Approximation Algorithms
Chapter 6: Review and Relay Placement

Sándor P. Fekete

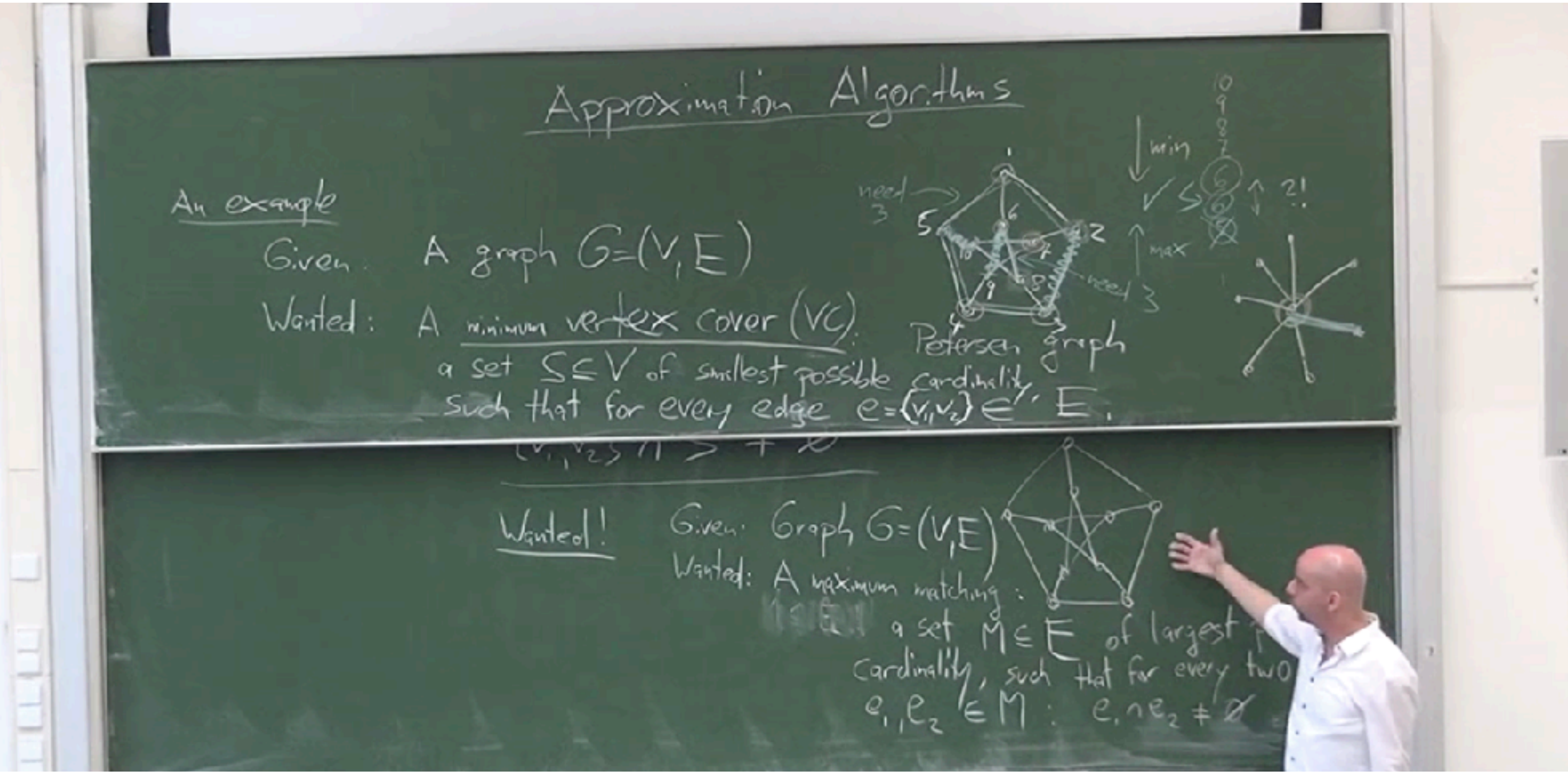
Algorithms Division
Department of Computer Science
TU Braunschweig

1. Introduction
2. Review
3. Extra Packing: Dispersion
4. Extra Tours: Lawn Mowing
5. Relay Placement
6. Coordinated Motion Planning

1. Introduction
2. Review
3. Extra Packing: Dispersion
4. Extra Tours: Lawn Mowing
5. Relay Placement
6. Coordinated Motion Planning

Review: Matching & Vertex Cover

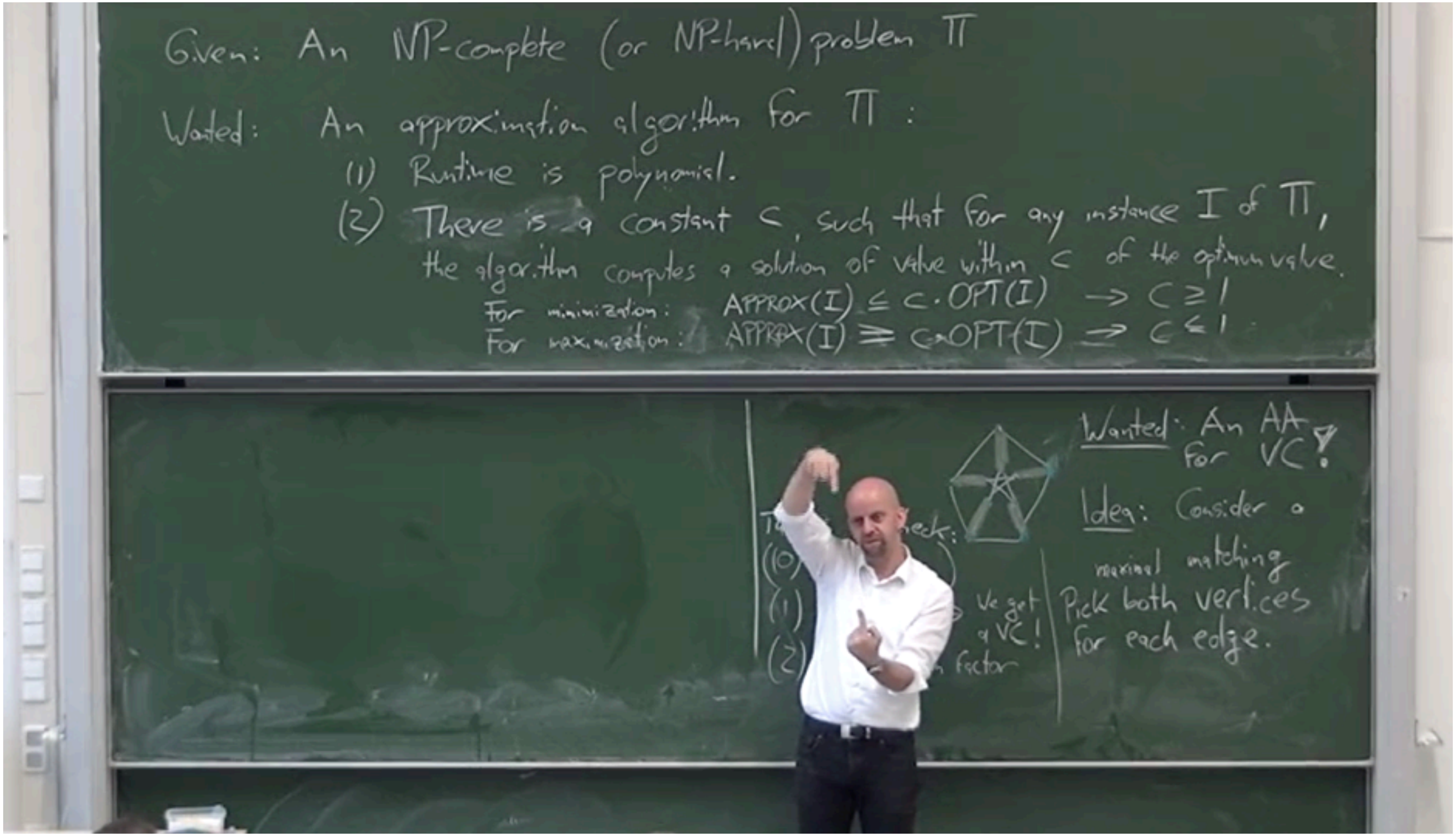
Review: Matching & Vertex Cover



Review: Approximation Algorithms

Technische Universität Braunschweig

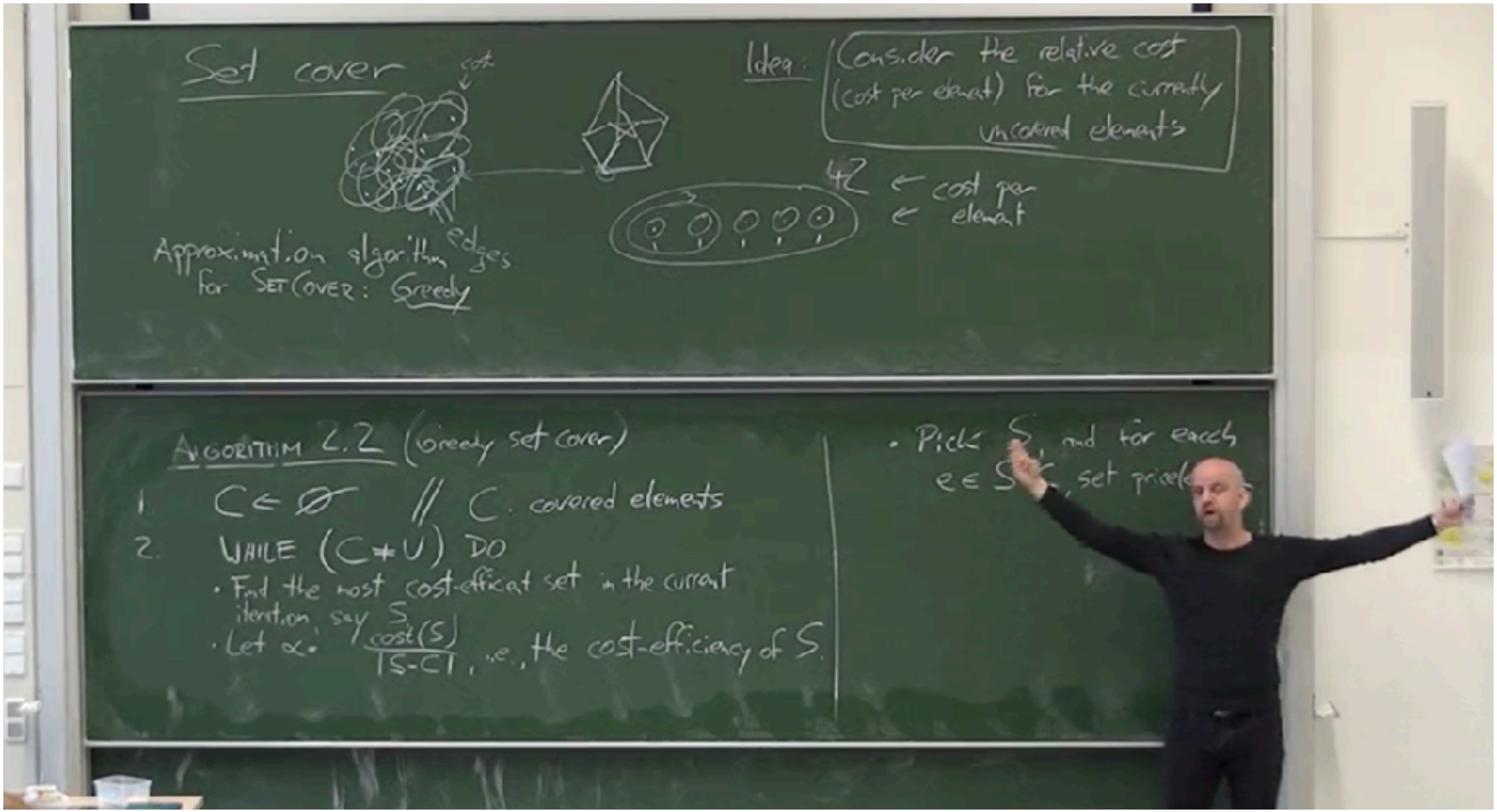
Review: Approximation Algorithms



Review: Set Cover

Technische Universität Braunschweig

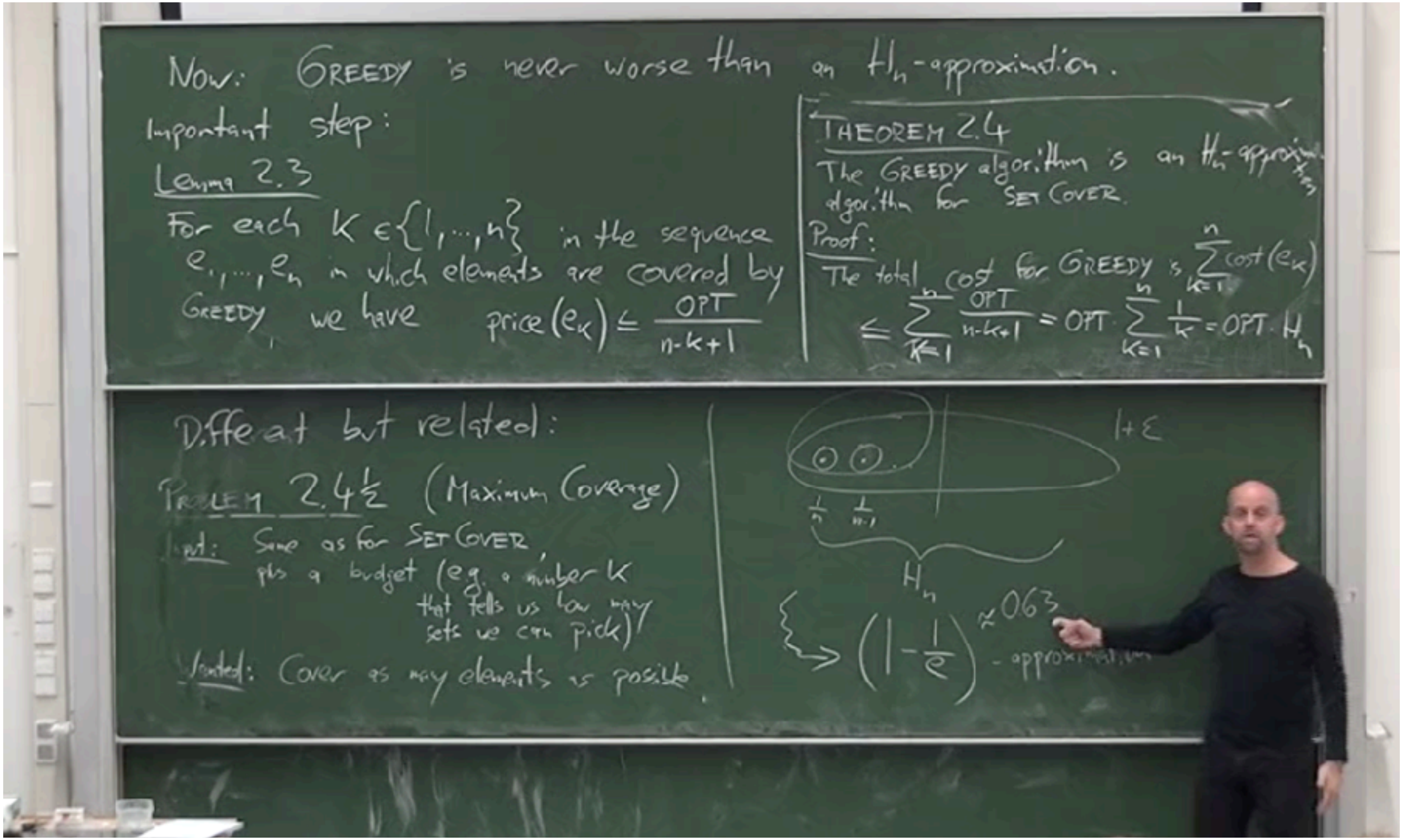
Review: Set Cover



Review: Maximum Coverage

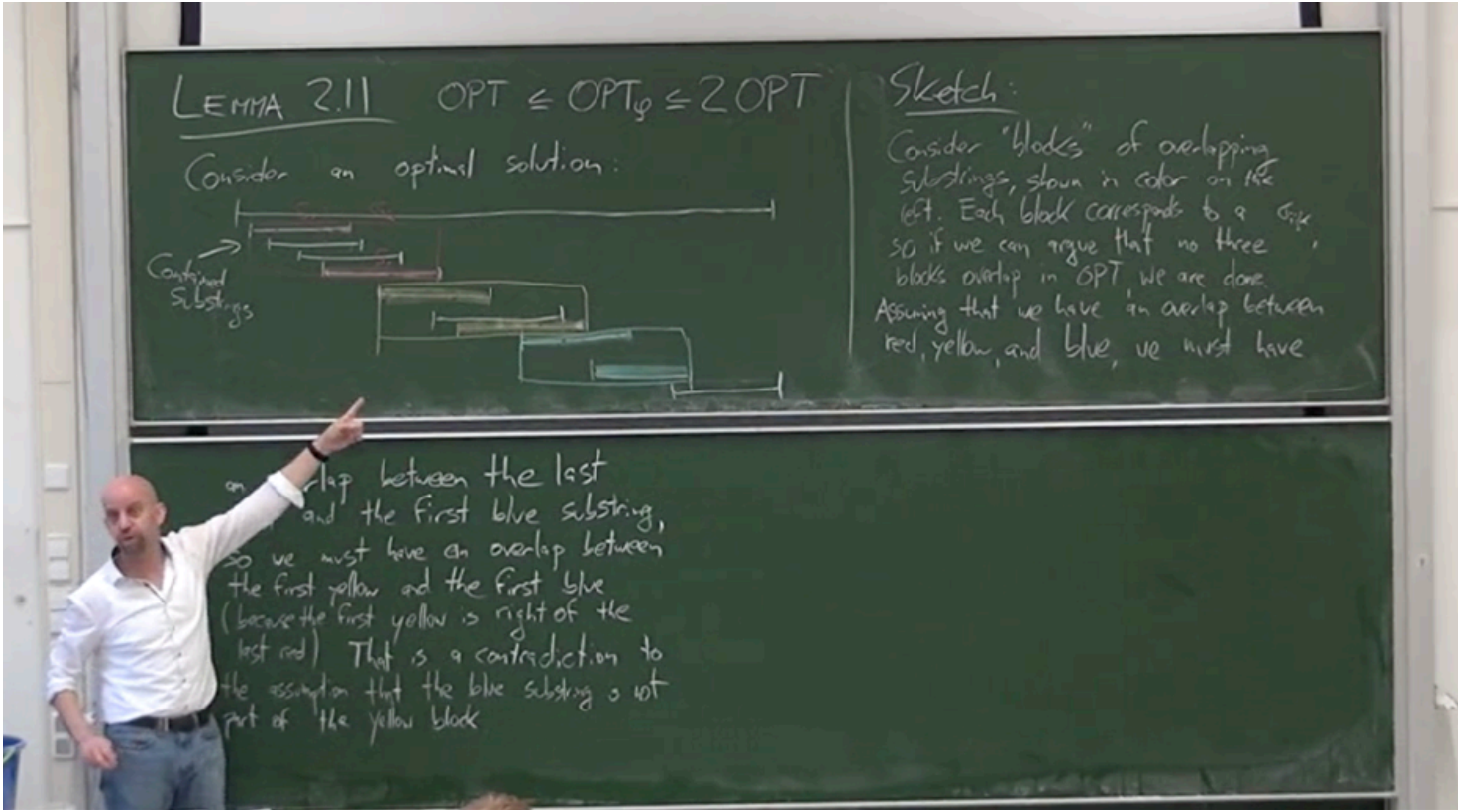
Technische Universität Braunschweig

Review: Maximum Coverage



Review: Shortest Superstring

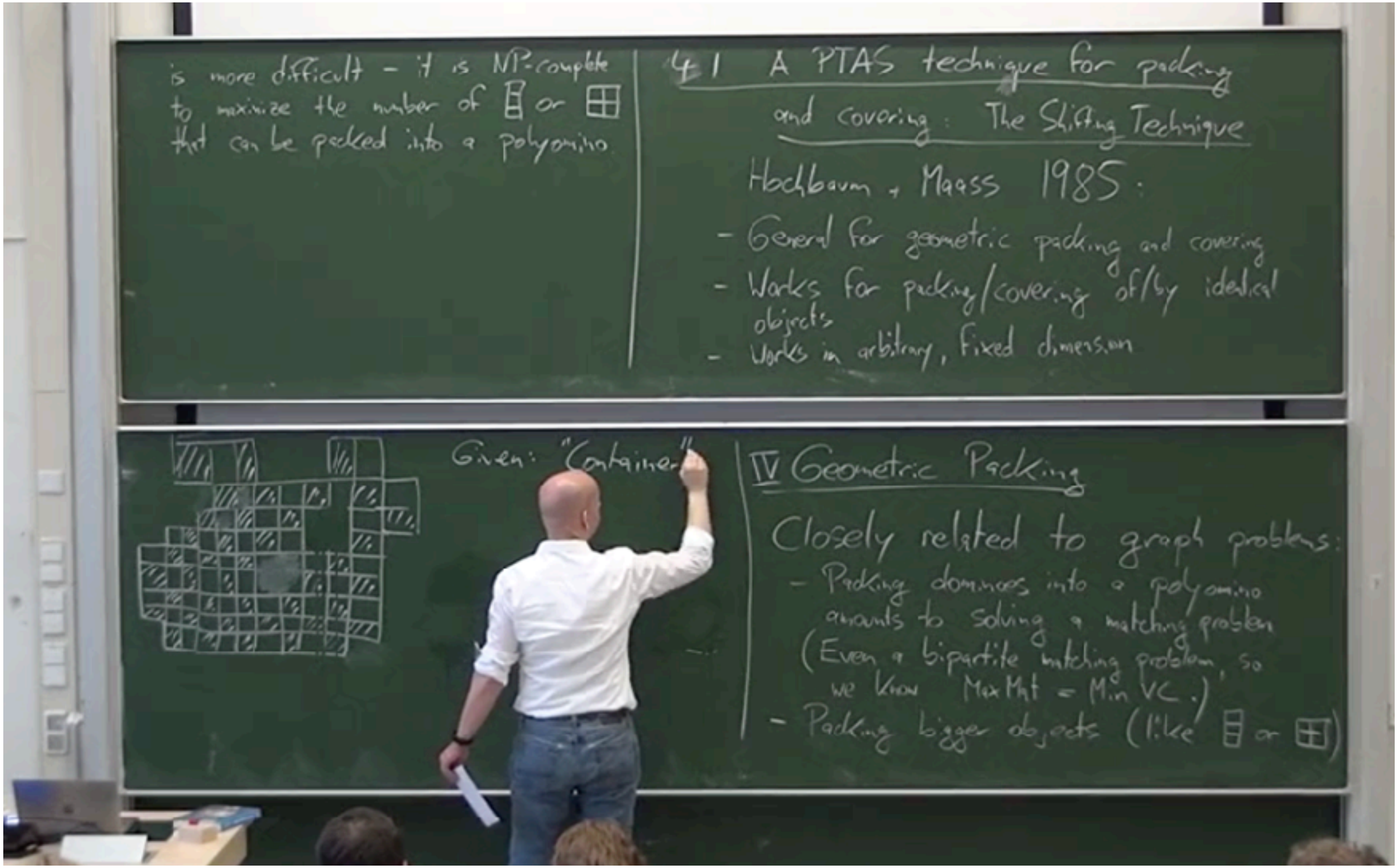
Review: Shortest Superstring



Technische Universität Braunschweig

Review: Packing

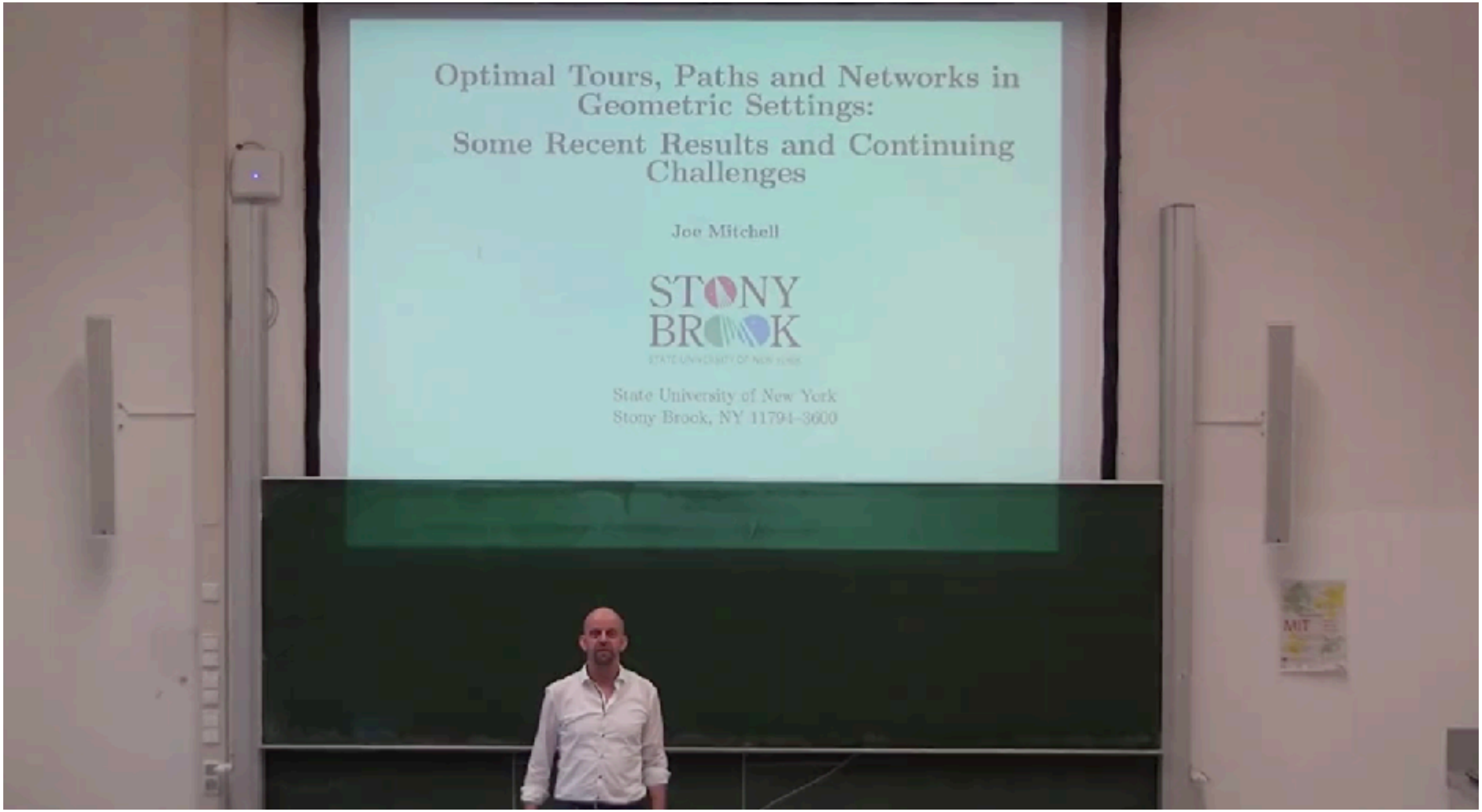
Review: Packing



Technische Universität Braunschweig

Review: Tour Problems

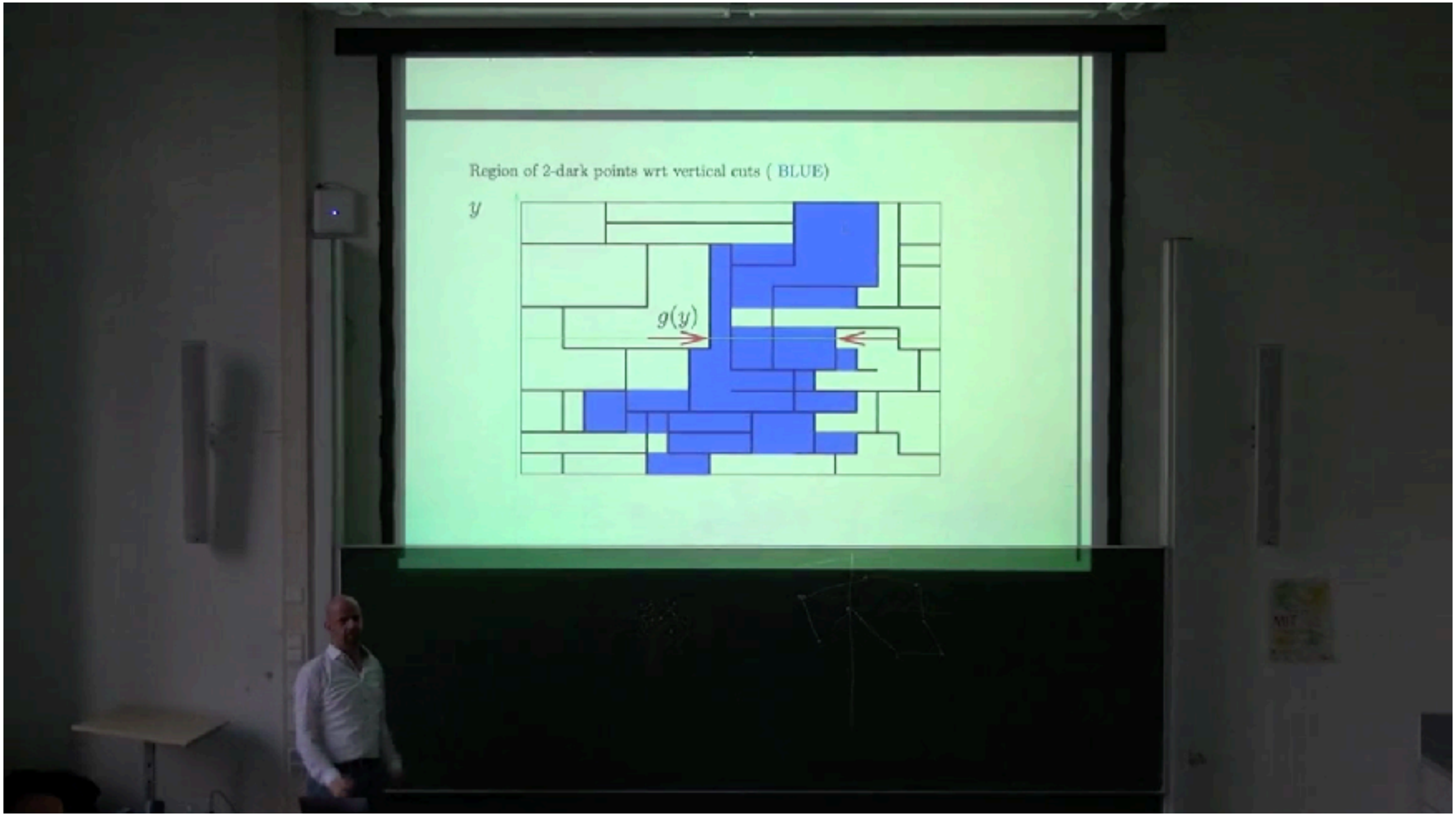
Review: Tour Problems



Technische Universität Braunschweig

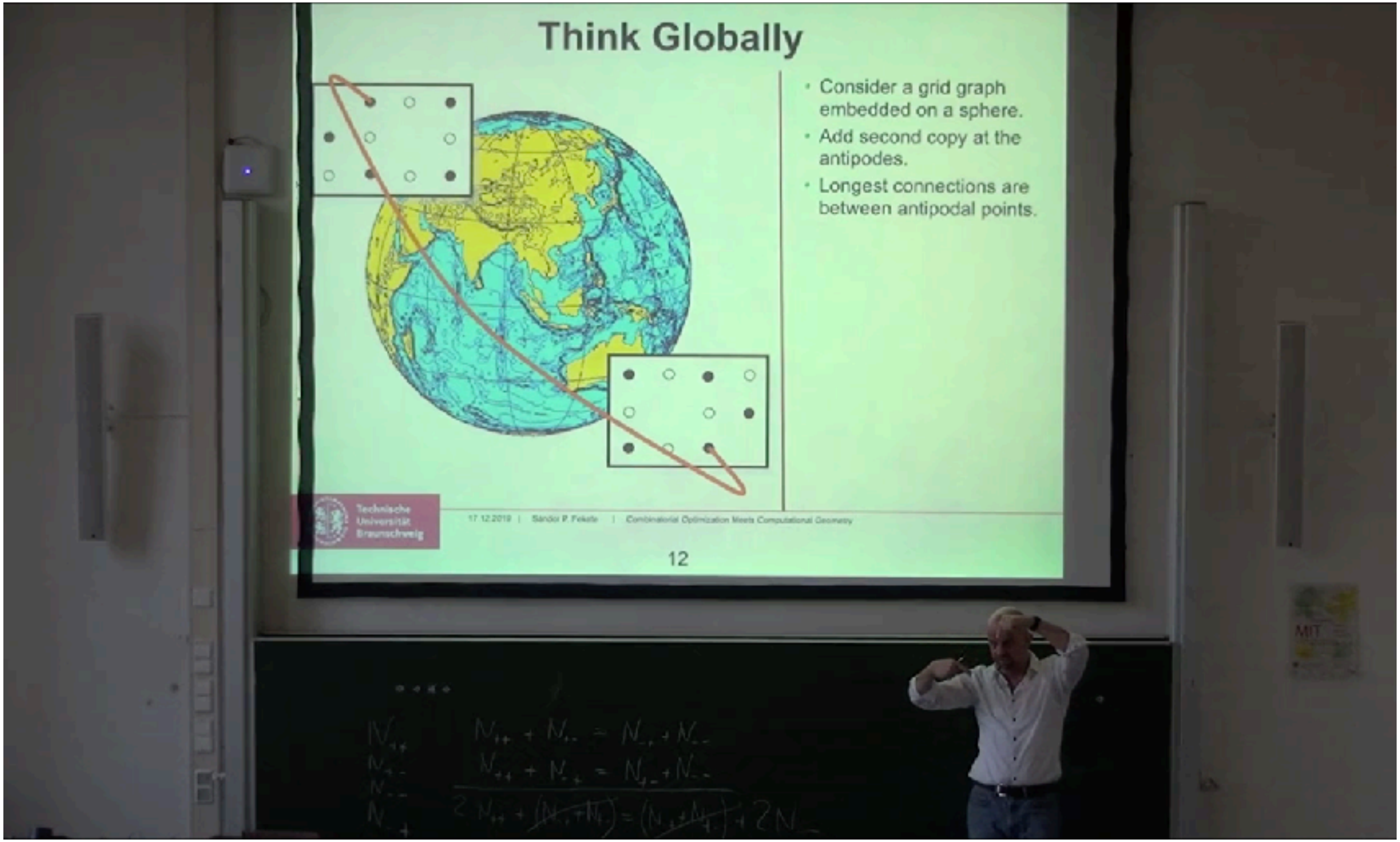
Review: Tour Problems

Review: Tour Problems



Review: Tour Problems

Review: Tour Problems

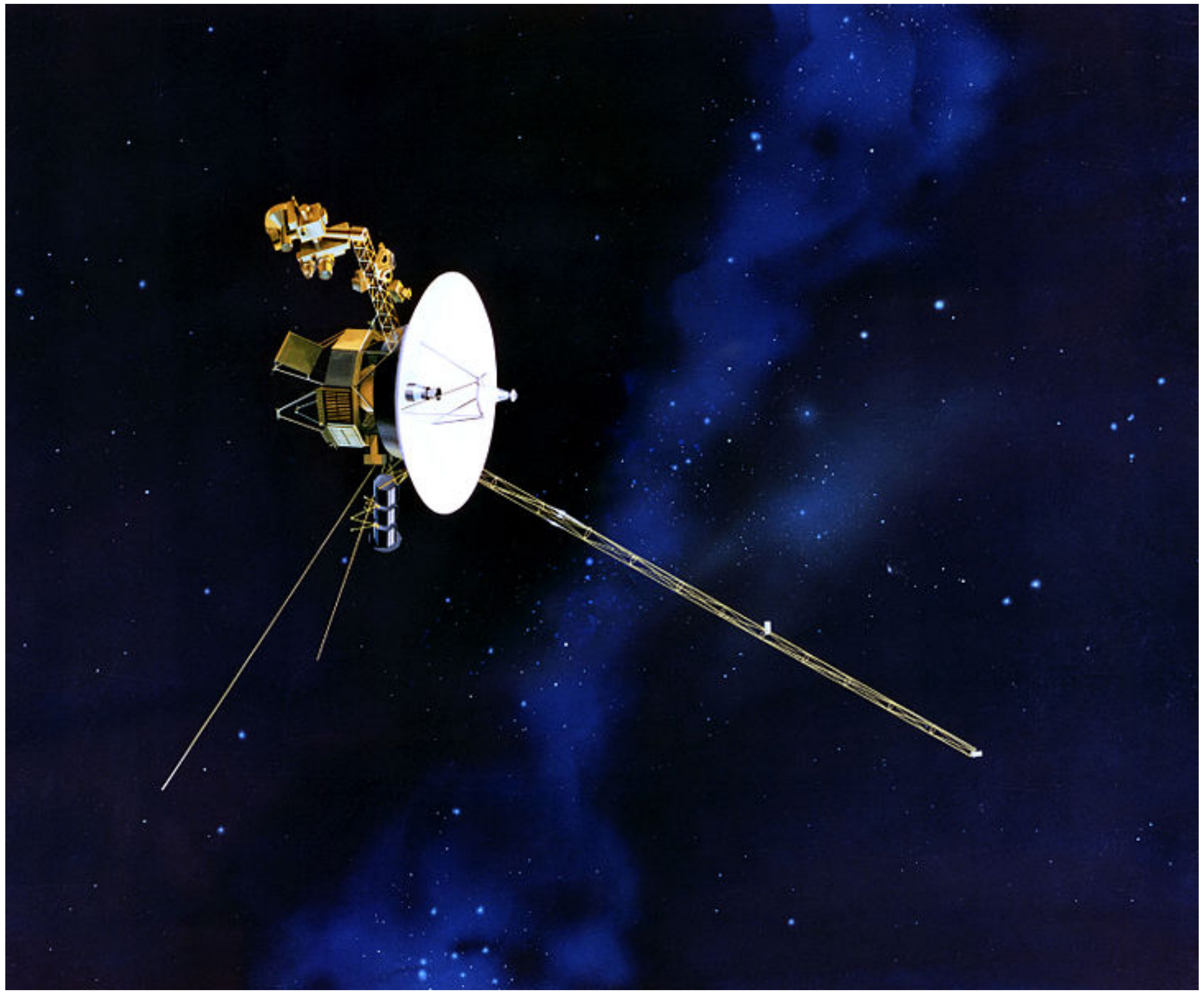


The Freeze-Tag Problem: How to Wake Up a Swarm of Robots

Estie Arkin[1]
Michael Bender[1]
Sándor Fekete[2]
Joe Mitchell[1]
Martin Skutella[3]

[1] University at Stony Brook
[2] TU Braunschweig
[3] TU Berlin

Motivation



- Highly focused antennas.
- Expensive rotations.

How can we quickly distribute information from one satellite to all others?

d.krupke@tu-bs.de
Institute of Operating Systems and Computer Networks

Technische Universität Braunschweig

Student Presentation

Minimum Scan Cover with Angular Transition Costs



Sándor Fekete

Linda Kleist

Dominik Krupke

1. Introduction
2. Review
3. Extra Packing: Dispersion
4. Extra Tours: Lawn Mowing
5. Relay Placement
6. Coordinated Motion Planning

Dispersion (1)

Algorithmica (2001) 30: 451–470
DOI: 10.1007/s00453-001-0022-x

Algorithmica

Approximation of Geometric Dispersion Problems[1]

C. Baur[2] and S. P. Fekete[3]

Abstract. We consider problems of distributing a number of points within a polygonal region P, such that the points are "far away" from each other. Problems of this type have been considered before for the case where the possible locations form a discrete set. Dispersion problems are closely related to packing problems. While Hochbaum and Maass [20] have given a polynomial-time approximation scheme for packing, we show that geometric dispersion problems cannot be approximated arbitrarily well in polynomial time, unless P = NP. A special case of this observation solves an open problem by Rosenkrantz et al. [31]. We give a $\frac{2}{3}$ approximation algorithm for one version of the geometric dispersion problem. This algorithm is strongly polynomial in the size of the input, i.e., its running time does not depend on the area of P. We also discuss extensions and open problems.

Key Words. Packing, Dispersion, Location problems, Geometric optimization, Bounds on approximation factors.

Technische Universität Braunschweig

Dispersion (2)

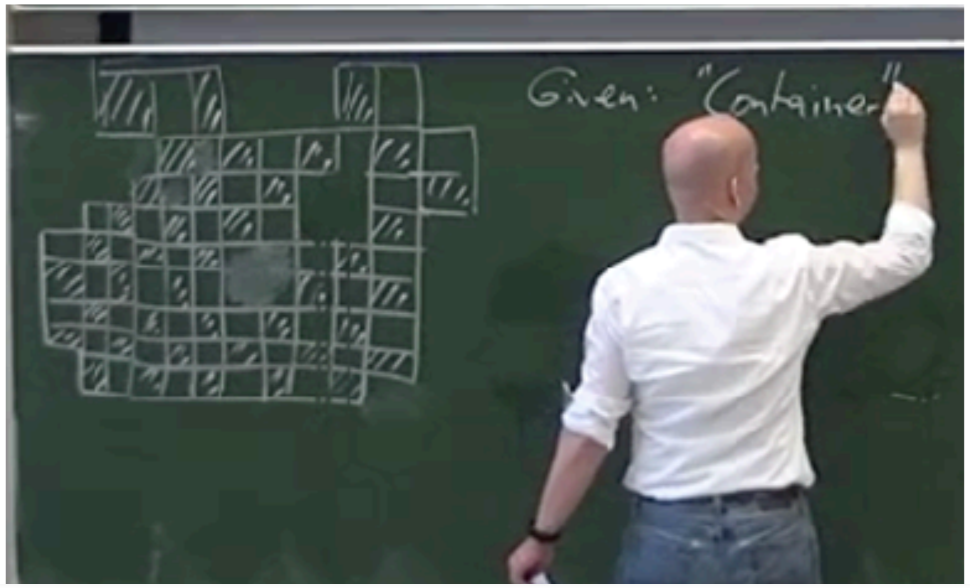
Dispersion (2)

Dispersion (2)

PACK(k, L)
Input: A polygonal region P with n vertices, a parameter k, a parameter L.
Question: Can k many L-squares be packed into P?

Dispersion (2)

PACK(k, L)
Input: A polygonal region P with n vertices, a parameter k, a parameter L.
Question: Can k many L-squares be packed into P?

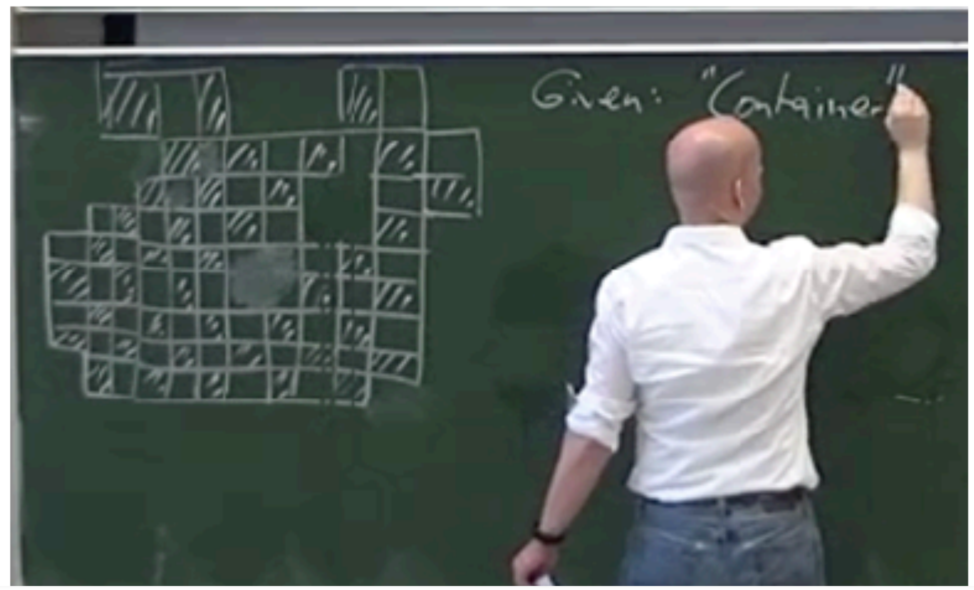


$\max_k$PACK(L)
Input: A polygonal region P with n vertices.
Task: Pack k many L-squares into P, such that k is as big as possible.

Technische Universität Braunschweig

Dispersion (2)

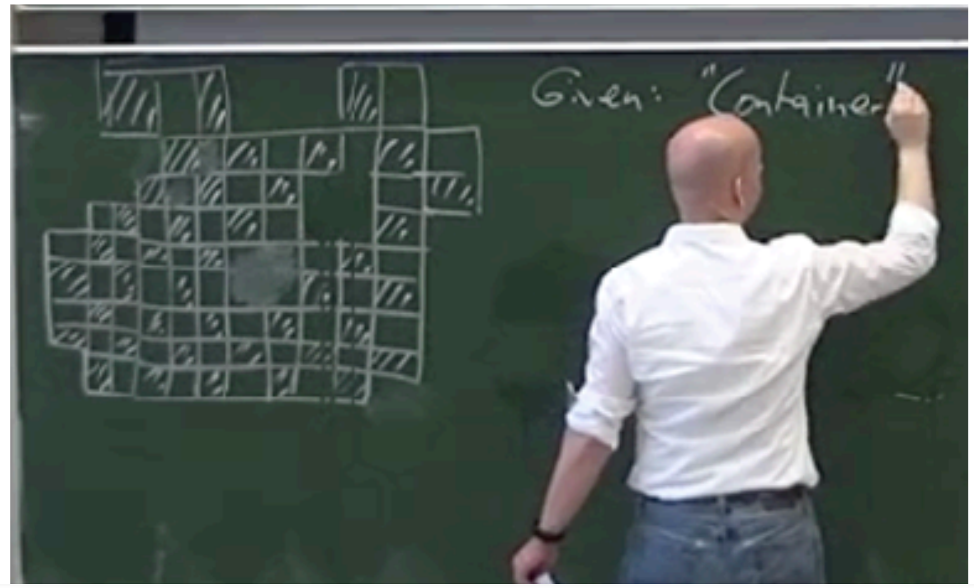
PACK(k, L)
Input: A polygonal region P with n vertices, a parameter k, a parameter L.
Question: Can k many L-squares be packed into P?



$\max_k$PACK(L)
Input: A polygonal region P with n vertices.
Task: Pack k many L-squares into P, such that k is as big as possible.

Dispersion (2)

$\text{PACK}(k, L)$
Input: A polygonal region P with n vertices, a parameter k, a parameter L.
Question: Can k many L-squares be packed into P?

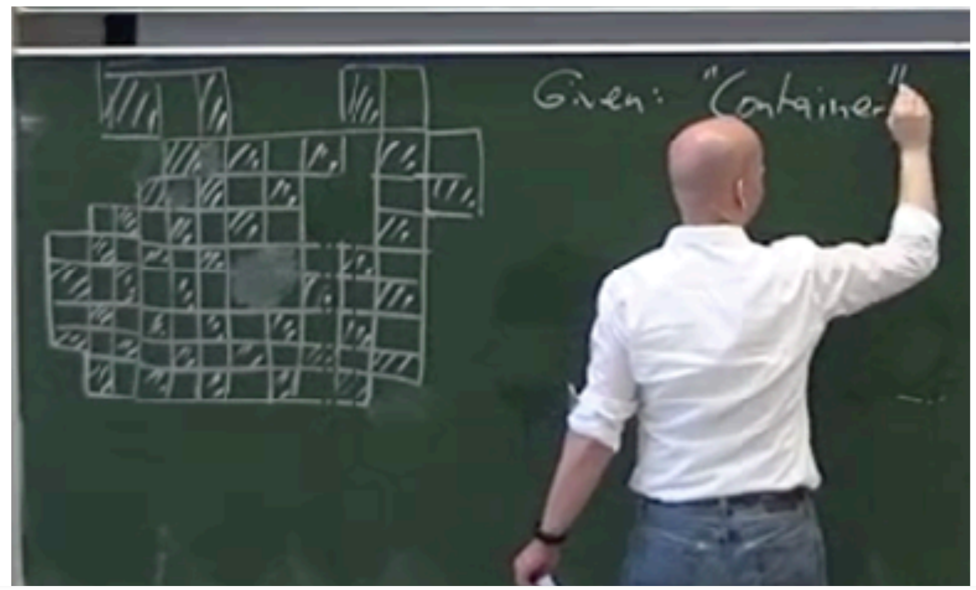


$\max_k \text{PACK}(L)$
Input: A polygonal region P with n vertices.
Task: Pack k many L-squares into P, such that k is as big as possible.

$\max_L \text{PACK}(k)$
Input: A polygonal region P with n vertices.
Task: Pack k many $L \times L$ squares into P, such that L is as big as possible.

Dispersion (2)

PACK(k, L)
Input: A polygonal region P with n vertices, a parameter k, a parameter L.
Question: Can k many L-squares be packed into P?



$\max_k$PACK(L)
Input: A polygonal region P with n vertices.
Task: Pack k many L-squares into P, such that k is as big as possible.

$\max_L$PACK(k)
Input: A polygonal region P with n vertices.
Task: Pack k many $L \times L$ squares into P, such that L is as big as possible.

