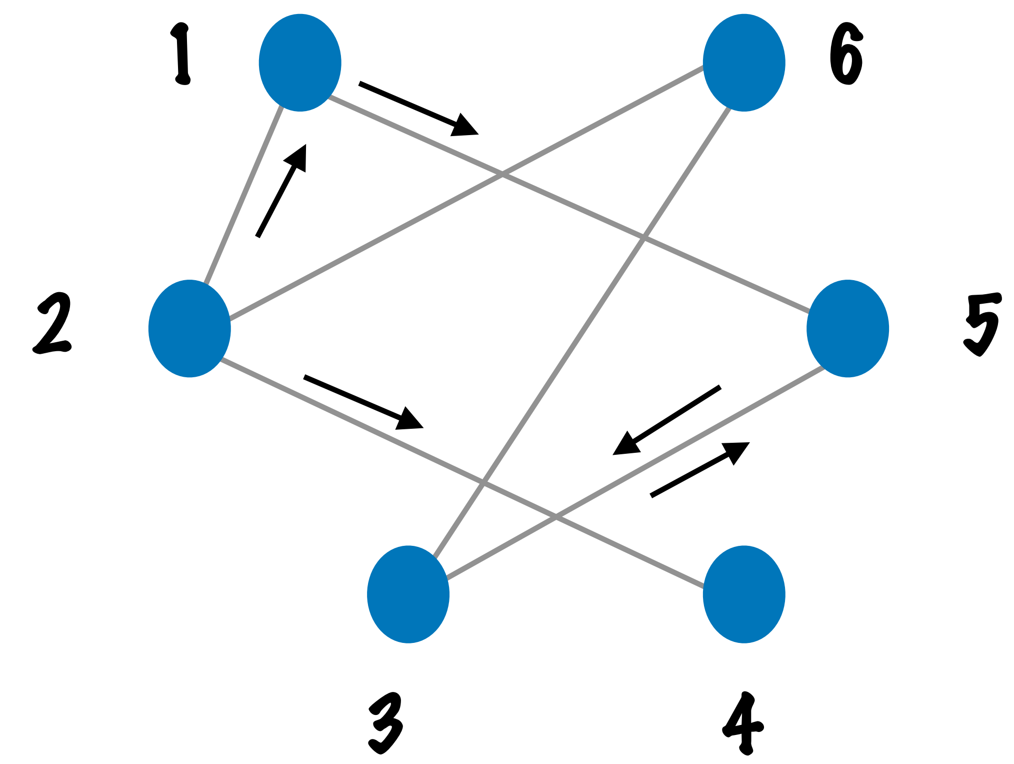
- In one round, each vertex
  - Computes
    - Local computation is unlimited
  - Sends/receives messages
    - Msg sent to a nbr is assumed to be at the beginning of the round and is received by the nbr at the end of the same round with no delay
  - LOCAL - no bound on msg size

Distributed Algorithms



* In one round, each vertex
    * Computes
        * Local computation is unlimited
    * Sends/receives messages
        * Msg sent to a nbr is assumed to be at the beginning of the round and is received by the nbr at the end of the same round with no delay
    * LOCAL - no bound on msg size
    * CONGEST - each msg is O(log n) bits

Distributed Algorithms



- In one round, each vertex
  - Computes
    - Local computation is unlimited
  - Sends/receives messages
    - Msg sent to a nbr is assumed to be at the beginning of the round and is received by the nbr at the end of the same round with no delay
  - LOCAL - no bound on msg size
  - CONGEST - each msg is O(log n) bits
    - Only O(log n) sized message can be sent per edge per time step

Broadcast

BFS Tree

Leader Election

Learning the Network

Broadcast

Given a network G and a vertex v, send a message from v to all nodes

BFS Tree

Leader Election

Learning the Network

Broadcast

Given a network G and a vertex v, send a message from v to all nodes

BFS Tree

Construct a BFS tree of G rooted at a particular vertex s

Leader Election

Learning the Network

Broadcast

Given a network G and a vertex v, send a message from v to all nodes

BFS Tree

Construct a BFS tree of G rooted at a particular vertex s

Leader Election

Elect an unique leader of the entire network

Learning the Network

Broadcast

Given a network G and a vertex v, send a message from v to all nodes

BFS Tree

Construct a BFS tree of G rooted at a particular vertex s

Leader Election

Elect an unique leader of the entire network

Learning the Network

Have all nodes learn the entire graph

Broadcast

CONGEST: O(dia) rounds

Given a network G and a vertex v, send a message from v to all nodes

BFS Tree

CONGEST: O(dia) rounds

Construct a BFS tree of G rooted at a particular vertex s

Leader Election

CONGEST: O(dia) rounds

Elect an unique leader of the entire network

Learning the Network

Have all nodes learn the entire graph

Broadcast

CONGEST: O(dia) rounds

Given a network G and a vertex v, send a message from v to all nodes

BFS Tree

CONGEST: O(dia) rounds

Construct a BFS tree of G rooted at a particular vertex s

Leader Election

CONGEST: O(dia) rounds

Elect an unique leader of the entire network

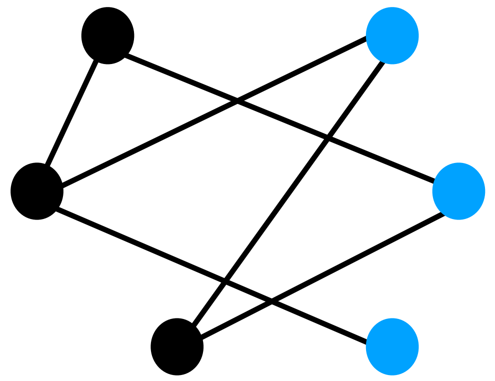
Learning the Network

CONGEST: O(m) rounds

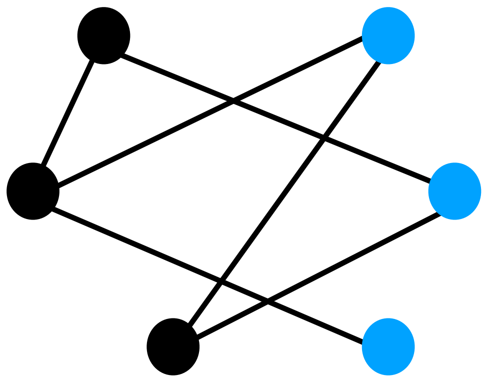
LOCAL: O(dia) rounds

Have all nodes learn the entire graph

Distributed Vertex Cover



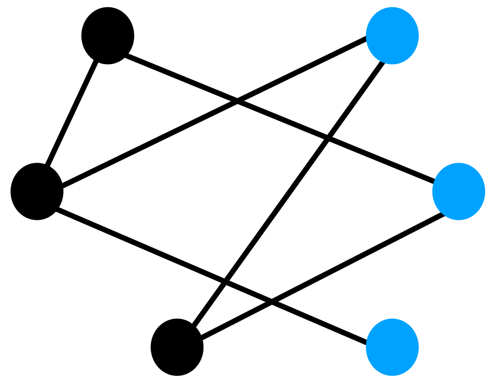
Distributed Vertex Cover



Input: Graph G on n vertices m edges

Question: Find a minimum sized set S of vertices such that for each edge (u,v), either u is in S or v is in S

Distributed Vertex Cover

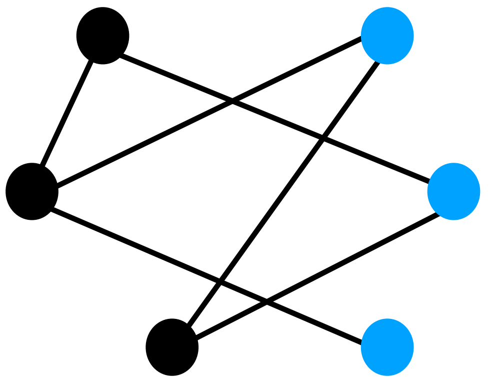


Input: Graph G on n vertices m edges

Question: Find a minimum sized set S of vertices such that for each edge (u,v), either u is in S or v is in S

* **Communication graph** is same as the graph on which the vertex cover is required
* At the end of the computation **each vertex knows if it is in the solution or not**

Distributed Vertex Cover



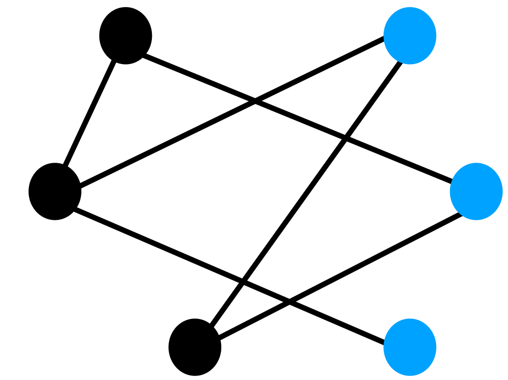
Input: Graph G on n vertices m edges

Question: Find a minimum sized set S of vertices such that for each edge (u,v), either u is in S or v is in S

* Communication graph is same as the graph on which the vertex cover is required
* At the end of the computation each vertex knows if it is in the solution or not