Technische Universität Braunschweig

Dispersion (3)

Dispersion (3)

THEOREM 1. *Unless $P = NP$, there is no polynomial algorithm that finds a solution within more than $\frac{13}{14}$ of the optimum for rectilinear geometric dispersion with boundaries.*

Dispersion (3)

THEOREM 1. *Unless P = NP, there is no polynomial algorithm that finds a solution within more than $\frac{13}{14}$ of the optimum for rectilinear geometric dispersion with boundaries.*

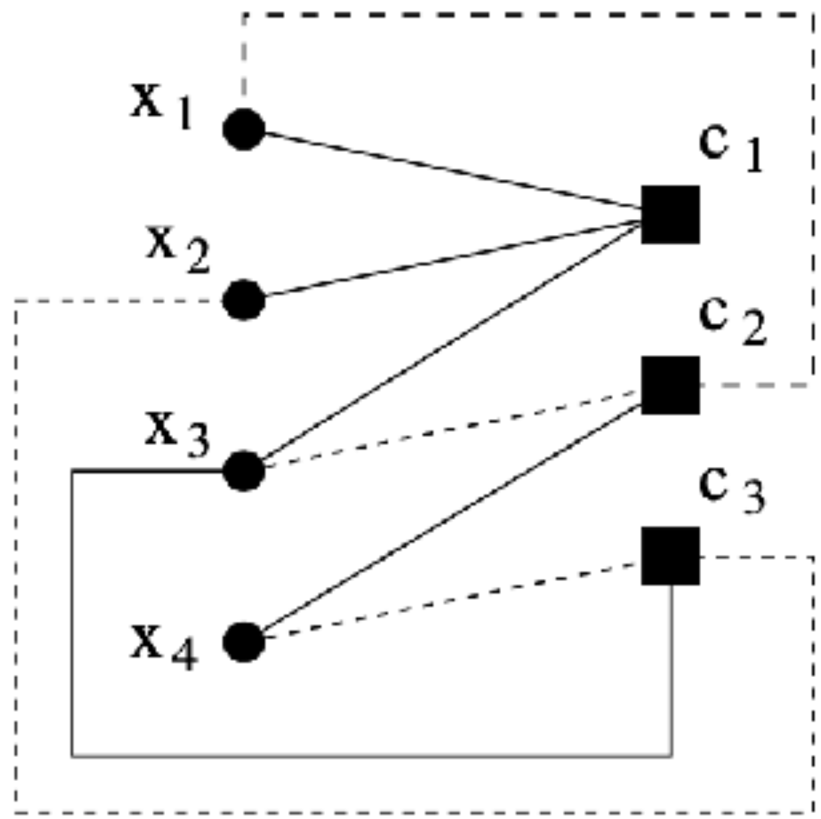


Fig. 1. The graph G_I representing the PLANAR 3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. Edges are distinguished according to the logical parity of the corresponding literals.

Dispersion (3)

THEOREM 1. *Unless P = NP, there is no polynomial algorithm that finds a solution within more than $\frac{13}{14}$ of the optimum for rectilinear geometric dispersion with boundaries.*

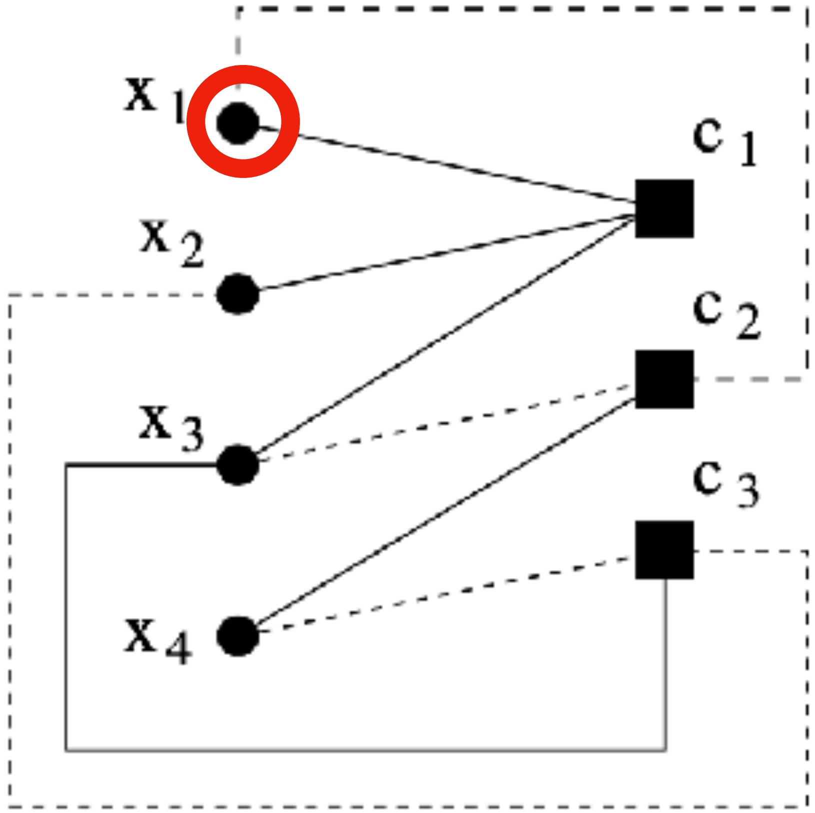


Fig. 1. The graph G_I representing the PLANAR 3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. Edges are distinguished according to the logical parity of the corresponding literals.

Technische Universität Braunschweig

Dispersion (3)

THEOREM 1. *Unless $P = NP$, there is no polynomial algorithm that finds a solution within more than $\frac{13}{14}$ of the optimum for rectilinear geometric dispersion with boundaries.*

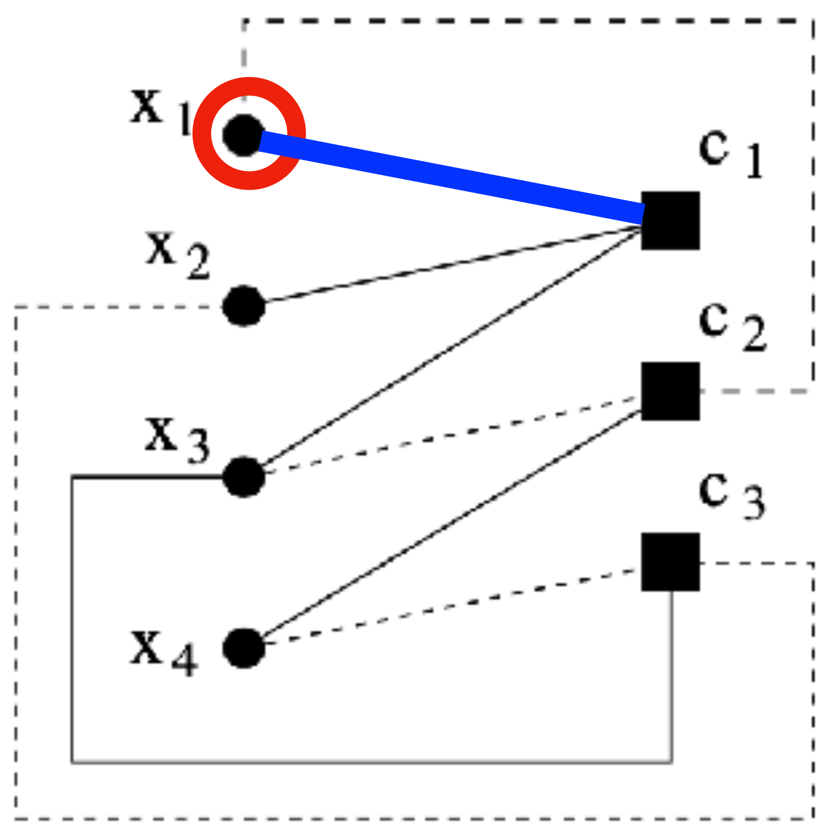


Fig. 1. The graph G_I representing the PLANAR 3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. Edges are distinguished according to the logical parity of the corresponding literals.

Technische Universität Braunschweig

Dispersion (3)

THEOREM 1. *Unless P = NP, there is no polynomial algorithm that finds a solution within more than $\frac{13}{14}$ of the optimum for rectilinear geometric dispersion with boundaries.*

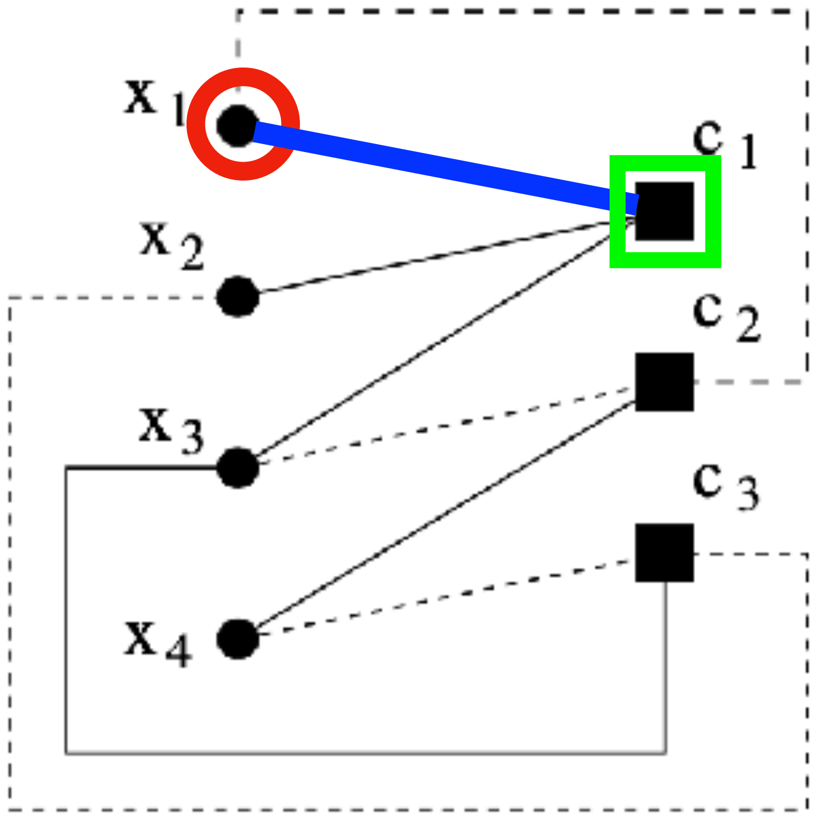
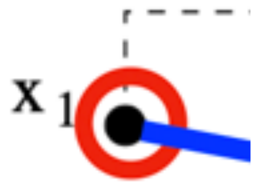


Fig. 1. The graph G_I representing the PLANAR 3SAT instance $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_2 \vee x_3 \vee \bar{x}_4)$. Edges are distinguished according to the logical parity of the corresponding literals.

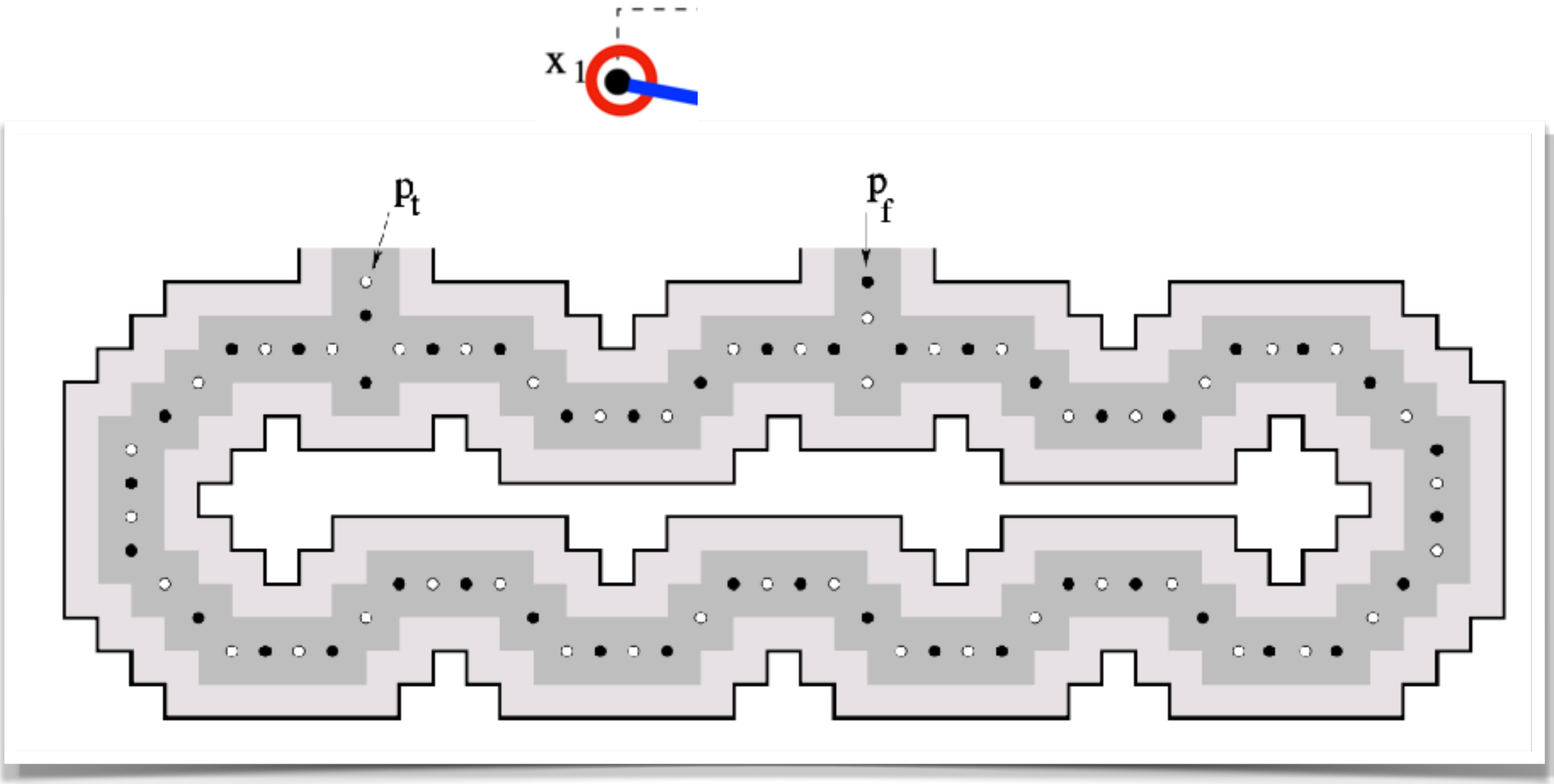
Variables (1)

Technische Universität Braunschweig

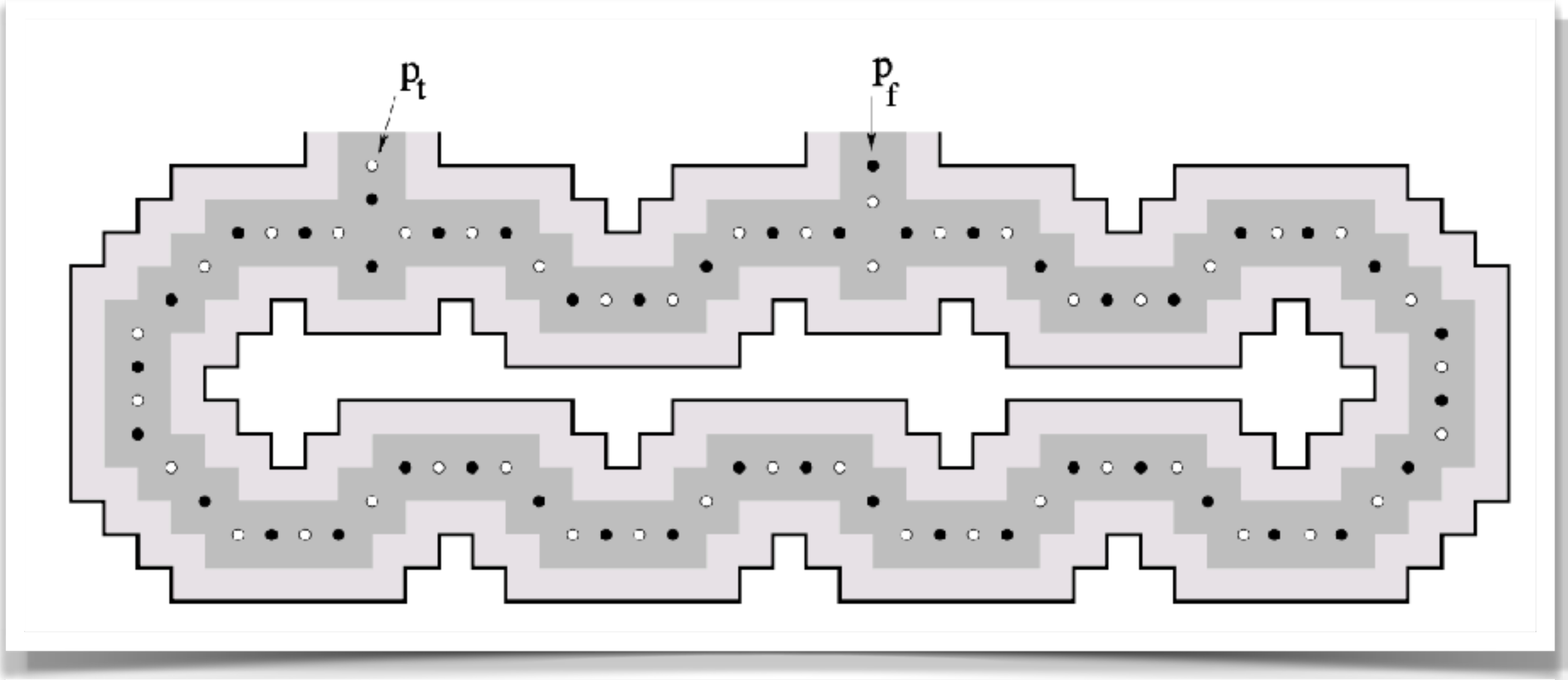
Variables (1)

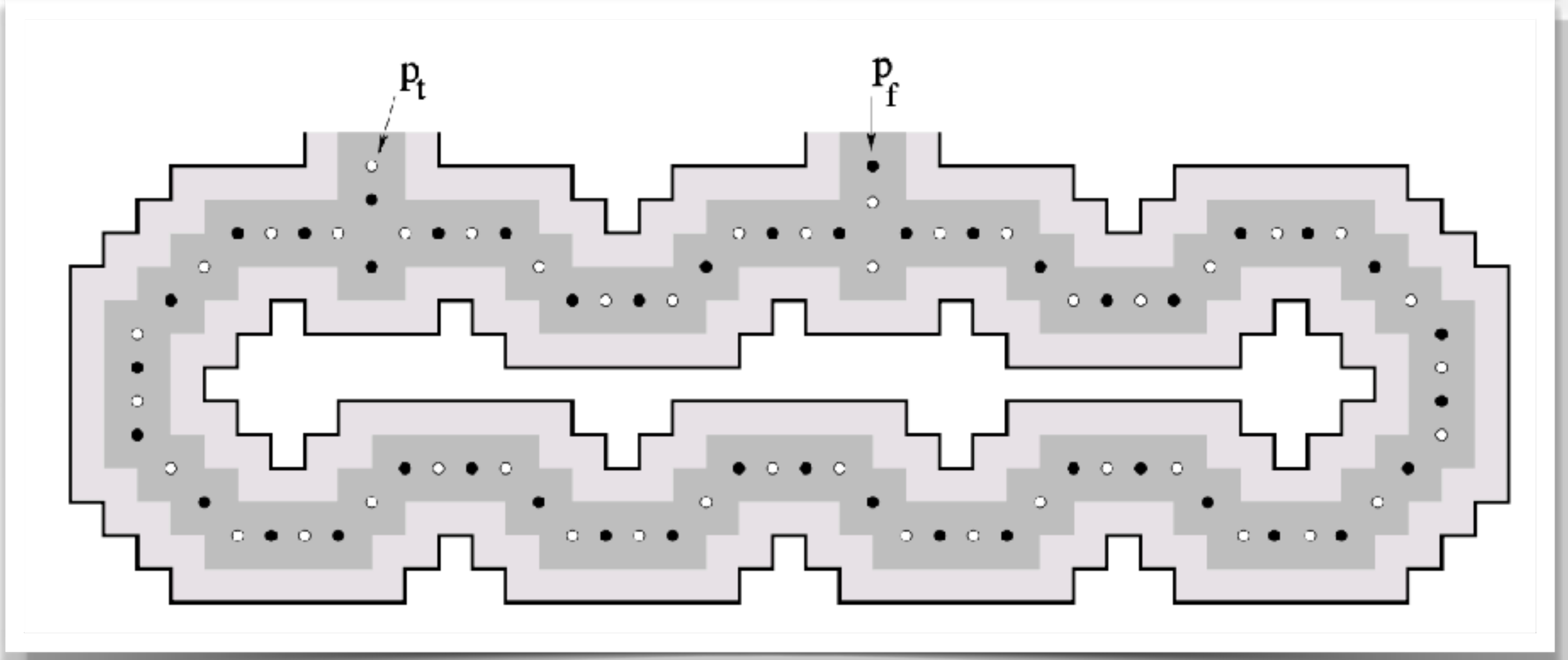


Variables (1)

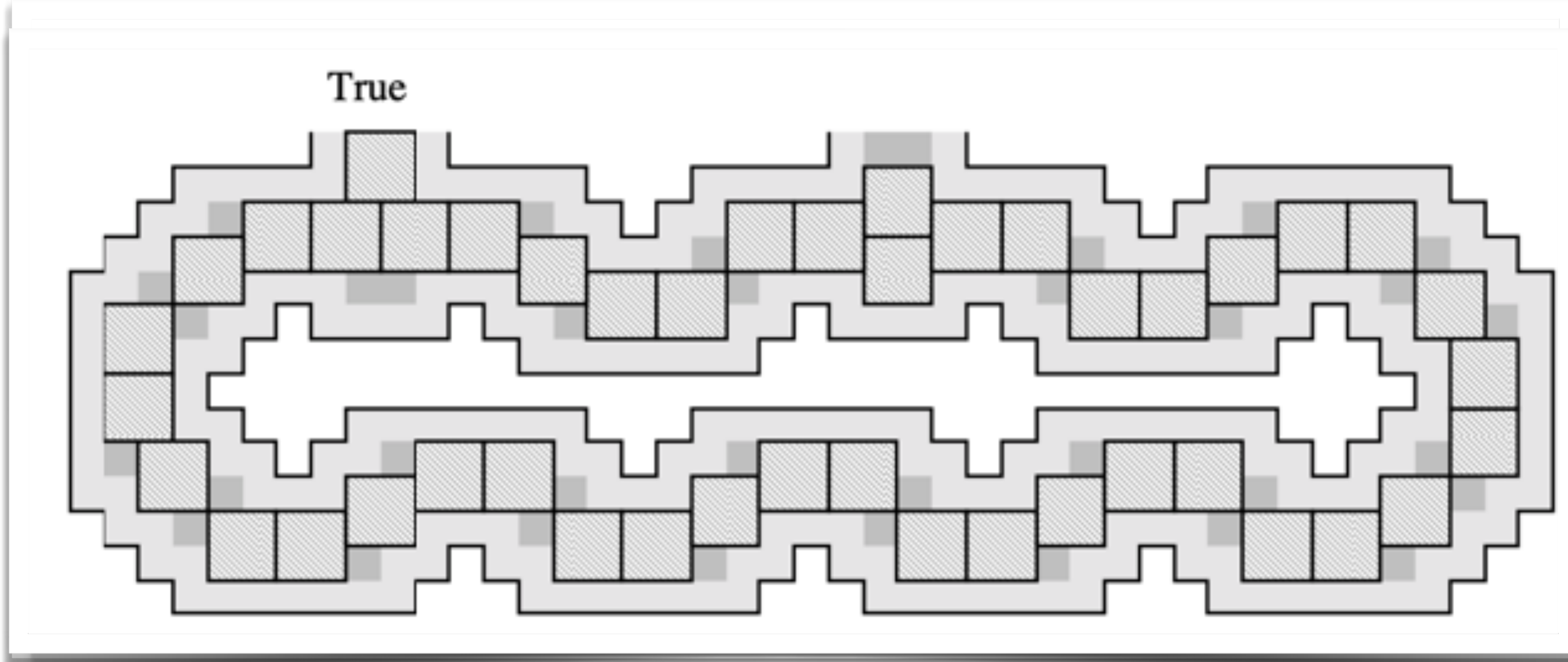


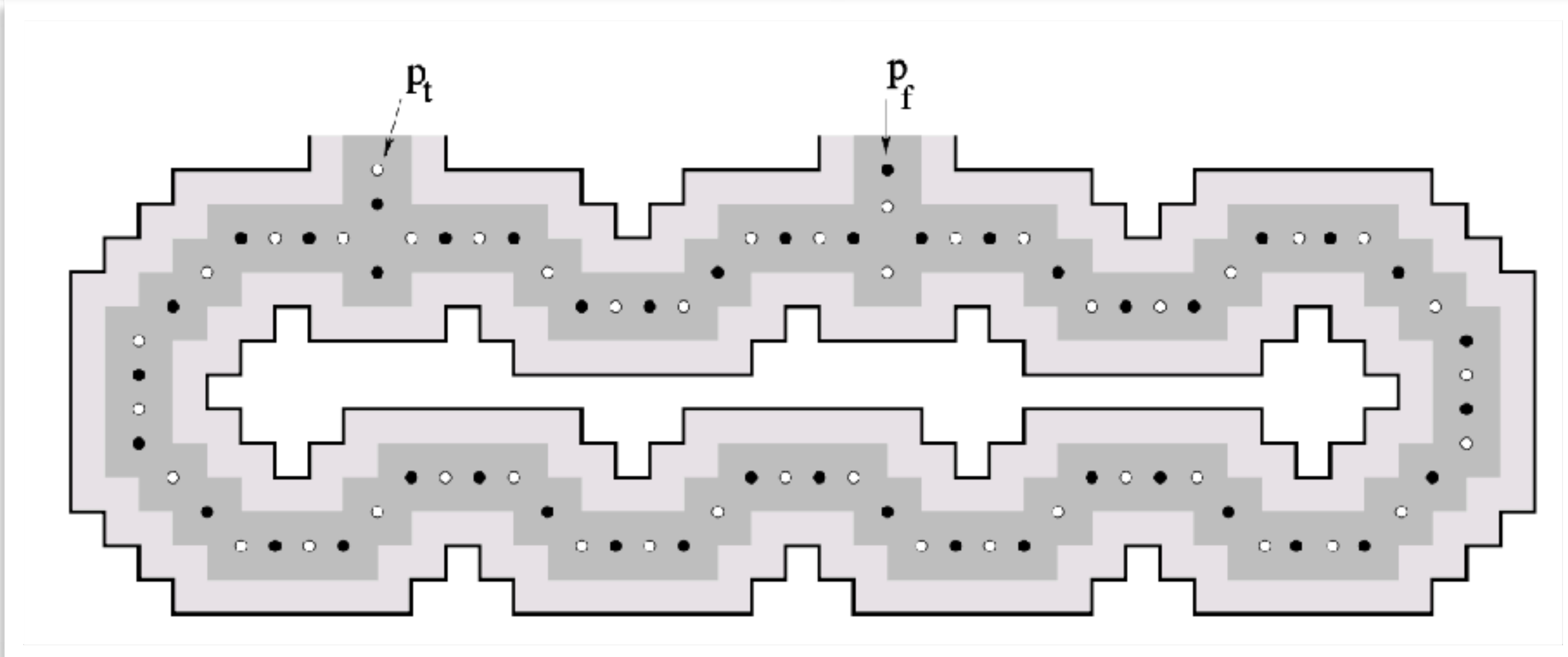
Variables (2)





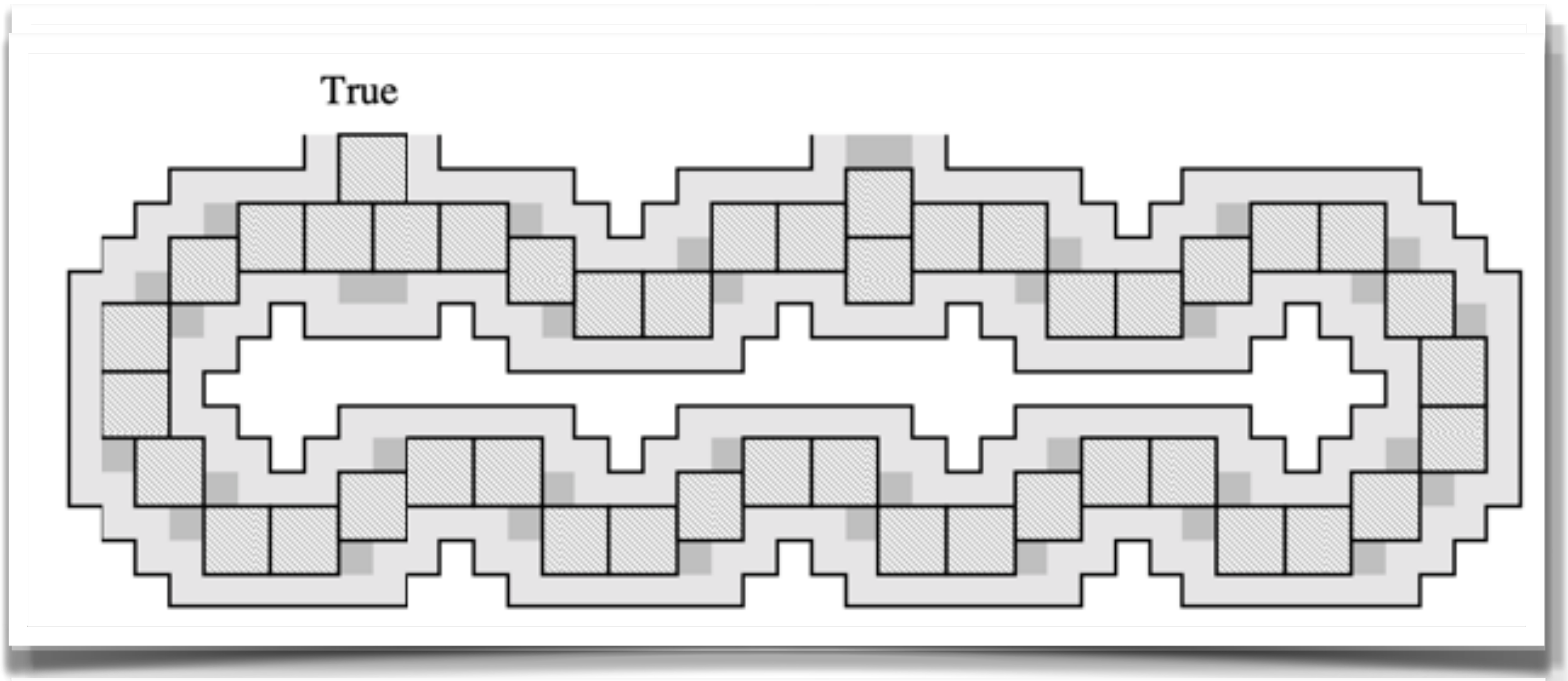
Variables (2)

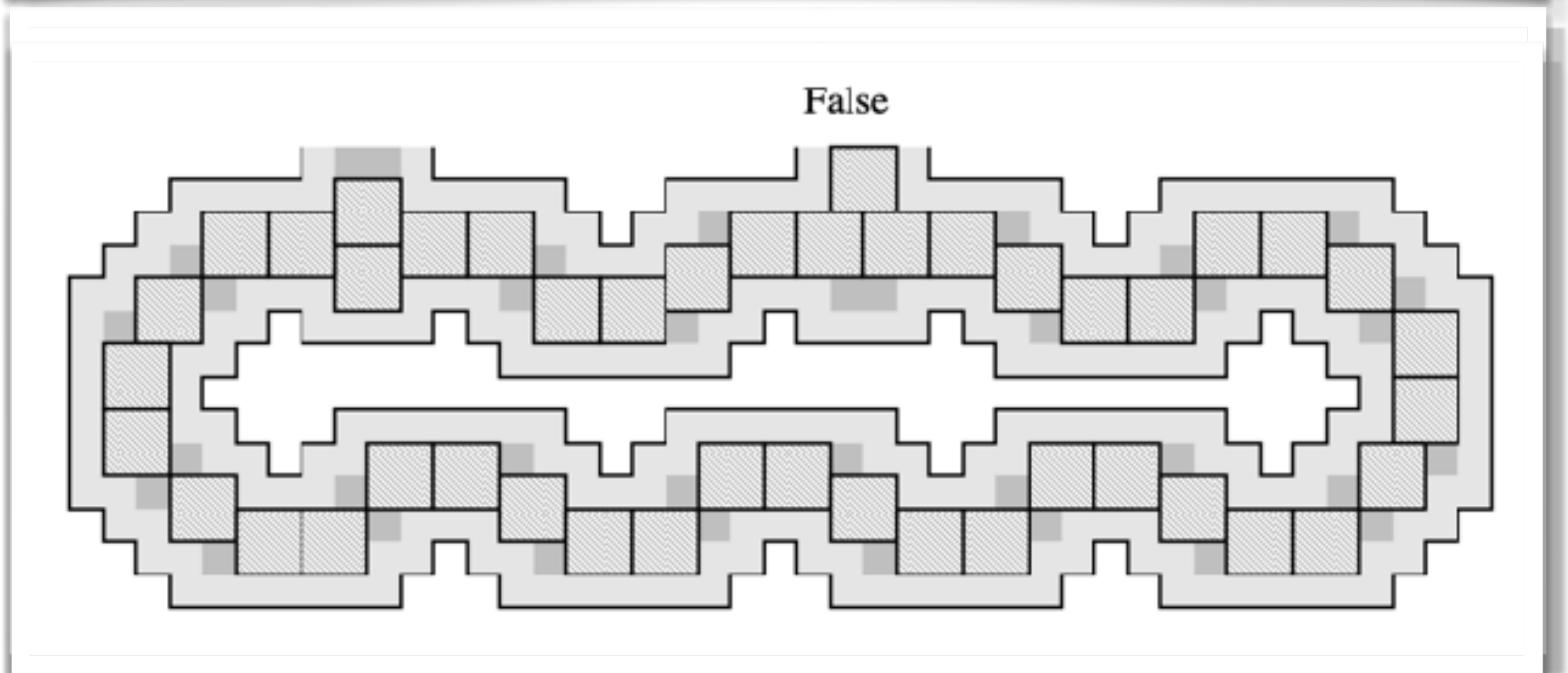




Technische Universität Braunschweig

Variables (2)

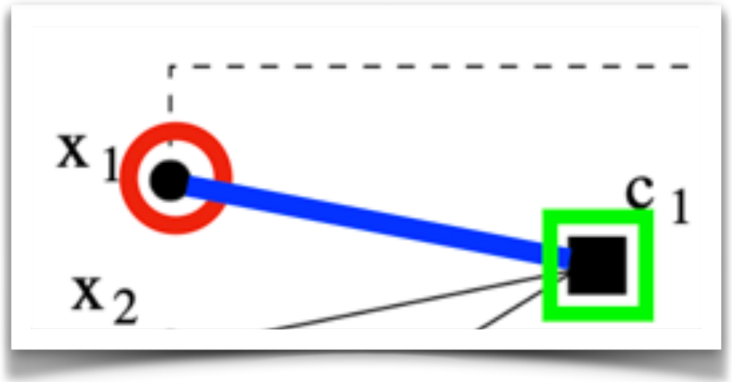




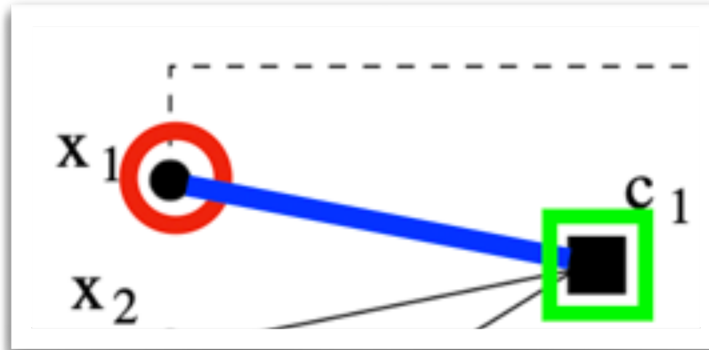
Technische Universität Braunschweig

Connectors

Connectors



Connectors



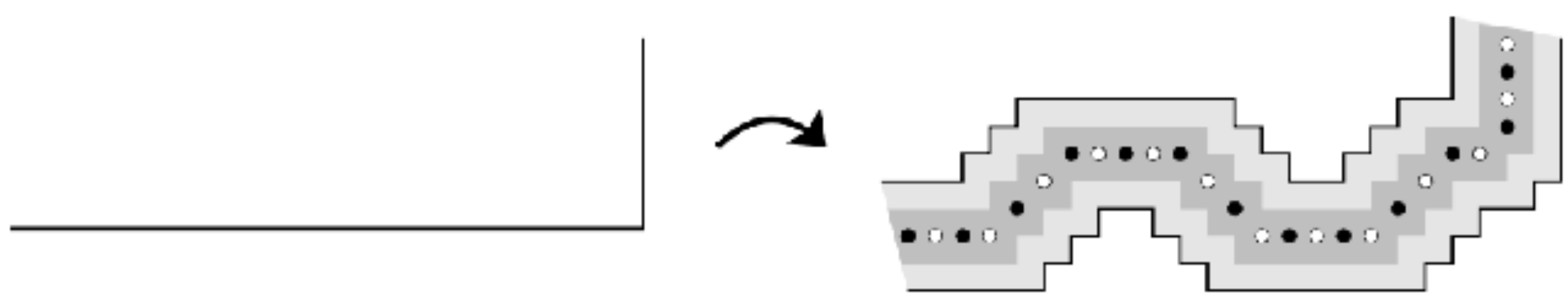
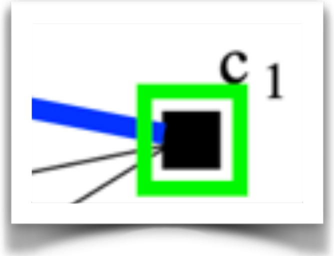


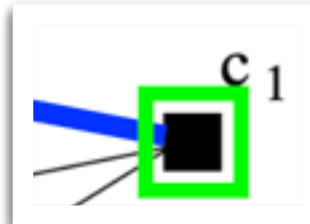
Fig. 3. Part of a connector component (right) for representing an edge between variable nodes and clause nodes (left).

Clauses

☐ Clauses



Clauses



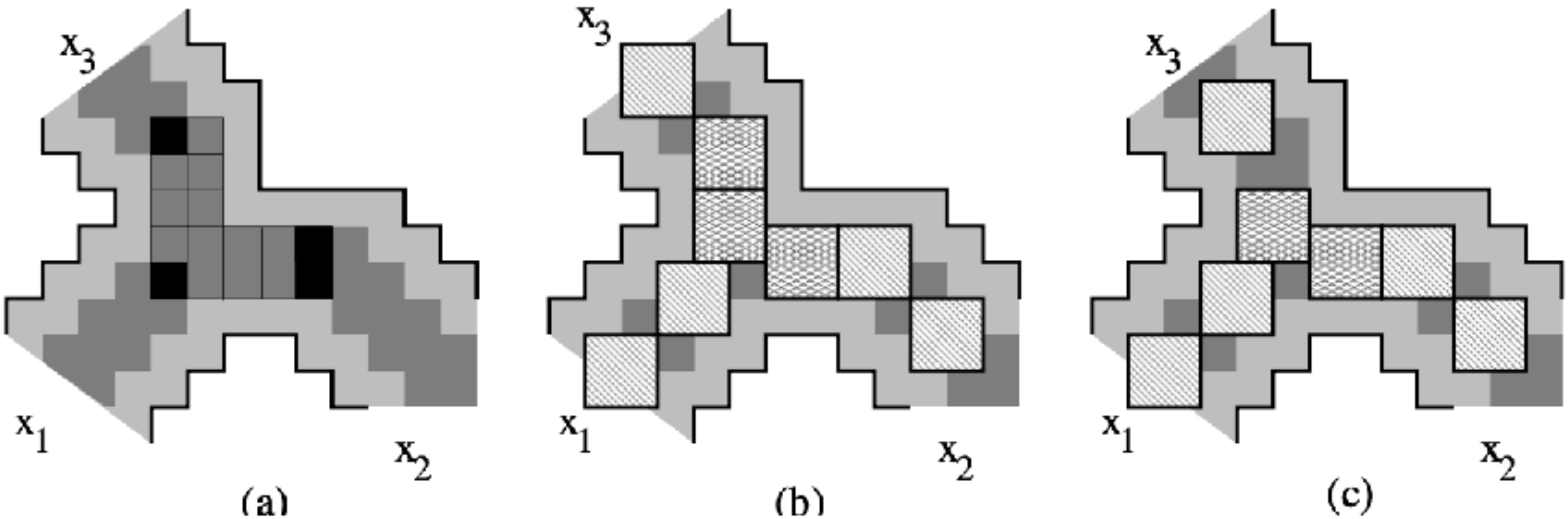
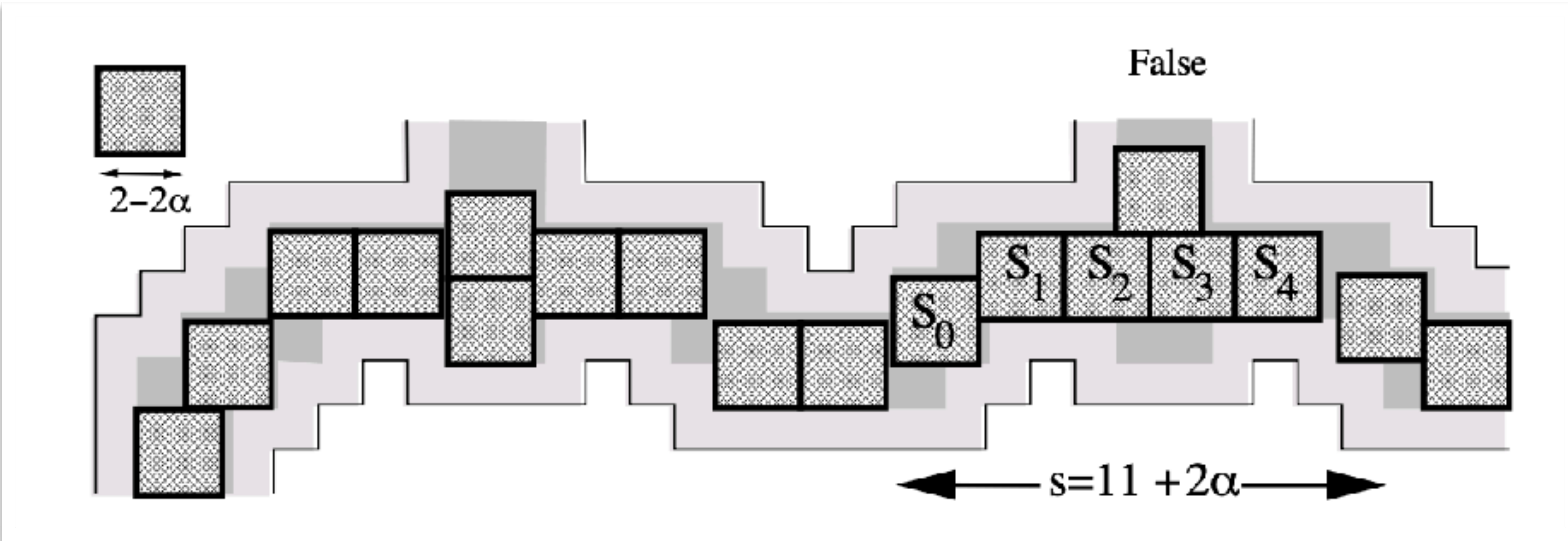


Fig. 4. A clause component for dispersion with boundaries and its receptor region (a); a satisfying placement (b); and an unsatisfying placement (c).

Technische Universität Braunschweig

Smaller Squares (1)

Smaller Squares (1)



Technische Universität Braunschweig

Smaller Squares (2)

Smaller Squares (2)

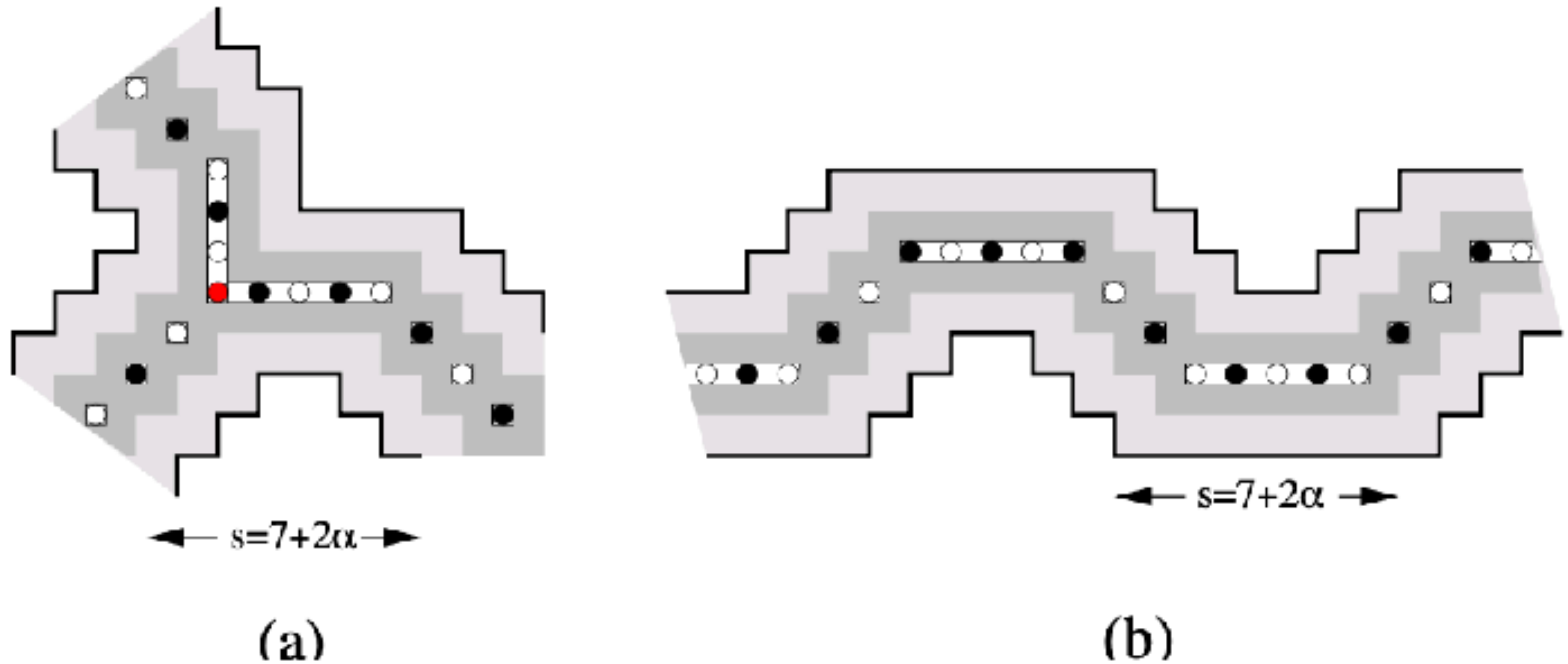


Fig. 7. An upper bound on the approximation factor: clause components (a) and connector components (b).

Other Metrics

Other Metrics

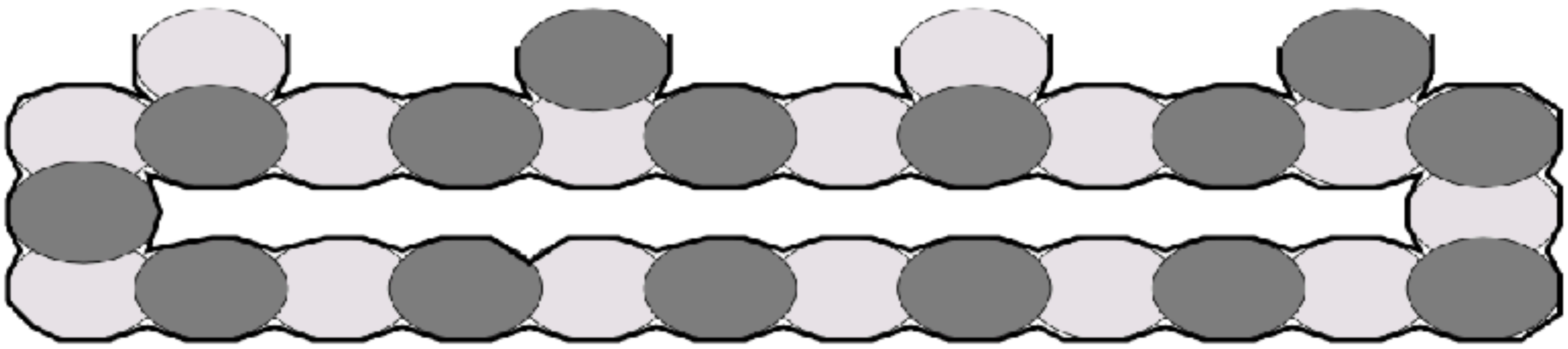


Fig. 9. A variable gadget for the case where the unit ball is an ellipse.