An easy algorithm: have the leader learn the entire graph

Distributed Vertex Cover



Input: Graph G on n vertices m edges

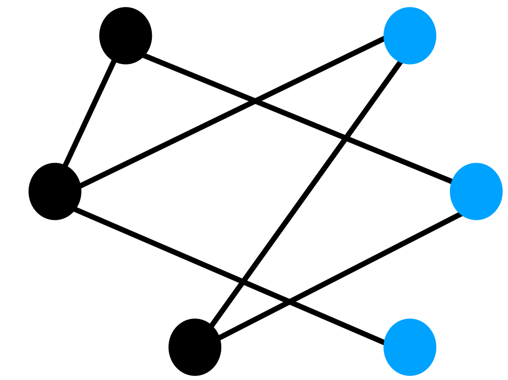
Question: Find a minimum sized set S of vertices such that for each edge (u,v), either u is in S or v is in S

* Communication graph is same as the graph on which the vertex cover is required
* At the end of the computation each vertex knows if it is in the solution or not

An easy algorithm: have the leader learn the entire graph

LOCAL: O(diam) rounds CONGEST: O(m) rounds

Distributed Vertex Cover



Input: Graph G on n vertices m edges

Question: Find a minimum sized set S of vertices such that for each edge (u,v), either u is in S or v is in S

* Communication graph is same as the graph on which the vertex cover is required
* At the end of the computation each vertex knows if it is in the solution or not

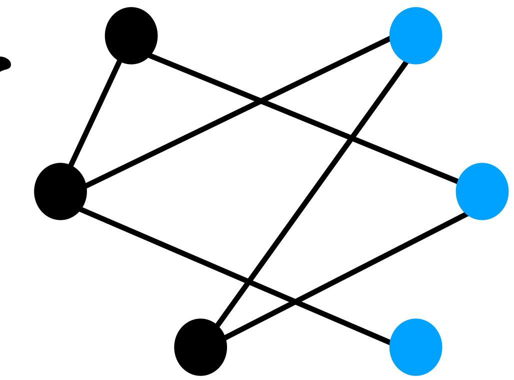
An easy algorithm: have the leader learn the entire graph

LOCAL: O(diam) rounds CONGEST: O(m) rounds

Lower Bounds

* $\Omega(n^2/\log^2 n)$ rounds to compute min vertex cover in CONGEST [Censor-Hillel et al. 17]
* $\Omega(\min \{(\log n / \log\log n)^{1/2}, \log \Delta / \log\log \Delta\}$ to compute constant factor approx. to min vertex cover in LOCAL [Kuhn et al. 16]

Parameterized Distributed Vertex Cover

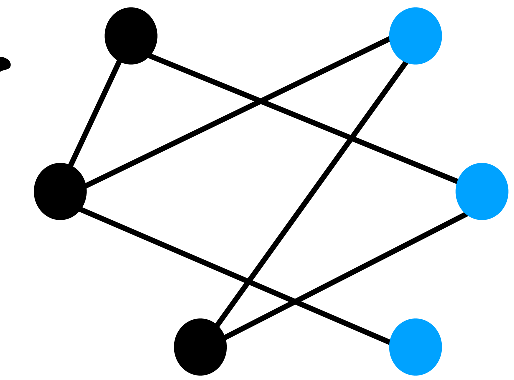


Input: Graph G and positive integer k

Question: Does G have a set S of at most k vertices such that for each edge (u,v), either u is in S or v is in S

- Communication graph is same as the graph on which the vertex cover is required
- At the end of the computation
  - NO instance: Each vertex knows that no k-solution exists
  - YES instance: Each vertex knows if it is in the k-solution or not
    - flag(v) = 1 if v is in the solution
    - flag(v) = 0 if v is not in the solution

Parameterized Distributed Vertex Cover

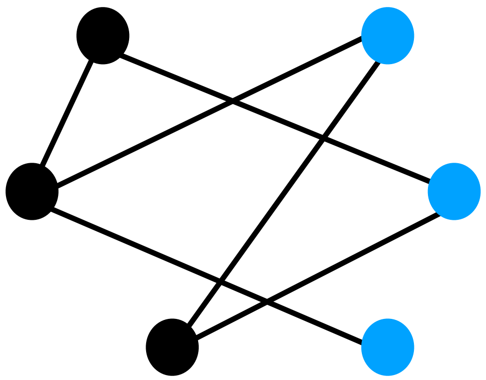


Input: Graph G and positive integer k **Parameter:** k

Question: Does G have a set S of at most k vertices such that for each edge (u,v), either u is in S or v is in S

* Communication graph is same as the graph on which the vertex cover is required
* At the end of the computation
  * NO instance: Each vertex knows that no k-solution exists
  * YES instance: Each vertex knows if it is in the k-solution or not
    * flag(v) = 1 if v is in the solution
    * flag(v) = 0 if v is not in the solution

Parameterized Distributed Vertex Cover



Input: Graph G and positive integer k Parameter: k

Question: Does G have a set S of at most k vertices such that for each edge (u,v), either u is in S or v is in S

* Communication graph is same as the graph on which the vertex cover is required
* At the end of the computation
    * NO instance: Each vertex knows that no k-solution exists
    * YES instance: Each vertex knows if it is in the k-solution or not
        * flag(v) = 1 if v is in the solution
        * flag(v) = 0 if v is not in the solution

Goal: $O(f(k))$ rounds

Parameterized Diameter Approximation

CONGEST

Parameterized Diameter Approximation

CONGEST

* O(k) rounds algorithm that terminates with all vertices declaring

Parameterized Diameter Approximation

CONGEST

* O(k) rounds algorithm that terminates with all vertices declaring
  * SMALL if diameter $\leq k$

Parameterized Diameter Approximation

CONGEST

* O(k) rounds algorithm that terminates with all vertices declaring
  * SMALL if diameter $\leq k$
  * LARGE if diameter $> 2k$

Parameterized Diameter Approximation

CONGEST

* $O(k)$ rounds algorithm that terminates with all vertices declaring
  * SMALL if diameter $\leq k$
  * LARGE if diameter $> 2k$
  * SMALL or LARGE unanimously if diameter is between $k+1$ and $2k$

Parameterized Diameter Approximation

CONGEST

* O(k) rounds algorithm that terminates with all vertices declaring
  * SMALL if diameter $\leq$ k
  * LARGE if diameter > 2k
  * SMALL or LARGE unanimously if diameter is between k+1 and 2k

* Vertices report LARGE
  * dia $\geq$ k+1

Parameterized Diameter Approximation

CONGEST

* O(k) rounds algorithm that terminates with all vertices declaring
  * SMALL if diameter $\leq k$
  * LARGE if diameter $> 2k$
  * SMALL or LARGE unanimously if diameter is between $k+1$ and $2k$

* Vertices report LARGE
  * dia $\geq k+1$

* Vertices report SMALL
  * dia $\leq 2k$

Parameterized Distributed Vertex Cover

LOCAL

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds
    * Vertices report LARGE: dia ≥ 2k+1

Parameterized Distributed Vertex Cover

LOCAL

- Check if diameter is bounded by 2k in O(k) rounds
  - Vertices report LARGE: dia ≥ 2k+1
    - No k-vertex cover exists

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds
    * Vertices report LARGE: dia ≥ 2k+1
        * No k-vertex cover exists
    * Vertices report SMALL: dia ≤ 4k

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds
  * Vertices report LARGE: dia ≥ 2k+1
    * No k-vertex cover exists
  * Vertices report SMALL: dia ≤ 4k
    * Learn the entire graph in O(k) rounds

Parameterized Distributed Vertex Cover

LOCAL

- Check if diameter is bounded by 2k in O(k) rounds
  - Vertices report LARGE: dia ≥ 2k+1
    - No k-vertex cover exists
  - Vertices report SMALL: dia ≤ 4k
    - Learn the entire graph in O(k) rounds
    - Elect a leader in O(k) rounds

Parameterized Distributed Vertex Cover

LOCAL

- Check if diameter is bounded by 2k in O(k) rounds
  - Vertices report LARGE: dia ≥ 2k+1
    - No k-vertex cover exists
  - Vertices report SMALL: dia ≤ 4k
    - Learn the entire graph in O(k) rounds
    - Elect a leader in O(k) rounds
    - Leader computes min VC and in O(k) rounds,

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds
    * Vertices report LARGE: dia ≥ 2k+1
        * No k-vertex cover exists
    * Vertices report SMALL: dia ≤ 4k
        * Learn the entire graph in O(k) rounds
        * Elect a leader in O(k) rounds
        * Leader computes min VC and in O(k) rounds,
            * Either notifies each vertex if it is in the k-solution or not

Parameterized Distributed Vertex Cover

LOCAL

- Check if diameter is bounded by 2k in O(k) rounds
  - Vertices report LARGE: dia ≥ 2k+1
    - No k-vertex cover exists
  - Vertices report SMALL: dia ≤ 4k
    - Learn the entire graph in O(k) rounds
    - Elect a leader in O(k) rounds
    - Leader computes min VC and in O(k) rounds,
      - Either notifies each vertex if it is in the k-solution or not
      - Or informs each vertex that no k-solution exists