Technische Universität Braunschweig

Approximation

Approximation

THEOREM 9. *For rectilinear geometric dispersion with boundaries of k locations in a rectilinear polygon P with n vertices, Algorithm 8 computes a solution $A_{Dis}(P,k)$, such that*

$$A_{Dis}(P,k) \geq \frac{2}{3} OPT(P,k).$$

The running time is strongly polynomial.

Approximation

THEOREM 9. *For rectilinear geometric dispersion with boundaries of k locations in a rectilinear polygon P with n vertices, Algorithm 8 computes a solution $A_{Dis}(P,k)$, such that*

$$A_{Dis}(P,k) \geq \frac{2}{3} OPT(P,k).$$

The running time is strongly polynomial.

ALGORITHM 8.
Input: rectilinear polygon P, positive integer k.
Output: a set of k locations, such that $A_{Dis}(P,k) := d$ is the minimum L_∞ distance between a location and the boundary, or between two locations.

1. **For all** $(e_i, e_j) \in Par(P)$ **do**
   (a) Perform binary search for the smallest integer m, $2 \leq m \leq k+1$, with the following property:
      - For $d_{ijm} := Dist(e_i, e_j)/m$, $AS(P - d_{ijm}/2, d_{ijm}, 6)$ returns a feasible solution for at least k locations at distance d_{ijm}.

   (b) Let d_{ij} be the distance d_{ijm} for the critical value m.
2. Let d be the maximum d_{ij} for any (e_i, e_j).

Technische Universität Braunschweig

Approximation

THEOREM 9. *For rectilinear geometric dispersion with boundaries of k locations in a rectilinear polygon P with n vertices, Algorithm 8 computes a solution $A_{Dis}(P,k)$, such that*

$$A_{Dis}(P,k) \geq \frac{2}{3} OPT(P,k).$$

The running time is strongly polynomial.

Technische Universität Braunschweig

Approximation

THEOREM 9. *For rectilinear geometric dispersion with boundaries of k locations in a rectilinear polygon P with n vertices, Algorithm 8 computes a solution $A_{Dis}(P,k)$, such that*

$$A_{Dis}(P,k) \geq \frac{2}{3} OPT(P,k).$$

The running time is strongly polynomial.

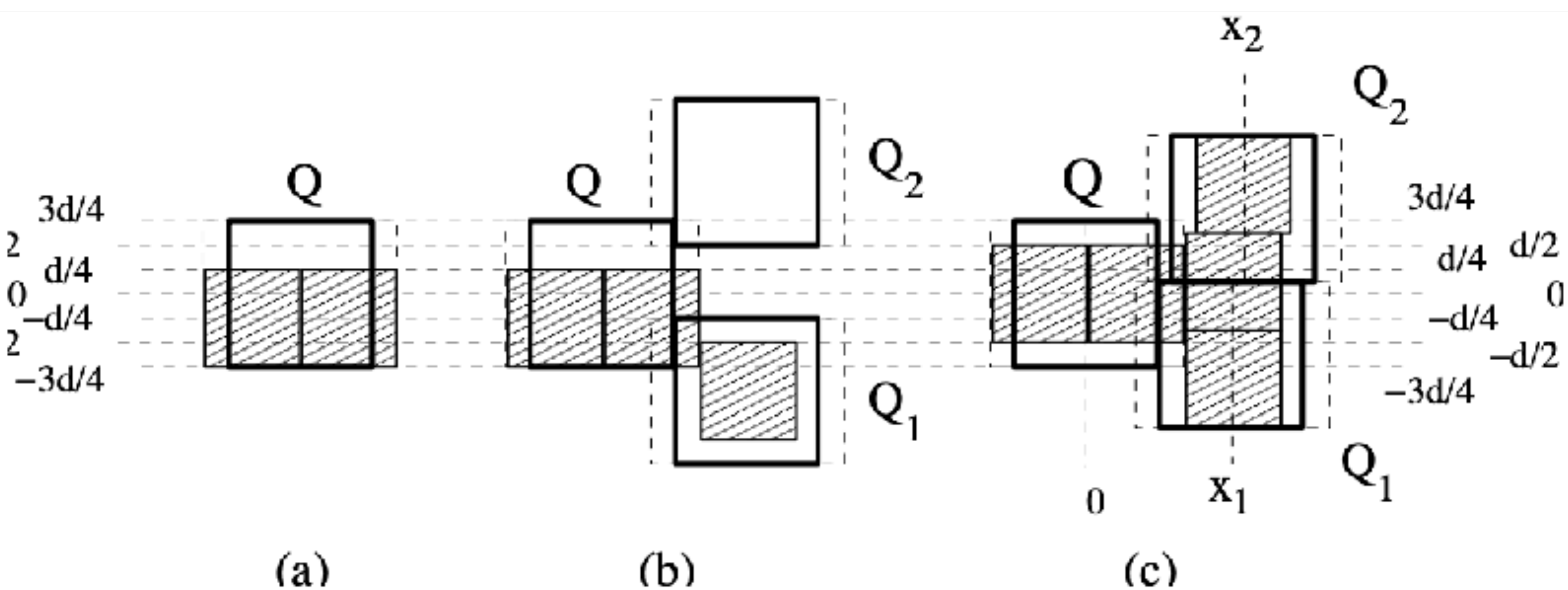


Fig. 12. Constructing a packing of *d*-squares.

Average Distance

Average Distance

Algorithmica (2004) 38: 501–511
DOI: 10.1007/s00453-003-1074-x

Algorithmica

Maximum Dispersion and Geometric Maximum Weight Cliques[1]

Sándor P. Fekete[2] and Henk Meijer[3]

Abstract. We consider a facility location problem, where the objective is to "disperse" a number of facilities, i.e., select a given number k of locations from a discrete set of n candidates, such that the average distance between selected locations is maximized. In particular, we present algorithmic results for the case where vertices are represented by points in d-dimensional space, and edge weights correspond to rectilinear distances. Problems of this type have been considered before, with the best result being an approximation algorithm with performance ratio 2. For the case where k is fixed, we establish a linear-time algorithm that finds an optimal solution. For the case where k is part of the input, we present a polynomial-time approximation scheme.

Key Words. Dispersion, Facility location, Maximum weight cliques, Remote clique, Heaviest subgraph, Geometric optimization, Approximation, PTAS.

Average Distance

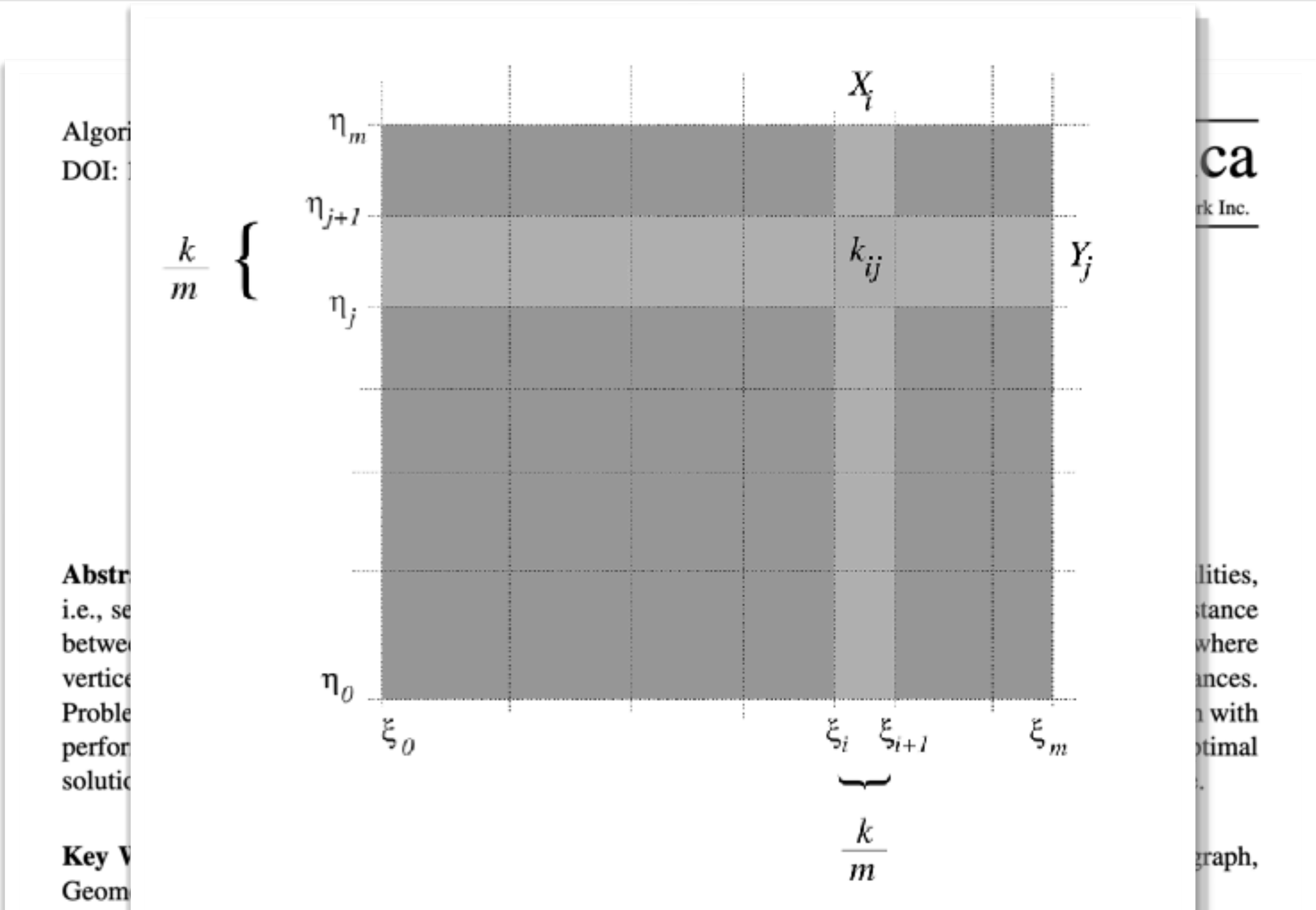


Fig. 1. Subdividing the plane into cells.

Average Distance

Algorithmica (2004) 38: 501–511
DOI: 10.1007/s00453-003-1074-x

Algorithmica

Maximum Dispersion and Geometric Maximum Weight Cliques[1]

Sándor P. Fekete[2] and Henk Meijer[3]

Abstract. We consider a facility location problem, where the objective is to "disperse" a number of facilities, i.e., select a given number k of locations from a discrete set of n candidates, such that the average distance between selected locations is maximized. In particular, we present algorithmic results for the case where vertices are represented by points in d-dimensional space, and edge weights correspond to rectilinear distances. Problems of this type have been considered before, with the best result being an approximation algorithm with performance ratio 2. For the case where k is fixed, we establish a linear-time algorithm that finds an optimal solution. For the case where k is part of the input, we present a polynomial-time approximation scheme.

Key Words. Dispersion, Facility location, Maximum weight cliques, Remote clique, Heaviest subgraph, Geometric optimization, Approximation, PTAS.

Technische Universität Braunschweig

1. Introduction
2. Review
3. Extra Packing: Dispersion
4. Extra Tours: Lawn Mowing
5. Relay Placement
6. Coordinated Motion Planning

Technische Universität Braunschweig

Lawn Mowing (1)

Technische Universität Braunschweig

Lawn Mowing (1)

ELSEVIER

Computational Geometry 17 (2000) 25–50

Computational Geometry
Theory and Applications

www.elsevier.nl/locate/comgeo

Approximation algorithms for lawn mowing and milling ☆

Esther M. Arkin [a,1], Sándor P. Fekete [b,*], Joseph S.B. Mitchell [a,2]

[a] *Department of Applied Mathematics and Statistics, SUNY Stony Brook, NY 11794-3600, USA*
[b] *Department of Mathematics, TU Berlin, 10623 Berlin, Germany*

Communicated by K. Mehlhorn; received 18 March 1997; accepted 12 August 1999

Abstract

We study the problem of finding shortest tours/paths for "lawn mowing" and "milling" problems: Given a region in the plane, and given the shape of a "cutter" (typically, a circle or a square), find a shortest tour/path for the cutter such that every point within the region is covered by the cutter at some position along the tour/path. In the milling version of the problem, the cutter is constrained to stay within the region. The milling problem arises naturally in the area of automatic tool path generation for NC pocket machining. The lawn mowing problem arises in optical inspection, spray painting, and optimal search planning.

Lawn Mowing (1)

ELSEVIER

Computational Geometry 17 (2000) 25–50

Computational Geometry
Theory and Applications
www.elsevier.nl/locate/comgeo

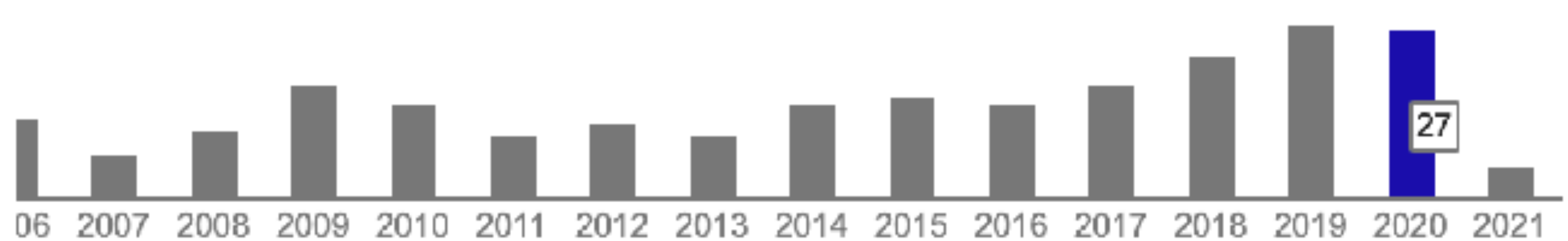
Approximation algorithms for lawn mowing and milling ☆

Esther M. Arkin [a,1], Sándor P. Fekete [b,*], Joseph S.B. Mitchell [a,2]

[a] *Department of Applied Mathematics and Statistics, SUNY Stony Brook, NY 11794-3600, USA*
[b] *Department of Mathematics, TU Berlin, 10623 Berlin, Germany*

Communicated by K. Mehlhorn; received 18 March 1997; accepted 12 August 1999

Total citations Cited by 283



Lawn Mowing (2)

Lawn Mowing (2)

E.M. Arkin et al. / Computational Geometry 17 (2000) 25–50

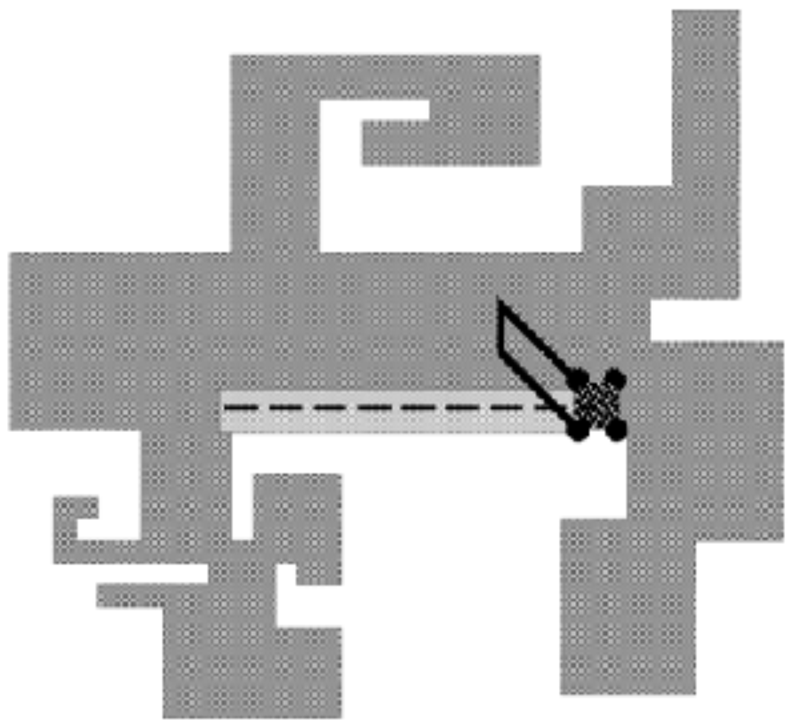


Fig. 1. The lawn mowing problem.

Lawn Mowing (2)

E.M. Arkin et al. / Computational Geometry 17 (2000) 25–50

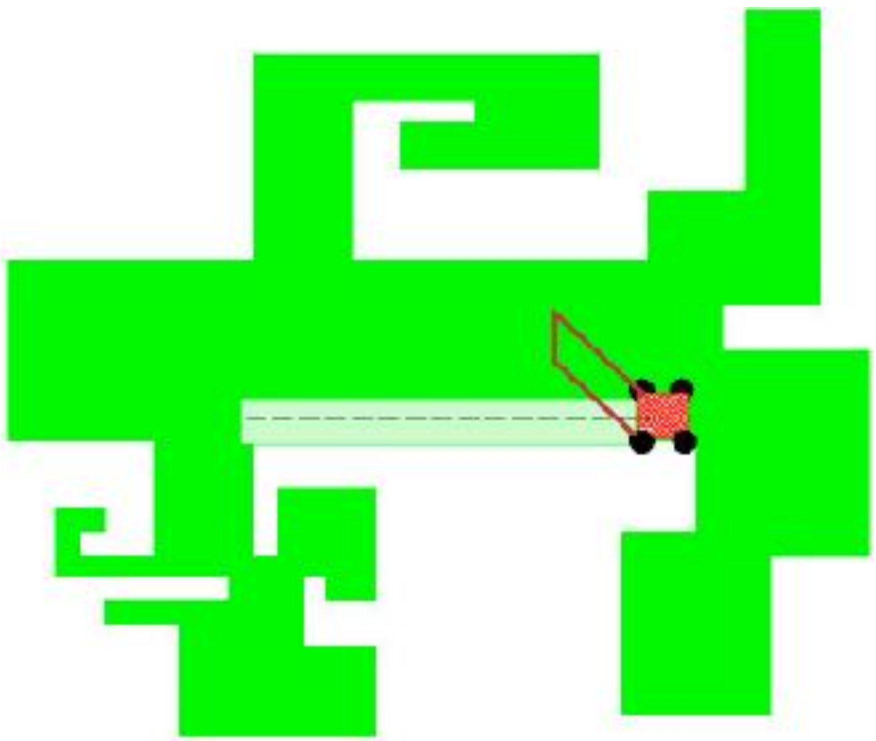


Fig. 1. The lawn mowing problem.

Technische Universität Braunschweig

Lawn Mowing (2)

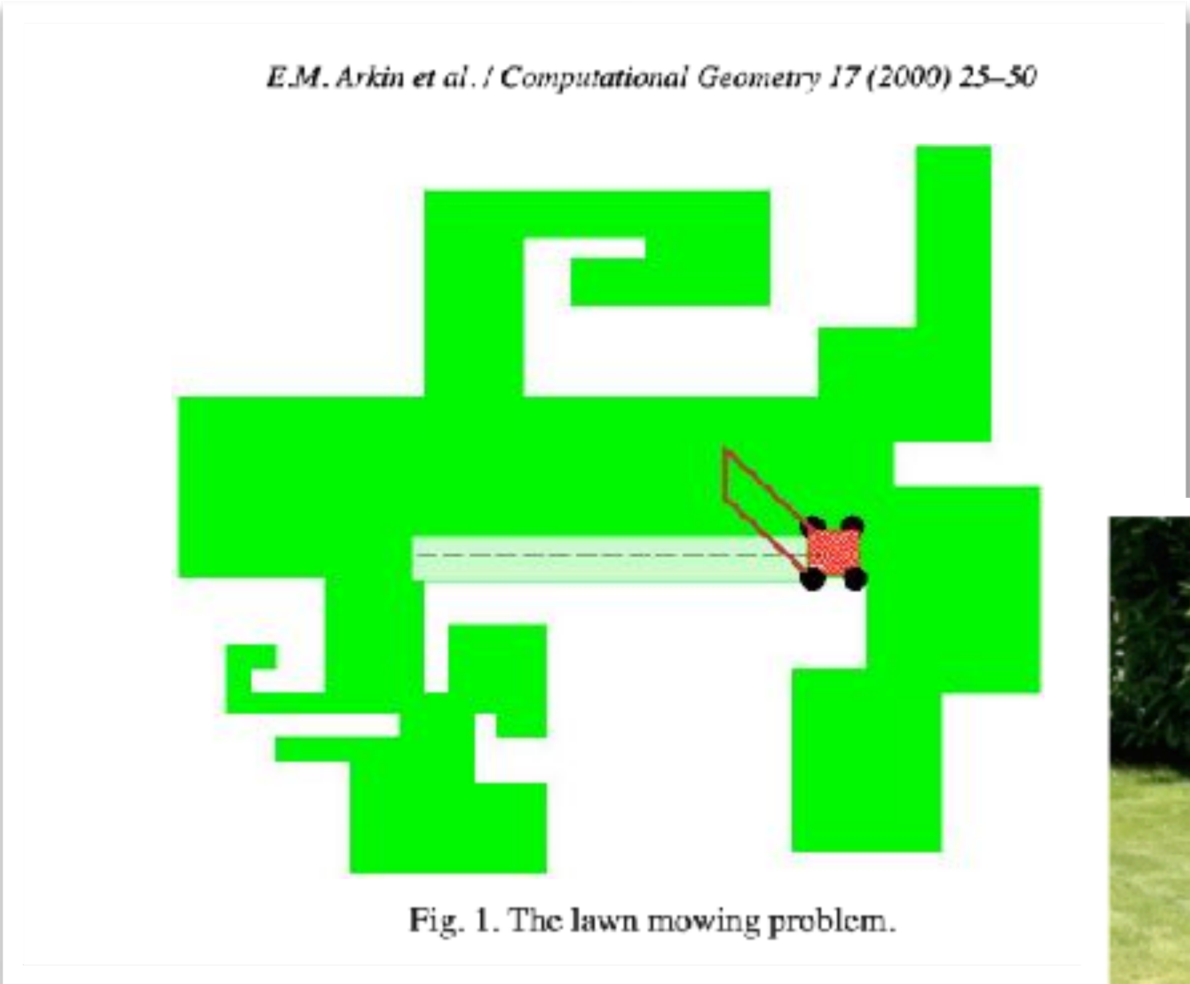
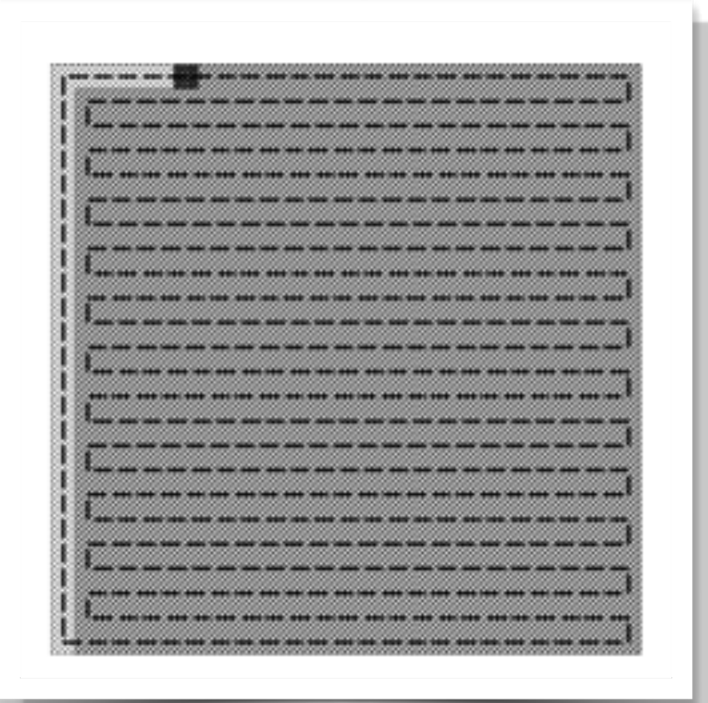
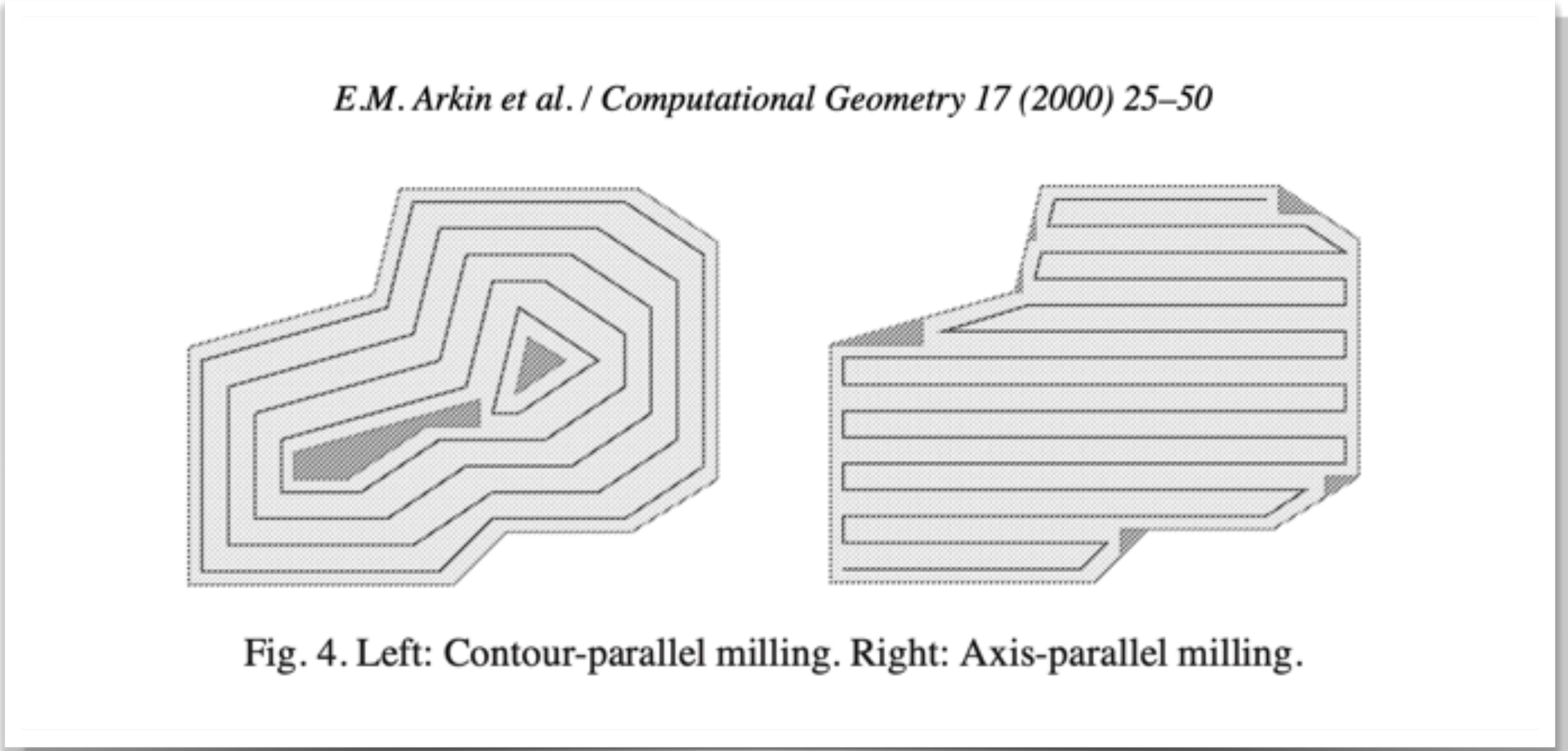

E.M. Arkin et al. / Computational Geometry 17 (2000) 25–50

Fig. 1. The lawn mowing problem.

Lawn Mowing (3)





Lawn Mowing (4)

Theorem 1. *The lawn mowing problem for a connected polygonal region is NP-hard for the case of an aligned unit square cutter* χ.

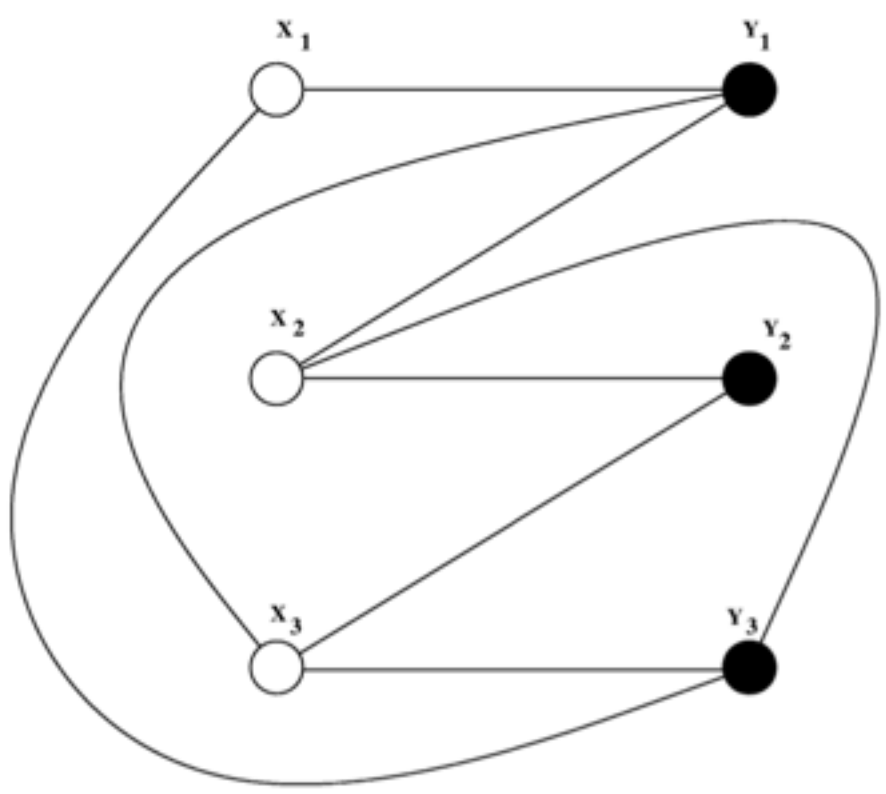
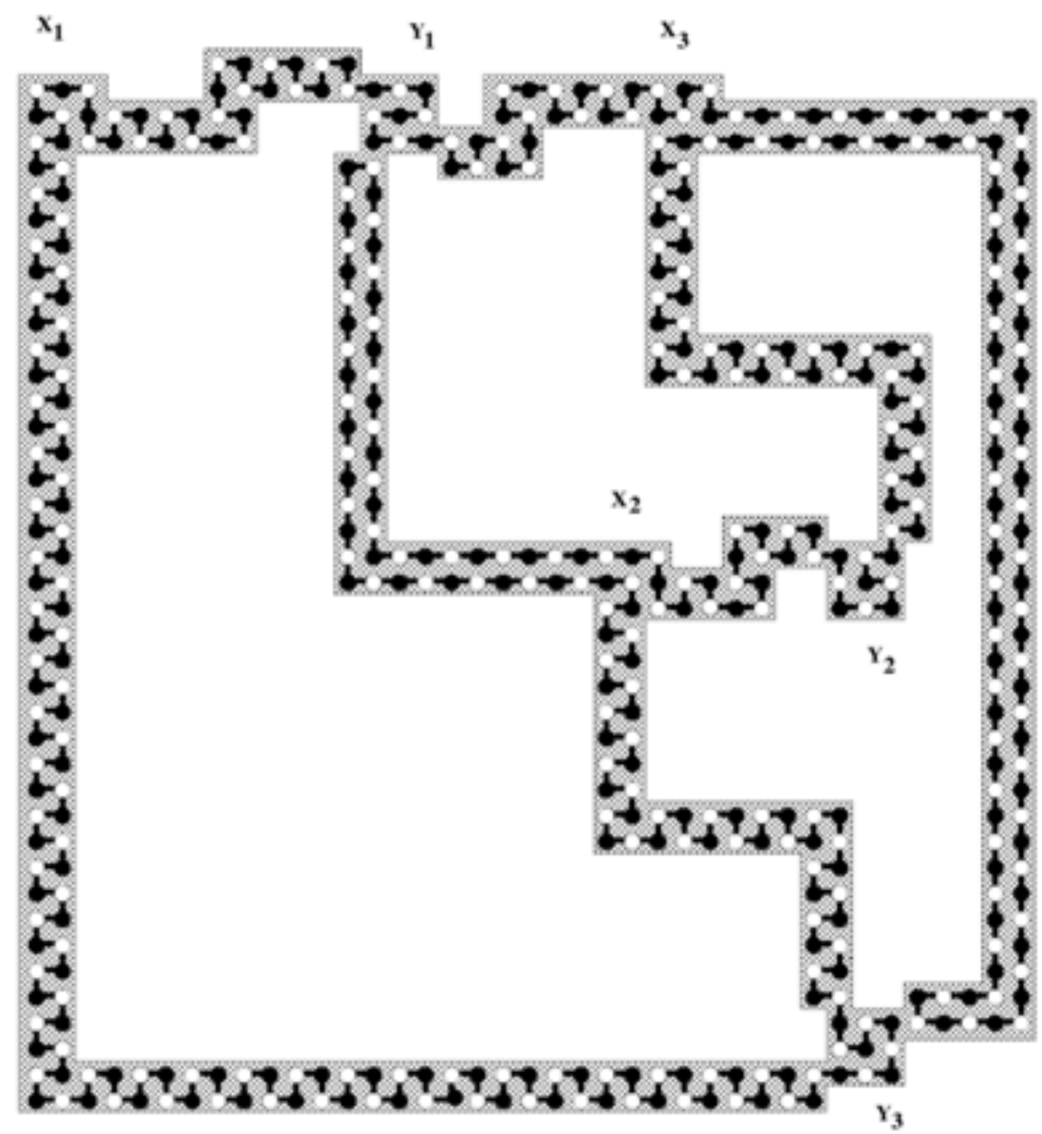


Fig. 5. A planar bipartite graph G, with maximum vertex degree 3.



Lawn Mowing (5)

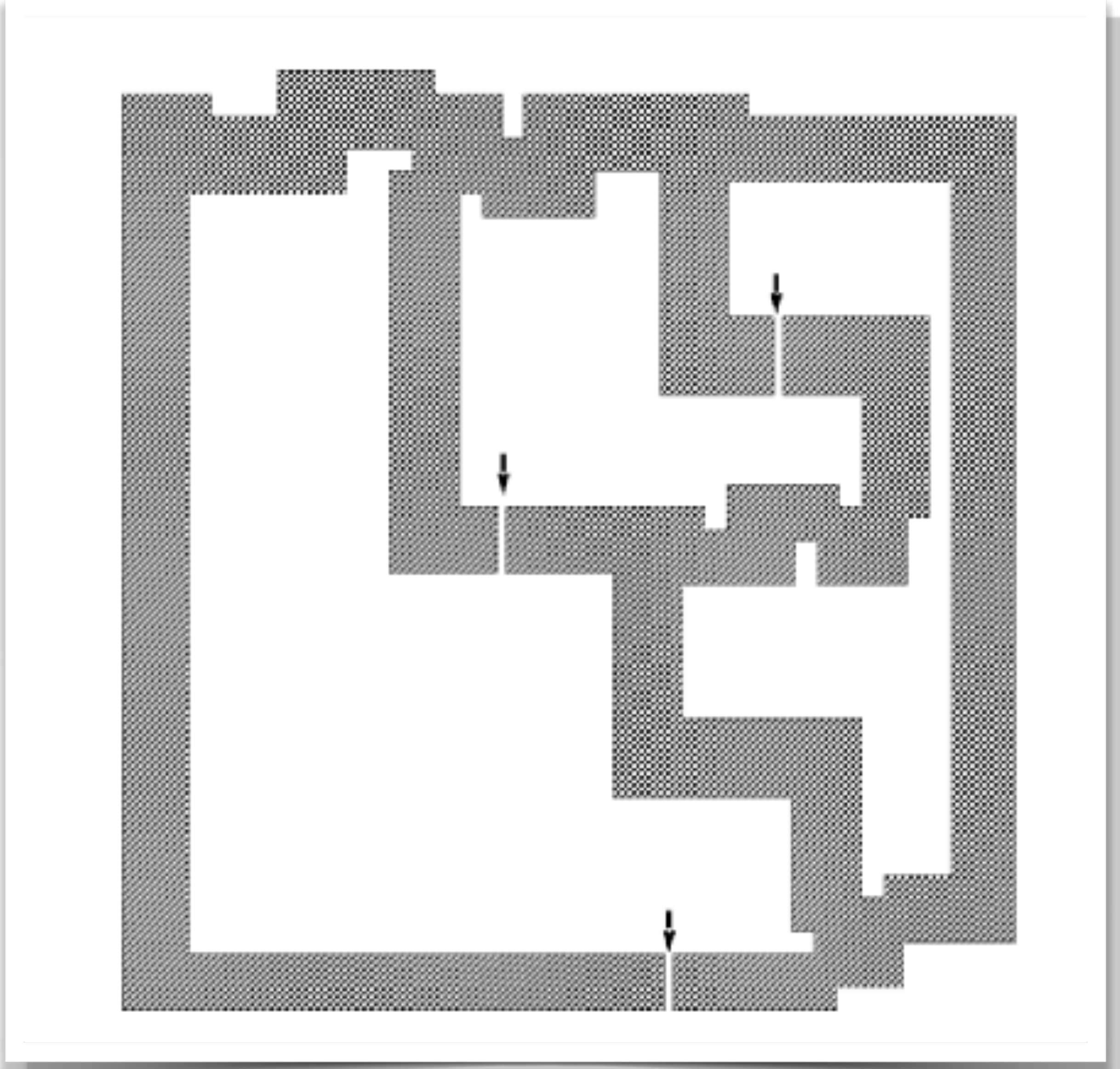
Lawn Mowing (5)

Corollary 1. *The lawn mowing problem is NP-hard even for simple polygonal regions R.*

Technische Universität Braunschweig

Lawn Mowing (5)

Corollary 1. *The lawn mowing problem is NP-hard even for simple polygonal regions R.*



Technische Universität Braunschweig

Lawn Mowing (6)

Lawn Mowing (6)

Theorem 2. *Finding a rectilinear TSP approximation (with factor α_{TSP}) on the set of centerpoints S yields a lawn mowing tour of length at most $\alpha_{\mathrm{TSP}}(3\ell^* + 6)$, where ℓ^* is the length of an optimal lawn mowing tour.*

Technische Universität Braunschweig

Lawn Mowing (6)

Theorem 2. *Finding a rectilinear TSP approximation (with factor α_{TSP}) on the set of centerpoints S yields a lawn mowing tour of length at most $\alpha_{\mathrm{TSP}}(3\ell^* + 6)$, where ℓ^* is the length of an optimal lawn mowing tour.*

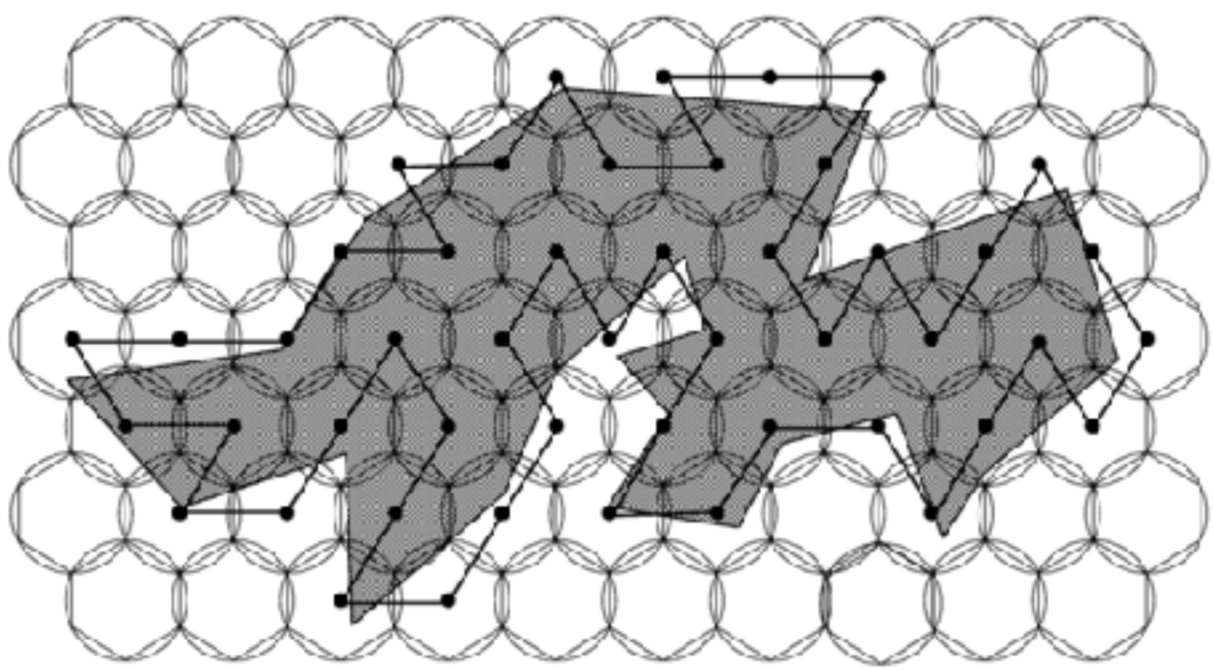
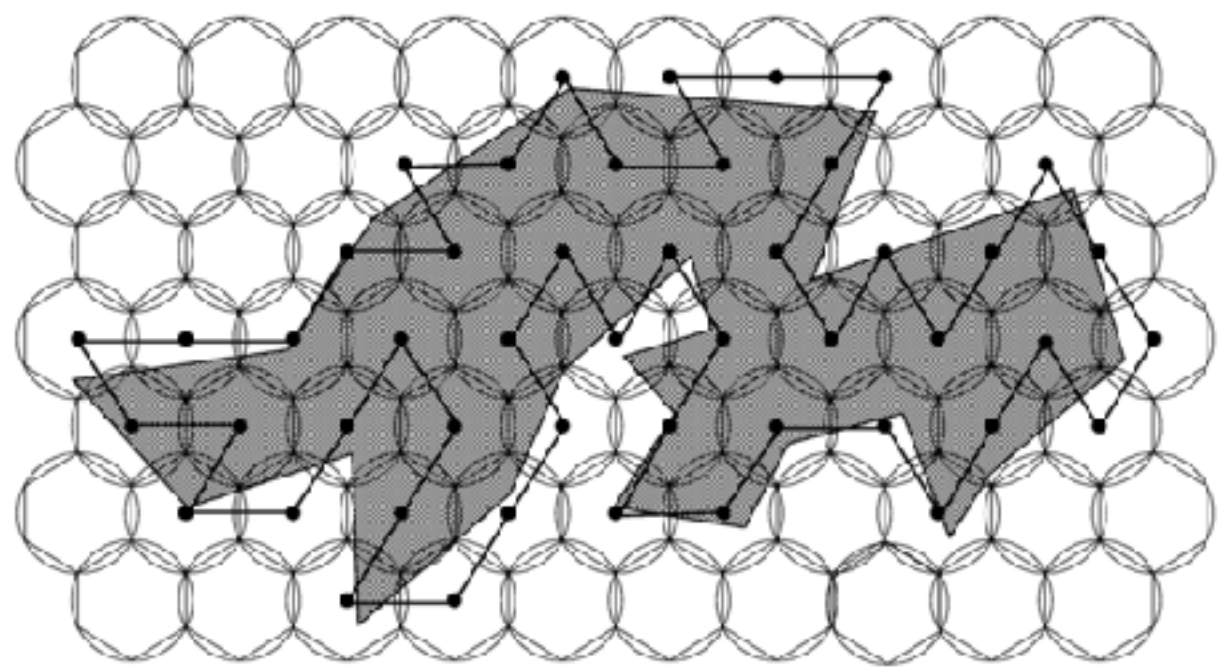


Fig. 10. Approximation for the case of a circular cutter χ.

Lawn Mowing (6)

Theorem 2. *Finding a rectilinear TSP approximation (with factor α_{TSP}) on the set of centerpoints S yields a lawn mowing tour of length at most $\alpha_{\mathrm{TSP}}(3\ell^* + 6)$, where ℓ^* is the length of an optimal lawn mowing tour.*



Theorem 3. *The lawn mowing problem has a constant-factor approximation algorithm that runs in polynomial time (dependent on the TSP heuristic employed). For the case of an aligned unit square cutter, the approximation factor is $3\alpha_{\mathrm{TSP}}$ for rectilinear motion, and is $3\beta\alpha_{\mathrm{TSP}}$ for arbitrary translational motion. For the case of a unit circular cutter, with arbitrary motion, the approximation factor is $3\gamma\alpha_{\mathrm{TSP}}$. Here, $\beta = \frac{2}{\sqrt{2+\sqrt{2}}} \approx 1.08$ and $\gamma = \frac{2\sqrt{3}}{3} \approx 1.15$.*

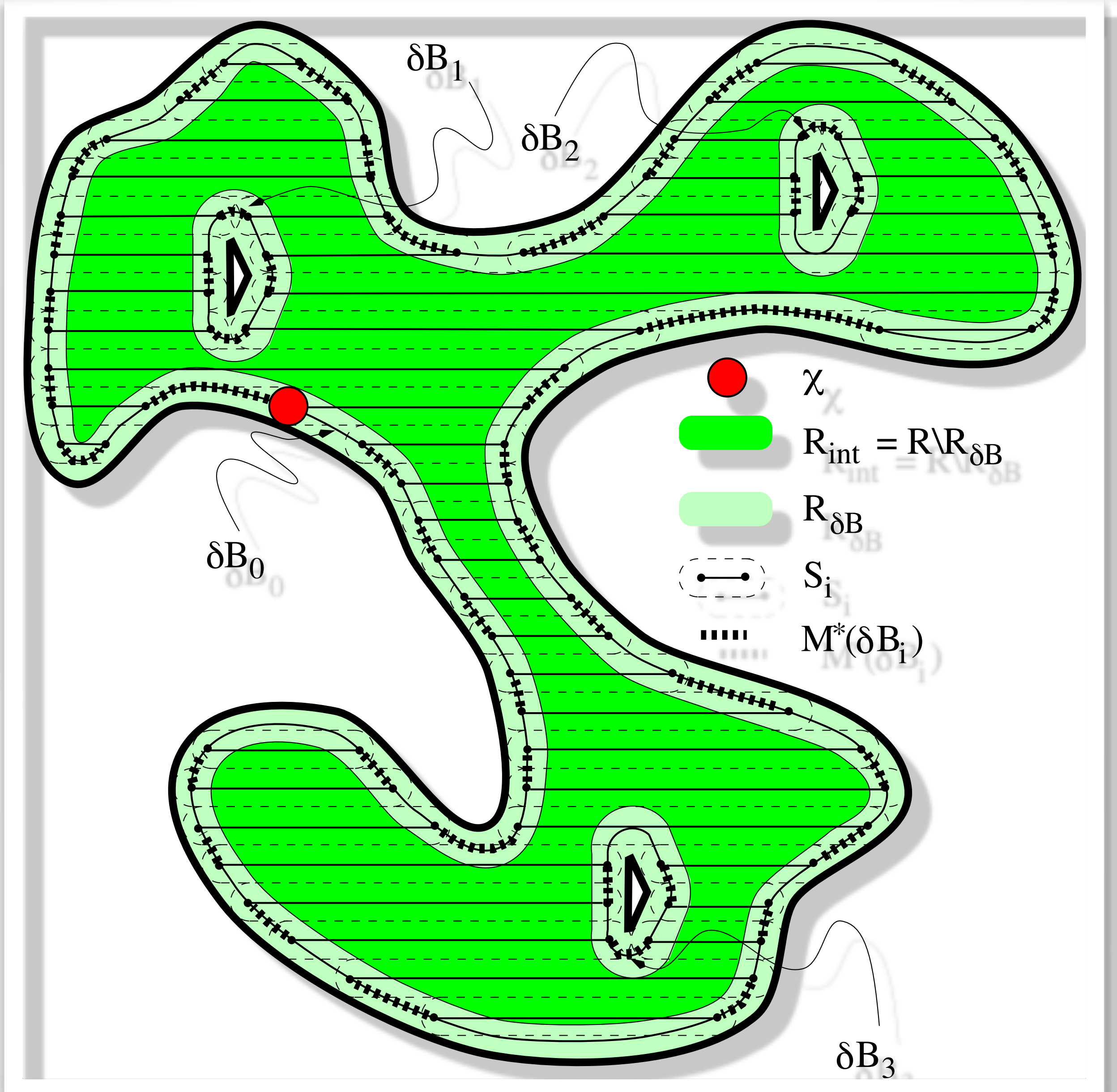
Technische Universität Braunschweig

Lawn Mowing (7)

Theorem 4. *In time* $O(n \log n)$, *one can decide whether a (multiply) connected region with n sides (straight or circular arc) can be milled by a unit disk or unit square, and, within the same time bound, one can construct a tour of length at most* $2\frac{1}{2}$ *times the length of an optimal milling tour.*

Lawn Mowing (7)

Theorem 4. *In time* $O(n \log n)$*, one can decide whether a (multiply) connected region with* n *sides (straight or circular arc) can be milled by a unit disk or unit square, and, within the same time bound, one can construct a tour of length at most* $2\frac{1}{2}$ *times the length of an optimal milling tour.*



Lawn Mowing (8)

Lawn Mowing (8)

Theorem 5. *Let G be a simple grid graph, having N nodes at the centerpoints, V, of pixels within a simple rectilinear polygon, R, having n (integer-coordinate) sides. Assume that G has no cut vertices. Then, in time $O(n)$, one can find a representation of a tour, T, that visits all N nodes of G, of length at most $\frac{6N-4}{5}$.*

Technische Universität Braunschweig

Lawn Mowing (8)

Theorem 5. *Let G be a simple grid graph, having N nodes at the centerpoints, V, of pixels within a simple rectilinear polygon, R, having n (integer-coordinate) sides. Assume that G has no cut vertices. Then, in time* O(*n*)*, one can find a representation of a tour, T, that visits all N nodes of G, of length at most* $\frac{6N-4}{5}$.