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds
  * Vertices report LARGE: dia ≥ 2k+1
    * No k-vertex cover exists
  * Vertices report SMALL: dia ≤ 4k
    * Learn the entire graph in O(k) rounds
    * Elect a leader in O(k) rounds
    * Leader computes min VC and in O(k) rounds,
      * Either notifies each vertex if it is in the k-solution or not
      * Or informs each vertex that no k-solution exists

O(k) rounds

Parameterized Distributed Vertex Cover

LOCAL

* Check if diameter is bounded by 2k in O(k) rounds
  * Vertices report LARGE: dia ≥ 2k+1
    * No k-vertex cover exists
  * Vertices report SMALL: dia ≤ 4k
    * Learn the entire graph in O(k) rounds
    * Elect a leader in O(k) rounds
    * Leader computes min VC and in O(k) rounds,
      * Either notifies each vertex if it is in the k-solution or not
      * Or informs each vertex that no k-solution exists

O(k) rounds

Matching, Dominating Set, Edge Dominating Set, Feedback Vertex Set, Feedback Edge Set

Parameterized Distributed Vertex Cover

CONGEST

* Check if diameter is bounded by 2k in O(k) rounds
  * Vertices report LARGE: dia ≥ 2k+1
    * No k-vertex cover exists
  * Vertices report SMALL: dia ≤ 4k
    * Learn the entire graph in O(m) rounds
    * Elect a leader in O(k) rounds
    * Leader computes min VC and in O(k) rounds,
      * Either notifies each vertex if it is in the k-solution or not
      * Or informs each vertex that no k-solution exists

Parameterized Distributed Vertex Cover

CONGEST

* Check if diameter is bounded by 2k in O(k) rounds
  * Vertices report LARGE: dia ≥ 2k+1
    * No k-vertex cover exists
  * Vertices report SMALL: dia ≤ 4k
    * Learn the entire graph in O(m) rounds
    * Elect a leader in O(k) rounds
    * Leader computes min VC and in O(k) rounds,
      * Either notifies each vertex if it is in the k-solution or not
      * Or informs each vertex that no k-solution exists

Parameterized Distributed Vertex Cover

CONGEST

* Check if diameter is bounded by 2k in O(k) rounds
    * Vertices report LARGE: dia ≥ 2k+1
        * No k-vertex cover exists
    * Vertices report SMALL: dia ≤ 4k
        * Learn the entire graph in O(m) rounds
        * Elect a leader in O(k) rounds
        * Leader computes min VC and in O(k) rounds,
            * Either notifies each vertex if it is in the k-solution or not
            * Or informs each vertex that no k-solution exists

How to improve this step?

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Diameter ≤ 2k

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Diameter ≤ 2k

Algorithm k-VC Guarantee

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Diameter ≤ 2k

Algorithm k-VC Guarantee

* Find a leader vertex v in O(k) rounds

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Diameter $\leq 2k$

Algorithm k-VC Guarantee

* Find a leader vertex v in $O(k)$ rounds
* If a vertex is of degree $> k$
  * Sets its flag to 1 and terminates after informing its neighbours (1 round)

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Diameter $\leq 2k$

Algorithm k-VC Guarantee

* Find a leader vertex v in $O(k)$ rounds
* If a vertex is of degree $> k$
  * Sets its flag to 1 and terminates after informing its neighbours (1 round)
* If a vertex is of degree 0
  * Sets its flag to 0 and terminates after informing its neighbours (1 round)

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Suppose k-vertex cover exists

Diameter ≤ 2k

Algorithm k-VC Guarantee

* Find a leader vertex v in O(k) rounds
* If a vertex is of degree > k
    * Sets its flag to 1 and terminates after informing its neighbours (1 round)
* If a vertex is of degree 0
    * Sets its flag to 0 and terminates after informing its neighbours (1 round)
* Any vertex that has not yet terminated has degree at most k

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k
* Graph has $O(k^2)$ (active) edges

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k
* Graph has $O(k^2)$ (active) edges
* Learn the entire graph in $O(k^2)$ rounds

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k
* Graph has $O(k^2)$ (active) edges
* Learn the entire graph in $O(k^2)$ rounds
* Leader computes min VC and in $O(k)$ rounds

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k
* Graph has $O(k^2)$ (active) edges
* Learn the entire graph in $O(k^2)$ rounds
* Leader computes min VC and in $O(k)$ rounds
  * Either notifies each vertex if it is in the k-solution or not

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k
* Graph has $O(k^2)$ (active) edges
* Learn the entire graph in $O(k^2)$ rounds
* Leader computes min VC and in $O(k)$ rounds
  * Either notifies each vertex if it is in the k-solution or not
  * Or informs each vertex that no k-solution exists

Parameterized Distributed Vertex Cover

CONGEST

Guarantee Version

Algorithm k-VC Guarantee (contd.)

* Any vertex that has not yet terminated has degree at most k
* Graph has $O(k^2)$ (active) edges
* Learn the entire graph in $O(k^2)$ rounds
* Leader computes min VC and in $O(k)$ rounds
  * Either notifies each vertex if it is in the k-solution or not
  * Or informs each vertex that no k-solution exists

Complexity: ck^2 rounds where c is a constant

Parameterized Distributed Vertex Cover

CONGEST

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

* Vertices report LARGE: dia $\geq$ 2k+1

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

* Vertices report LARGE: dia ≥ 2k+1
    * No k-vertex cover exists

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

* Vertices report LARGE: dia ≥ 2k+1
  * No k-vertex cover exists
* Vertices report SMALL: dia ≤ 4k

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

* Vertices report LARGE: dia ≥ 2k+1
  * No k-vertex cover exists
* Vertices report SMALL: dia ≤ 4k
  * Compute a leader and a BFS tree rooted at it in O(k) rounds

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

- Vertices report LARGE: dia ≥ 2k+1
    - No k-vertex cover exists
- Vertices report SMALL: dia ≤ 4k
    - Compute a leader and a BFS tree rooted at it in O(k) rounds
    - Run Algorithm k-VC Guarantee for ck^2 rounds

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in $O(k)$ rounds

* Vertices report LARGE: dia $\geq 2k+1$
  * No k-vertex cover exists
* Vertices report SMALL: dia $\leq 4k$
  * Compute a leader and a BFS tree rooted at it in $O(k)$ rounds
  * Run Algorithm k-VC Guarantee for ck^2 rounds
    * Not yet terminated: no k-vertex cover exists

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

* Vertices report LARGE: dia $\geq 2k+1$
  * No k-vertex cover exists
* Vertices report SMALL: dia $\leq 4k$
  * Compute a leader and a BFS tree rooted at it in O(k) rounds
  * Run Algorithm k-VC Guarantee for ck^2 rounds
    * Not yet terminated: no k-vertex cover exists
    * Terminated: determine if the size of the vertex cover computed is indeed at most k in O(k) rounds

Parameterized Distributed Vertex Cover

CONGEST

Check if diameter is bounded by 2k in O(k) rounds

* Vertices report LARGE: dia ≥ 2k+1
  * No k-vertex cover exists
* Vertices report SMALL: dia ≤ 4k
  * Compute a leader and a BFS tree rooted at it in O(k) rounds
  * Run Algorithm k-VC Guarantee for ck^2 rounds
    * Not yet terminated: no k-vertex cover exists
    * Terminated: determine if the size of the vertex cover computed is indeed at most k in O(k) rounds

Complexity: $O(k^2)$ rounds

Parameterized Diameter Approximation

CONGEST

* O(k) rounds algorithm that terminates with all vertices declaring
  * SMALL if diameter $\leq k$
  * LARGE if diameter $> 2k$
  * SMALL or LARGE unanimously if diameter is between $k+1$ and $2k$

* Vertices report LARGE
  * dia $\geq k+1$

* Vertices report SMALL
  * dia $\leq 2k$

Parameterized Diameter Approximation

CONGEST

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round,

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round,
  * Every vertex sends the min ID that it has seen so far to its neighbours

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round,
  * Every vertex sends the min ID that it has seen so far to its neighbours

At the end of Phase 1, each vertex v has the min ID x(v) in its k-hop neighbourhood

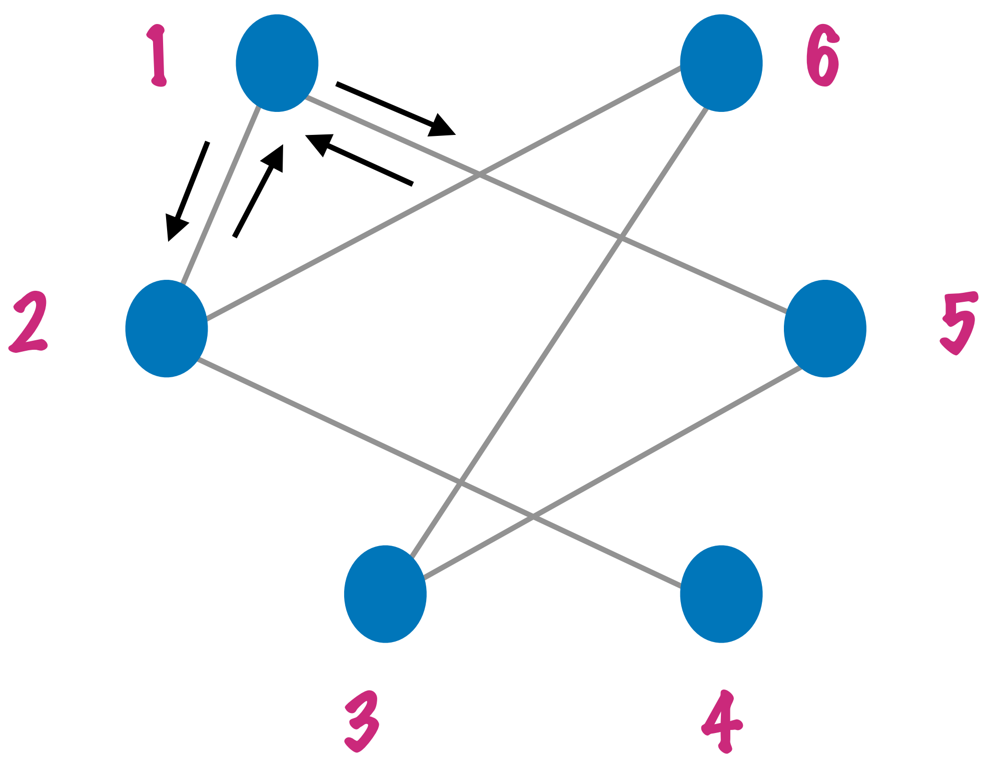
Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round,
  * Every vertex sends the min ID that it has seen so far to its neighbours

At the end of Phase 1, each vertex v has the min ID x(v) in its k-hop neighbourhood



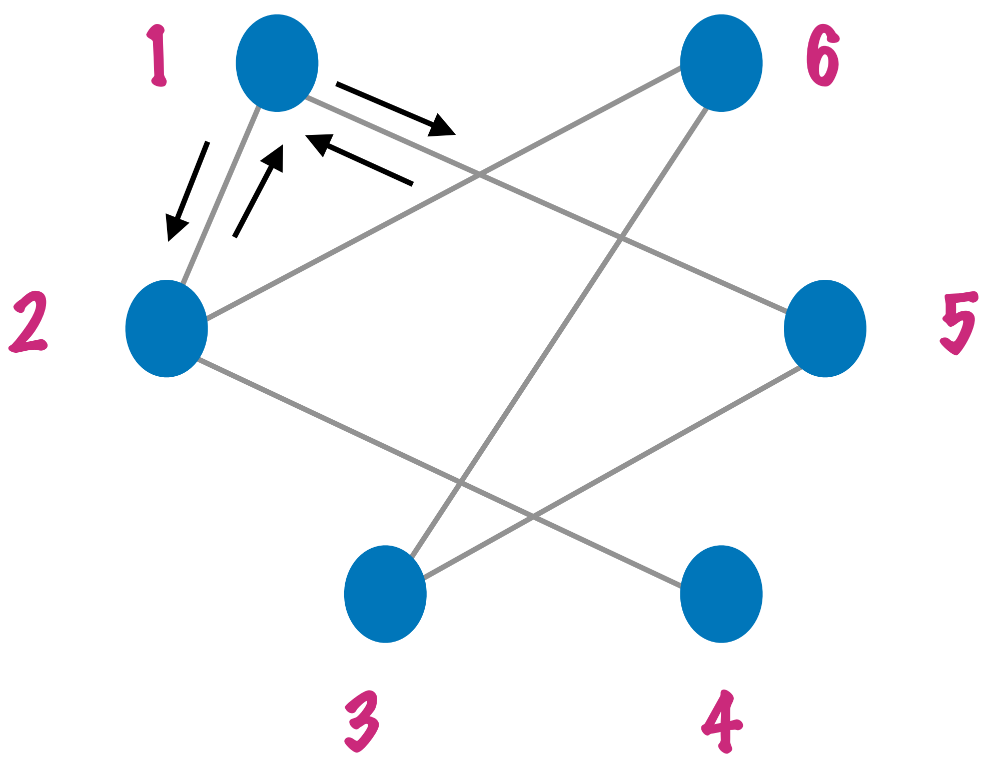
Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round,
  * Every vertex sends the min ID that it has seen so far to its neighbours

At the end of Phase 1, each vertex v has the min ID x(v) in its k-hop neighbourhood



. . .

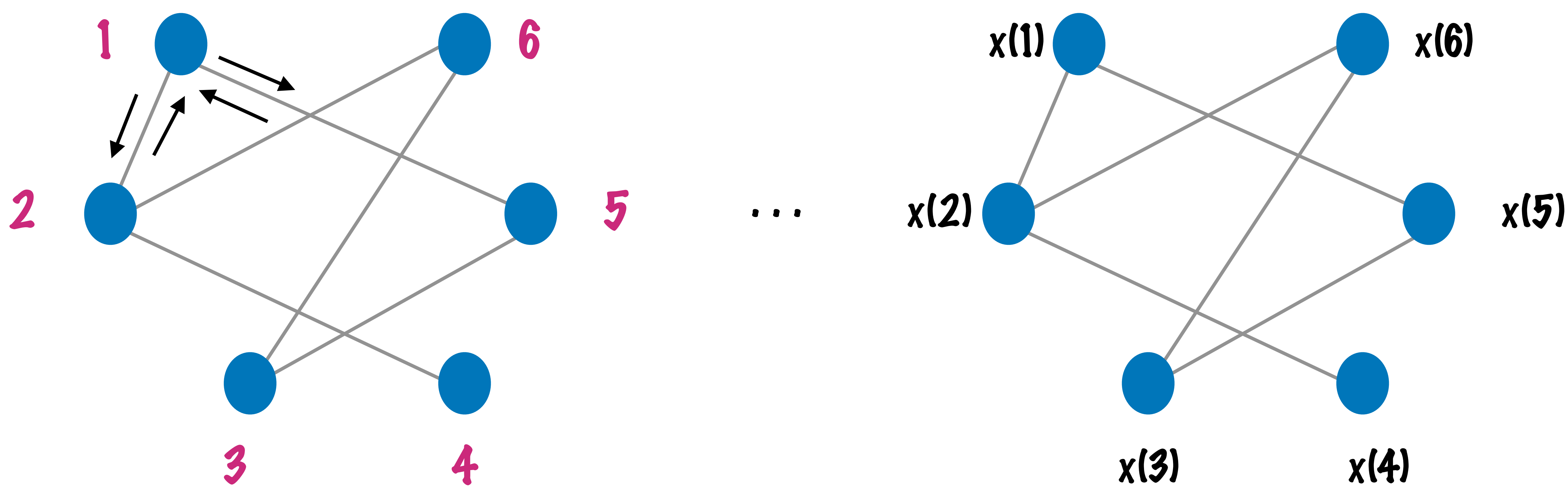
Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round,
  * Every vertex sends the min ID that it has seen so far to its neighbours

At the end of Phase 1, each vertex v has the min ID x(v) in its k-hop neighbourhood



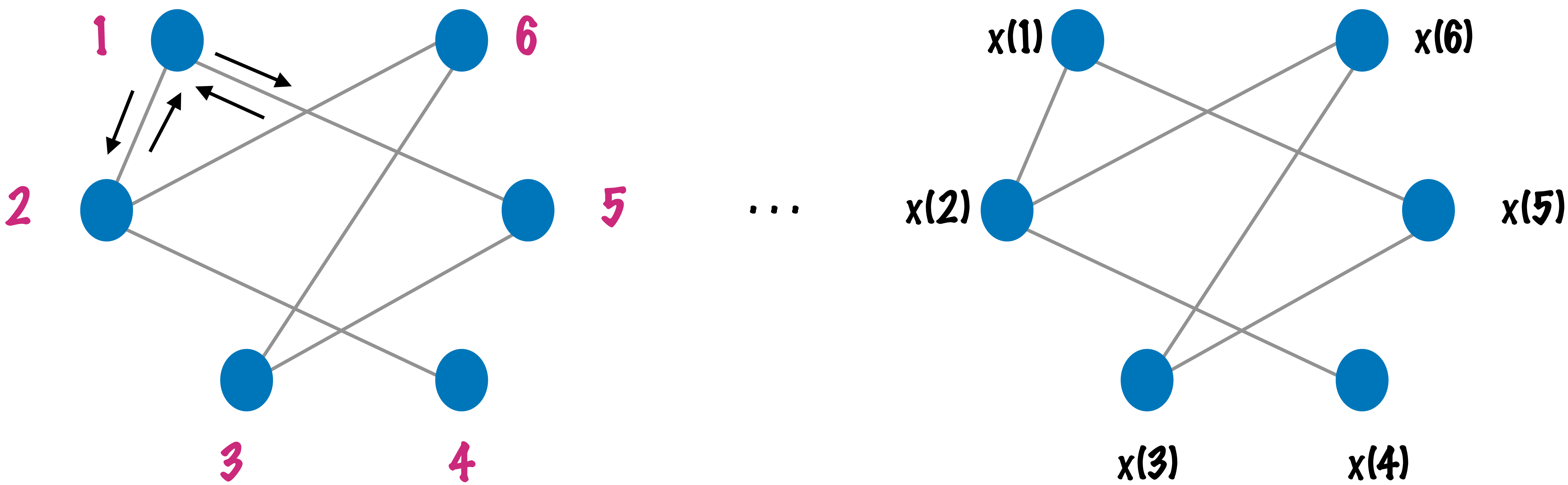
Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