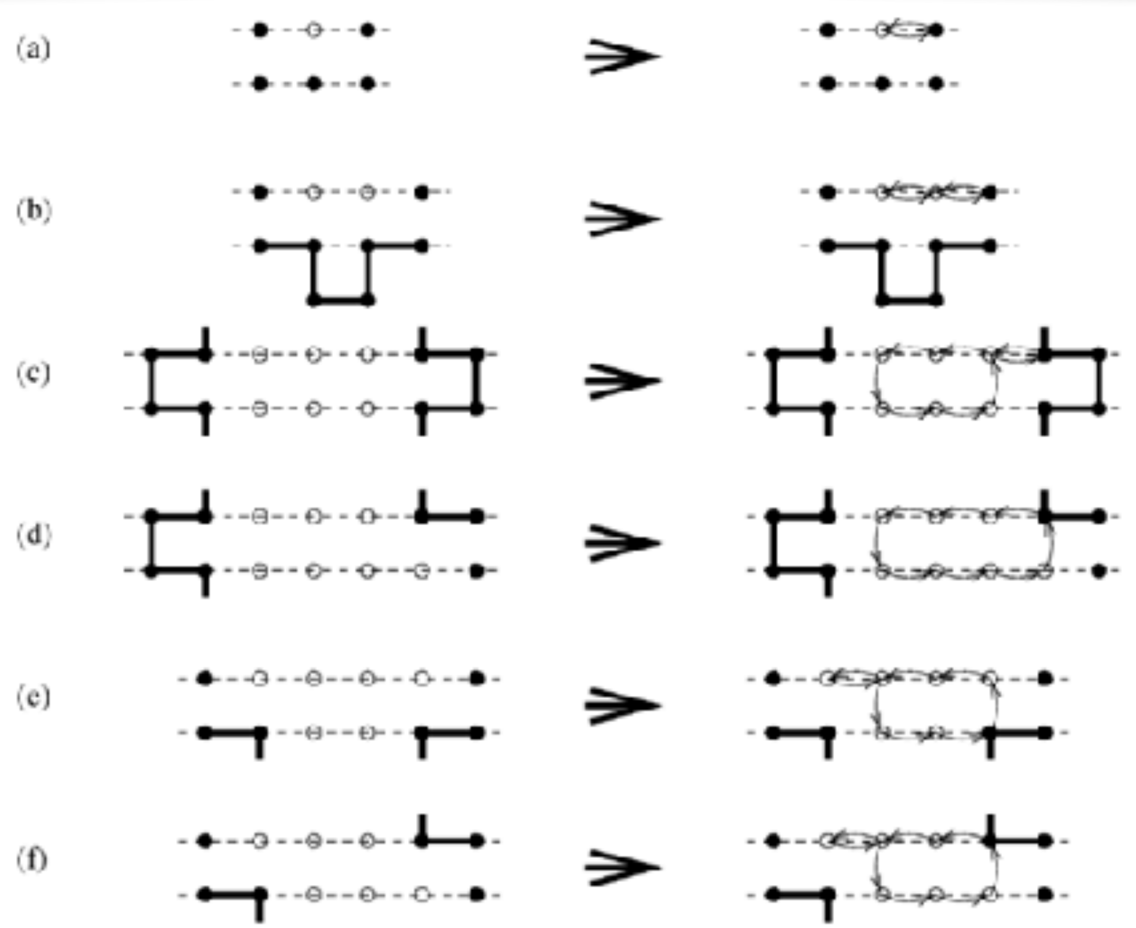
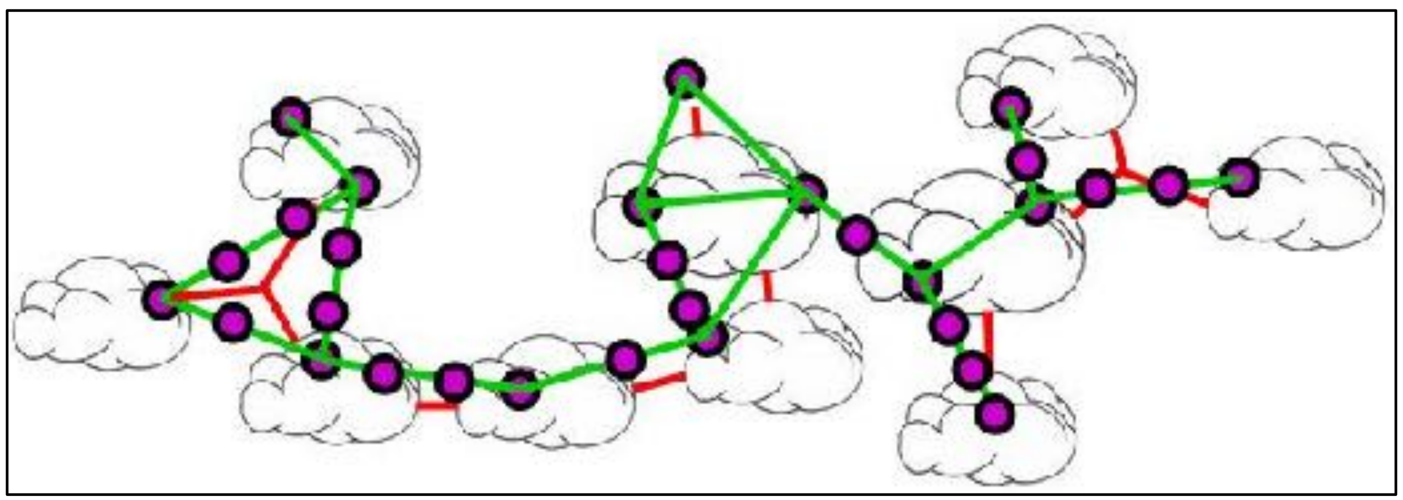


Fig. 15. Six cases for incorporating internal nodes into the modified contour tour. Hollow circles denote internal nodes, V_I', and solid circles denote nodes of C'. Solid edges are drawn where there *must* be edges of C'.

1. Introduction
2. Review
3. Extra Packing: Dispersion
4. Extra Tours: Lawn Mowing
5. Relay Placement
6. Coordinated Motion Planning

Approximation Algorithms for Relay Placement in the Plane



Alon Efrat, Sándor P. Fekete, Joe Mitchell, Valentin Polishchuk,
G. R. Poornananda, and Jukka Suomela

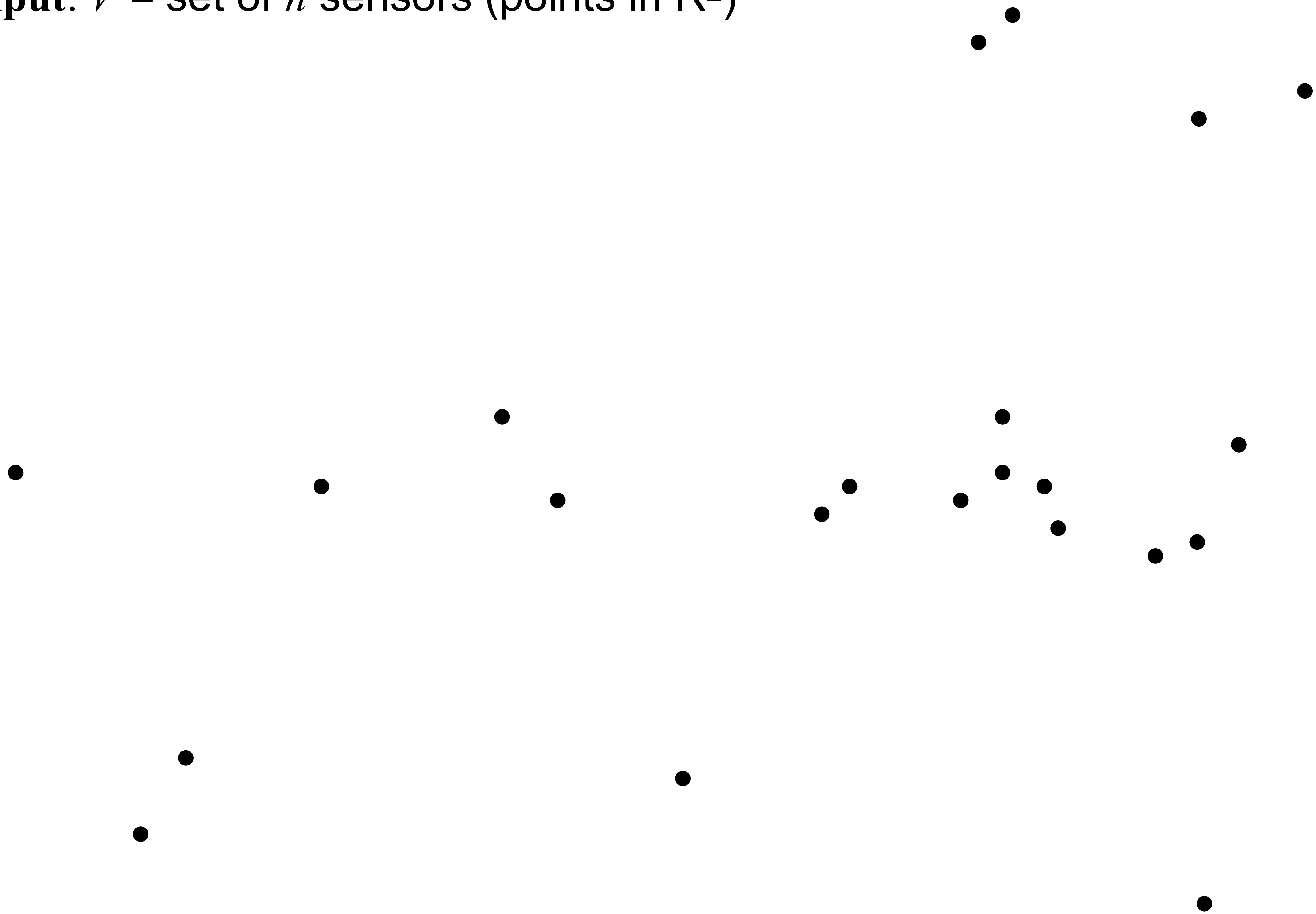
Relay Placement

Relay Placement

Input: V = set of n sensors (points in R^2)

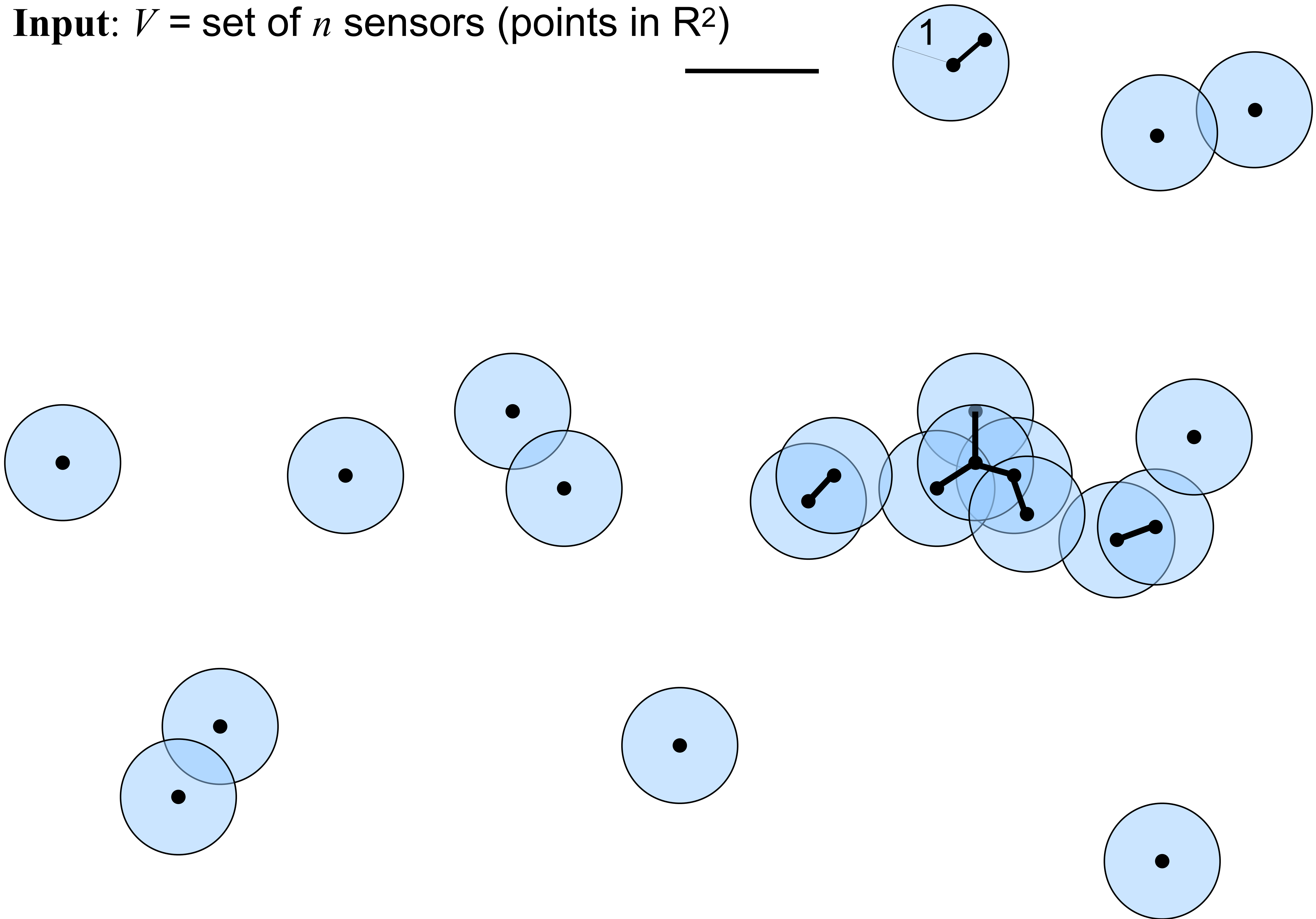
Relay Placement

Input: V = set of n sensors (points in R^2)



Relay Placement

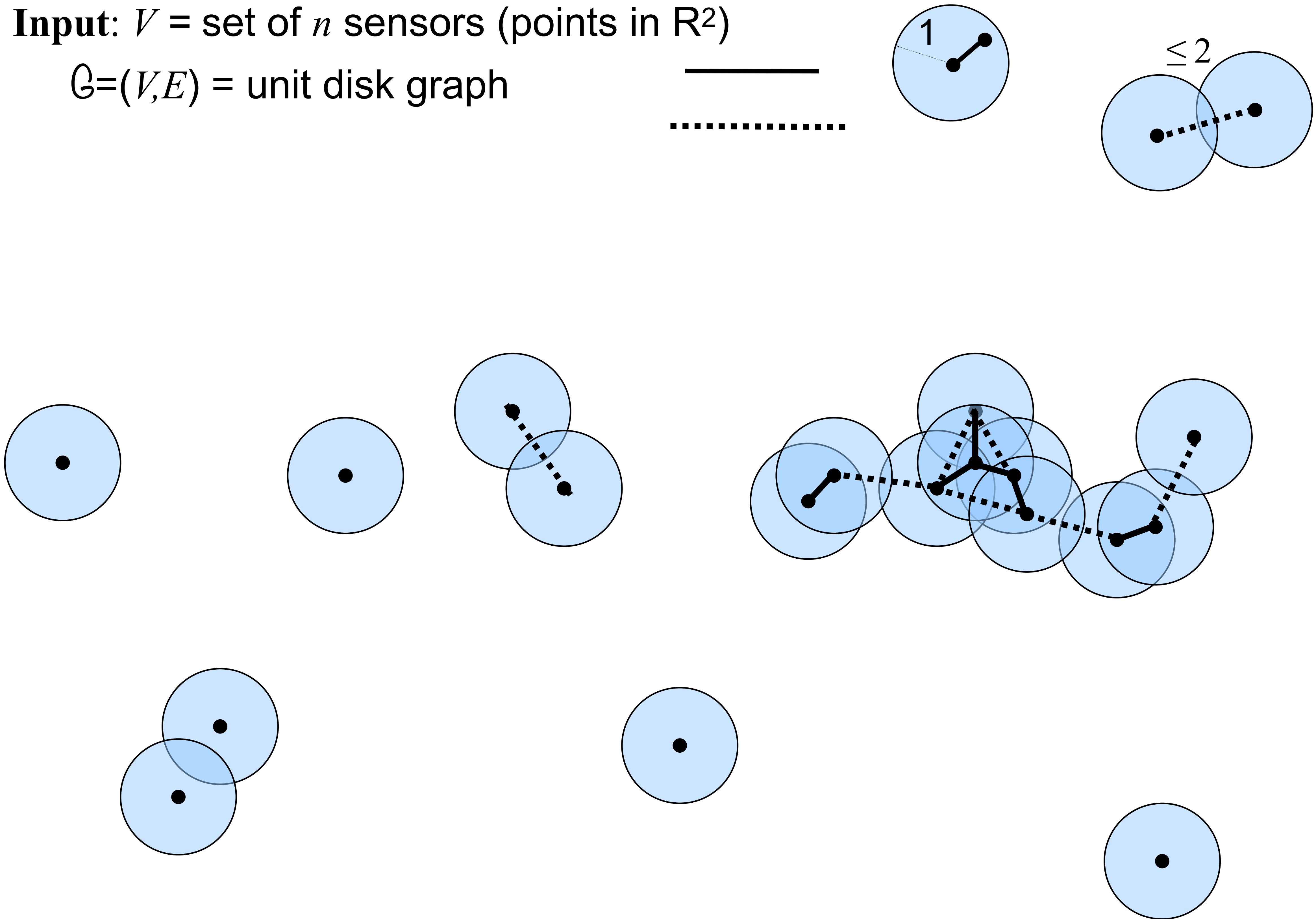
Input: V = set of n sensors (points in R^2)



Relay Placement

Input: V = set of n sensors (points in R^2)

$G=(V,E)$ = unit disk graph

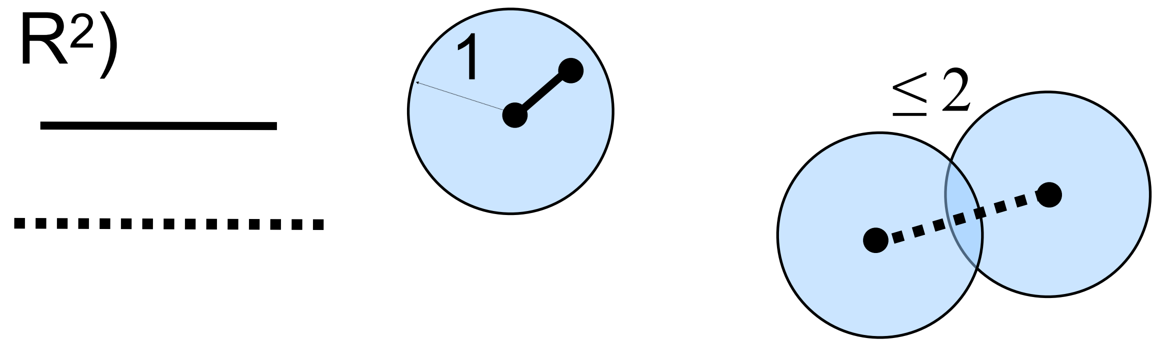


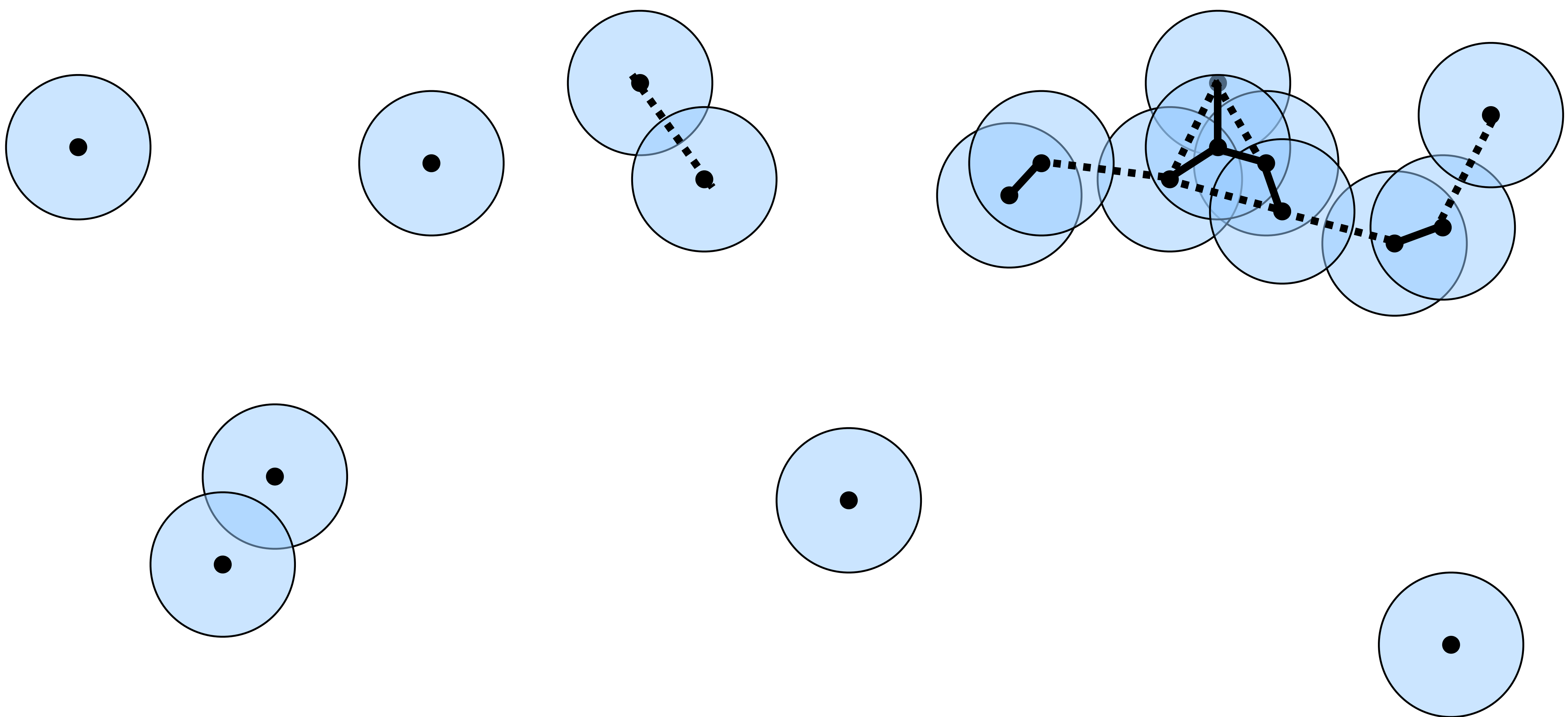
Relay Placement

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$ = unit disk graph

$\mathcal{F}=(V,F)$ = radius-2 disk graph



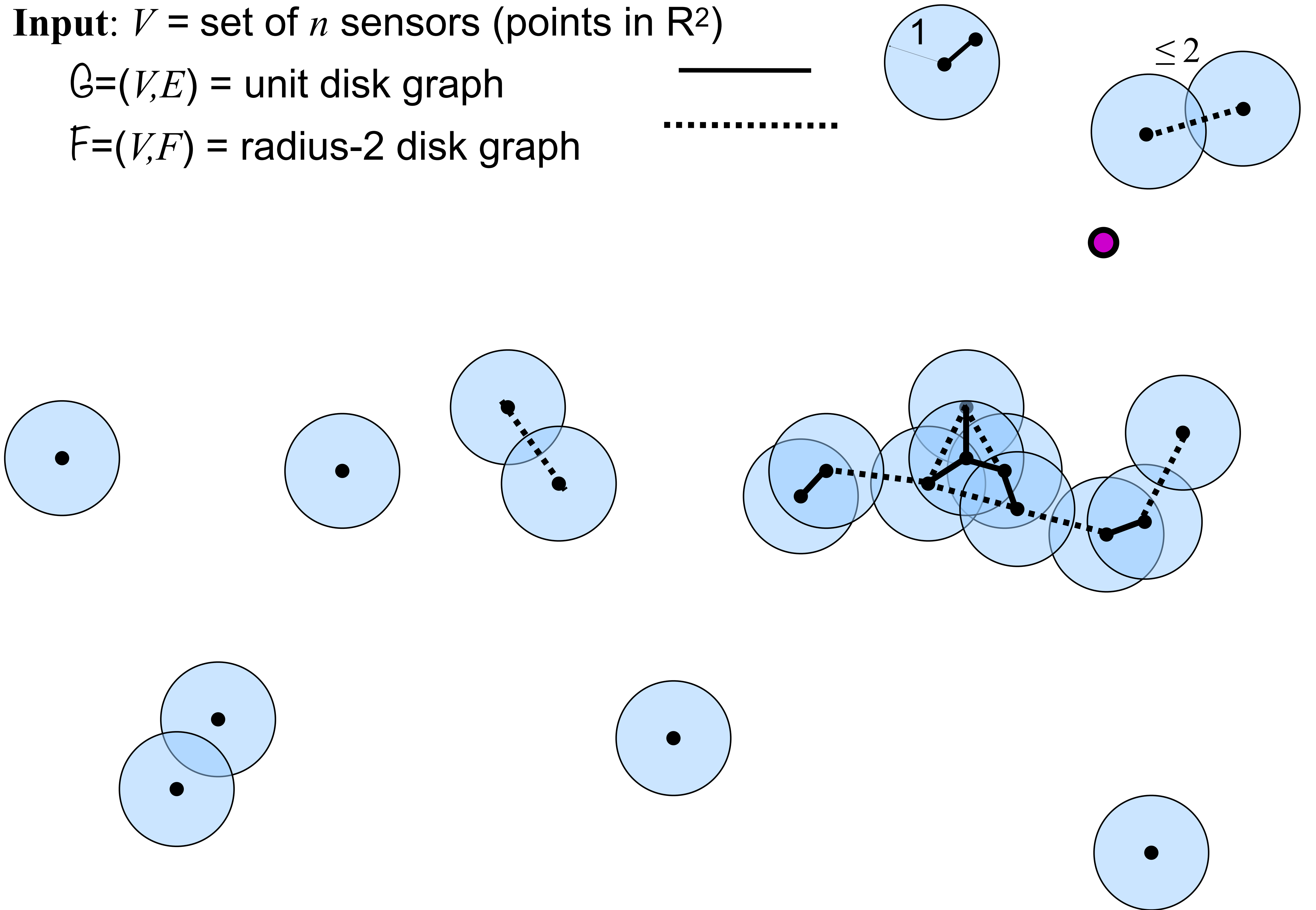


Relay Placement

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$ = unit disk graph

$\mathcal{F}=(V,F)$ = radius-2 disk graph

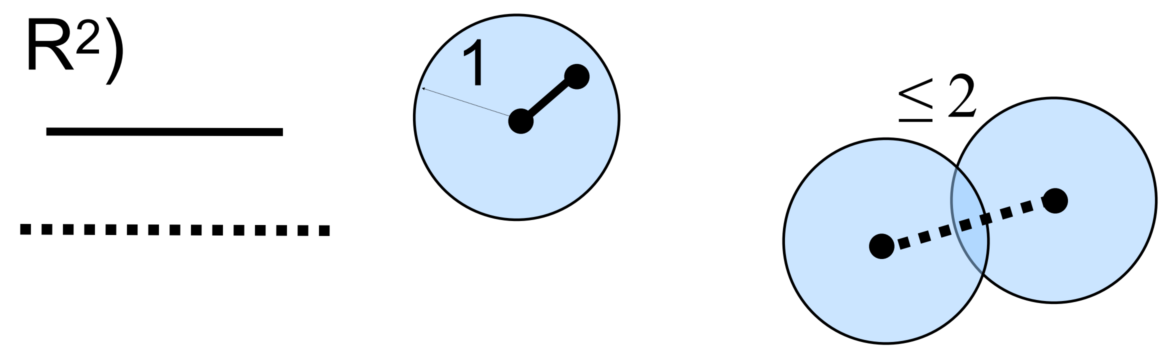


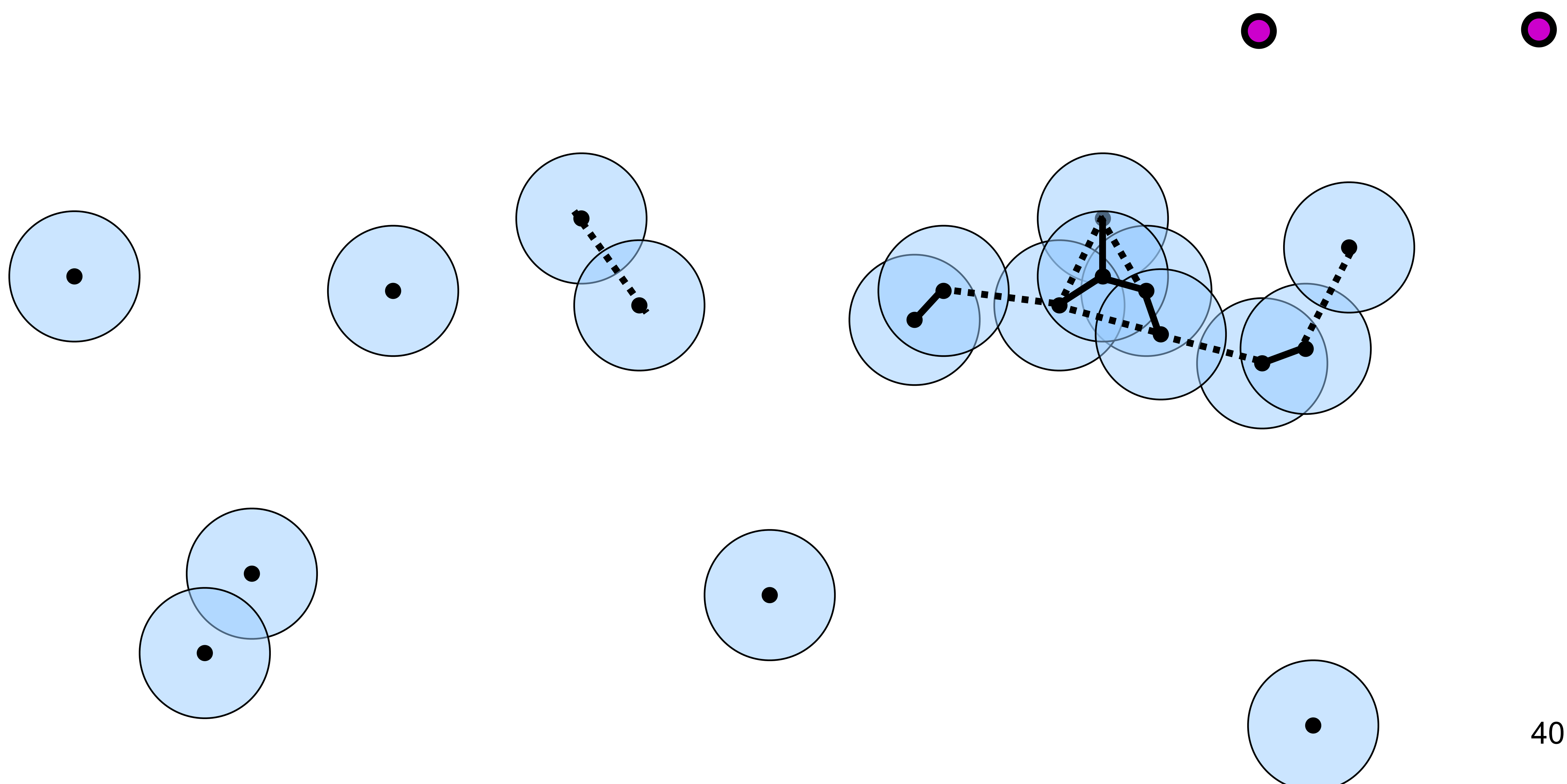
Relay Placement

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$ = unit disk graph

$\mathcal{F}=(V,F)$ = radius-2 disk graph



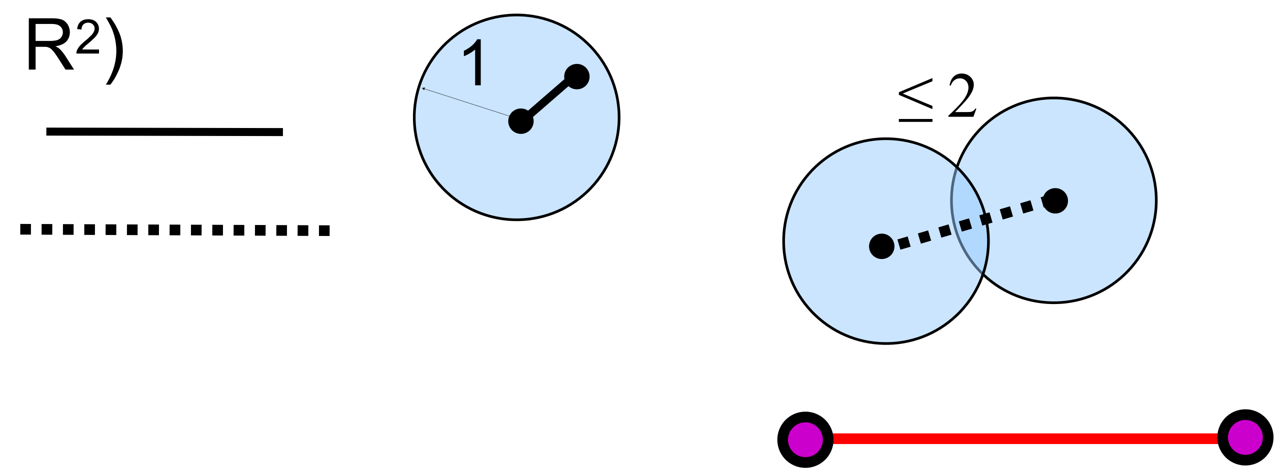


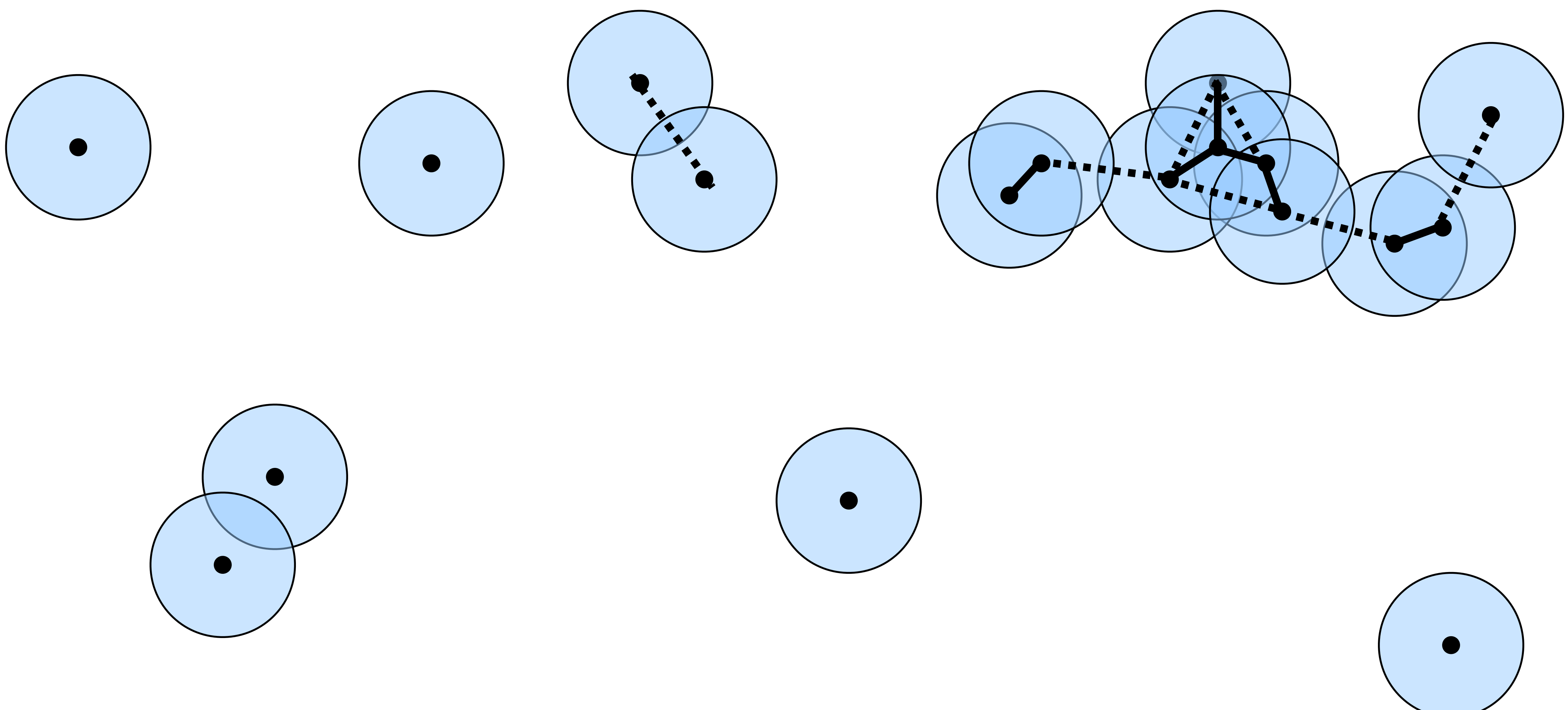
Relay Placement

Input: V = set of n sensors (points in R^2)

$G=(V,E)$ = unit disk graph

$F=(V,F)$ = radius-2 disk graph



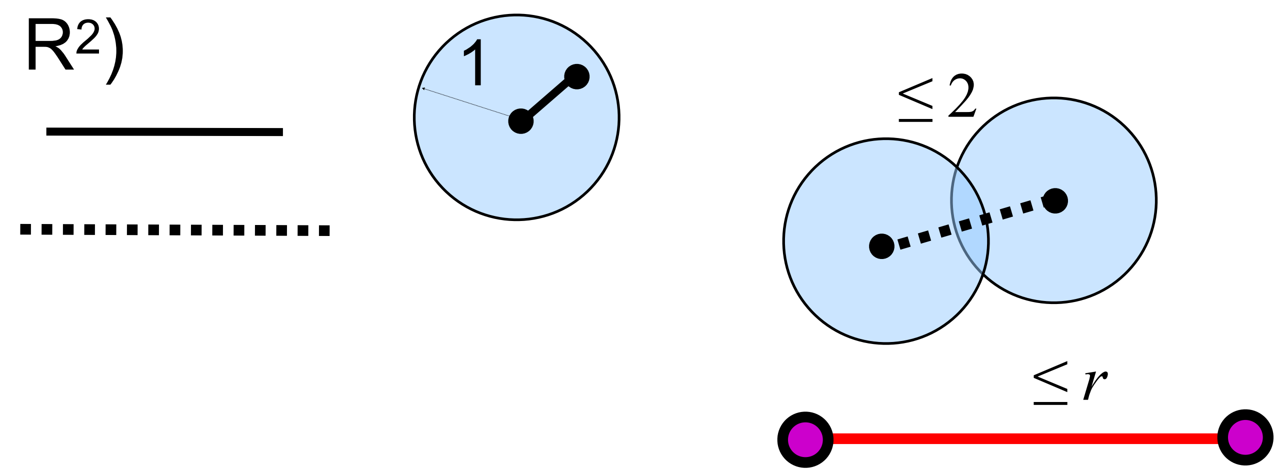


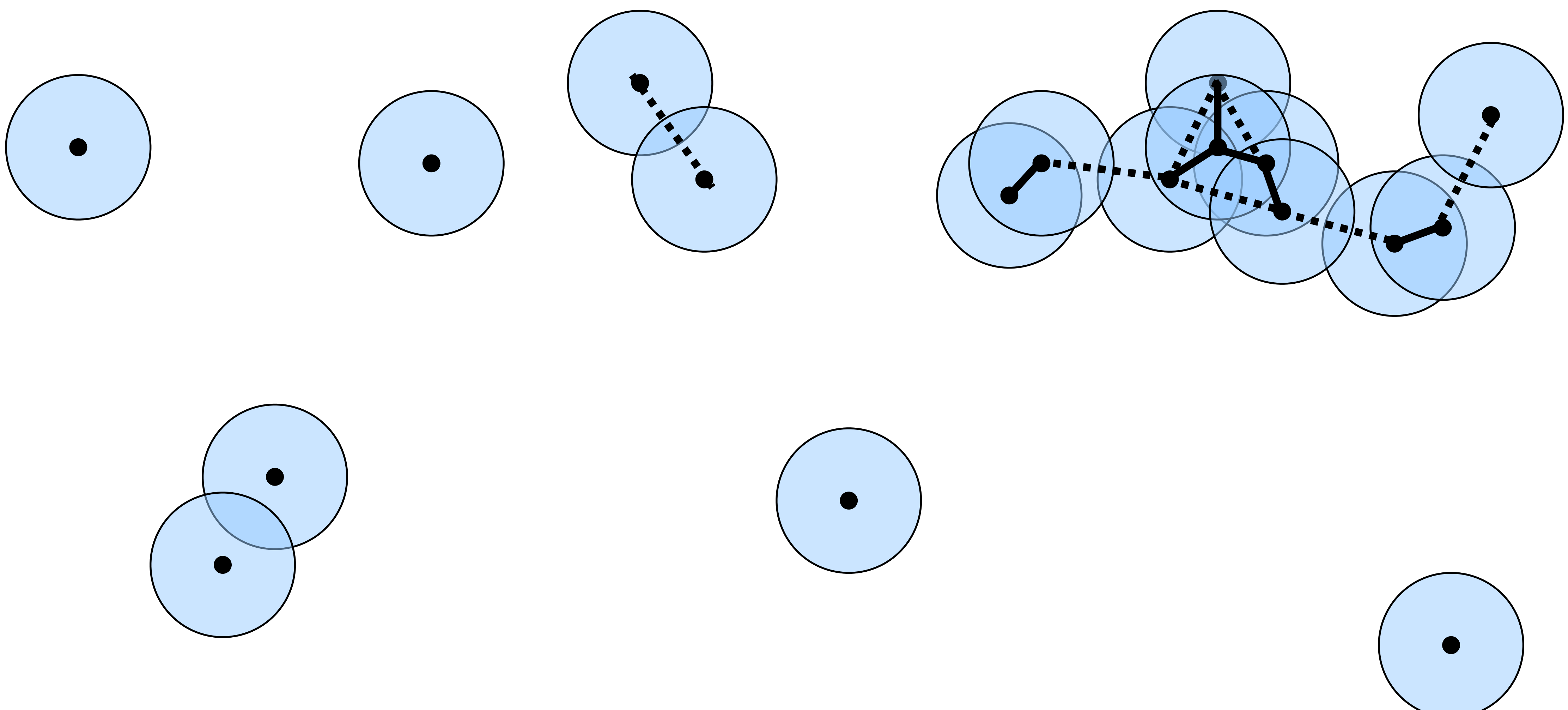
Relay Placement

Input: V = set of n sensors (points in R^2)

$G=(V,E)$ = unit disk graph

$F=(V,F)$ = radius-2 disk graph



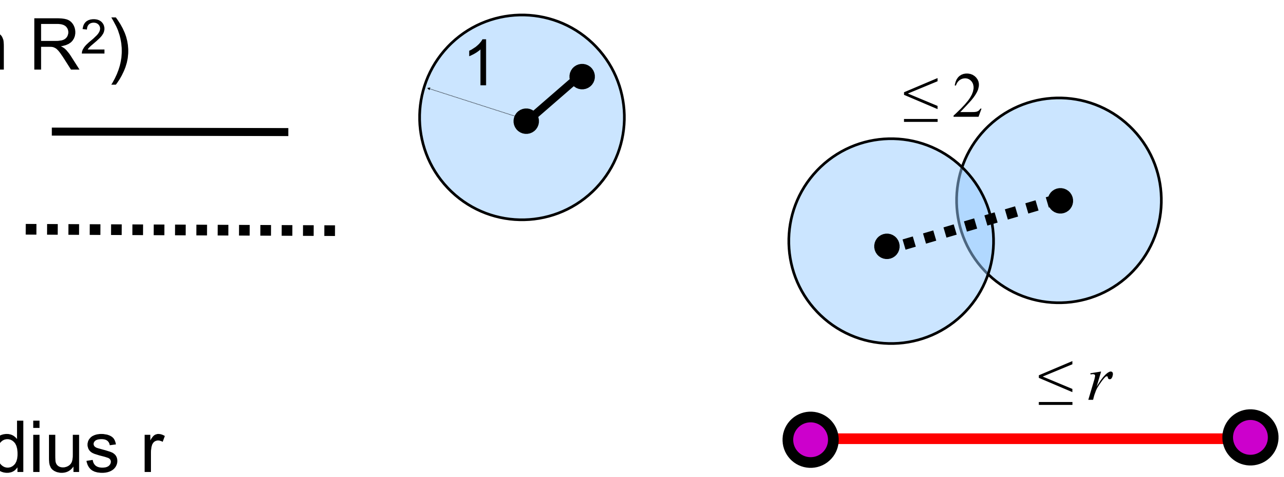


Relay Placement

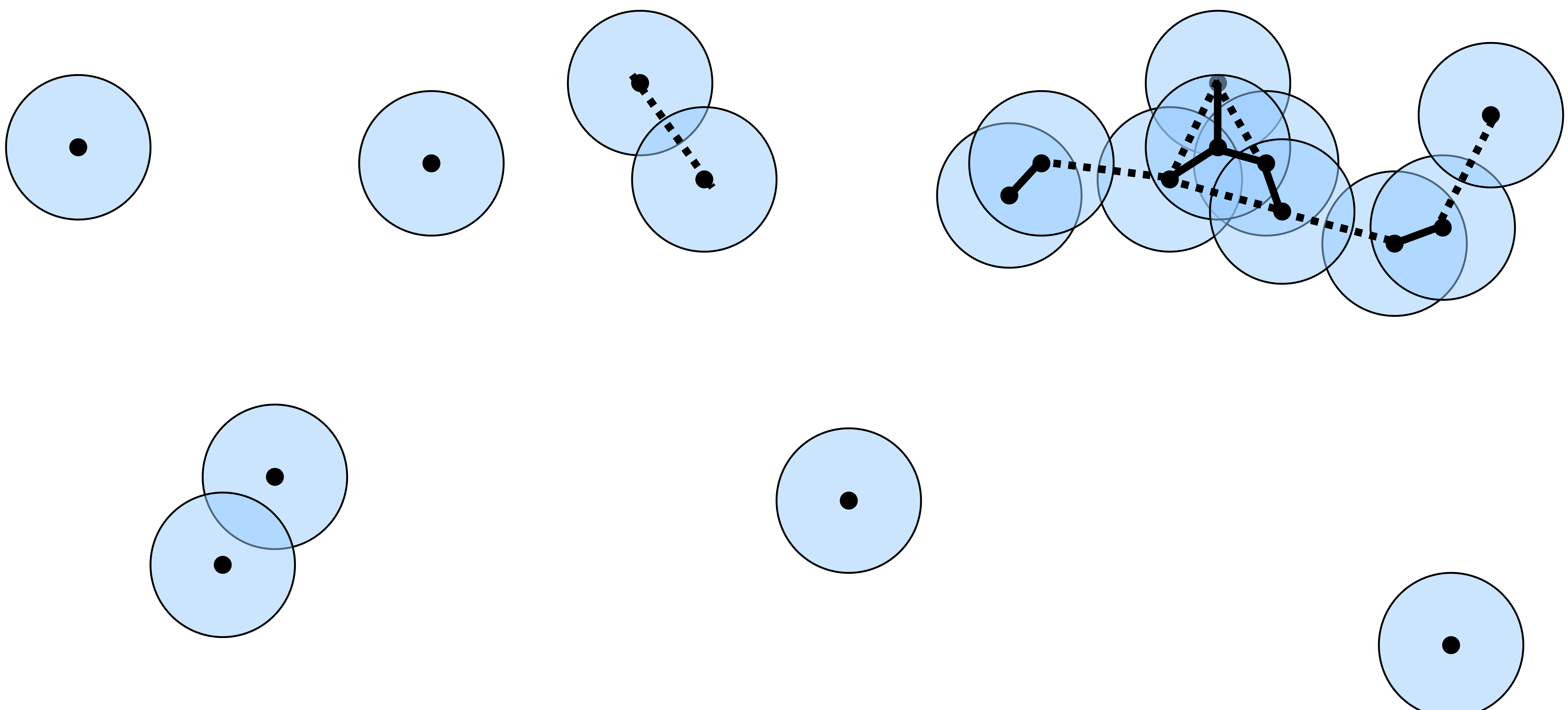
Input: V = set of n sensors (points in R^2)