- In each round,
  - Every vertex sends the min ID that it has seen so far to its neighbours

At the end of Phase 1, each vertex v has the min ID x(v) in its k-hop neighbourhood



If dia ≤ k, then x(v)s are identical

Parameterized Diameter Approximation

CONGEST

Parameterized Diameter Approximation

CONGEST

Phase 2: 2k+1 rounds

Parameterized Diameter Approximation

CONGEST

Phase 2: 2k+1 rounds

* In each round,

Parameterized Diameter Approximation

CONGEST

Phase 2: 2k+1 rounds

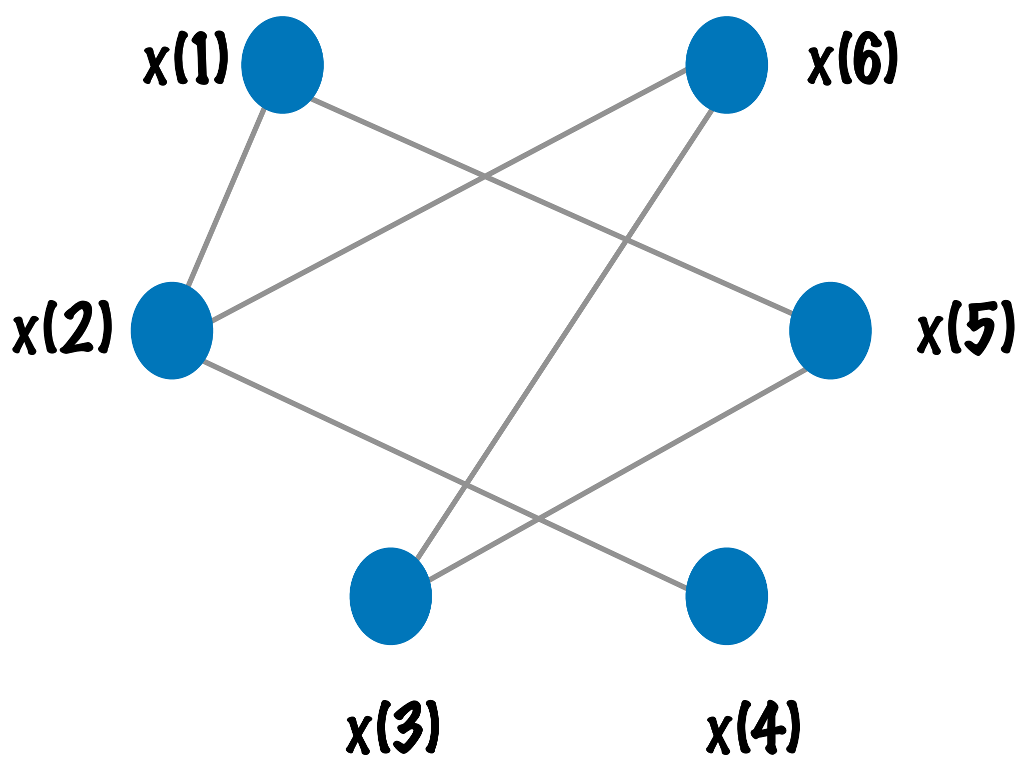
* In each round,
  * Every vertex v sends y(v) (the min x(u)) and z(v) (the max x(u)) that it has seen so far to its neighbours

Parameterized Diameter Approximation

CONGEST

Phase 2: 2k+1 rounds

- In each round,
  - Every vertex v sends y(v) (the min x(u)) and z(v) (the max x(u)) that it has seen so far to its neighbours

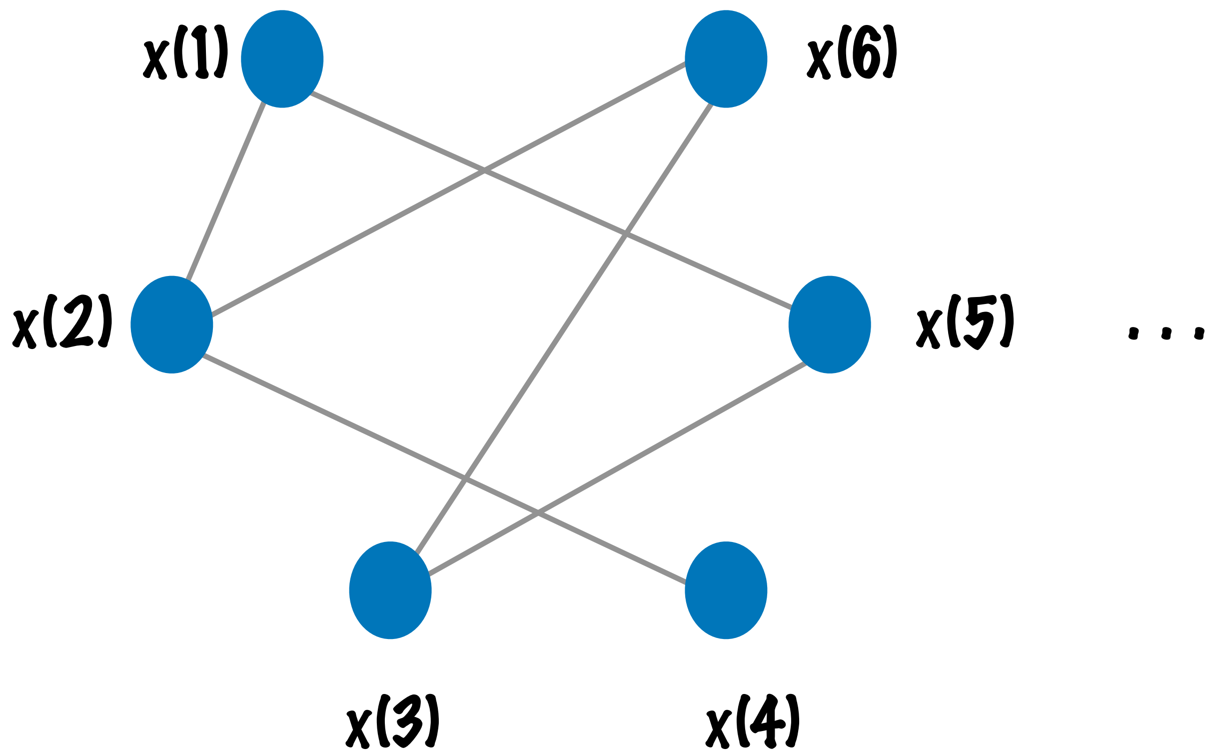


Parameterized Diameter Approximation

CONGEST

Phase 2: 2k+1 rounds

* In each round,
    * Every vertex v sends y(v) (the min x(u)) and z(v) (the max x(u)) that it has seen so far to its neighbours