$G=(V,E)$ = unit disk graph

$F=(V,F)$ = radius-2 disk graph



Add: relays with communication radius r

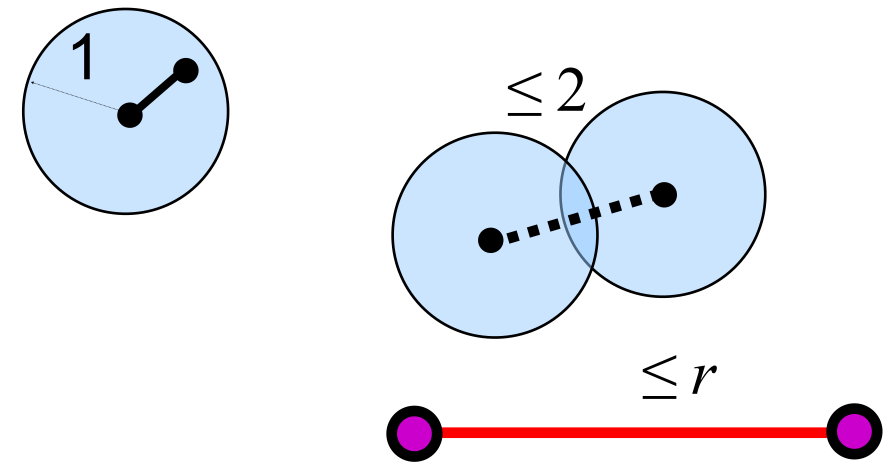


Relay Placement

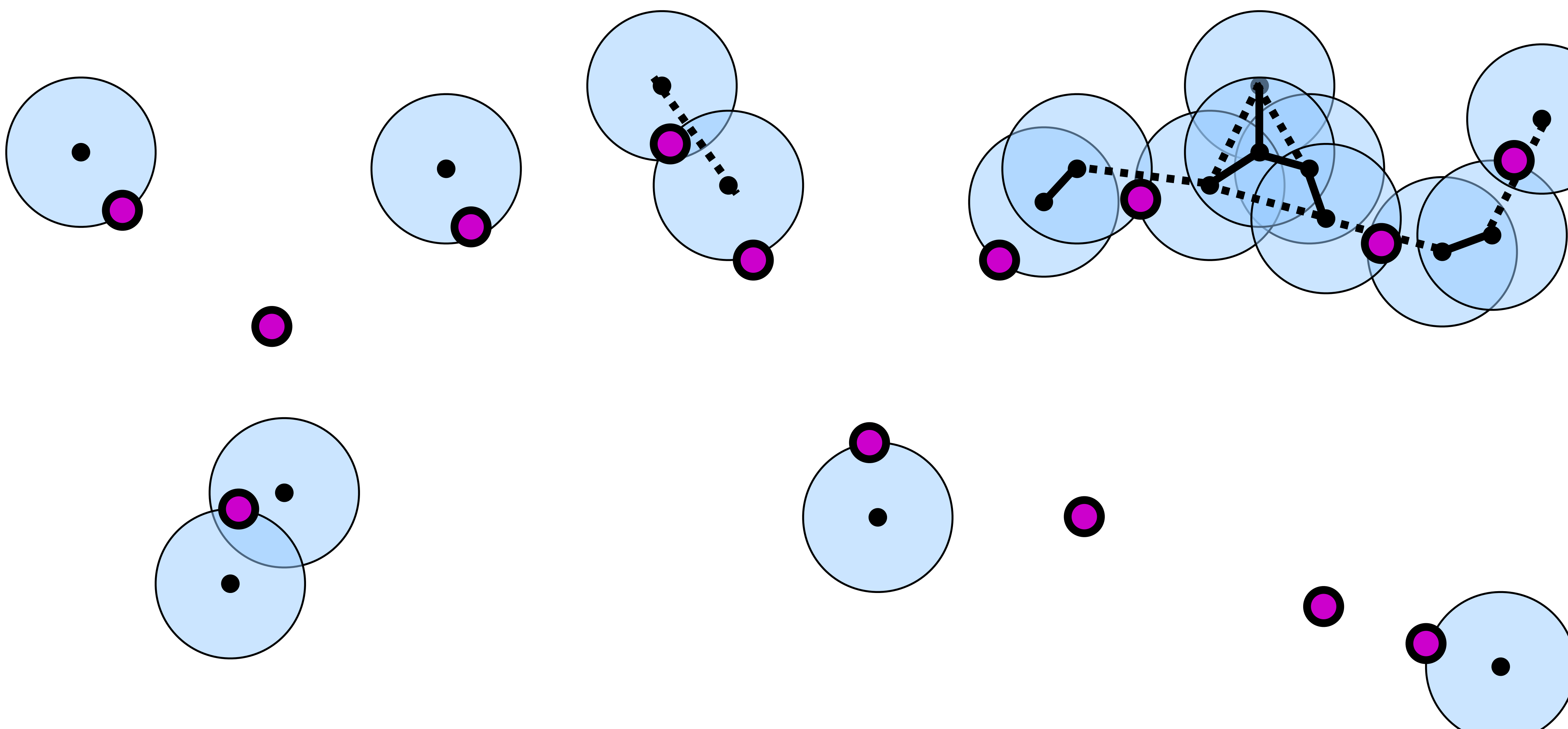
Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$ = unit disk graph

$\mathcal{F}=(V,F)$ = radius-2 disk graph



Add: relays with communication radius r

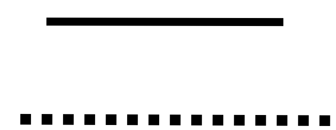


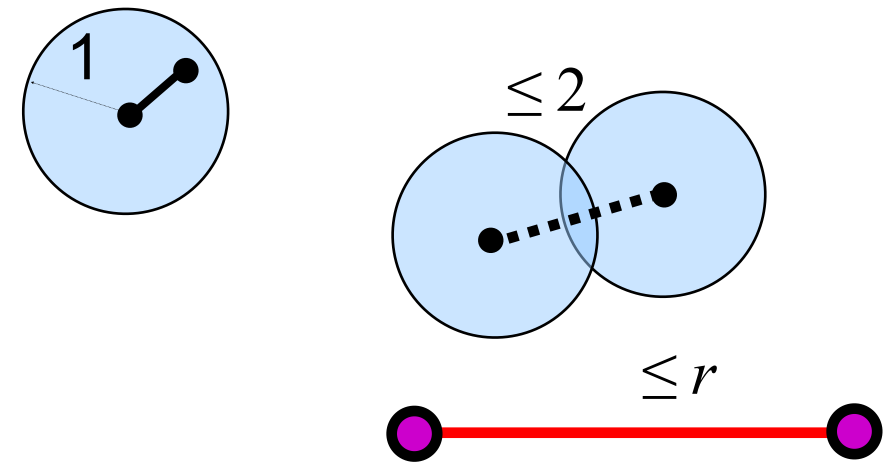
Relay Placement

Input: V = set of n sensors (points in R^2)

$G=(V,E)$ = unit disk graph

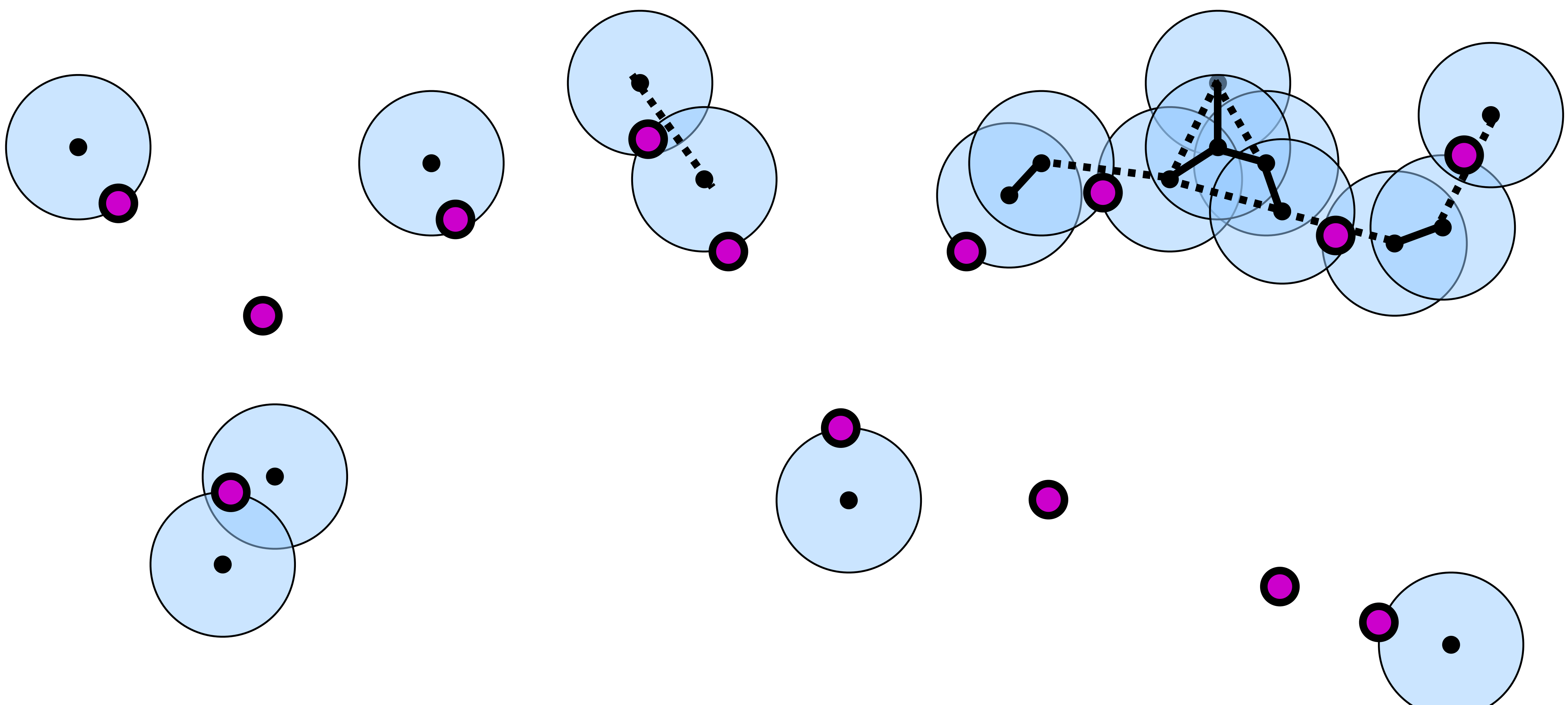
$F=(V,F)$ = radius-2 disk graph





Add: relays with communication radius r

Goal: min # relays (communication radius *r*)

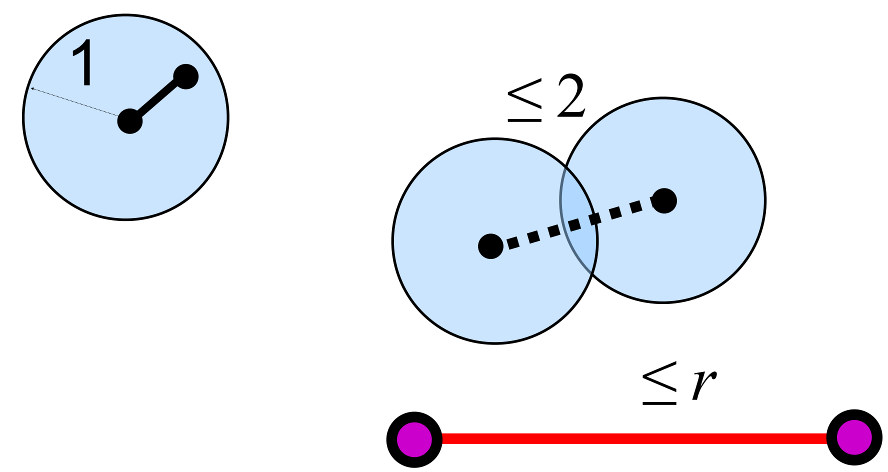


Relay Placement

Input: V = set of n sensors (points in R^2)

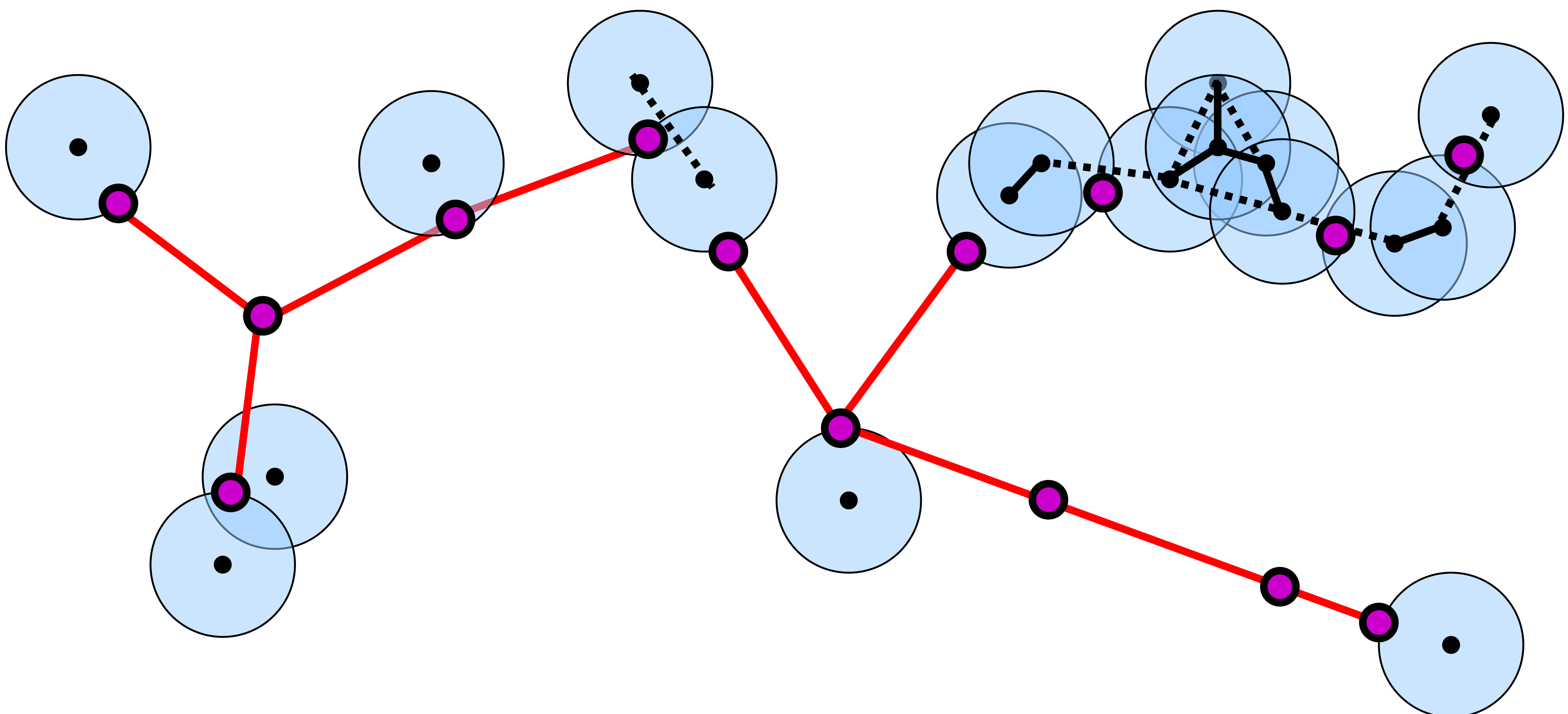
$\mathcal{G}=(V,E)$ = unit disk graph

$\mathcal{F}=(V,F)$ = radius-2 disk graph



Add: relays with communication radius r

Goal: min # relays (communication radius *r*)

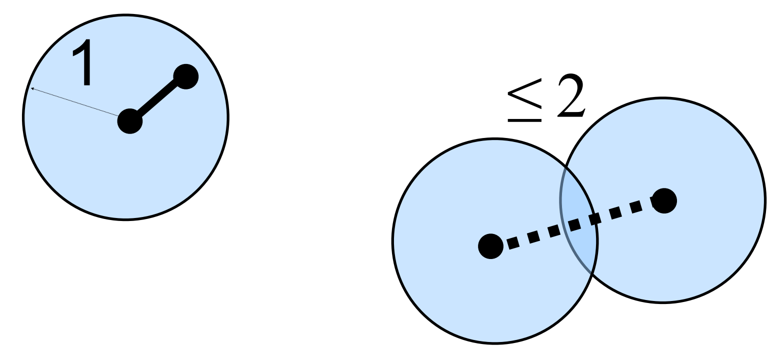


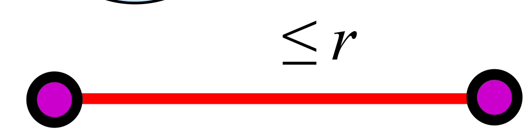
Relay Placement

Input: V = set of n sensors (points in R^2)

$G=(V,E)$ = unit disk graph

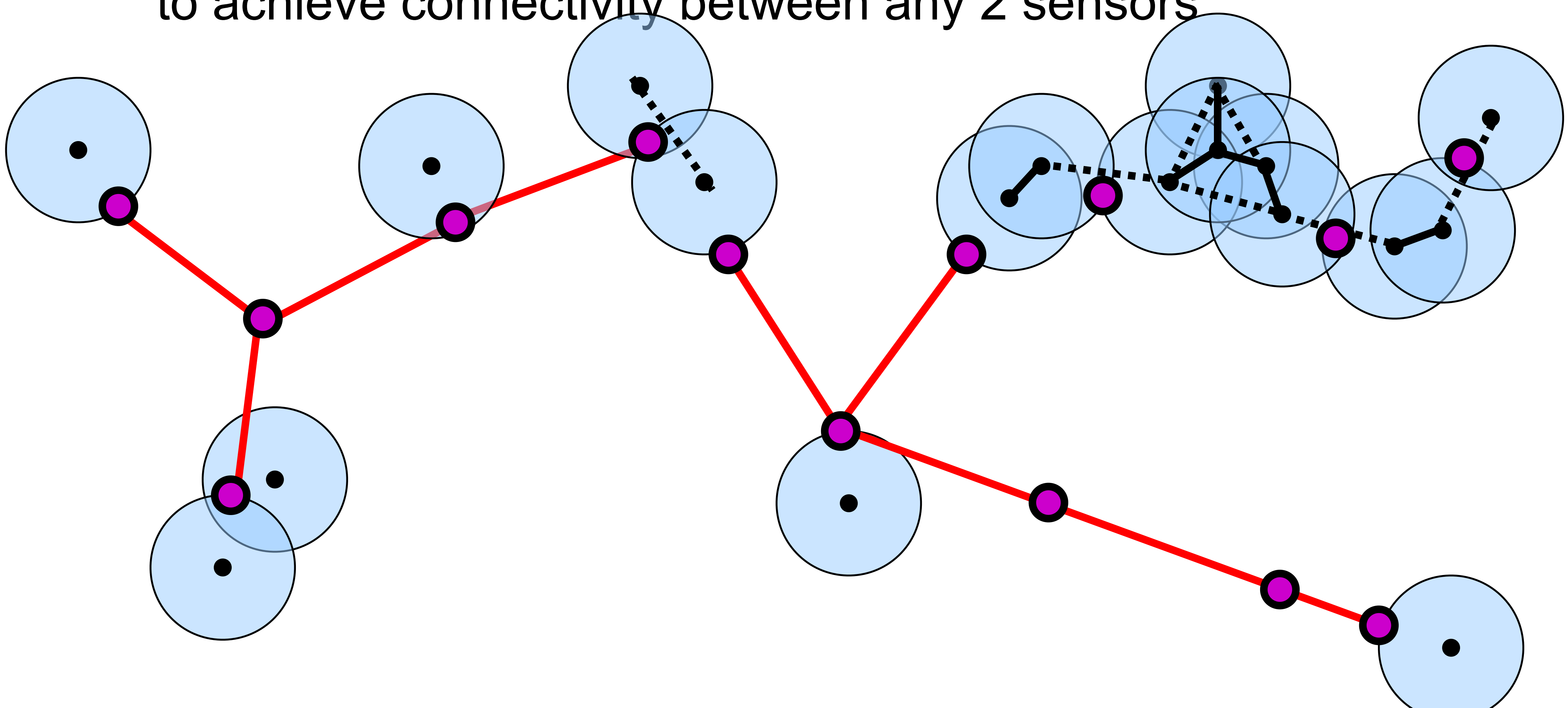
$F=(V,F)$ = radius-2 disk graph





Add: relays with communication radius r

Goal: min # relays (communication radius r)
to achieve connectivity between any 2 sensors



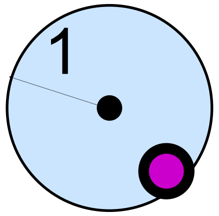
Model of Communication

Model of Communication

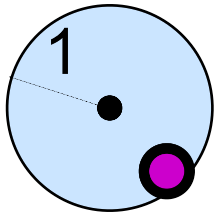
- Sensor and relay: communicate if dist ≤ 1

Model of Communication

- Sensor and relay: communicate if dist ≤ 1



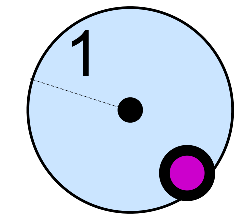
Model of Communication



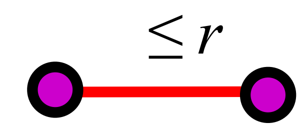
- Sensor and relay: communicate if dist ≤ 1
- Relay and relay: communicate if dist ≤ r

Model of Communication

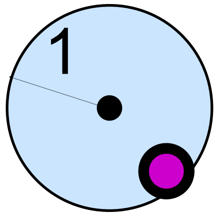
- Sensor and relay: communicate if dist ≤ 1



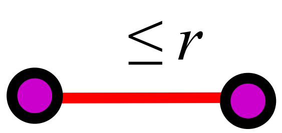
- Relay and relay: communicate if dist ≤ r



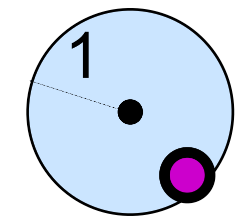
Model of Communication



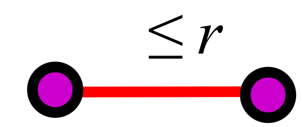
- Sensor and relay: communicate if dist ≤ 1
- Relay and relay: communicate if dist ≤ r
- Two sensors:
  - One-tier: if dist ≤ 1
  - Two-tier: no direct communication (only via a *path of relays*)



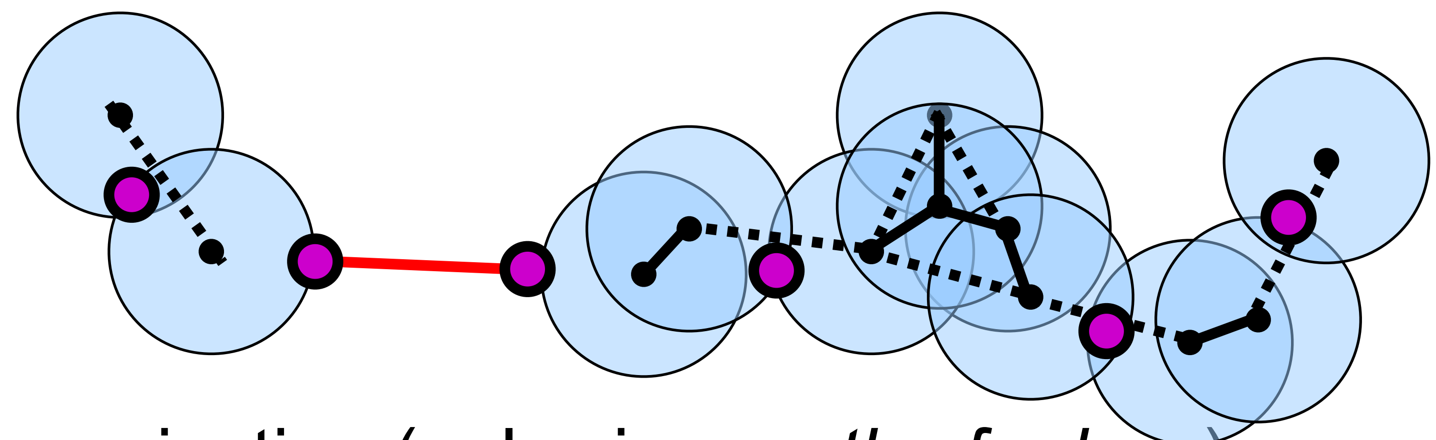
Model of Communication



- Sensor and relay: communicate if dist ≤ 1
- Relay and relay: communicate if dist ≤ r



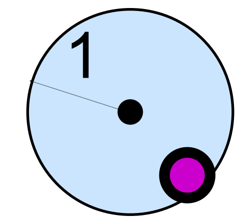
- Two sensors:
  – One-tier: if dist ≤ 1



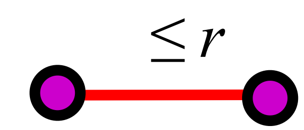
  – Two-tier: no direct communication (only via a *path of relays*)

Model of Communication

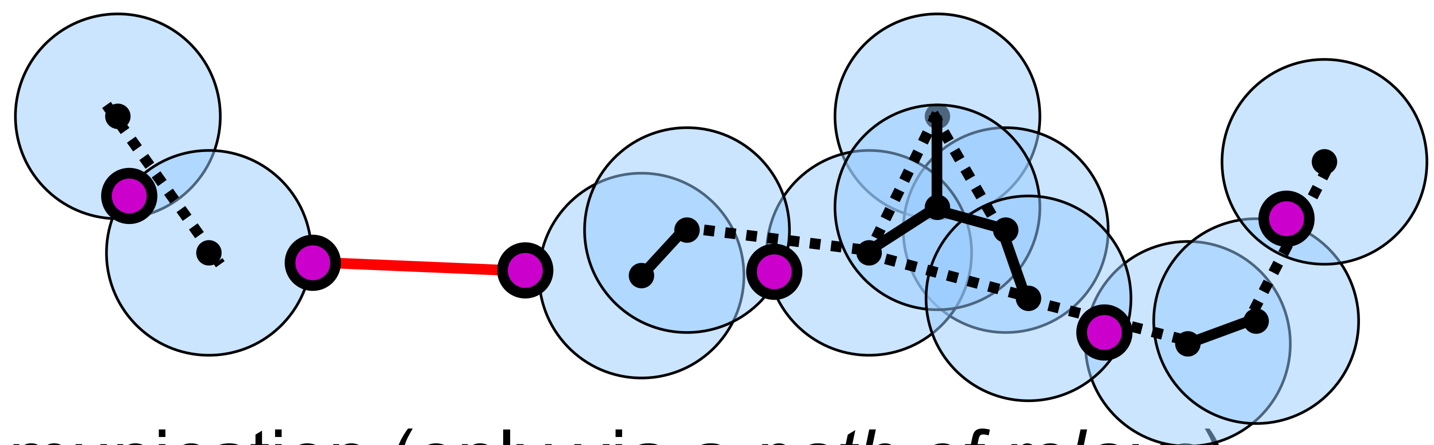
- Sensor and relay: communicate if dist ≤ 1



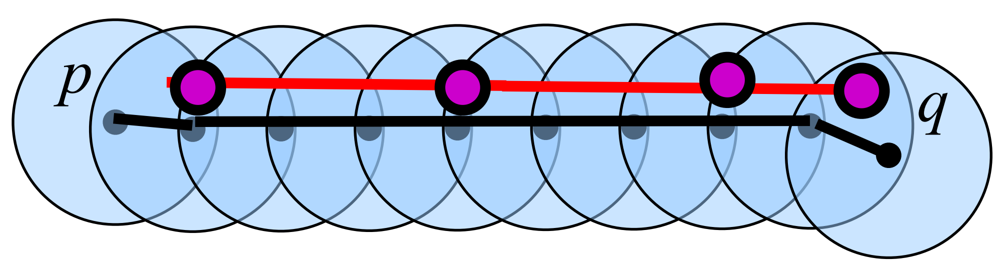
- Relay and relay: communicate if dist ≤ r



- Two sensors:
  - One-tier: if dist ≤ 1



  - Two-tier: no direct communication (only via a *path of relays*)



Prior Related Work

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1**

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1**
  (same as one-tier, $r=1$) :
  - NP-hard, 5-approx [Lin, Xue ’99]
  - [Lin,Xue] is actually a 4-approx; also give 3-approx [Chen et al ’00]
  - Faster 3-approx, randomized 2.5-approx [Cheng et al ’07]

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1** (same as one-tier, $r=1$) :
  – NP-hard, 5-approx [Lin, Xue ’99]
  – [Lin,Xue] is actually a 4-approx; also give 3-approx [Chen et al ’00]
  – Faster 3-approx, randomized 2.5-approx [Cheng et al ’07]
- **7-approx for one-tier, arbitrary *r*** (time $O(n \log n)$)

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1** (same as one-tier, $r=1$) :
  - NP-hard, 5-approx [Lin, Xue ’99]
  - [Lin,Xue] is actually a 4-approx; also give 3-approx [Chen et al ’00]
  - Faster 3-approx, randomized 2.5-approx [Cheng et al ’07]
- **7-approx for one-tier, arbitrary *r*** (time $O(n \log n)$)

  [Lloyd, Xue ’07]

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1** (same as one-tier, $r=1$) :
  - NP-hard, 5-approx [Lin, Xue '99]
  - [Lin,Xue] is actually a 4-approx; also give 3-approx [Chen et al '00]
  - Faster 3-approx, randomized 2.5-approx [Cheng et al '07]
- **7-approx for one-tier, arbitrary *r*** (time $O(n \log n)$) [Lloyd, Xue '07]
- **(5+ε)-approx for two-tier, any r ≥ 1** [Lloyd, Xue '07]

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1** (same as one-tier, $r=1$) :
  - NP-hard, 5-approx [Lin, Xue '99]
  - [Lin,Xue] is actually a 4-approx; also give 3-approx [Chen et al '00]
  - Faster 3-approx, randomized 2.5-approx [Cheng et al '07]
- **7-approx for one-tier, arbitrary *r*** (time $O(n \log n)$) [Lloyd, Xue '07]
- **($5+\varepsilon$)-approx for two-tier, any $r \geq 1$** [Lloyd, Xue '07]
- **($4+\varepsilon$)-approx for two-tier, any $r \geq 2$**

Prior Related Work

- **Steiner trees with min # Steiner points, edge length ≤ 1** (same as one-tier, $r=1$) :
  - NP-hard, 5-approx [Lin, Xue '99]
  - [Lin,Xue] is actually a 4-approx; also give 3-approx [Chen et al '00]
  - Faster 3-approx, randomized 2.5-approx [Cheng et al '07]
- **7-approx for one-tier, arbitrary *r*** (time $O(n \log n)$)
  [Lloyd, Xue '07]
- **(5+ε)-approx for two-tier, any r ≥ 1** [Lloyd, Xue '07]
- **(4+ε)-approx for two-tier, any r ≥ 2**
  [Srinivas, Zussman, Modiano '06]

Our Contributions

Our Contributions

- Simple O(n log n)-time 6.73-approx for one-tier relay placement [vs. previous 7-approx]

Our Contributions

- Simple O(n log n)-time 6.73-approx for one-tier relay placement [vs. previous 7-approx]
- Poly-time 3.11-approx for one-tier, any $r \geq 1$

Our Contributions

- Simple O(n log n)-time 6.73-approx for one-tier relay placement [vs. previous 7-approx]
- Poly-time 3.11-approx for one-tier, any $r \geq 1$
- No PTAS for one-tier (r is part of input), assuming P $\neq$ NP

Our Contributions

- Simple O(n log n)-time 6.73-approx for one-tier relay placement [vs. previous 7-approx]
- Poly-time 3.11-approx for one-tier, any $r \geq 1$
- No PTAS for one-tier (r is part of input), assuming P ≠ NP
- PTAS for two-tier relay placement

Our Contributions

- Simple O(n log n)-time 6.73-approx for one-tier relay placement [vs. previous 7-approx]
- Poly-time 3.11-approx for one-tier, any $r \geq 1$
- No PTAS for one-tier (r is part of input), assuming P ≠ NP
- PTAS for two-tier relay placement

  [vs. previous (5+ε)-approx]

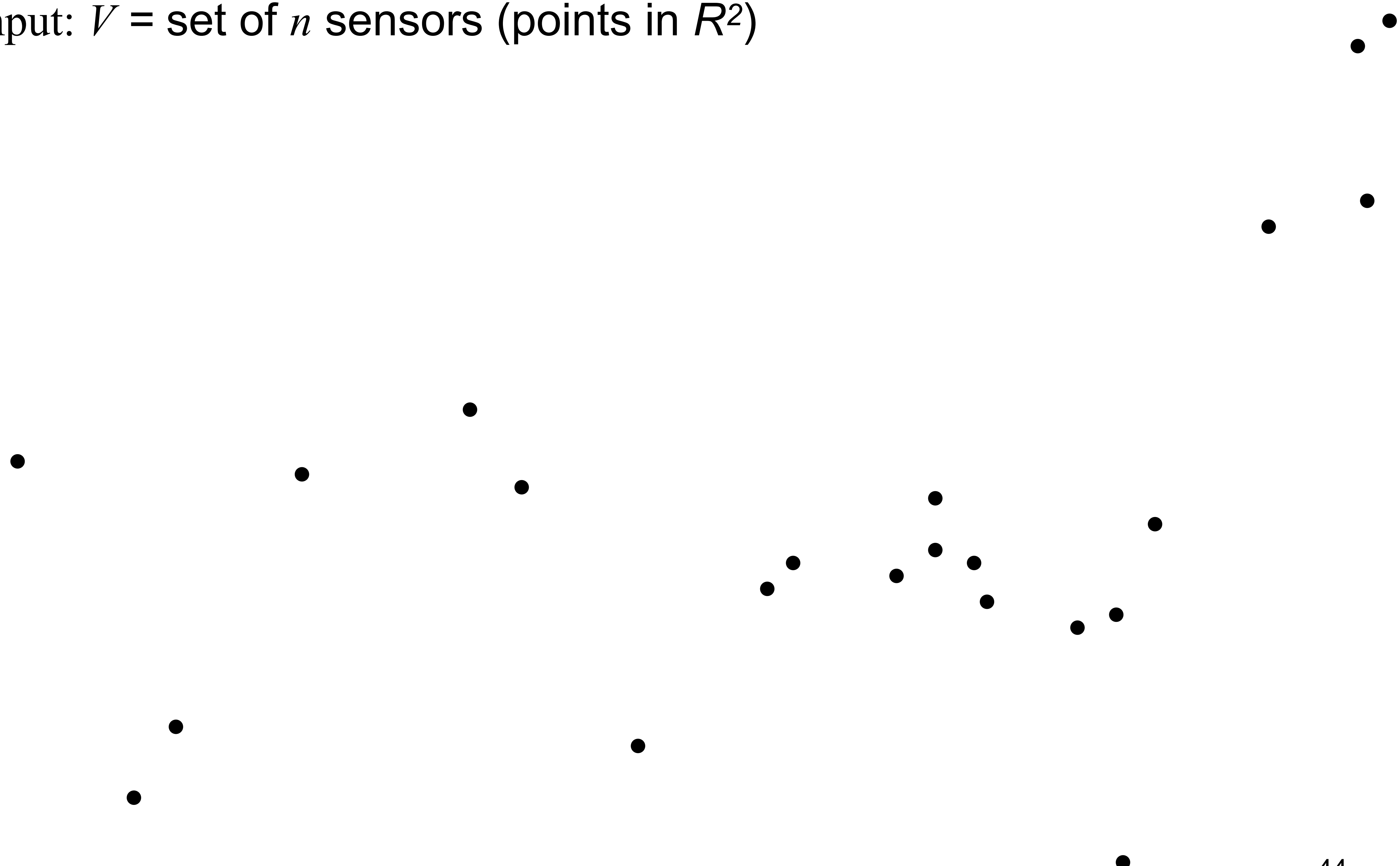
Terminology: Blobs and Clouds

Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

Terminology: Blobs and Clouds

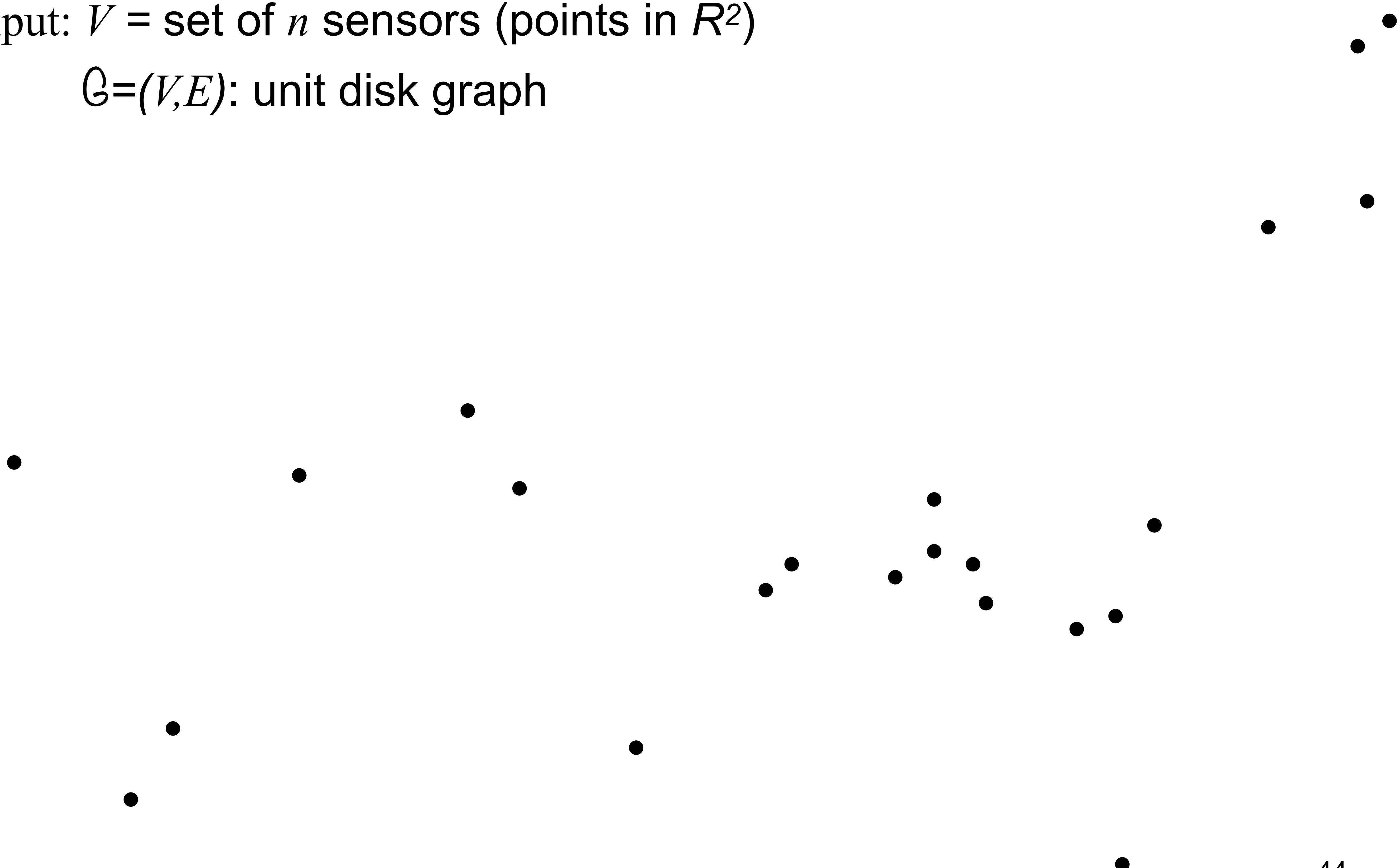
Input: V = set of n sensors (points in R^2)



Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

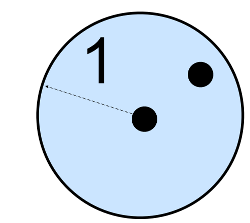
$G=(V,E)$: unit disk graph

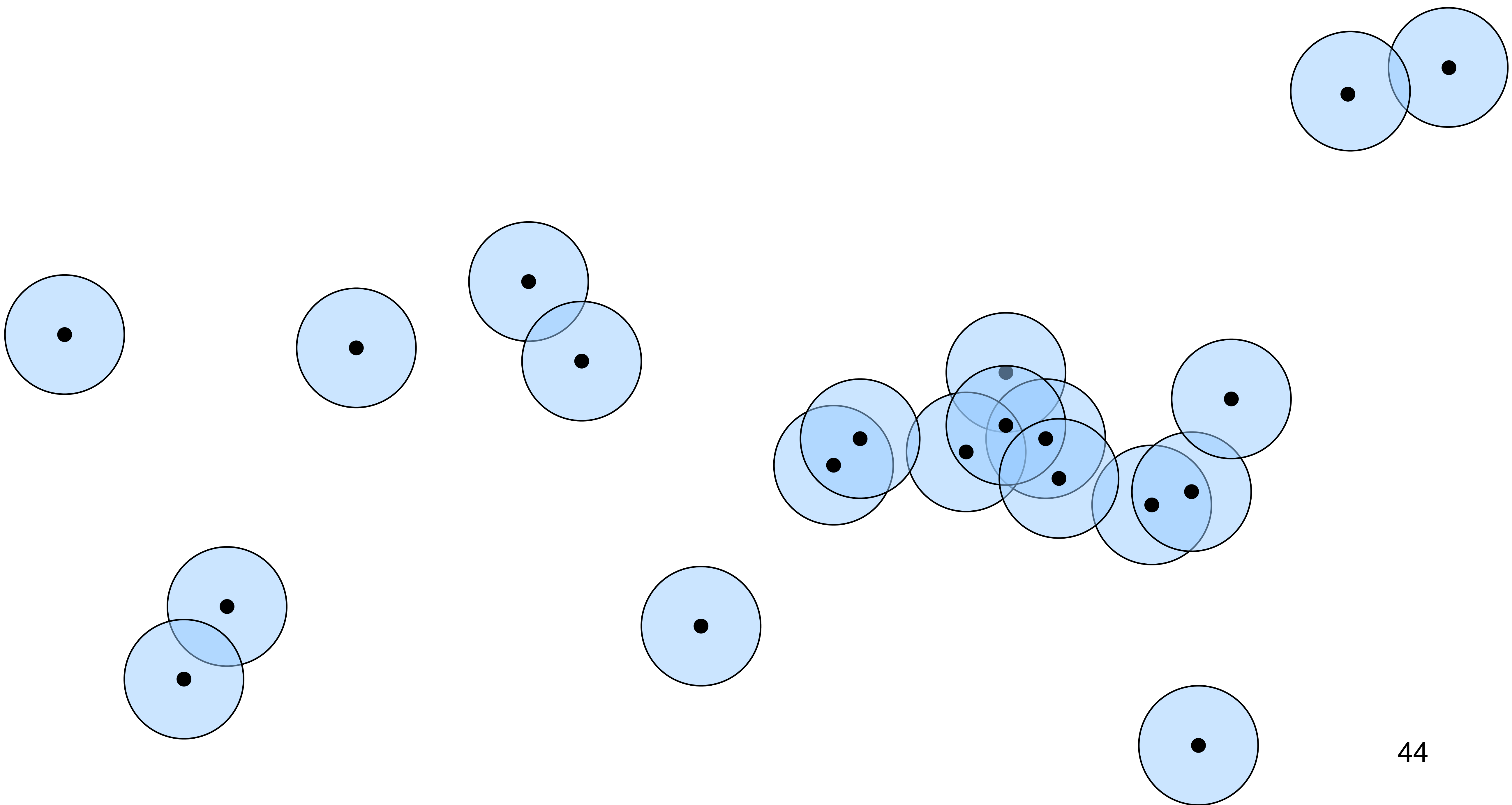


Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

$G=(V,E)$: unit disk graph

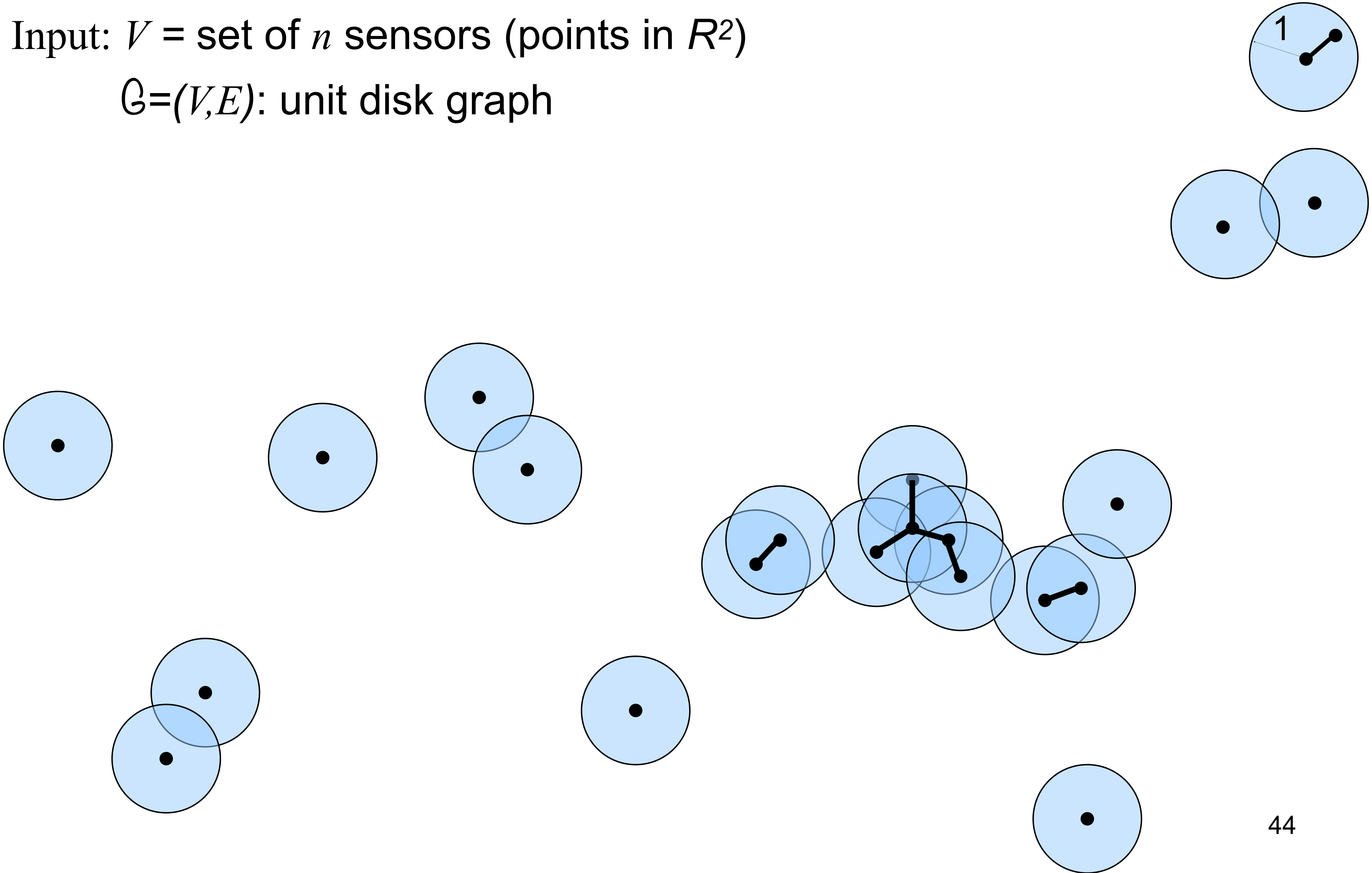




Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

$G=(V,E)$: unit disk graph

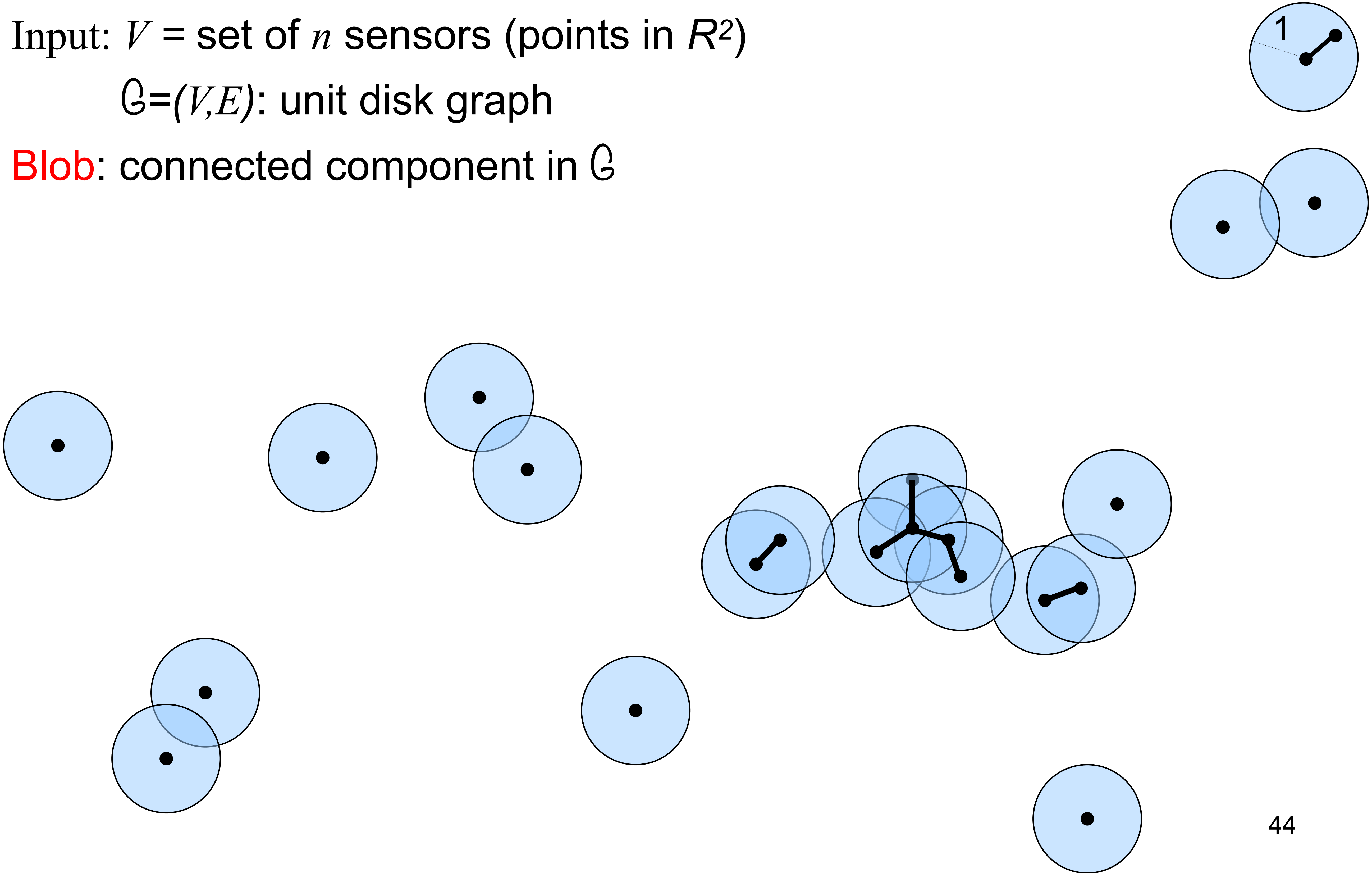


Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$: unit disk graph

Blob: connected component in $\mathcal{G}$

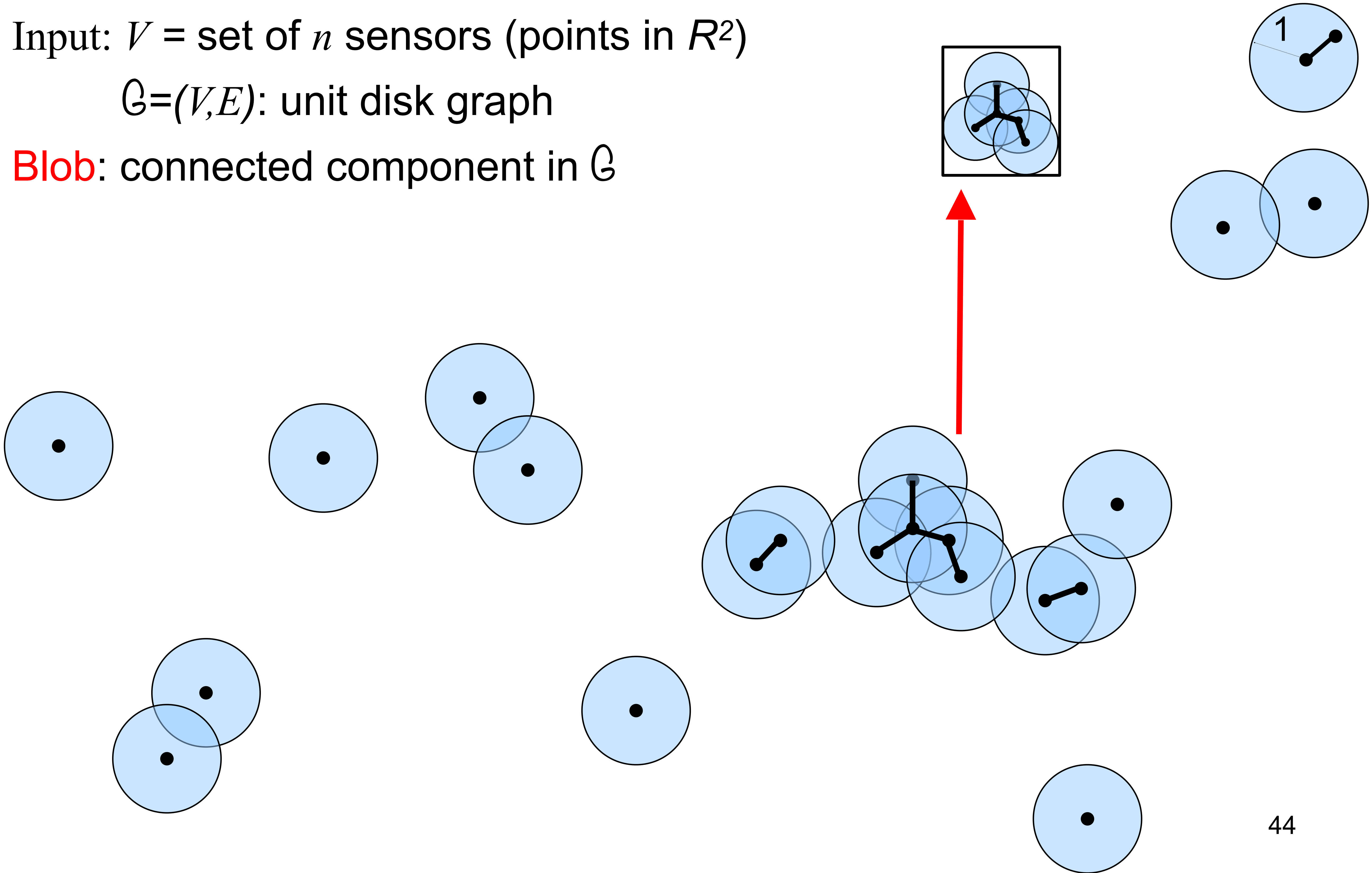


Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$: unit disk graph

Blob: connected component in $\mathcal{G}$

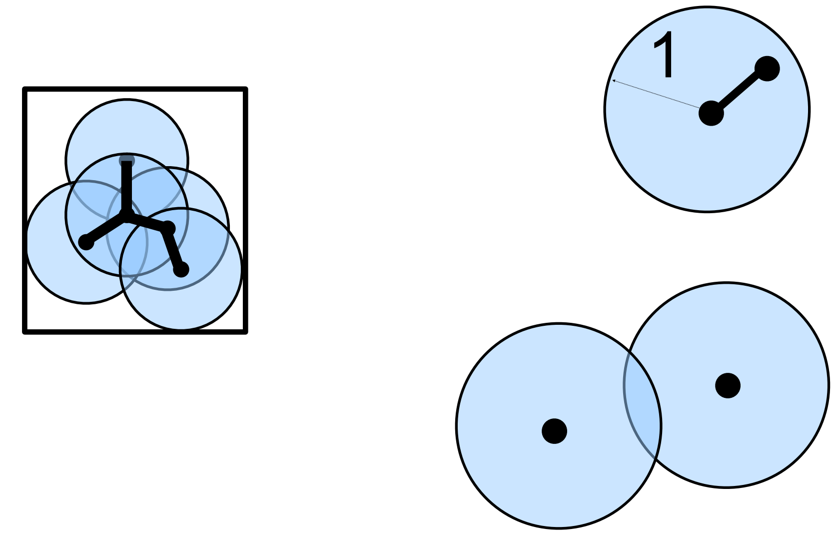


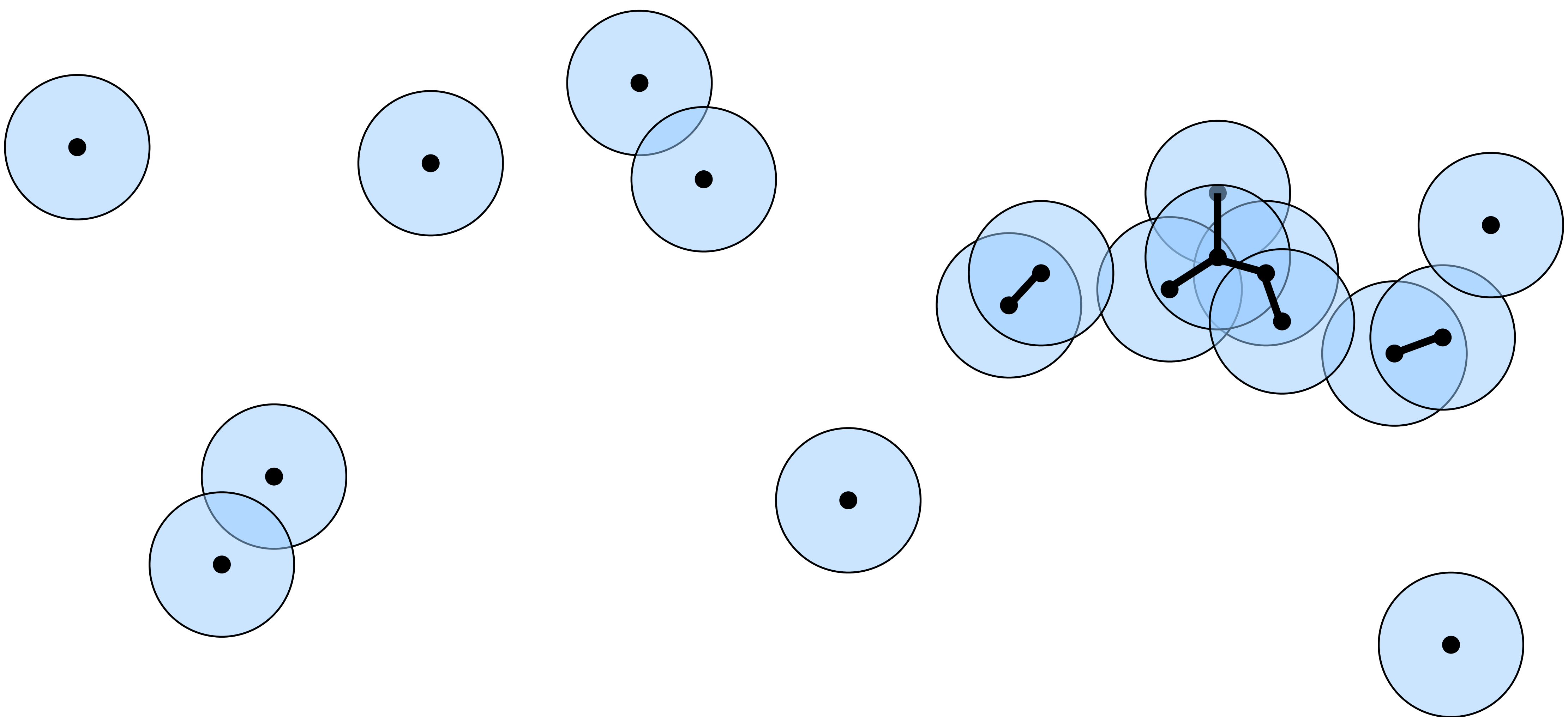
Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$: unit disk graph

Blob: connected component in $\mathcal{G}$





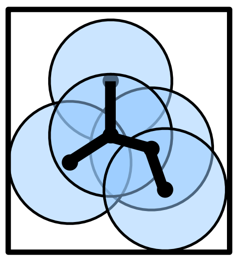
Terminology: Blobs and Clouds

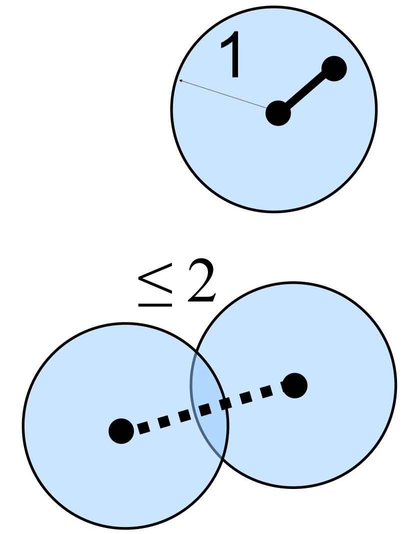
Input: V = set of n sensors (points in R^2)

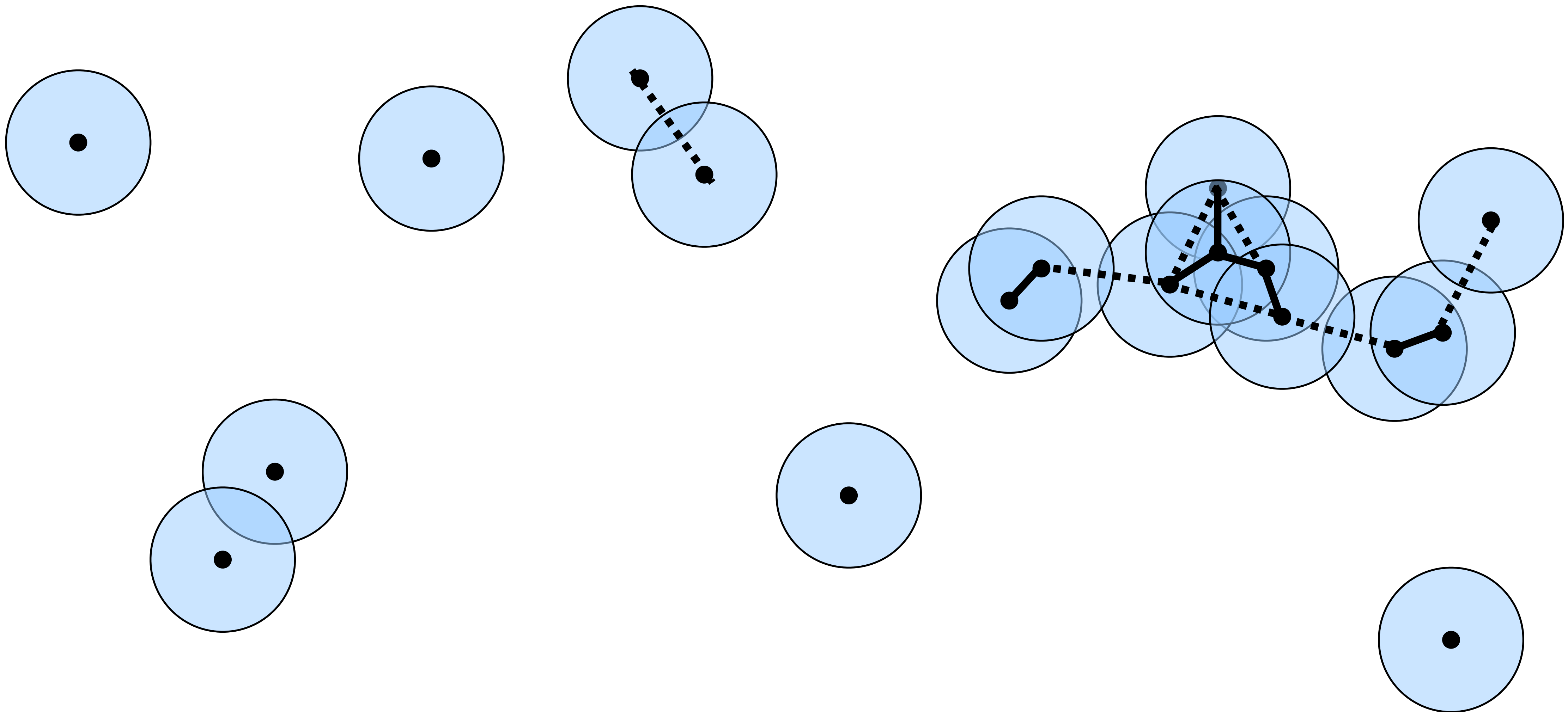
$\mathcal{G}=(V,E)$: unit disk graph

Blob: connected component in $\mathcal{G}$

$\mathcal{F}=(V,F)$: radius-2 disk graph







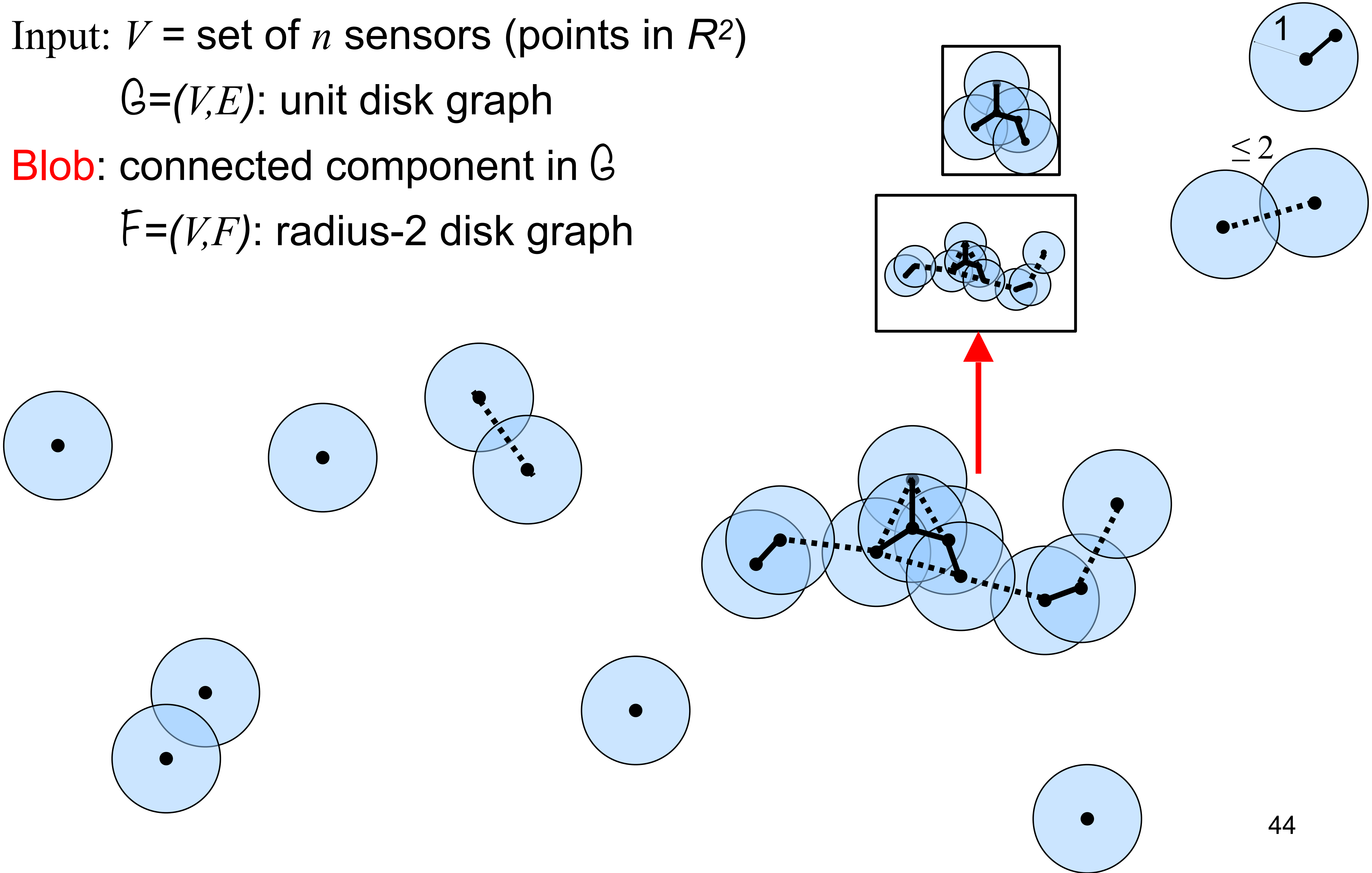
Terminology: Blobs and Clouds

Input: V = set of n sensors (points in R^2)

$\mathcal{G}=(V,E)$: unit disk graph

Blob: connected component in $\mathcal{G}$

$\mathcal{F}=(V,F)$: radius-2 disk graph



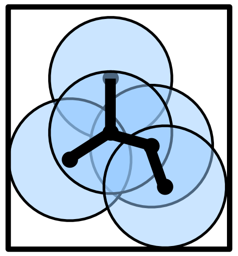
Terminology: Blobs and Clouds

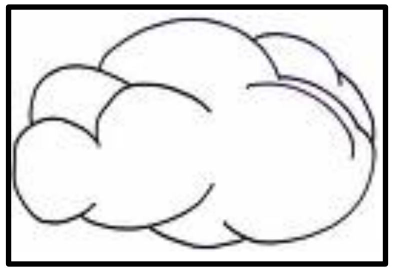
Input: V = set of n sensors (points in R^2)

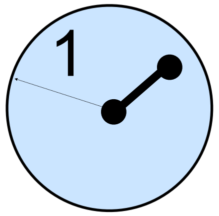
$\mathcal{G}=(V,E)$: unit disk graph

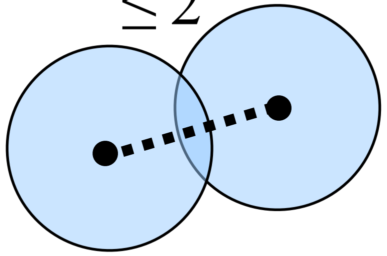
Blob: connected component in $\mathcal{G}$

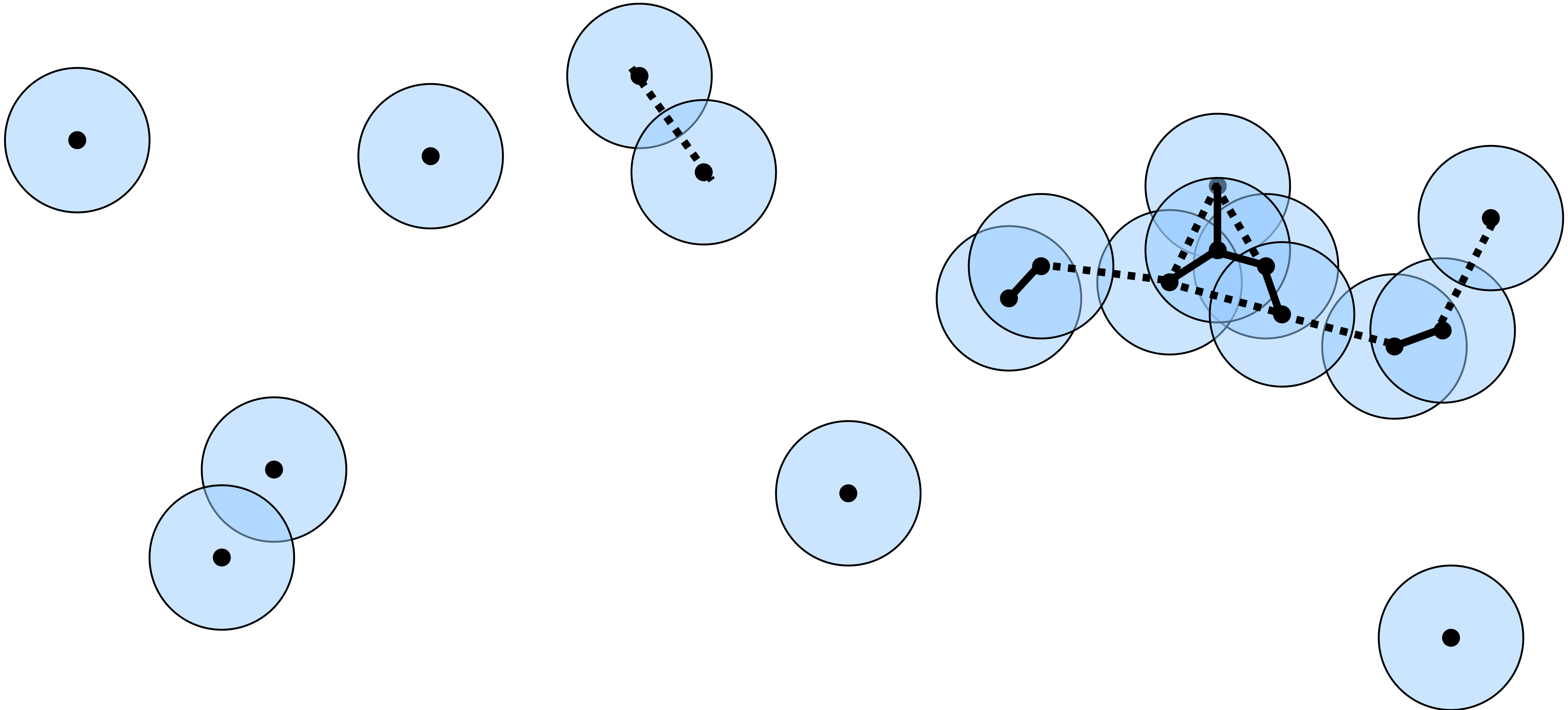
$\mathcal{F}=(V,F)$: radius-2 disk graph











Terminology: Blobs and Clouds

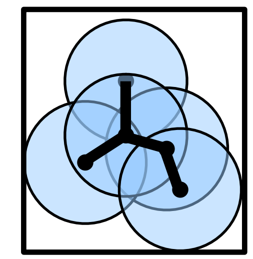
Input: V = set of n sensors (points in R^2)

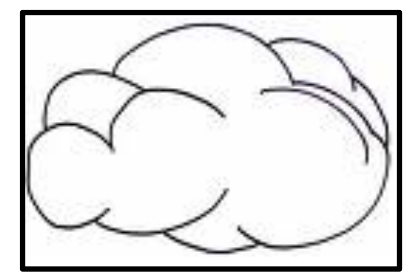
$\mathcal{G}=(V,E)$: unit disk graph

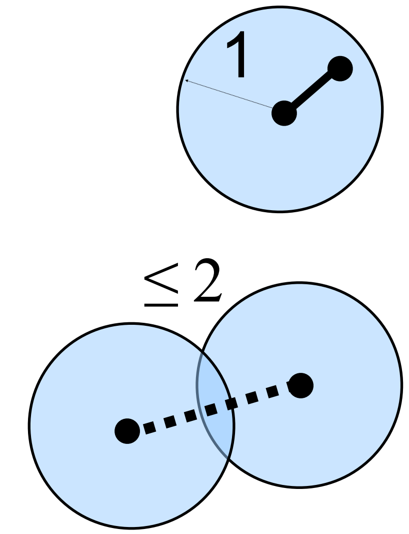
Blob: connected component in $\mathcal{G}$

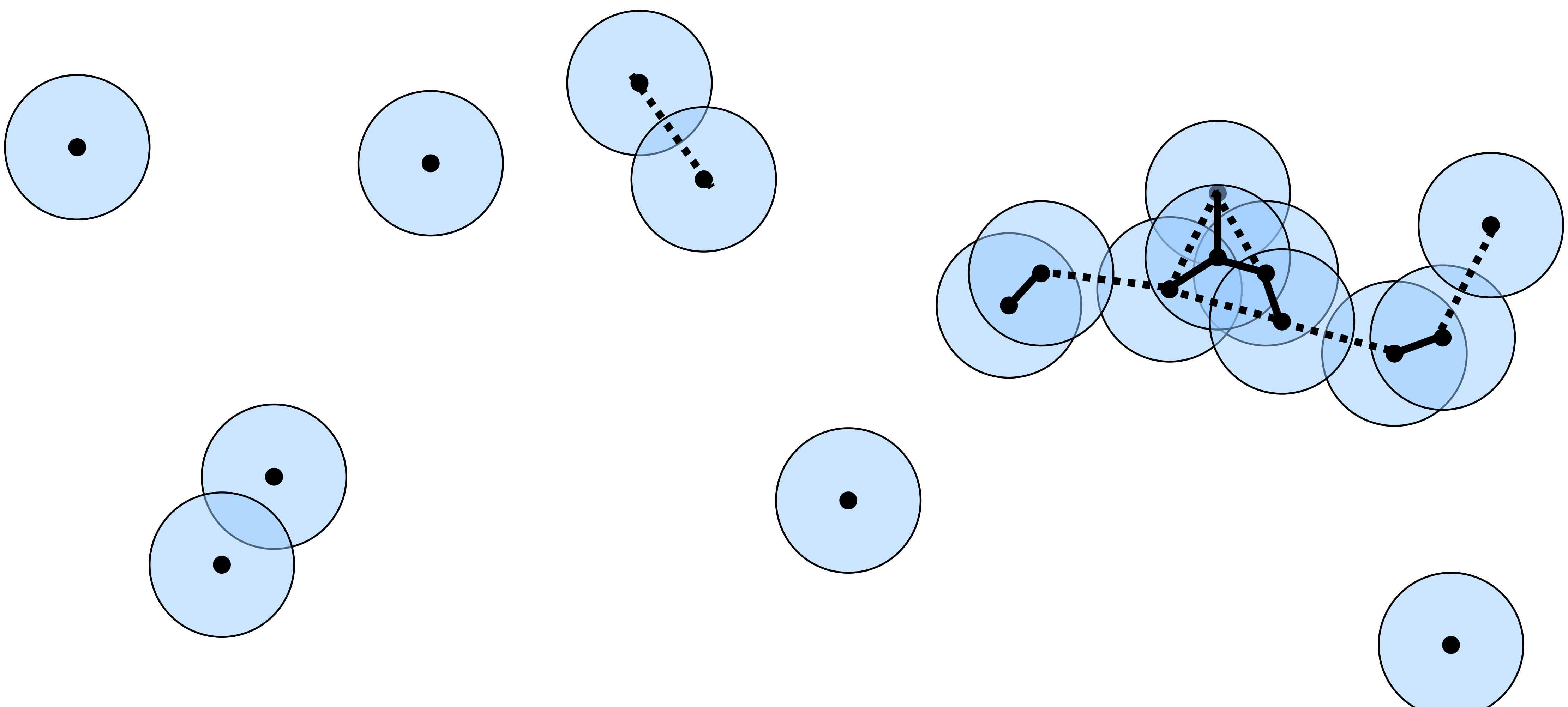
$\mathcal{F}=(V,F)$: radius-2 disk graph

Cloud: connected component in $\mathcal{F}$









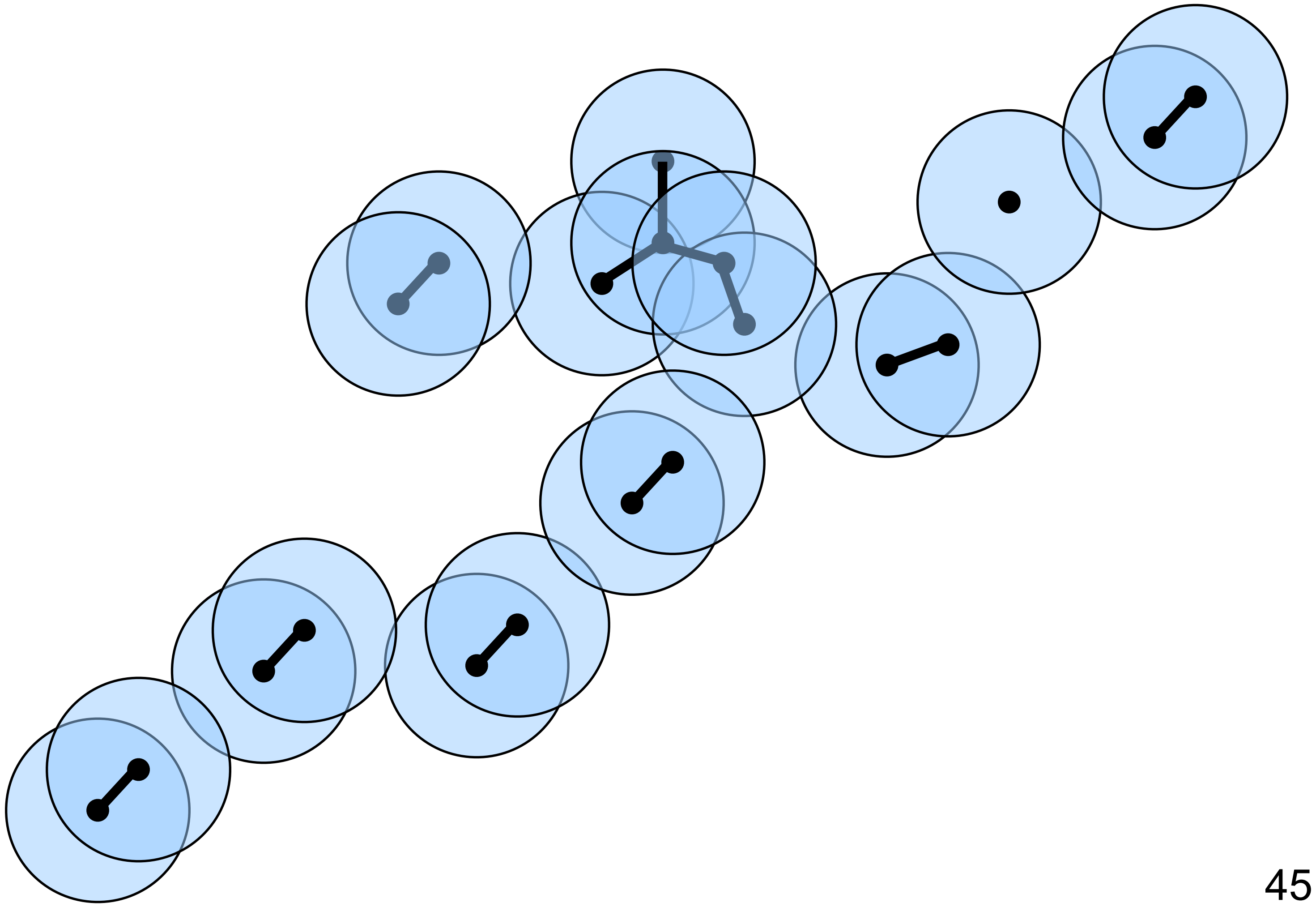
Lemma 1

Lemma 1

Lemma 1: To connect all sensors within a cloud C,

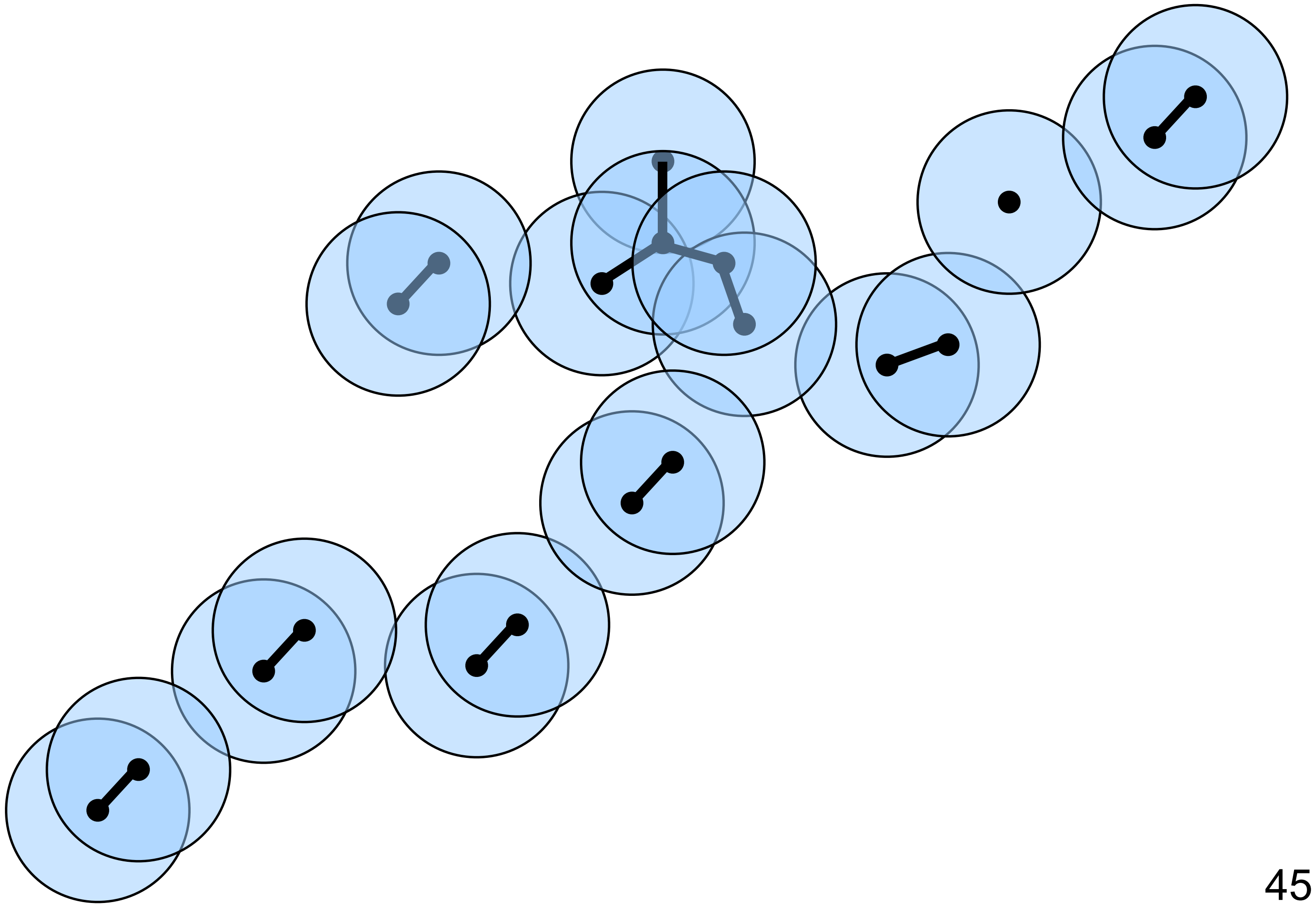
Lemma 1

Lemma 1: To connect all sensors within a cloud C,



Lemma 1

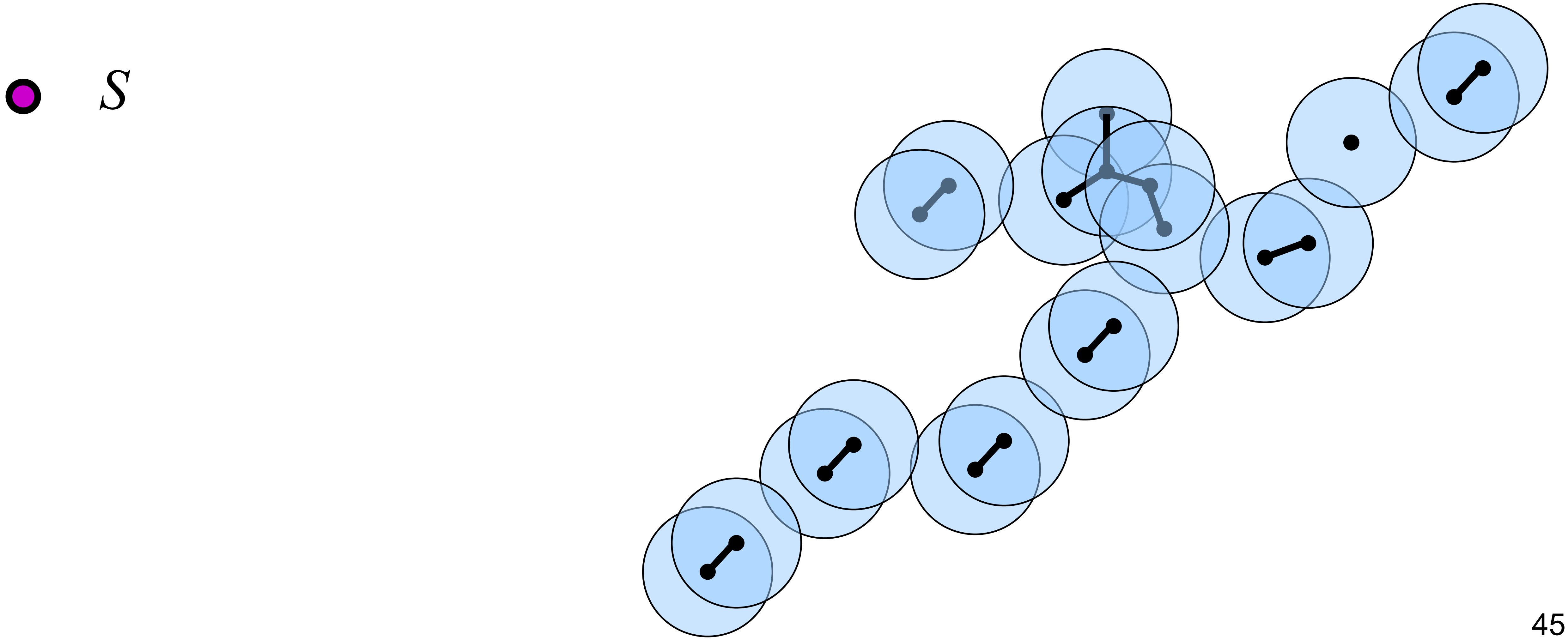
Lemma 1: To connect all sensors within a cloud C,
it suffices to take a set S that stabs all blobs,



Lemma 1

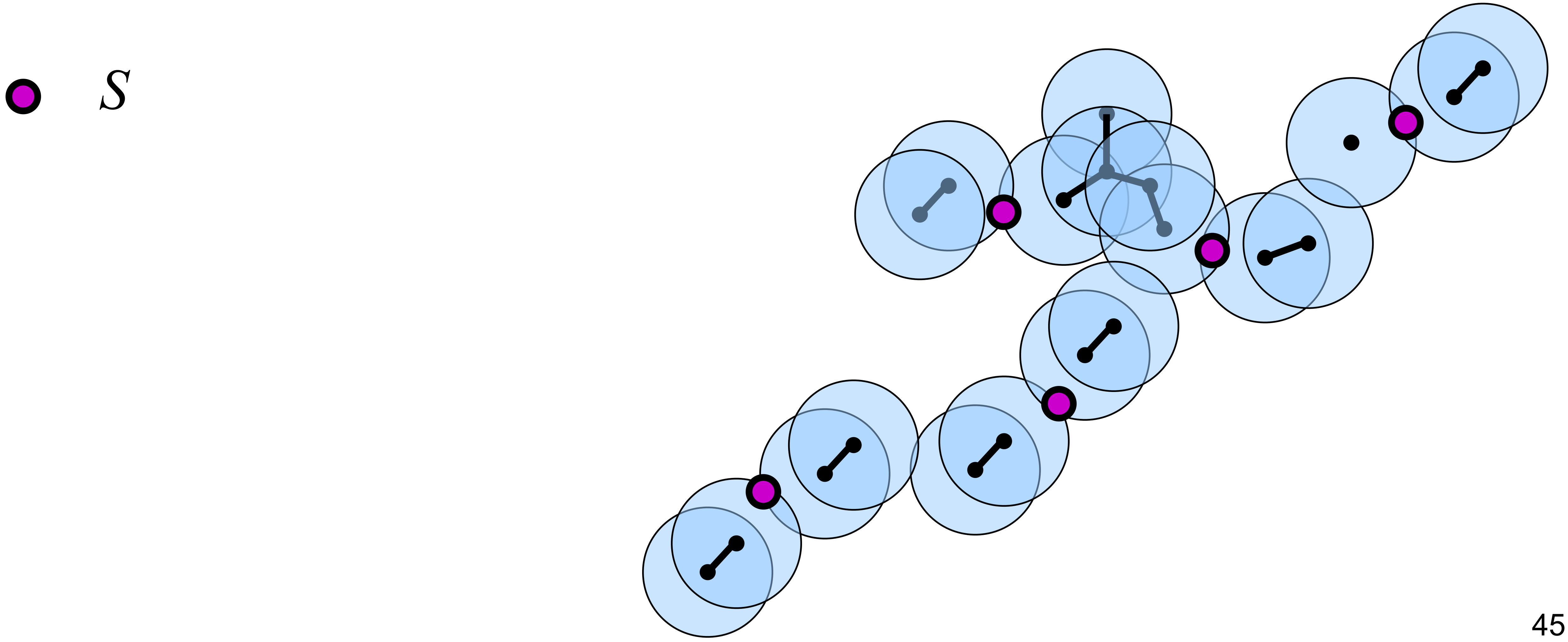
Lemma 1: To connect all sensors within a cloud C,

it suffices to take a set S that stabs all blobs,



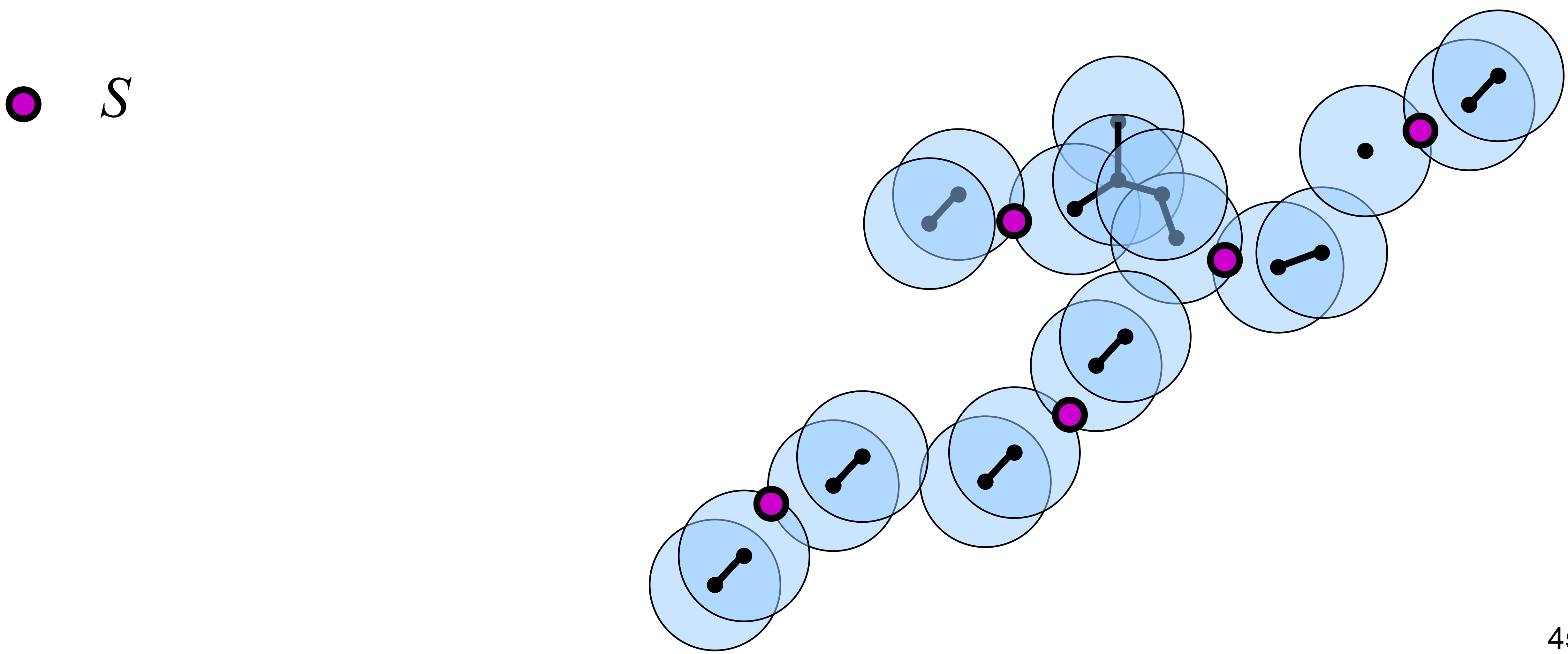
Lemma 1

Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs,



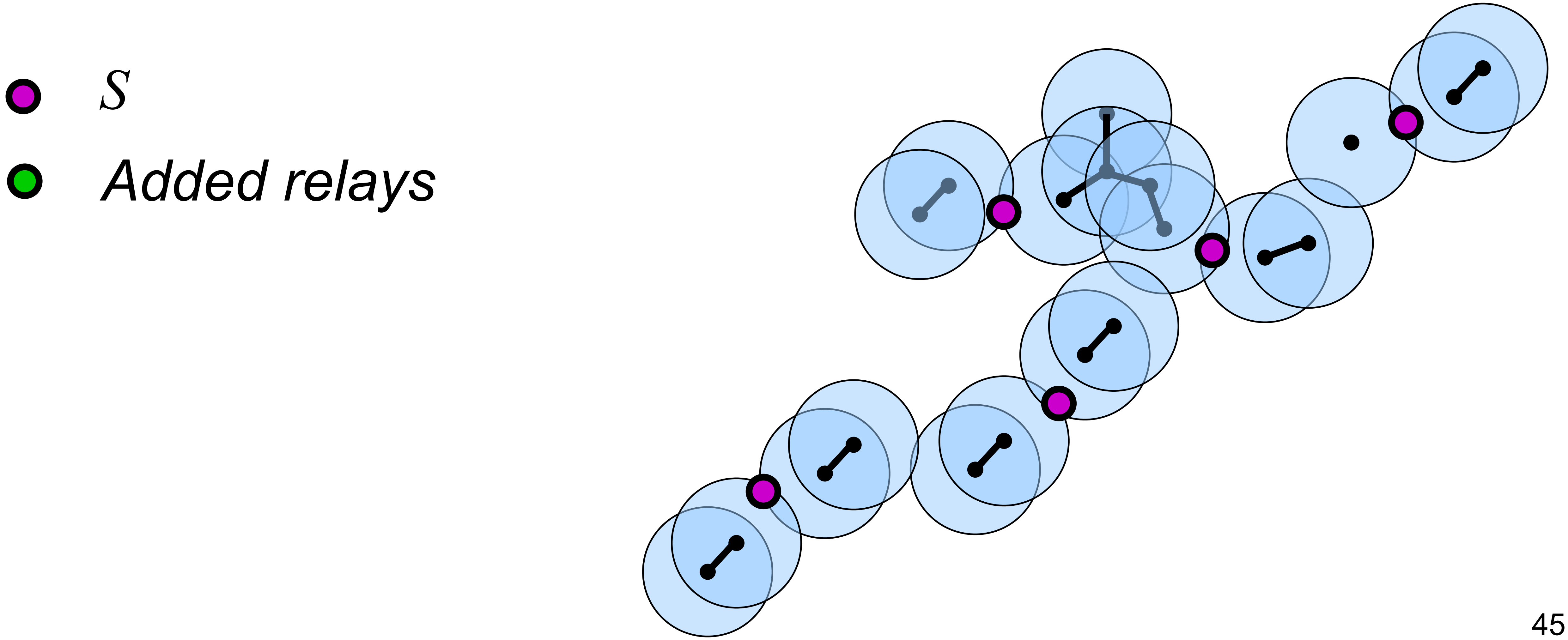
Lemma 1

Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.