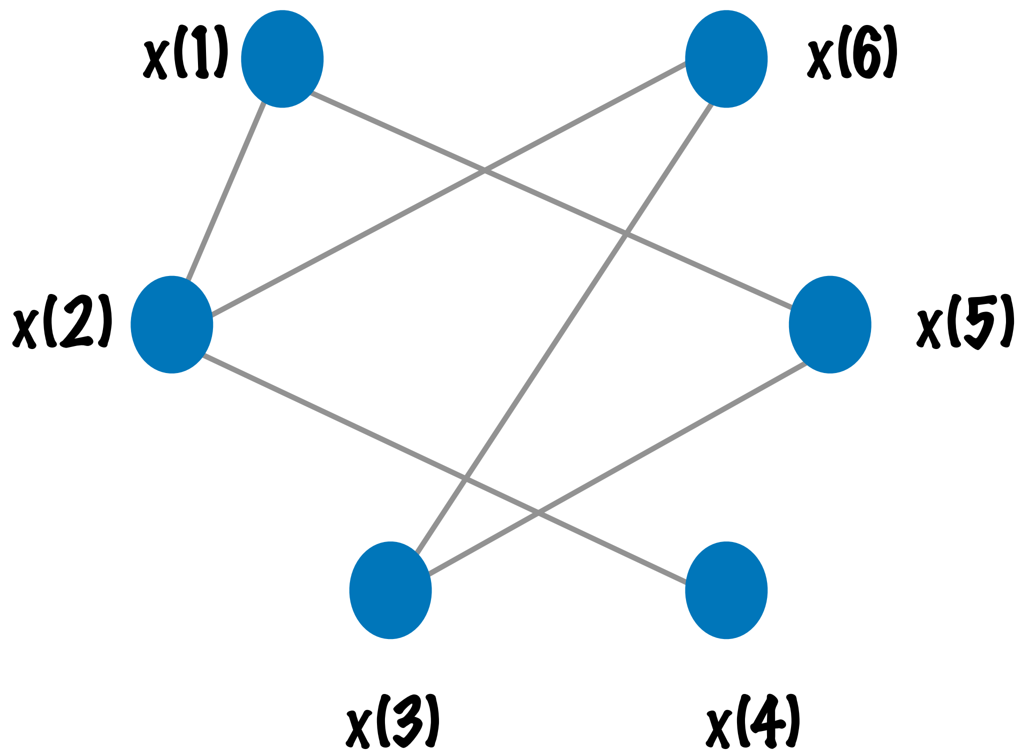


Parameterized Diameter Approximation

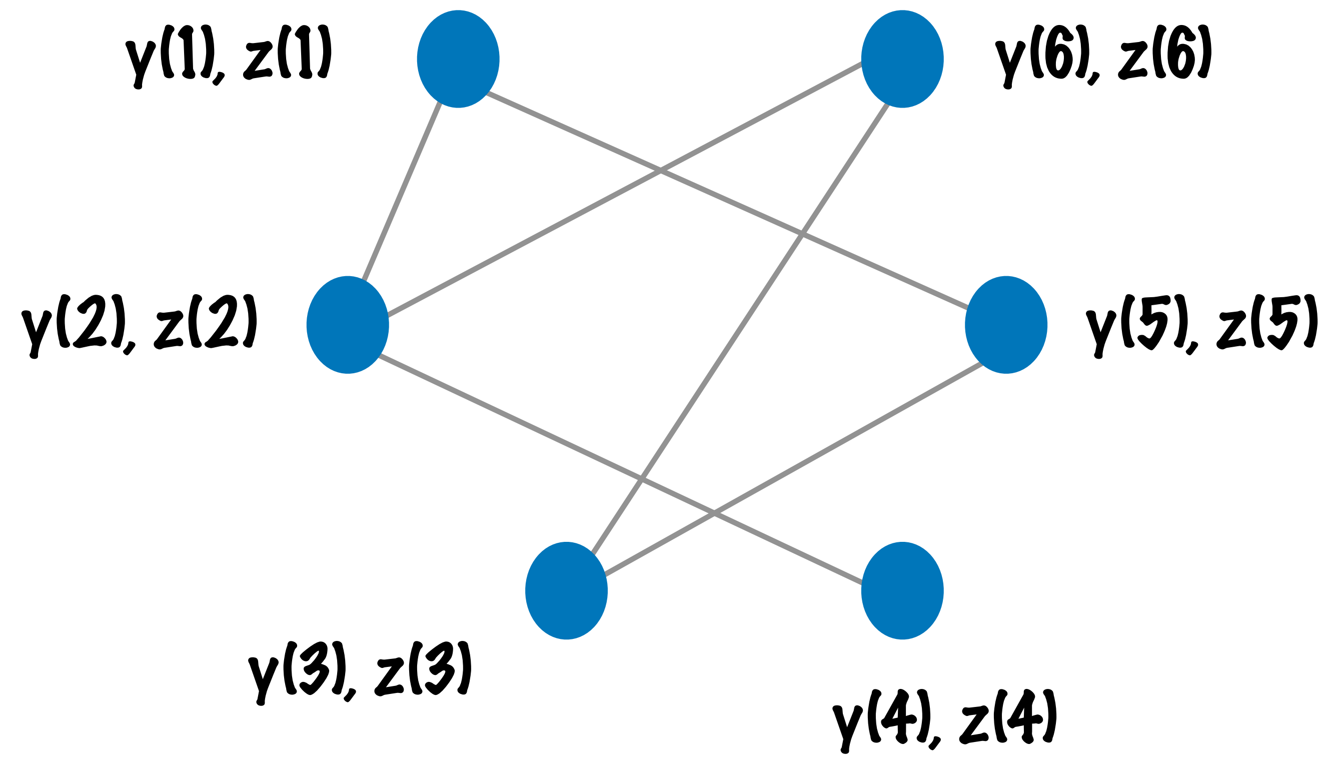
CONGEST

Phase 2: 2k+1 rounds

* In each round,
    * Every vertex v sends y(v) (the min x(u)) and z(v) (the max x(u)) that it has seen so far to its neighbours



. . .

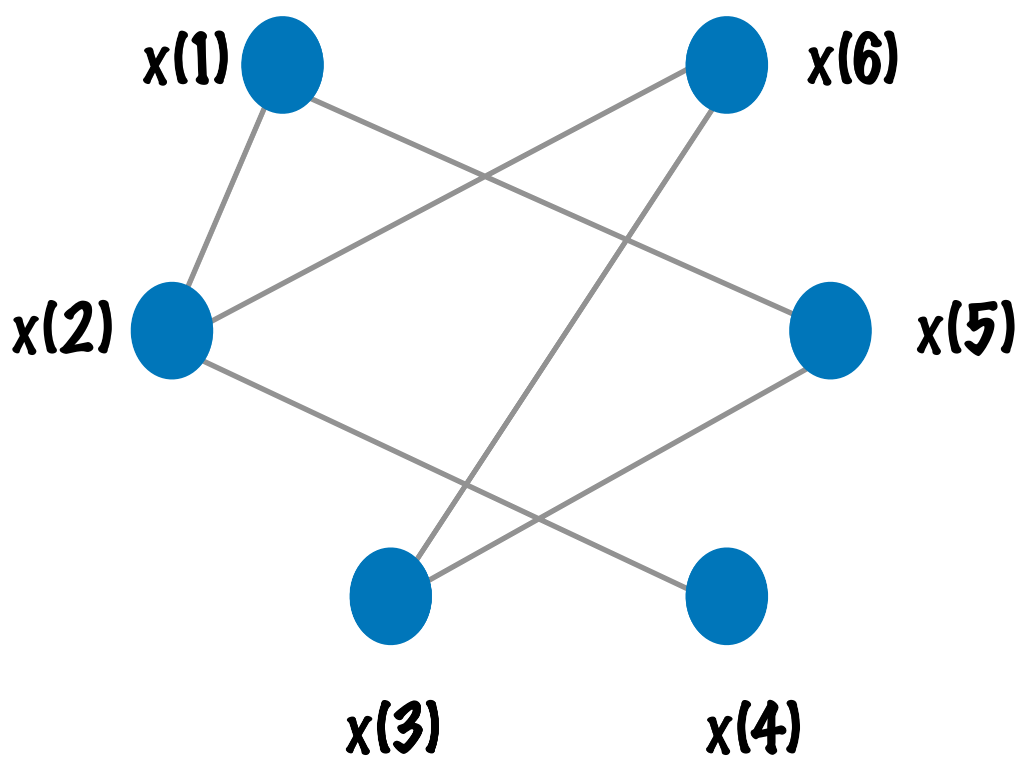


Parameterized Diameter Approximation

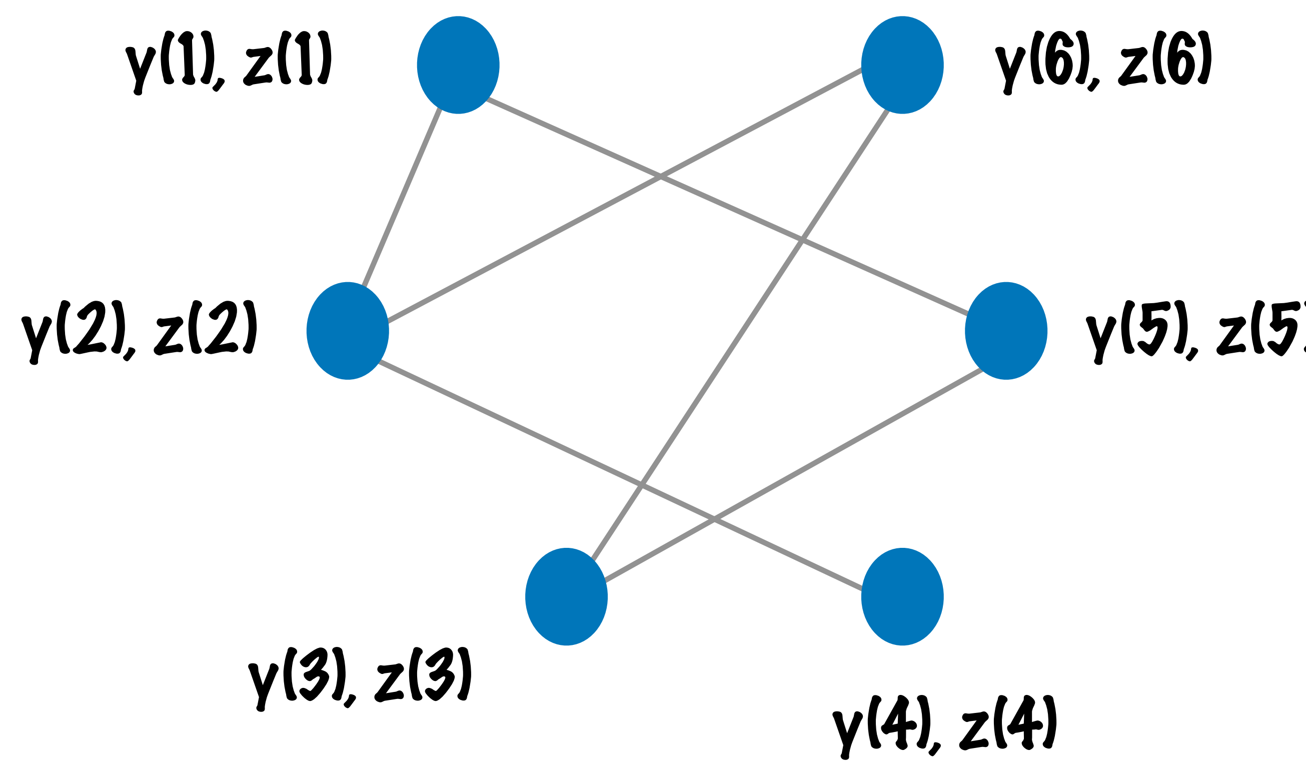
CONGEST

Phase 2: 2k+1 rounds

* In each round,
    * Every vertex v sends y(v) (the min x(u)) and z(v) (the max x(u)) that it has seen so far to its neighbours



. . .