Lemma 1

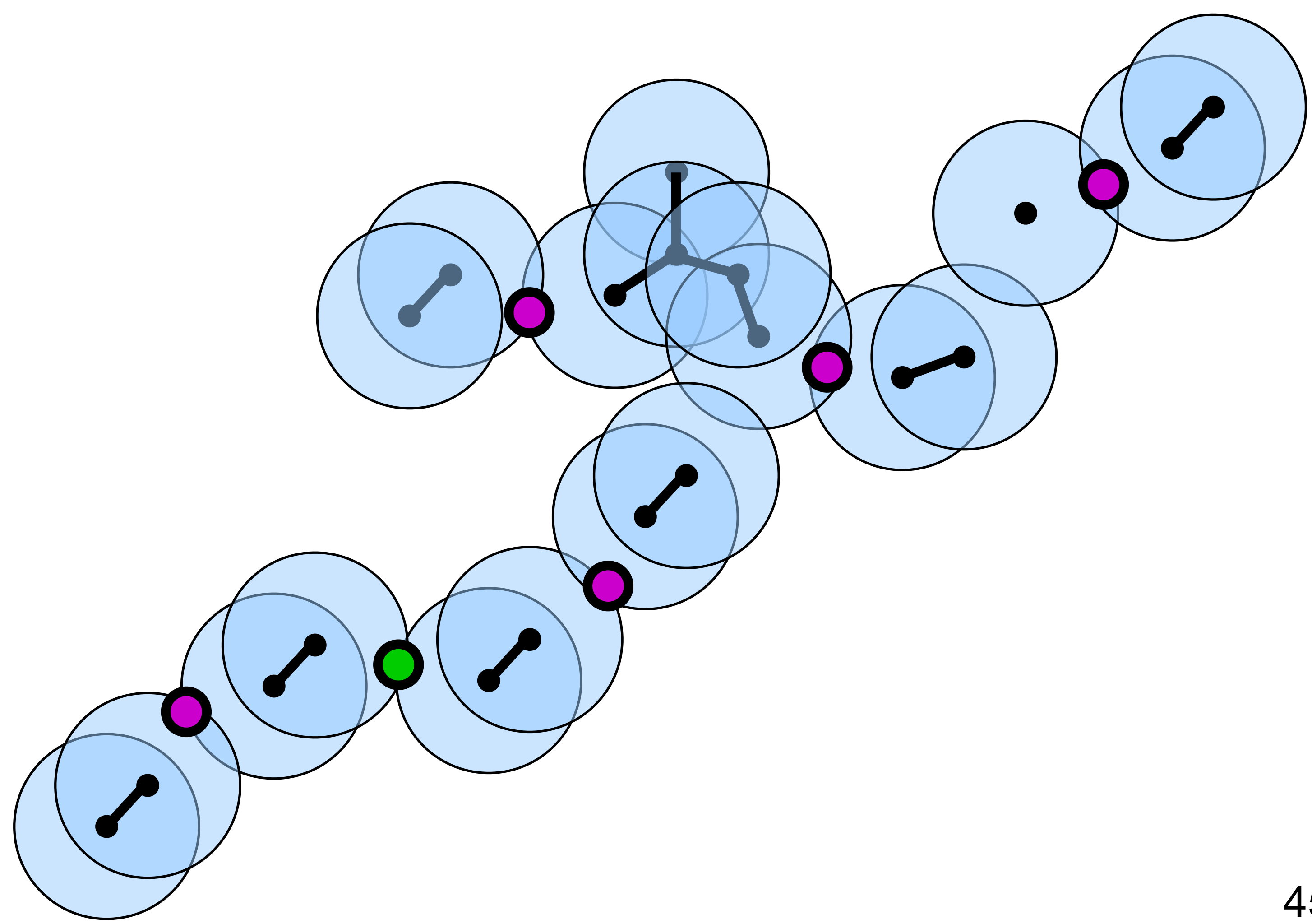
Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.



Lemma 1

Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.

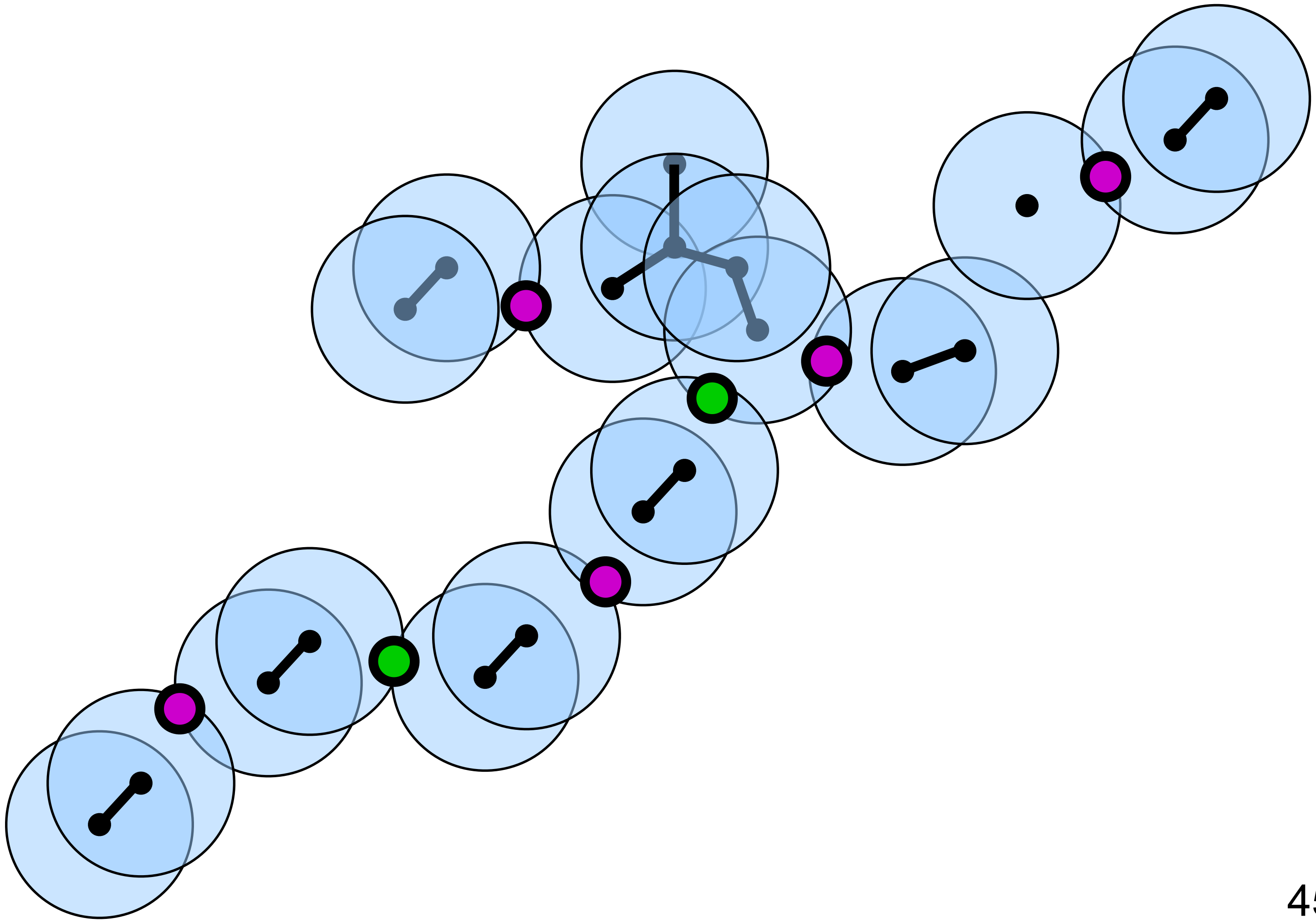
- S
- *Added relays*



Lemma 1

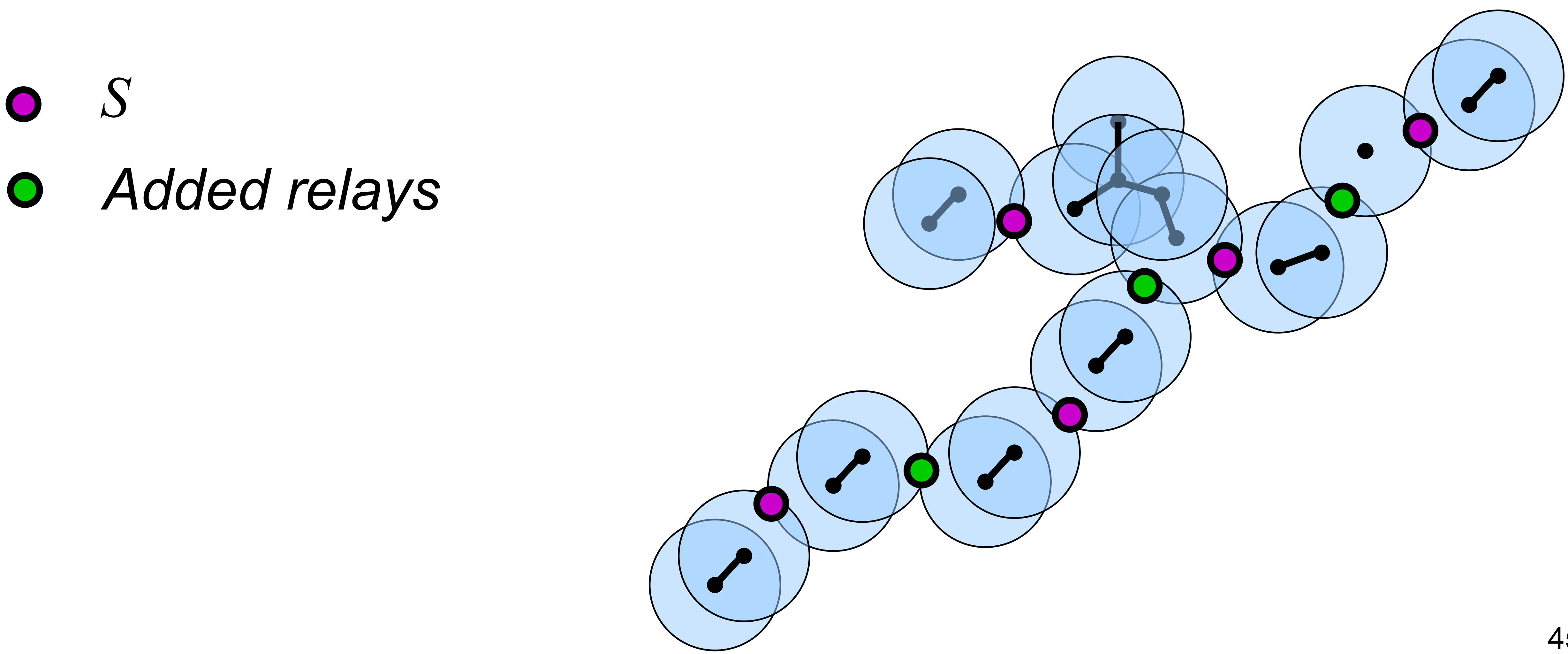
Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.

- S (magenta points)
- *Added relays* (green points)



Lemma 1

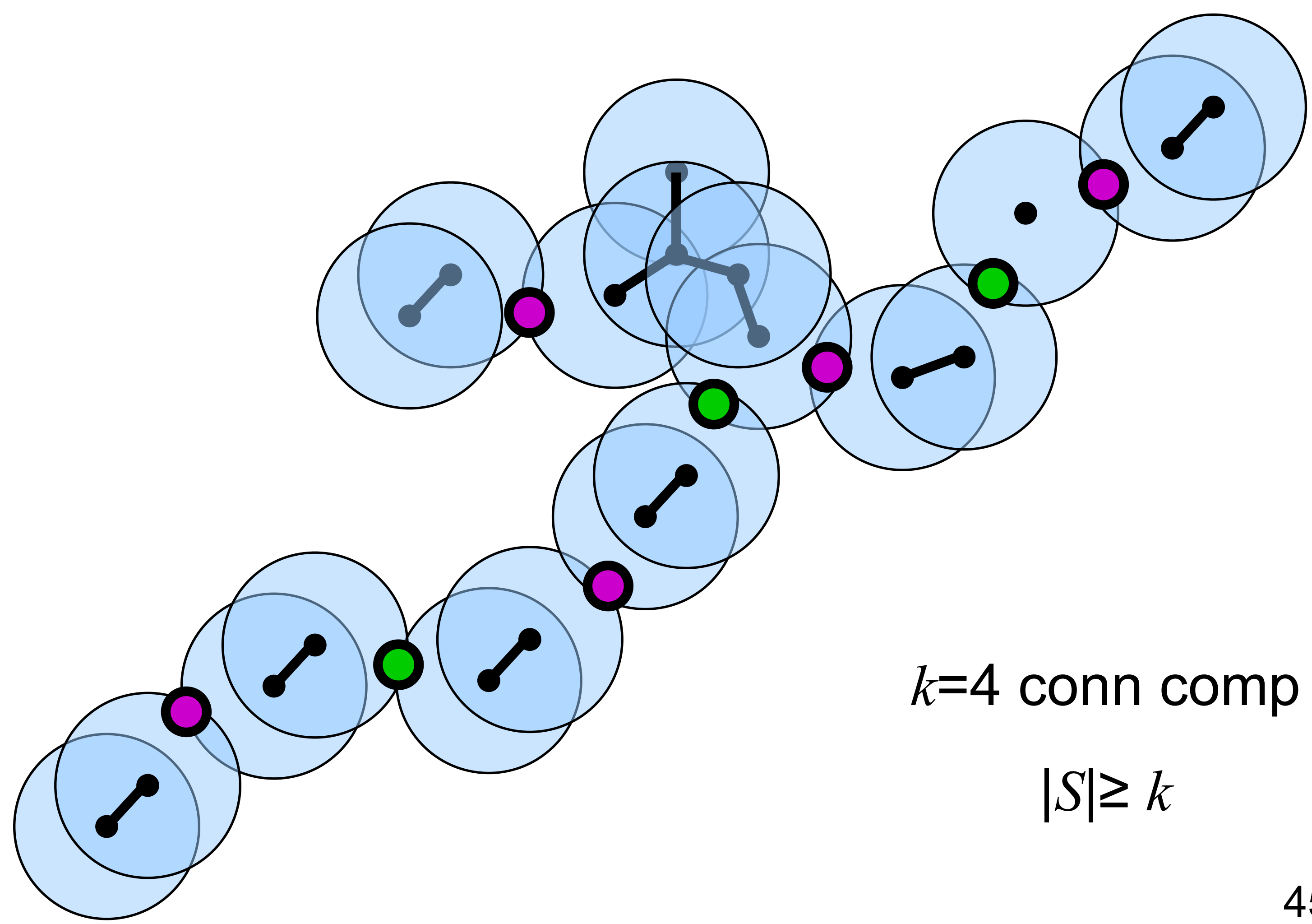
Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.



Lemma 1

Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.

- S
- *Added relays*



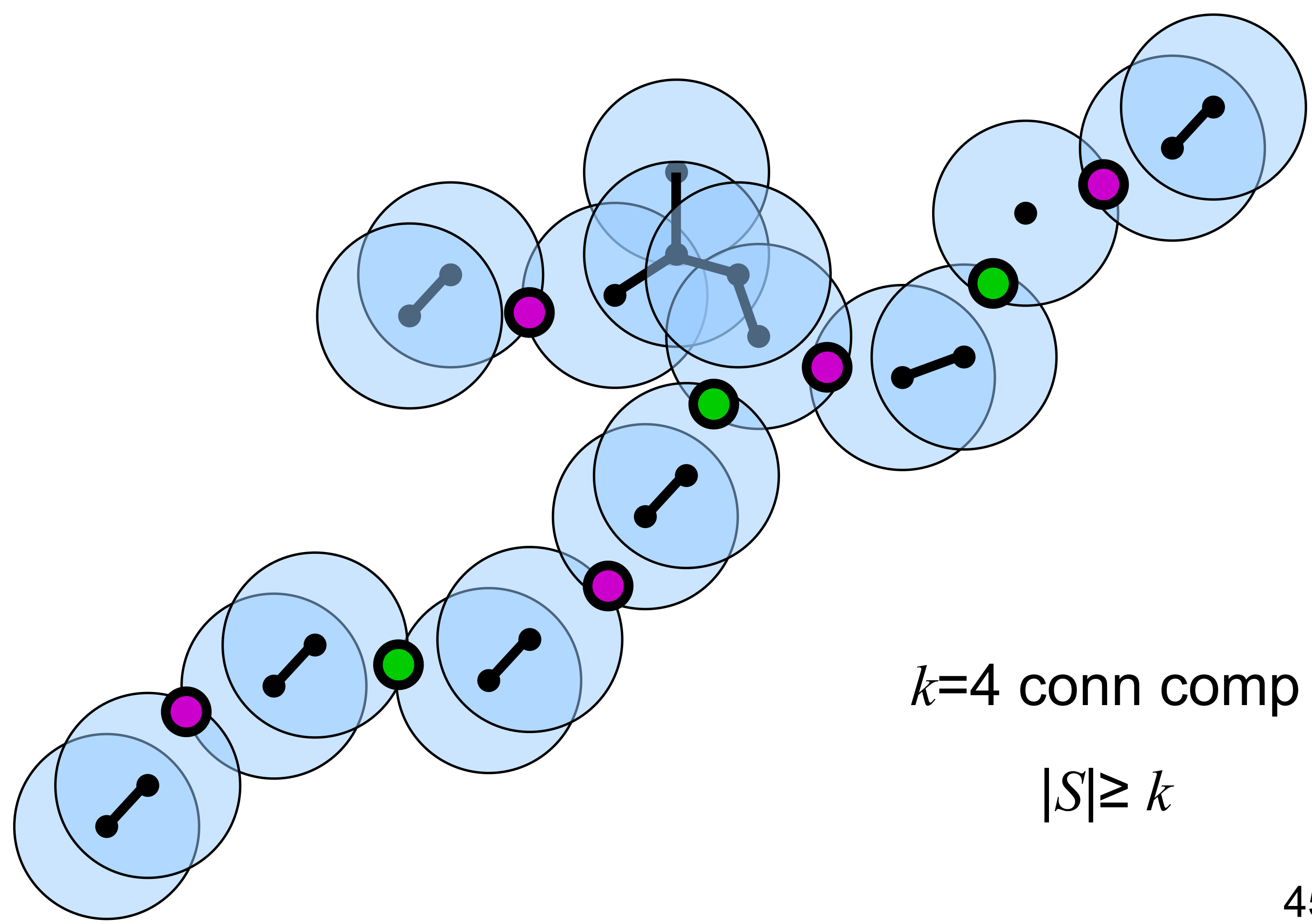
k=4 conn comp

$|S| \geq k$

Lemma 1

Lemma 1: To connect all sensors within a cloud C, it suffices to take a set S that stabs all blobs, and add at most $|S|$-1 relays.

- S
- *Added relays*



k=4 conn comp

$|S| \geq k$

Each added relay reduces the number of connected components by 1

Thus, at most $k\text{-}1 \leq |S|\text{-}1$ additional relays needed to connect the relays within C

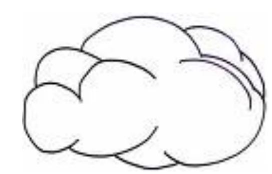
Steiner Forests and Spanning Trees with Neighborhoods

Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

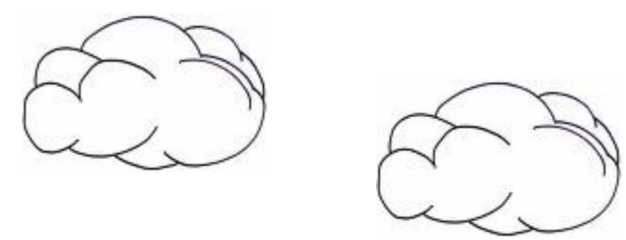
Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)



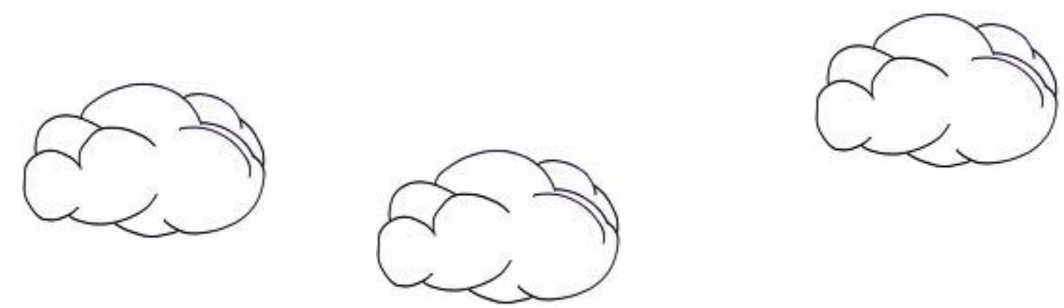
Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)



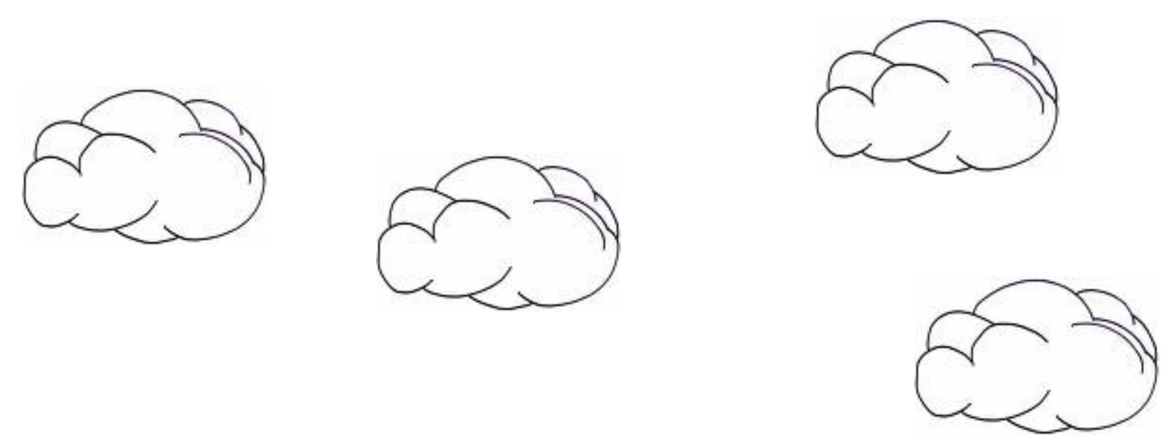
Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)



Steiner Forests and Spanning Trees with Neighborhoods

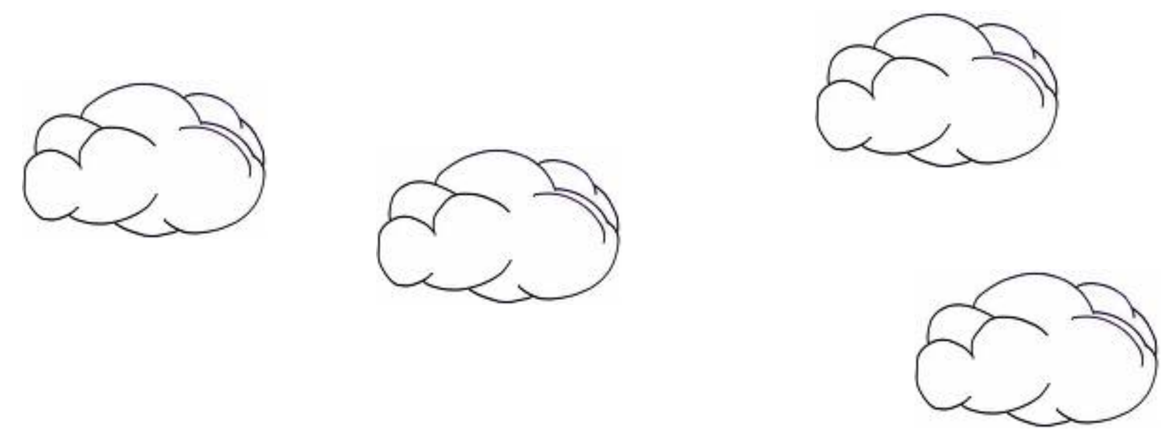
Let $\mathcal{P}$ = family of planar subsets (neighborhoods)



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

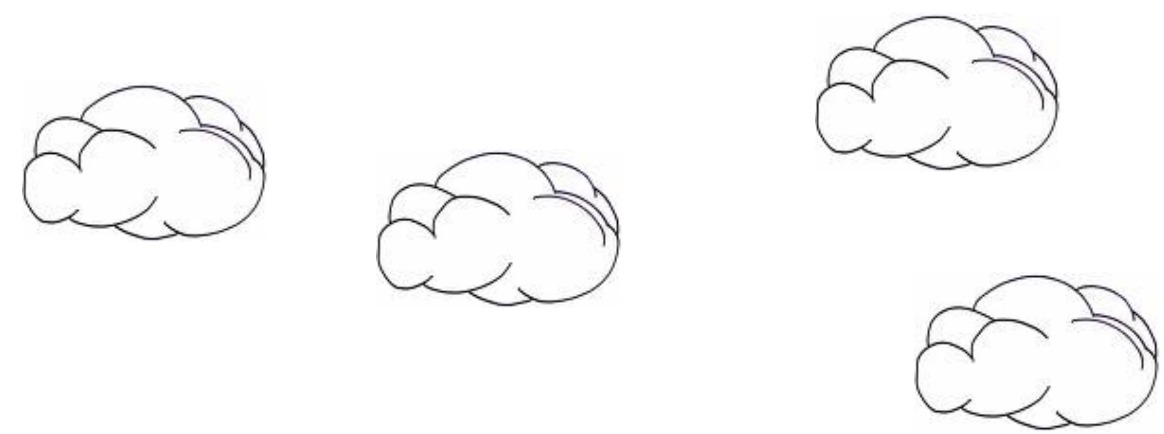


Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.



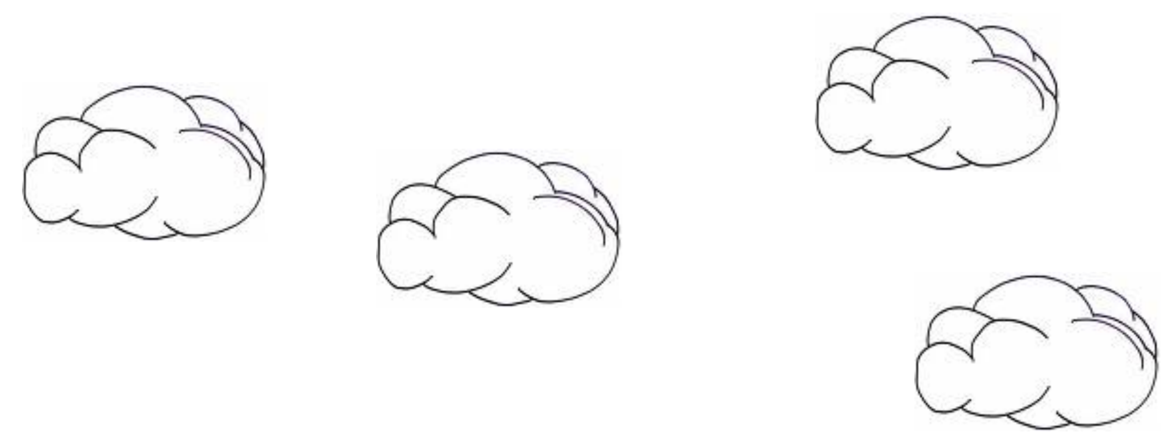
Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$\mathcal{G}_{\mathcal{P}}=(\mathcal{P},E(G))$ is connected



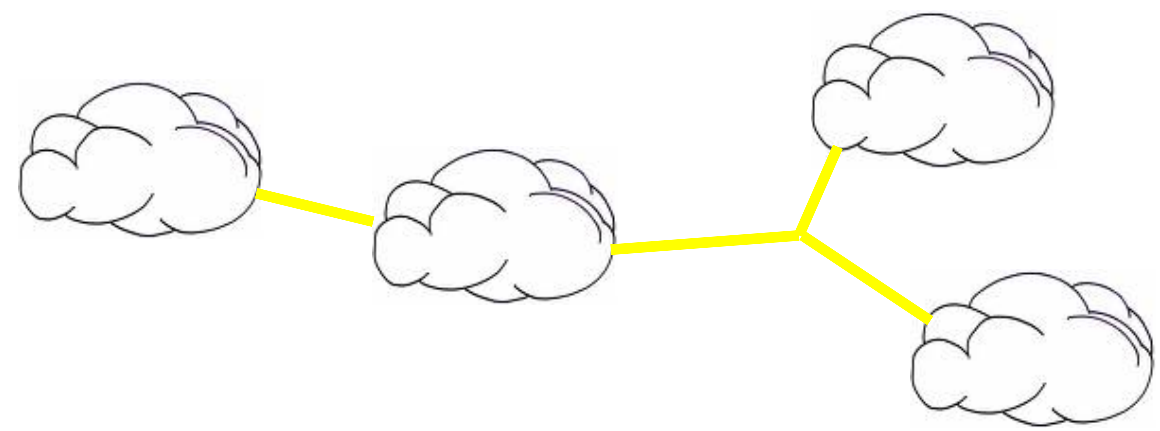
Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}$=($\mathcal{P}$,$E(G)$) is connected



Steiner Forests and Spanning Trees with Neighborhoods

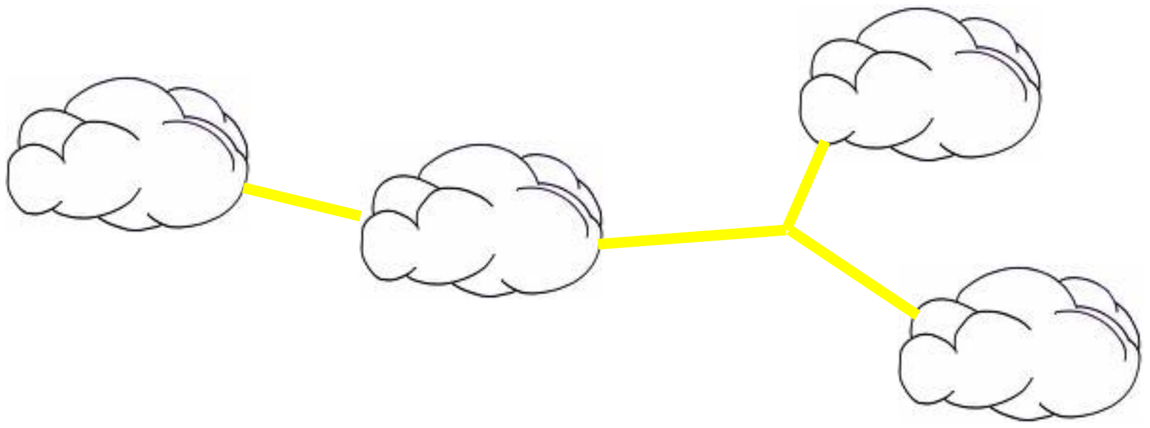
Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}=(\mathcal{P}, E(G))$ is connected

Counts only length *outside* $\mathcal{P}$



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

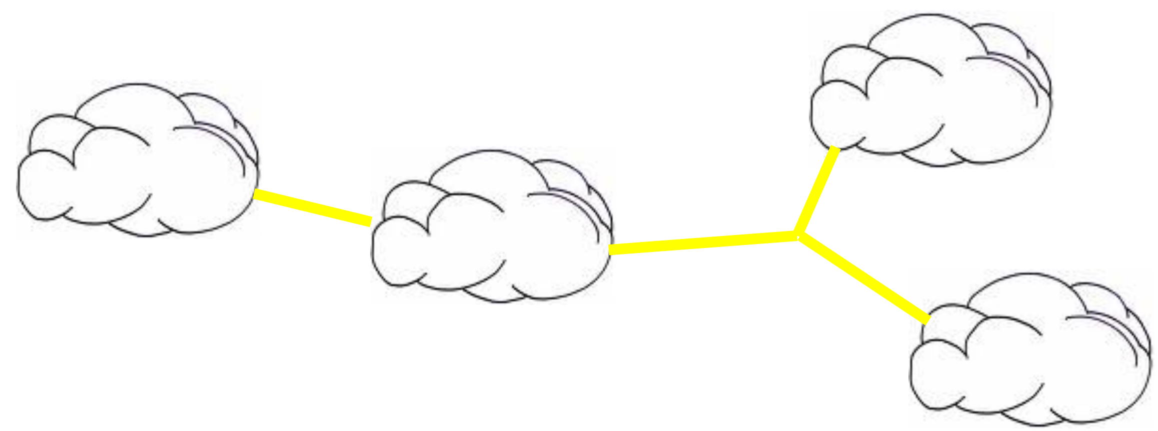
Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$\mathcal{G}_{\mathcal{P}}=(\mathcal{P},E(G))$ is connected

Counts only length *outside* $\mathcal{P}$

(vs. MStTN)



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

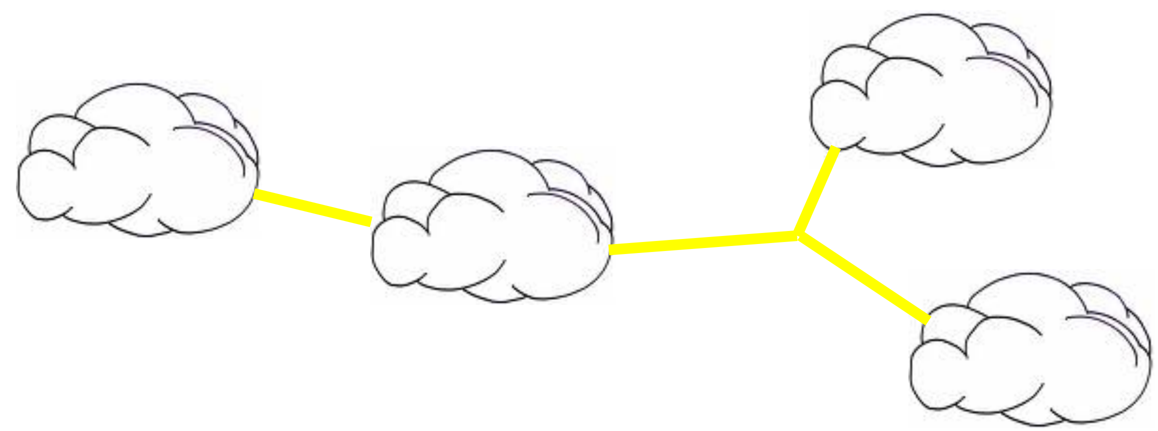
Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}$=($\mathcal{P}$,$E(G)$) is connected

Counts only length *outside* $\mathcal{P}$

(vs. MStTN)



Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$

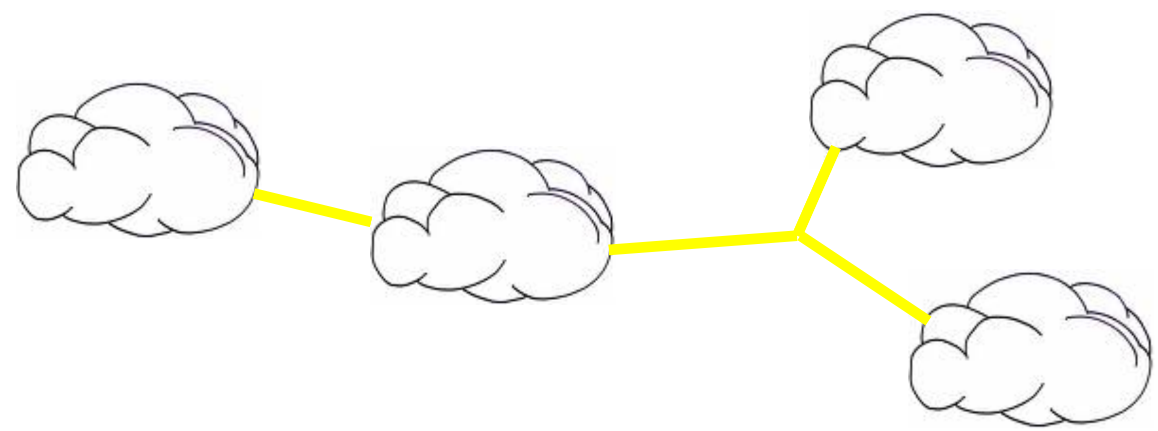
Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

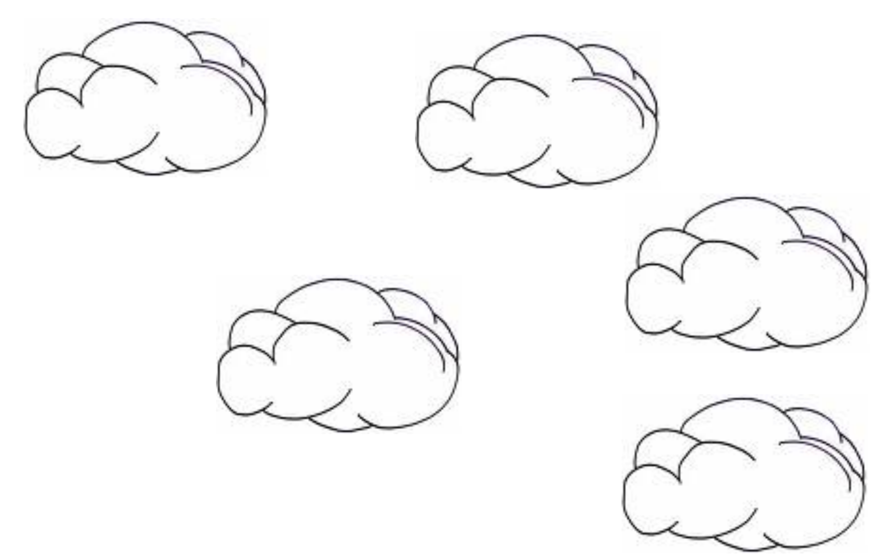
$G_{\mathcal{P}}=(\mathcal{P},E(G))$ is connected



Counts only length *outside* $\mathcal{P}$

(vs. MStTN)

Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

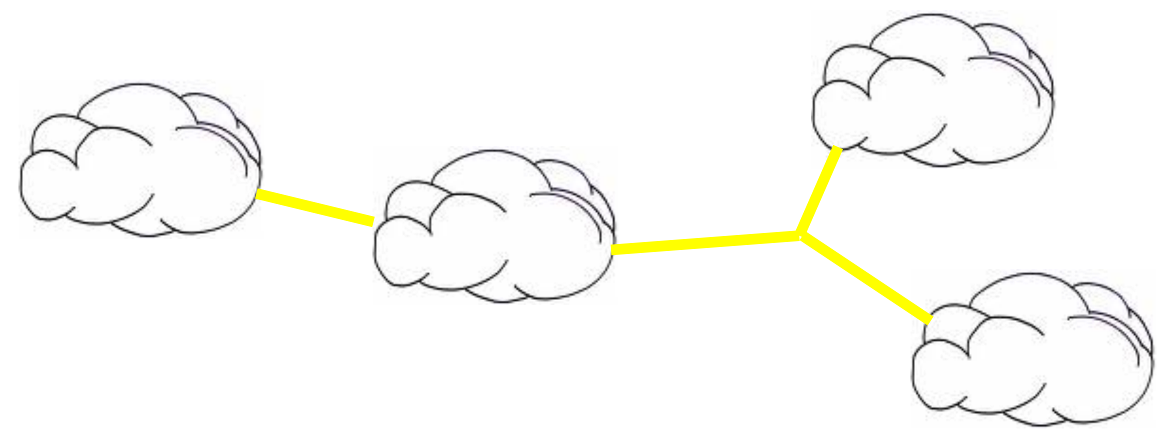
Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}$=($\mathcal{P}$,$E(G)$) is connected

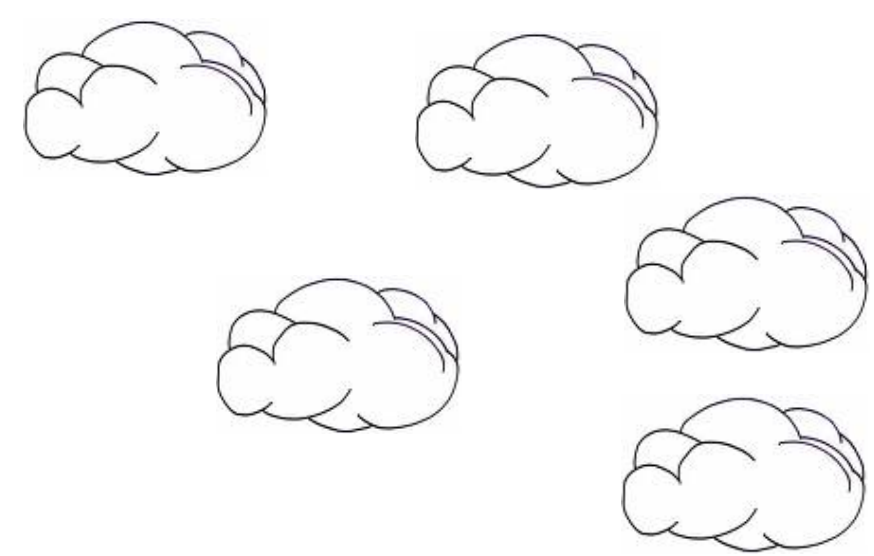
Counts only length *outside* $\mathcal{P}$

(vs. MStTN)



Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$

MSFN($\mathcal{P}$) : min spanning tree in graph on $\mathcal{P}$,



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

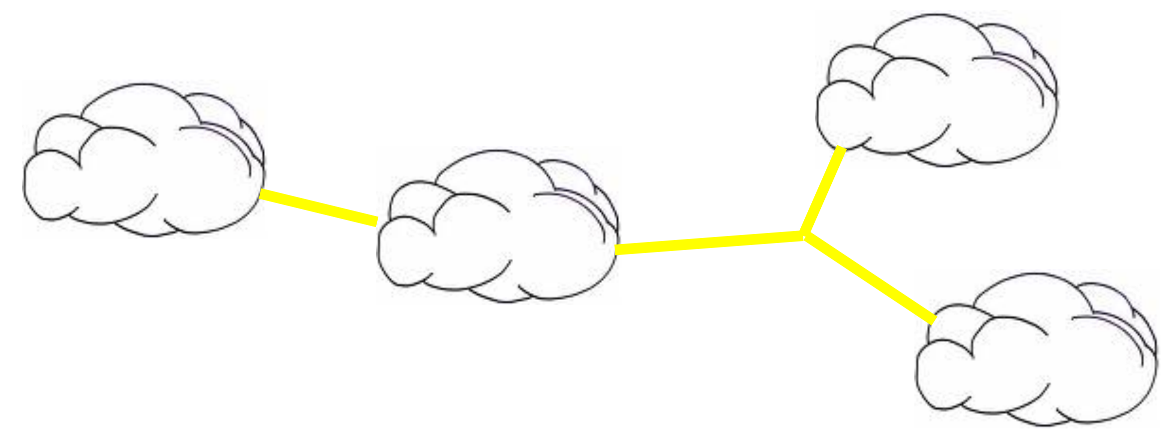
Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}=(\mathcal{P},E(G))$ is connected

Counts only length *outside* $\mathcal{P}$

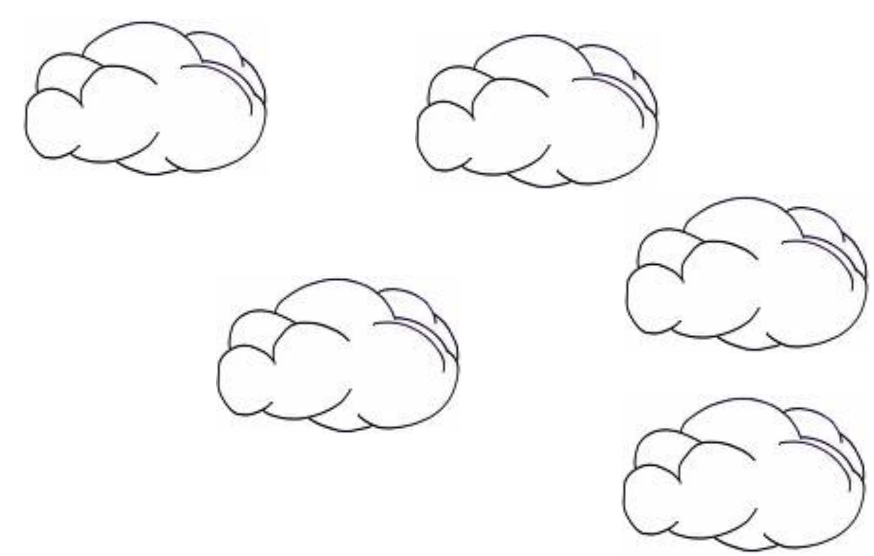
(vs. MStTN)



Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$

MSFN($\mathcal{P}$) : min spanning tree in graph on $\mathcal{P}$,

edge weights = distance between sets



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

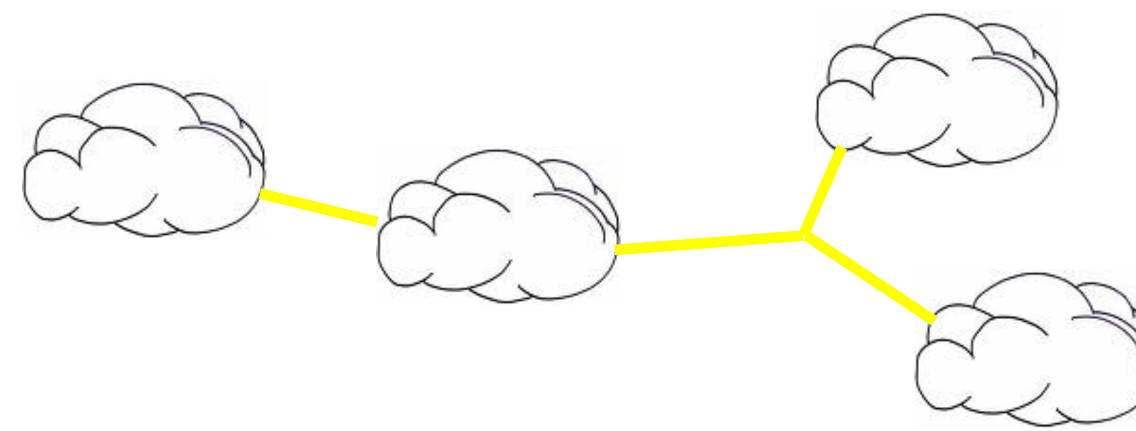
Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}=(\mathcal{P}, E(G))$ is connected

Counts only length *outside* $\mathcal{P}$

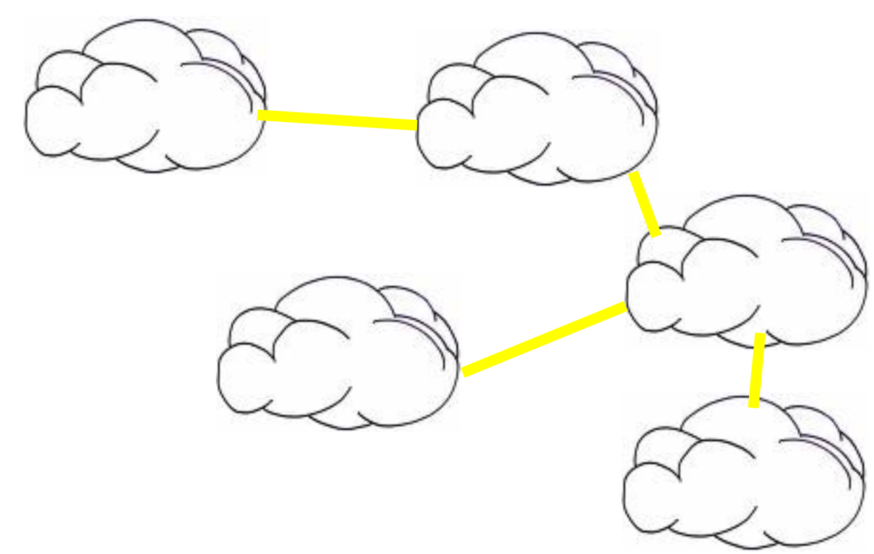
(vs. MStTN)



Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$

MSFN($\mathcal{P}$) : min spanning tree in graph on $\mathcal{P}$,

edge weights = distance between sets



Steiner Forests and Spanning Trees with Neighborhoods

Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

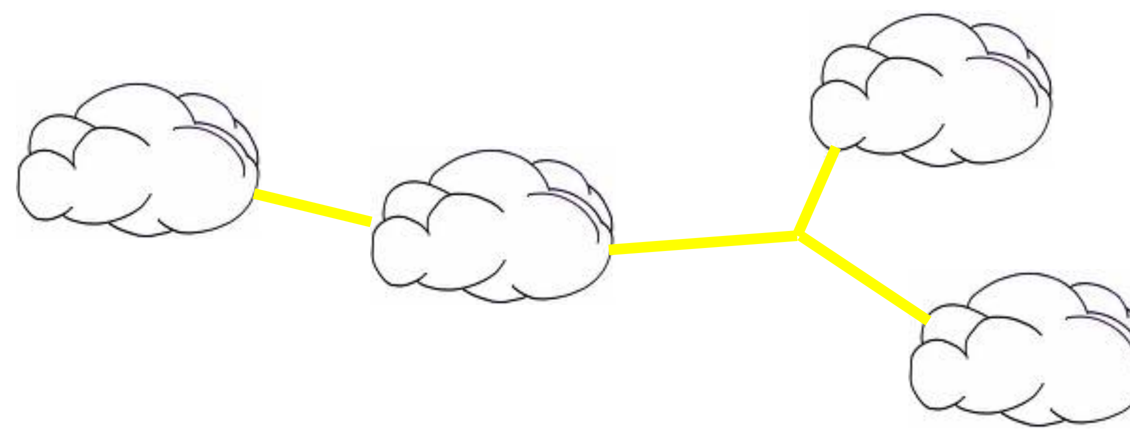
Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}=(\mathcal{P},E(G))$ is connected

Counts only length *outside* $\mathcal{P}$

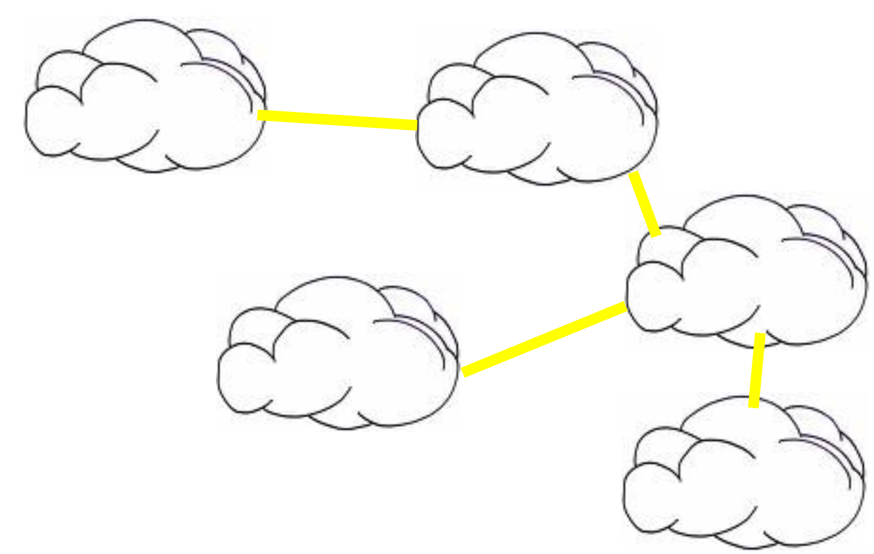
(vs. MStTN)



Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$

MSFN($\mathcal{P}$) : min spanning tree in graph on $\mathcal{P}$,

edge weights = distance between sets



Lemma 2: For any $\mathcal{P}$,

Steiner Forests and Spanning Trees with Neighborhoods

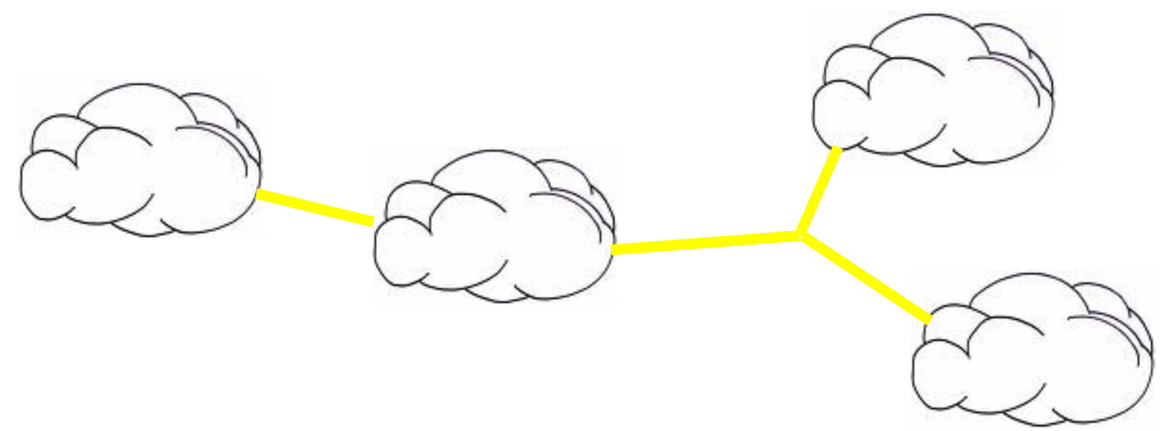
Let $\mathcal{P}$ = family of planar subsets (neighborhoods)

Minimum Steiner Forest for the Neighborhoods in P

MStFN($\mathcal{P}$) : min-length plane graph G s.t.