After the end of Phase 2, each vertex v returns

* SMALL if y(v) = z(v) and LARGE if y(v) ≠ z(v)

Parameterized Diameter Approximation

CONGEST

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if y(v) = z(v) & LARGE if y(v) ≠ z(v)

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

* If dia $\leq$ k, then x(v)s are identical

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if y(v) = z(v) & LARGE if y(v) ≠ z(v)

O(k) rounds

* If dia ≤ k, then x(v)s are identical
    * All vertices report SMALL

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID $x(v)$ in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends $y(v)$ (min $x(u)$) and $z(v)$ (max $x(u)$)

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

* If dia $\leq$ k, then $x(v)$s are identical
  * All vertices report SMALL
* If $k+1 \leq$ dia $\leq 2k$, then $y(v)$s are identical, $z(v)$s are identical

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID $x(v)$ in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends $y(v)$ (min $x(u)$) and $z(v)$ (max $x(u)$)

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

* If dia $\leq$ k, then $x(v)$s are identical
  * All vertices report SMALL
* If $k+1 \leq$ dia $\leq 2k$, then $y(v)$s are identical, $z(v)$s are identical
  * All vertices report SMALL or all vertices report LARGE

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID $x(v)$ in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends $y(v)$ (min $x(u)$) and $z(v)$ (max $x(u)$)

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

$O(k)$ rounds

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID $x(v)$ in its k-hop neighbourhood

Phase 2: 2k+1 rounds

* In each round, every v sends $y(v)$ (min $x(u)$) and $z(v)$ (max $x(u)$)

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

* If dia ≥ 2k+1, consider a vertex v

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

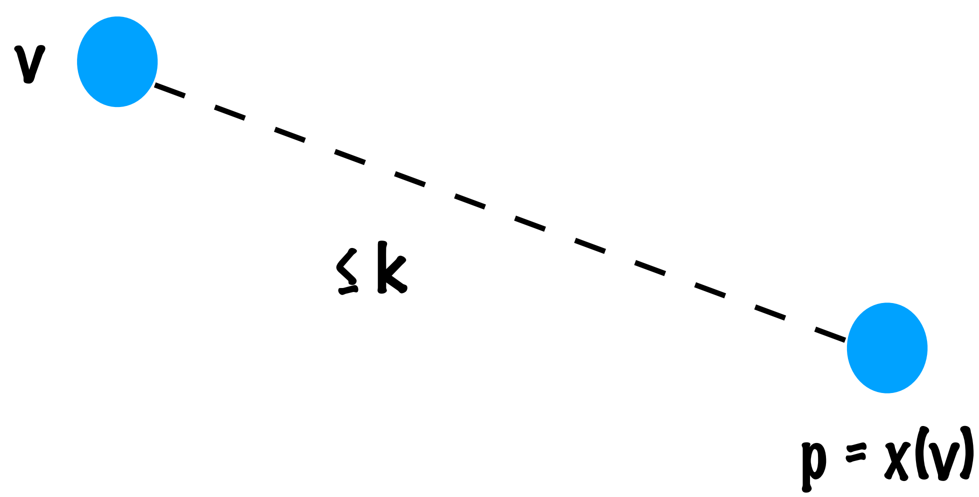
Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if y(v) = z(v) & LARGE if y(v) ≠ z(v)

O(k) rounds

* If dia ≥ 2k+1, consider a vertex v



Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID $x(v)$ in its k-hop neighbourhood

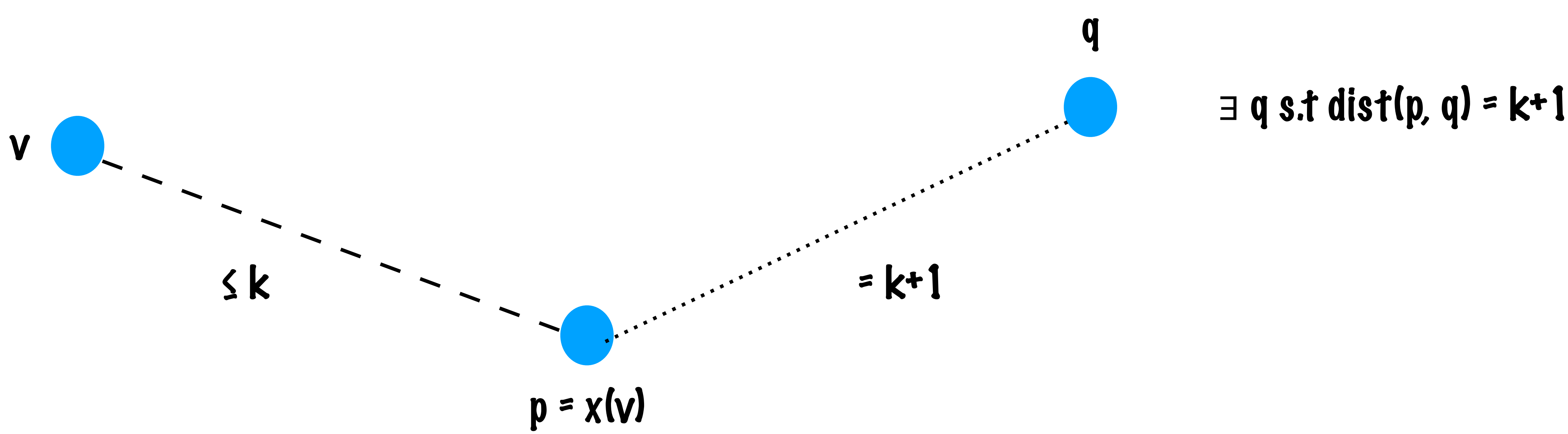
Phase 2: 2k+1 rounds

* In each round, every v sends $y(v)$ (min $x(u)$) and $z(v)$ (max $x(u)$)

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

* If dia ≥ 2k+1, consider a vertex v



Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

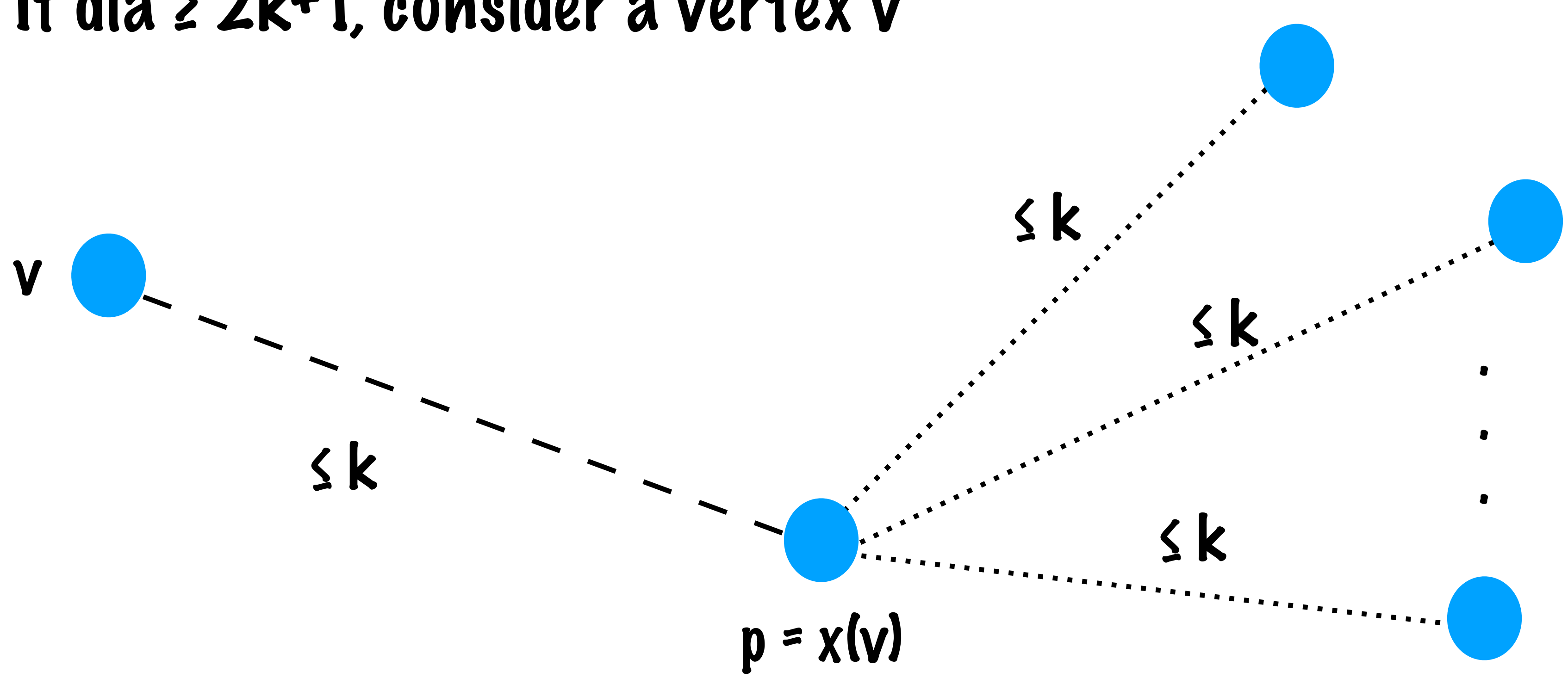
Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if y(v) = z(v) & LARGE if y(v) ≠ z(v)

O(k) rounds

* If dia ≥ 2k+1, consider a vertex v



dia ≤ 2k+1

Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID x(v) in its k-hop neighbourhood

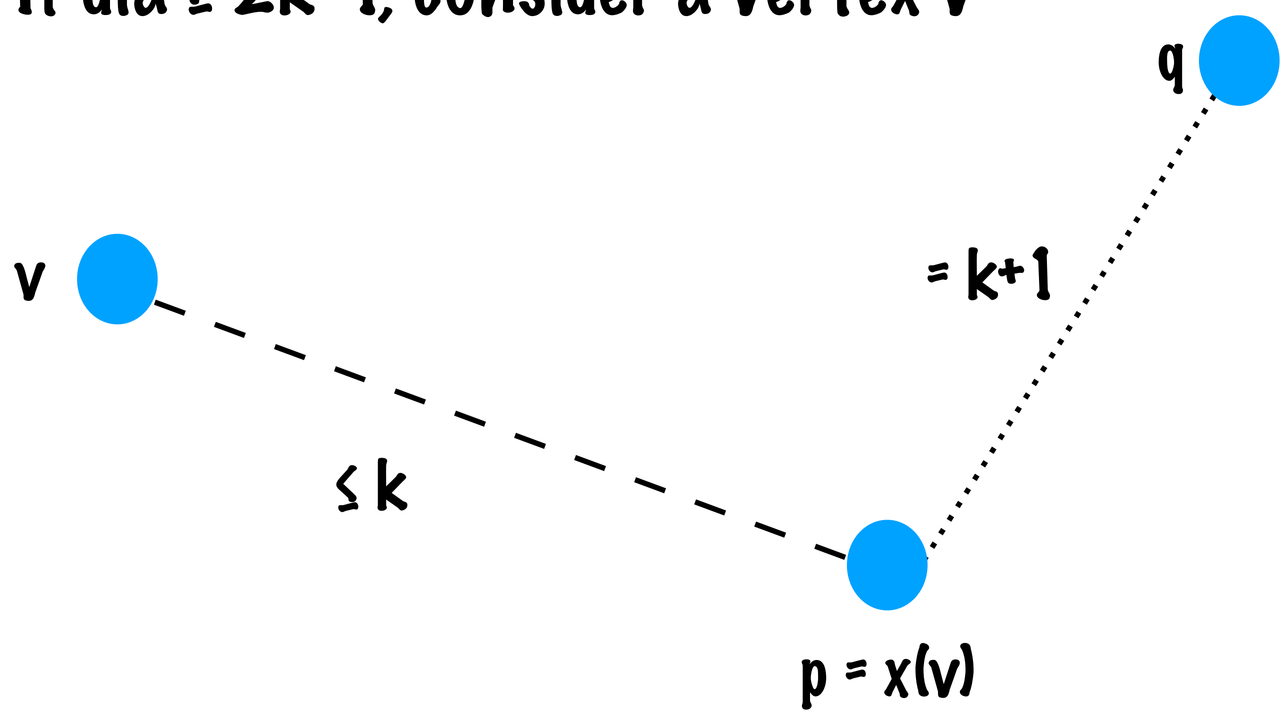
Phase 2: 2k+1 rounds

* In each round, every v sends y(v) (min x(u)) and z(v) (max x(u))

After the end of Phase 2, each v returns SMALL if y(v) = z(v) & LARGE if y(v) ≠ z(v)

O(k) rounds

* If dia ≥ 2k+1, consider a vertex v



Parameterized Diameter Approximation

CONGEST

Phase 1: k rounds

* In each round, every v sends min ID

At end of Phase 1, each v has min ID $x(v)$ in its k-hop neighbourhood

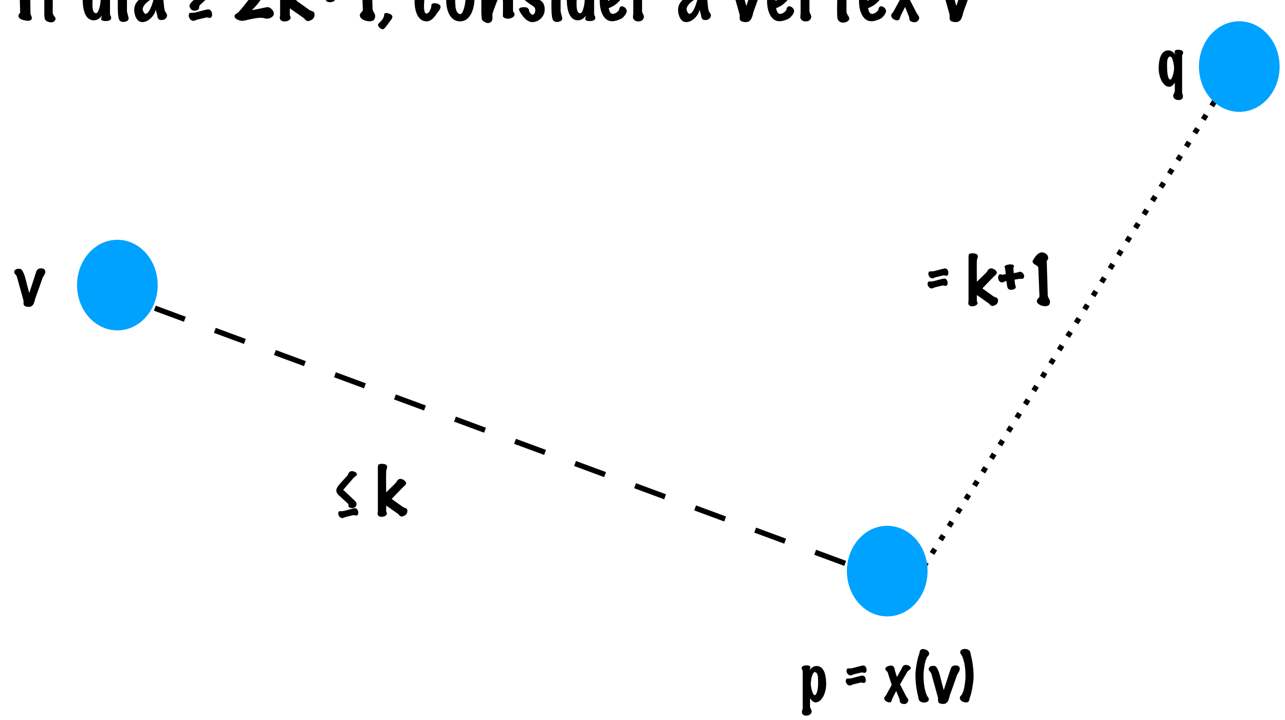
Phase 2: 2k+1 rounds

* In each round, every v sends $y(v)$ (min $x(u)$) and $z(v)$ (max $x(u)$)

After the end of Phase 2, each v returns SMALL if $y(v) = z(v)$ & LARGE if $y(v) \neq z(v)$

O(k) rounds

* If dia ≥ 2k+1, consider a vertex v



* After k rounds of Phase 1, $x(q) \neq p$
* After k+1 rounds of Phase 2,
  * $y(p) \neq z(p)$
* After next k rounds of Phase 2,
  * $y(v) \neq z(v)$
  * v outputs LARGE

Concluding Remarks

Problem	Model	Upper Bound	Lower Bound
Vertex Cover	CONGEST	$O(k^2)$	$\Omega(k^2/\log k \log n)$
	LOCAL	$O(k)$	$\Omega(k)$
Independent Set	CONGEST	$O(n^2)$	$\Omega(n^2/\log^2 n)$
	LOCAL	$O(k)$	$\Omega(k)$

Note: Lower bounds hold even when n is arbitrarily larger than k

Thank you!

Questions?