$G_{\mathcal{P}}=(\mathcal{P},E(G))$ is connected

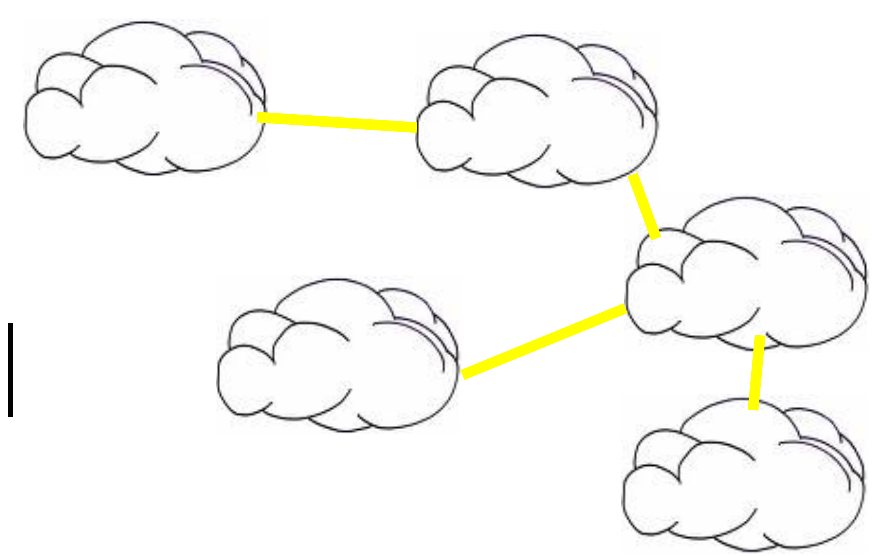
Counts only length *outside* $\mathcal{P}$ (vs. MStTN)



Minimum Spanning Forest for the Neighborhoods in $\mathcal{P}$

MSFN($\mathcal{P}$) : min spanning tree in graph on $\mathcal{P}$,

edge weights = distance between sets



Lemma 2: For any $\mathcal{P}$,

$$|\mathrm{MSFN}(\mathcal{P})| \leq \left(\frac{2}{\sqrt{3}}\right) |\mathrm{MStFN}(\mathcal{P})|$$

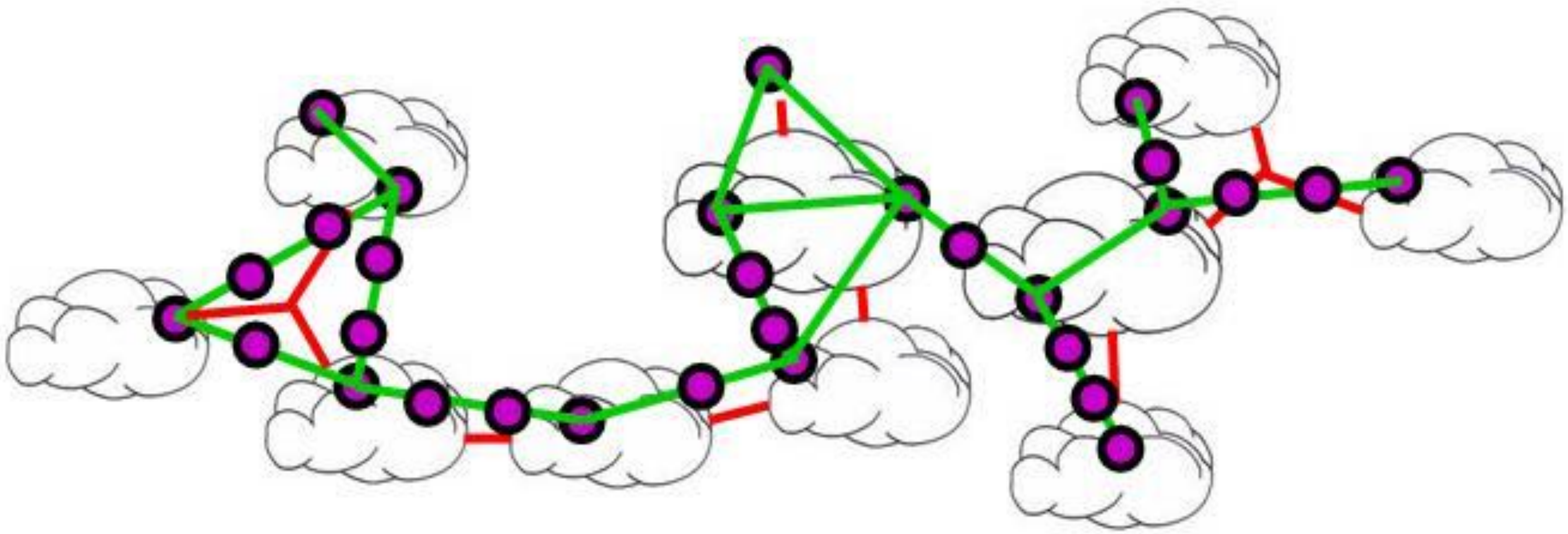
Lower Bounds on OPT

Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$

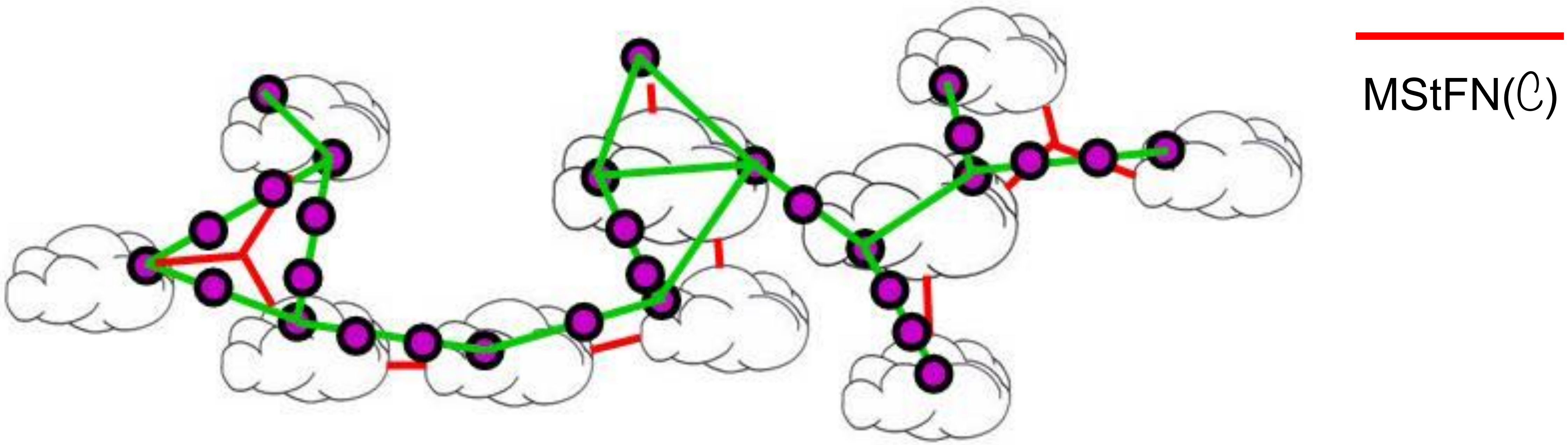
Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$



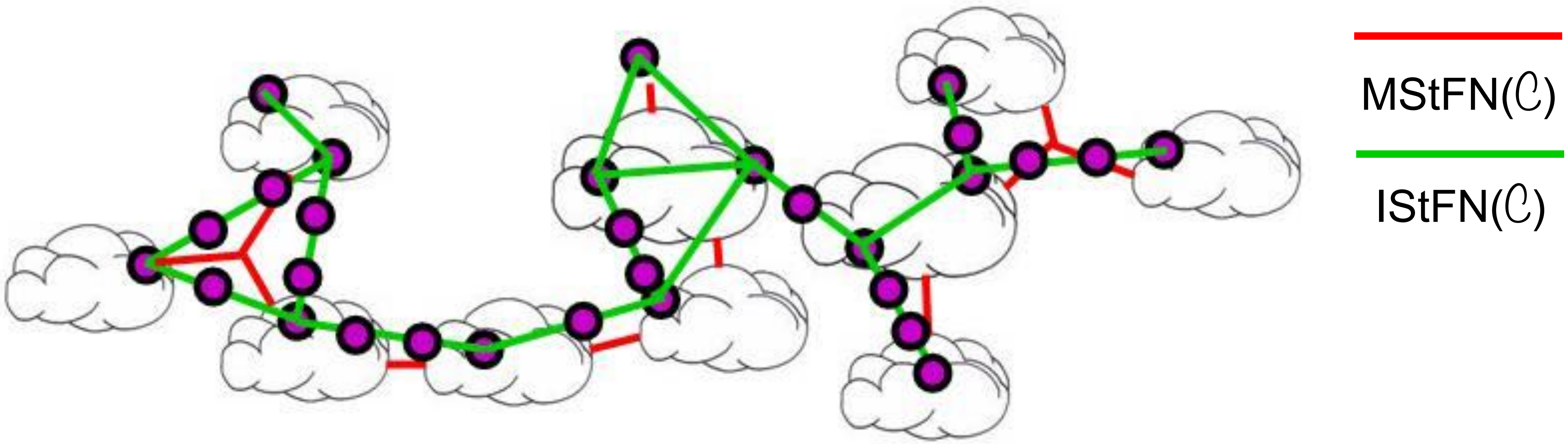
Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$



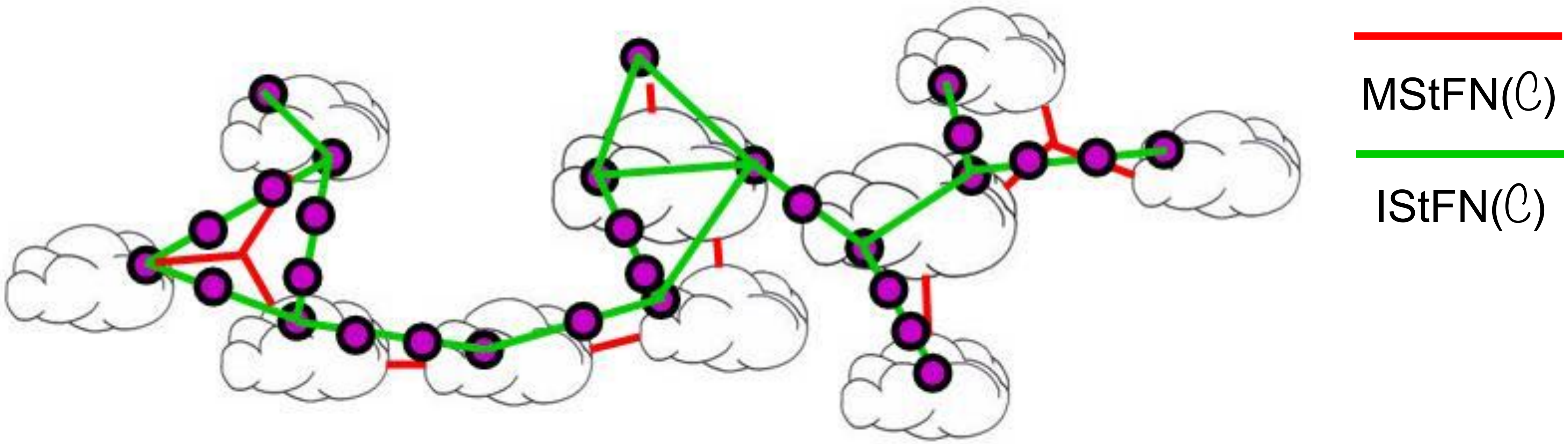
Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$



Lower Bounds on OPT

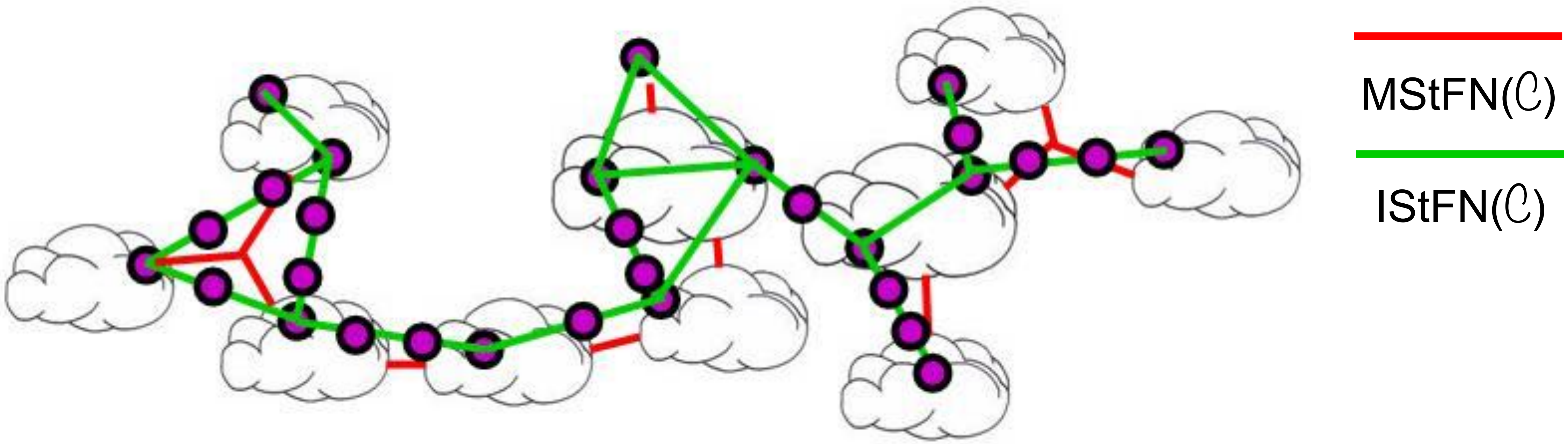
(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$



Optmal relays induce a Steiner Forest.

Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$

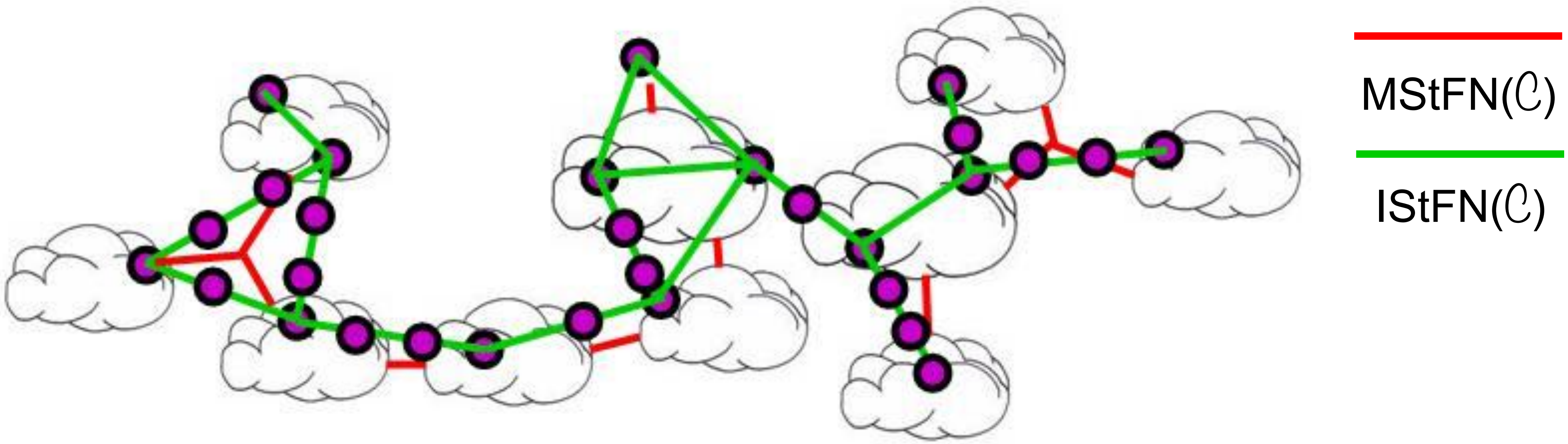


Optmal relays induce a Steiner Forest.

(2) $|R^*| \geq |\text{Stab}(\mathcal{B})|$

Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$

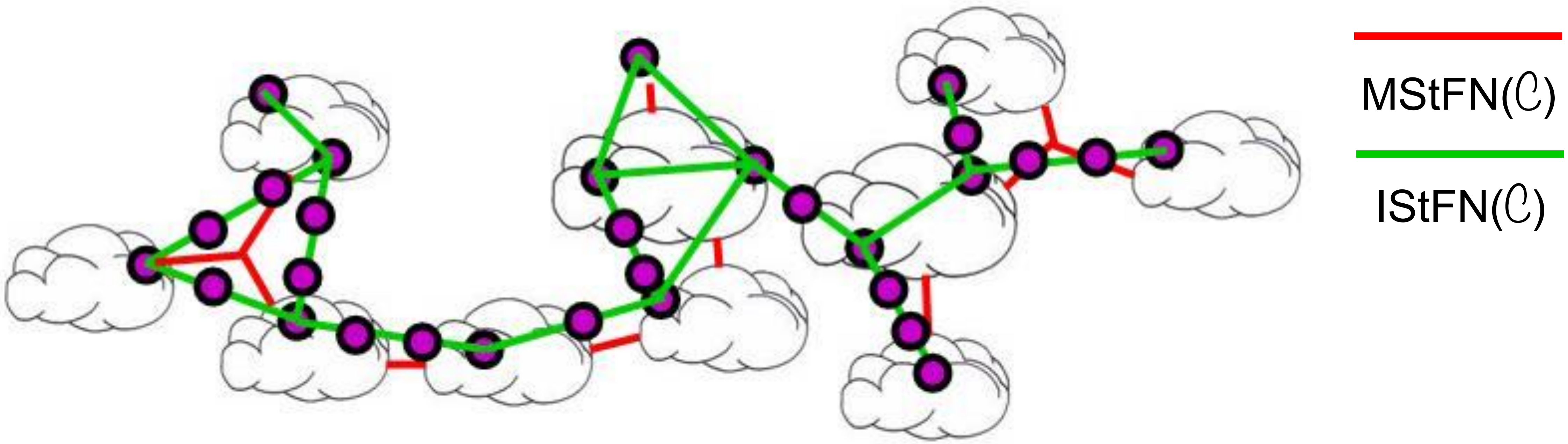


Optmal relays induce a Steiner Forest.

(2) $|R^*| \geq |\text{Stab}(\mathcal{B})|$ There must be a relay in every blob.

Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$



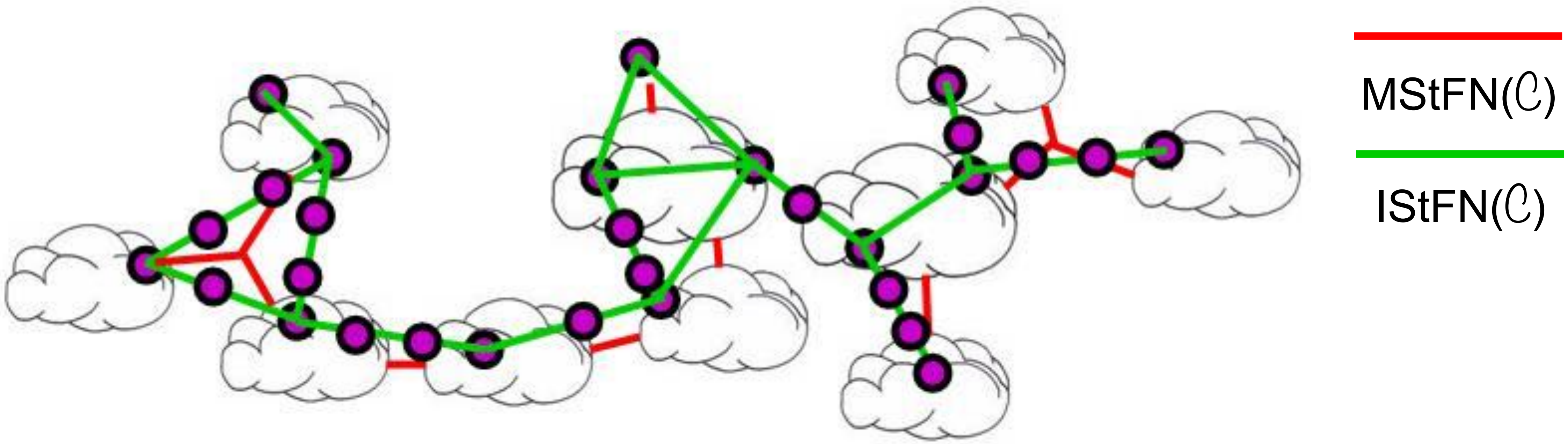
Optmal relays induce a Steiner Forest.

(2) $|R^*| \geq |\text{Stab}(\mathcal{B})|$ There must be a relay in every blob.

(3) $|R^*| \geq |\mathcal{C}|$

Lower Bounds on OPT

(1) $|R^*| \geq |\text{MStFN}(\mathcal{C})|/r$



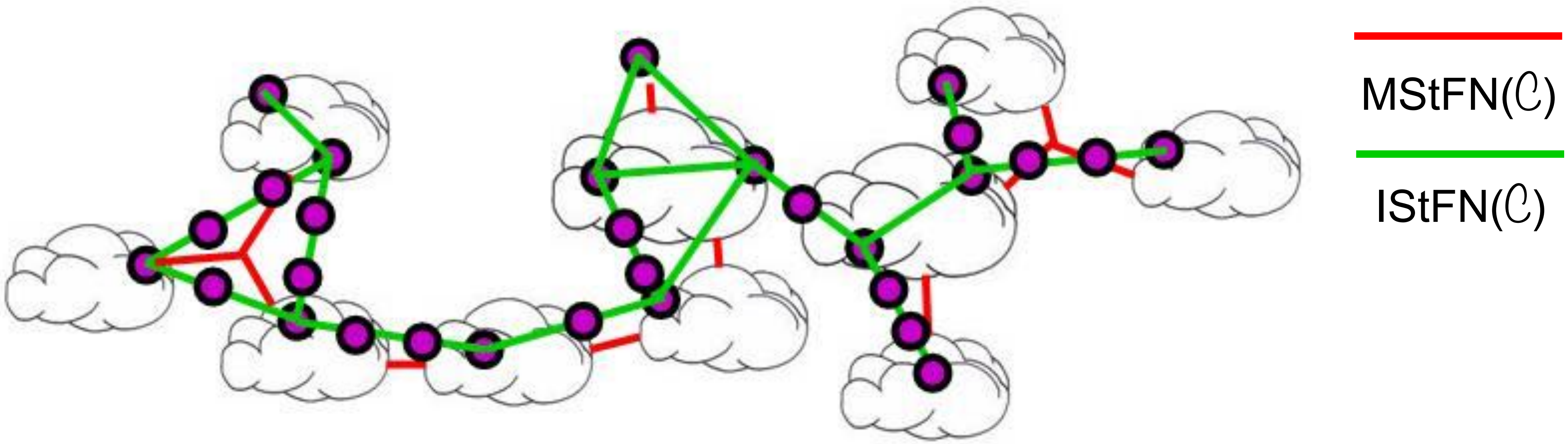
Optmal relays induce a Steiner Forest.

(2) $|R^*| \geq |\text{Stab}(\mathcal{B})|$ There must be a relay in every blob.

(3) $|R^*| \geq |\mathcal{C}|$ There must be a relay in every cloud, and clouds are disjoint.

Lower Bounds on OPT

(1) $|R^*| \geq |\mathrm{MStFN}(\mathcal{C})|/r$



Optmal relays induce a Steiner Forest.

(2) $|R^*| \geq |\mathrm{Stab}(\mathcal{B})|$ There must be a relay in every blob.

(3) $|R^*| \geq |\mathcal{C}|$ There must be a relay in every cloud, and clouds are disjoint.

Simple 6.73-Approximation (I)

Simple 6.73-Approximation (I)

- For each cloud C:

Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$

Simple 6.73-Approximation (I)

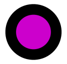
- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover

Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs

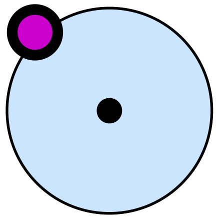
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs



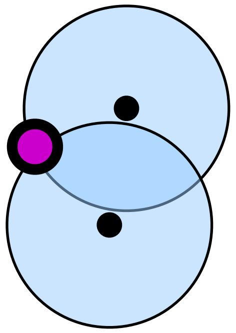
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs



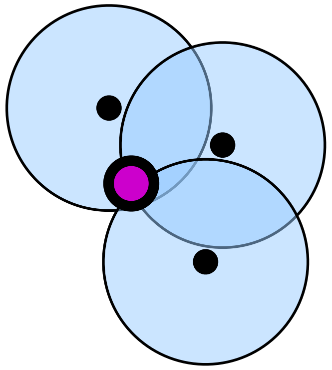
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs



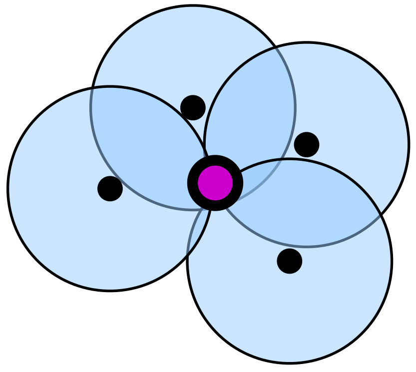
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs



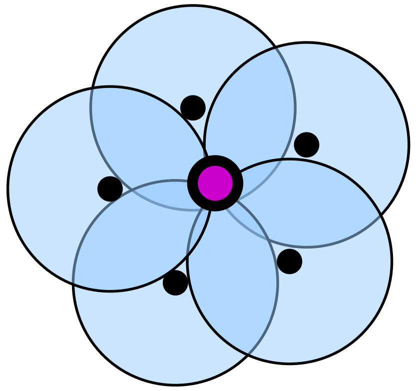
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs



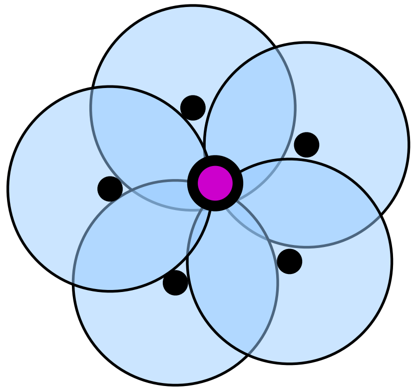
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs



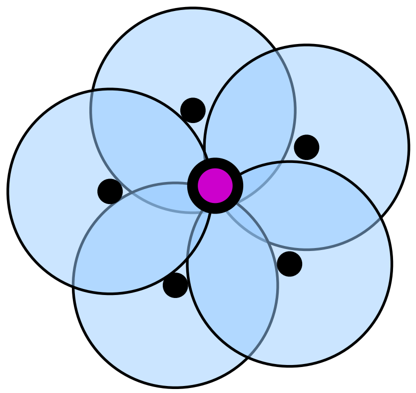
Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover
    - A stab pierces ≤ 5 blobs
    - Greedy gives approx factor:



Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover

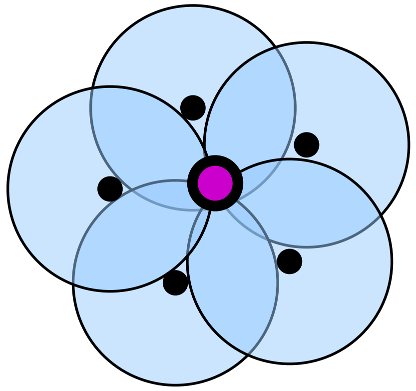


- A stab pierces ≤ 5 blobs
- Greedy gives approx factor:

$$|R_C| \le \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) |\text{Stab}(\mathcal{B})| = \frac{137}{60} |\text{Stab}(\mathcal{B})|$$

Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover

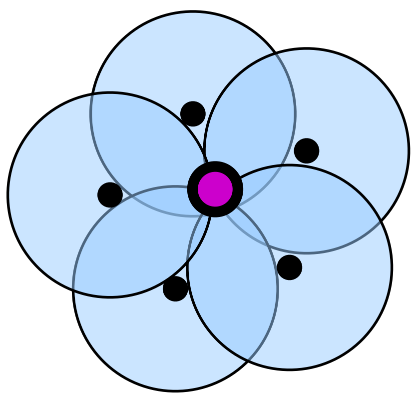


- A stab pierces ≤ 5 blobs
- Greedy gives approx factor:

$$|R_C| \leq \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) |\text{Stab}(\mathcal{B})| = \frac{137}{60} |\text{Stab}(\mathcal{B})|$$

Simple 6.73-Approximation (I)

- For each cloud C:
  - Find relays R_C to hit (stab) all blobs $B \in \mathcal{B}_C$ using Greedy Set Cover



- A stab pierces ≤ 5 blobs
- Greedy gives approx factor:

$$|R_C| \leq \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) |\mathrm{Stab}(\mathcal{B})| = \frac{137}{60} |\mathrm{Stab}(\mathcal{B})|$$

Simple 6.73-Approximation (II)

Simple 6.73-Approximation (II)

- (For each cloud C:)

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

$$|R'_C| \le 2|R_C| - 1 \le \frac{137}{30}|\text{Stab}(B_C)| - 1$$

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

$$|R'_C| \leq 2|R_C| - 1 \leq \frac{137}{30}|\mathrm{Stab}(B_C)| - 1$$

Total for all C:

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

$$|R'_C| \le 2|R_C| - 1 \le \frac{137}{30}|\mathrm{Stab}(\mathcal{B}_C)| - 1$$

Total for all C:

$$|R'| = \sum_{C \in \mathcal{C}} |R'_C| \le \sum_{C \in \mathcal{C}} \left(\frac{137}{30}|\mathrm{Stab}(\mathcal{B}_C)| - 1 \right) = \frac{137}{30}|\mathrm{Stab}(\mathcal{B})| - |\mathcal{C}|$$

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

$$|R'_C| \leq 2|R_C| - 1 \leq \frac{137}{30}|\mathrm{Stab}(\mathcal{B}_C)| - 1$$

Total for all C:

$$|R'| = \sum_{C \in \mathcal{C}} |R'_C| \leq \sum_{C \in \mathcal{C}} \left(\frac{137}{30}|\mathrm{Stab}(\mathcal{B}_C)| - 1 \right) = \frac{137}{30}|\mathrm{Stab}(\mathcal{B})| - |\mathcal{C}|$$

(since blobs $\mathcal{B}_C$ for different C are disjoint)

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

$$|R'_C| \le 2|R_C| - 1 \le \frac{137}{30}|\text{Stab}(\mathcal{B}_C)| - 1$$

Total for all C:

$$|R'| = \sum_{C \in \mathcal{C}} |R'_C| \le \sum_{C \in \mathcal{C}} \left(\frac{137}{30}|\text{Stab}(\mathcal{B}_C)| - 1 \right) = \frac{137}{30}|\text{Stab}(\mathcal{B})| - |\mathcal{C}|$$

(since blobs $\mathcal{B}_C$ for different C are disjoint)

Simple 6.73-Approximation (II)

- (For each cloud C:)
  - Apply **Lemma 1**: Obtain relays R'_C connecting cloud C.

$$|R'_C| \leq 2|R_C| - 1 \leq \frac{137}{30}|\mathrm{Stab}(\mathcal{B}_C)| - 1$$

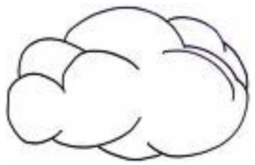
Total for all C:

$$|R'| = \sum_{C\in\mathcal{C}} |R'_C| \leq \sum_{C\in\mathcal{C}} \left(\frac{137}{30}|\mathrm{Stab}(\mathcal{B}_C)| - 1 \right) = \frac{137}{30}|\mathrm{Stab}(\mathcal{B})| - |\mathcal{C}|$$

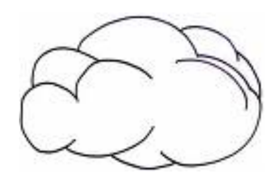
(since blobs $\mathcal{B}_C$ for different C are disjoint)

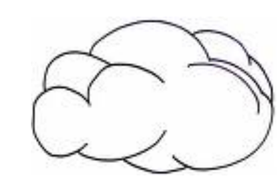
Simple 6.73-Approximation (III)

Simple 6.73-Approximation (III)



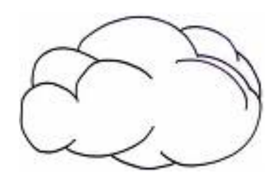
Simple 6.73-Approximation (III)

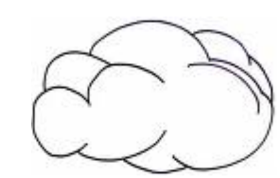




Simple 6.73-Approximation (III)

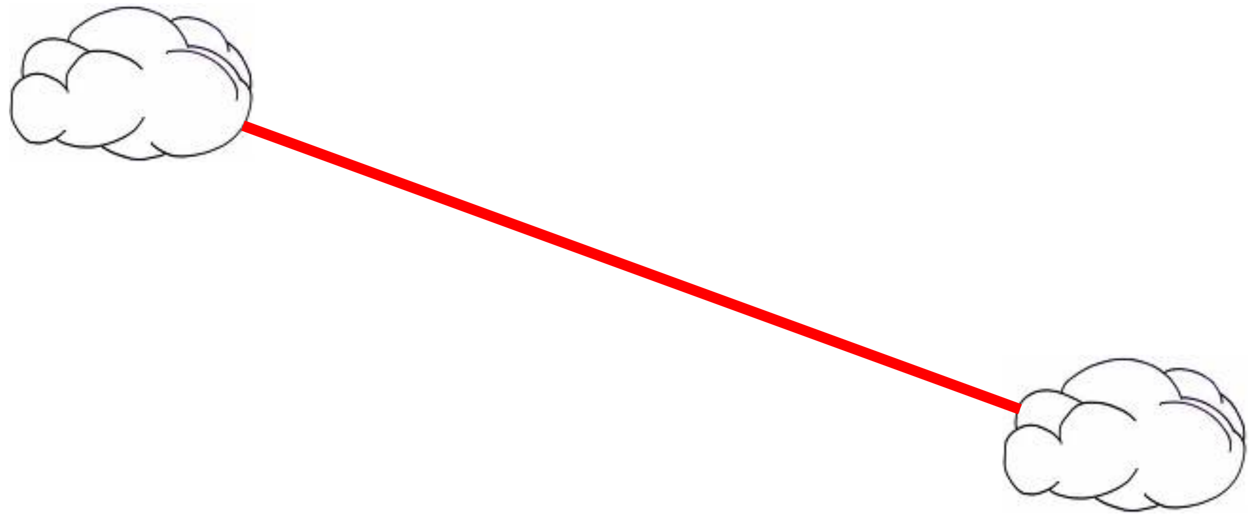
- Compute MSFN($\mathcal{C}$).





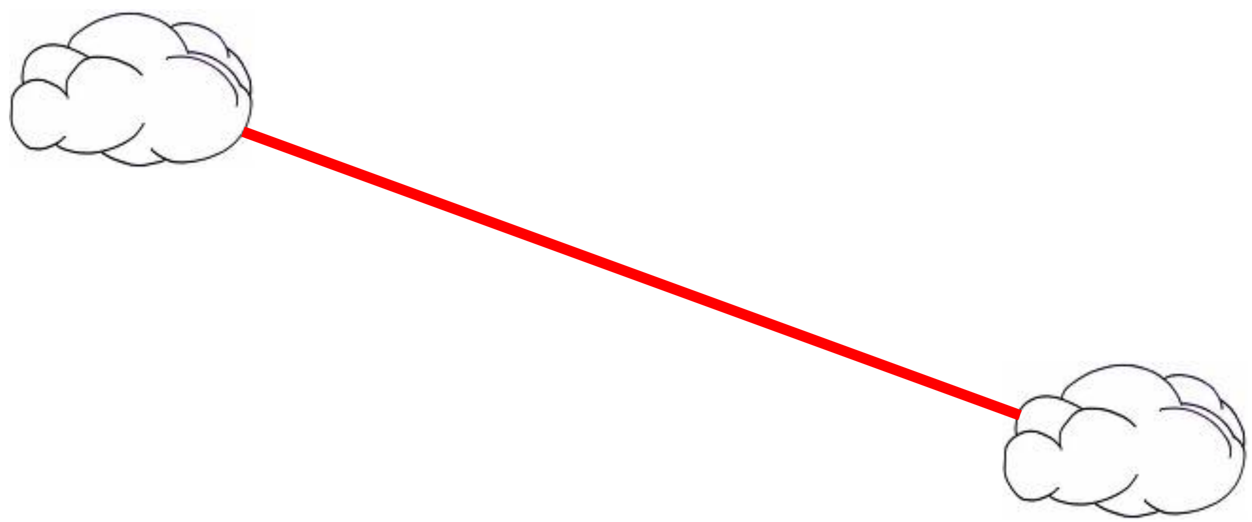
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).



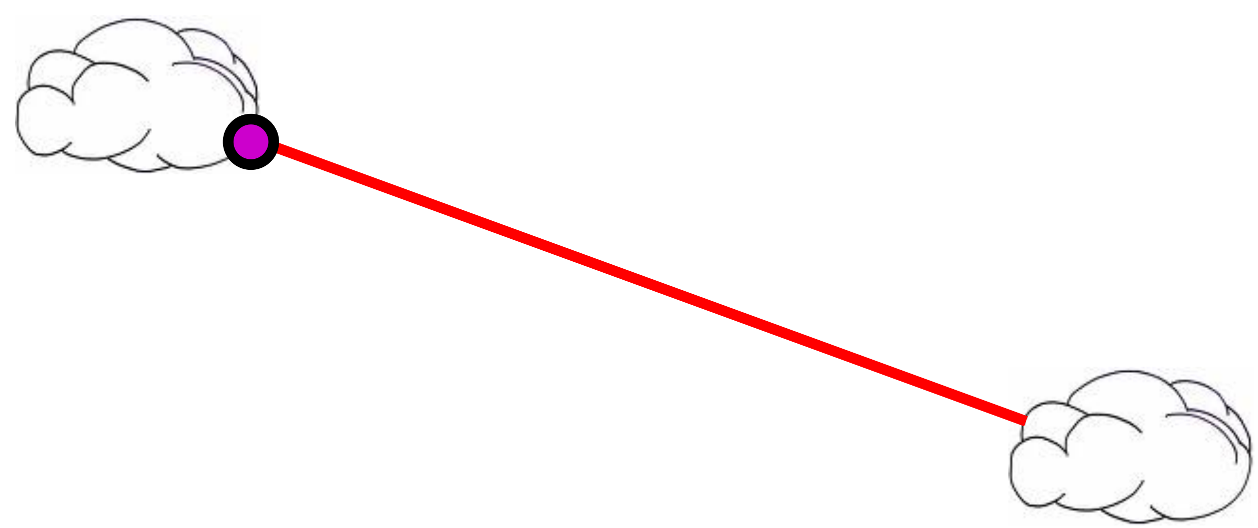
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - along each edge e



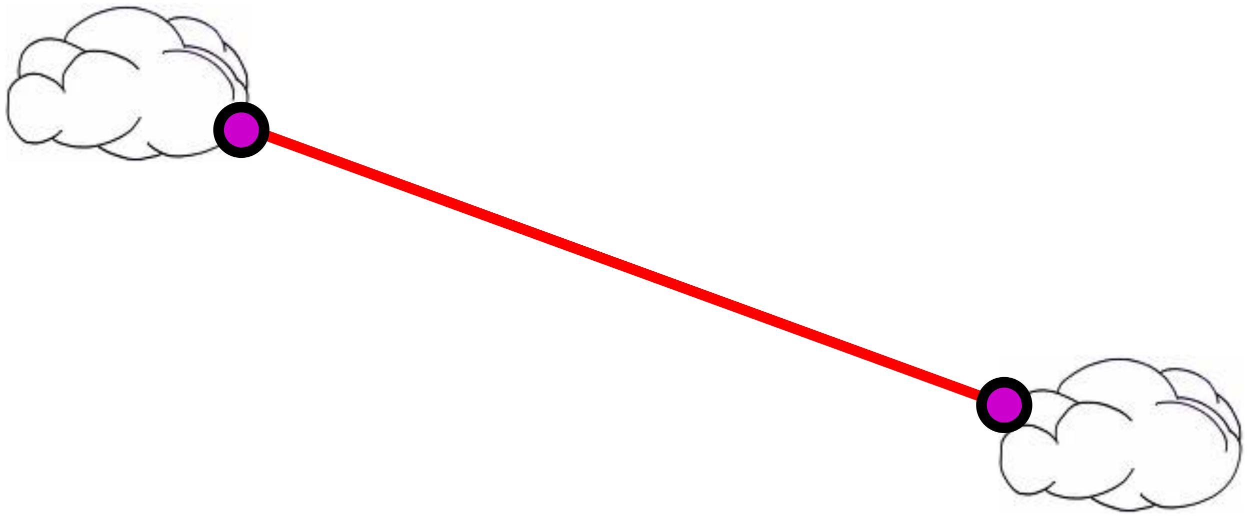
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - along each edge e



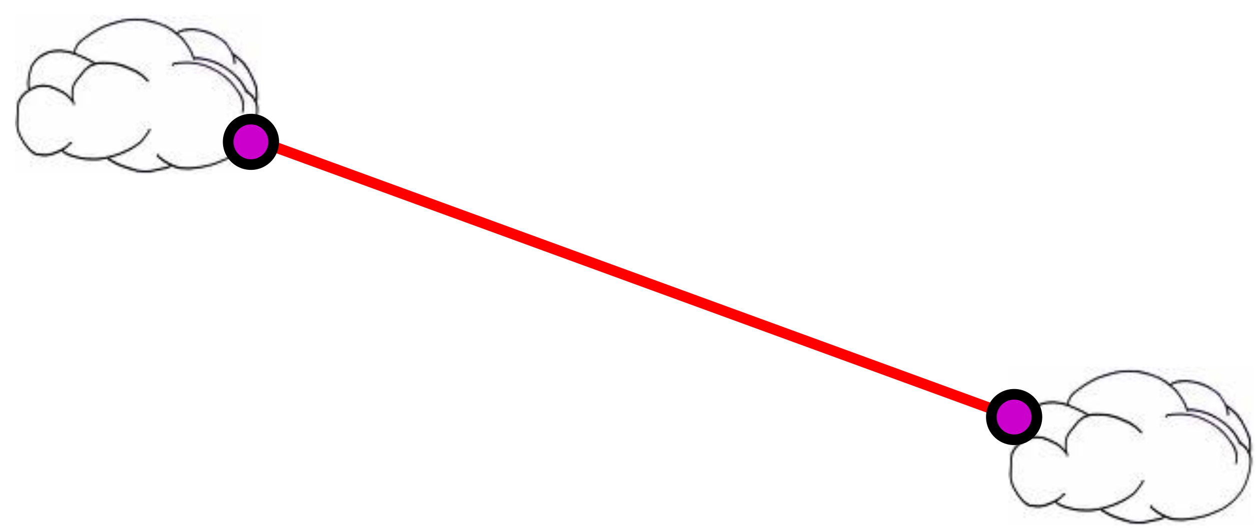
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - along each edge e



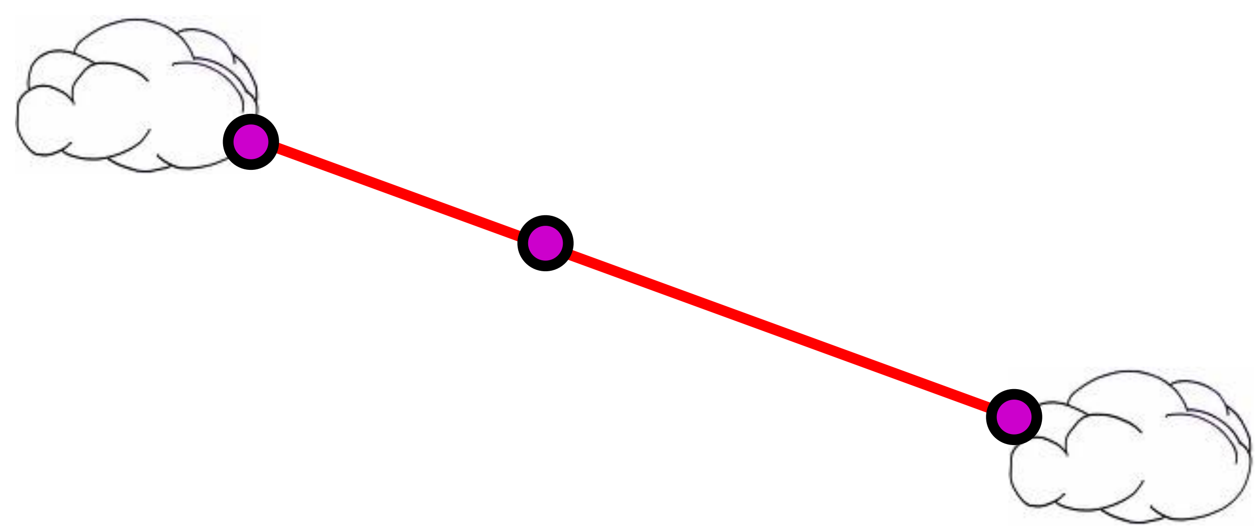
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



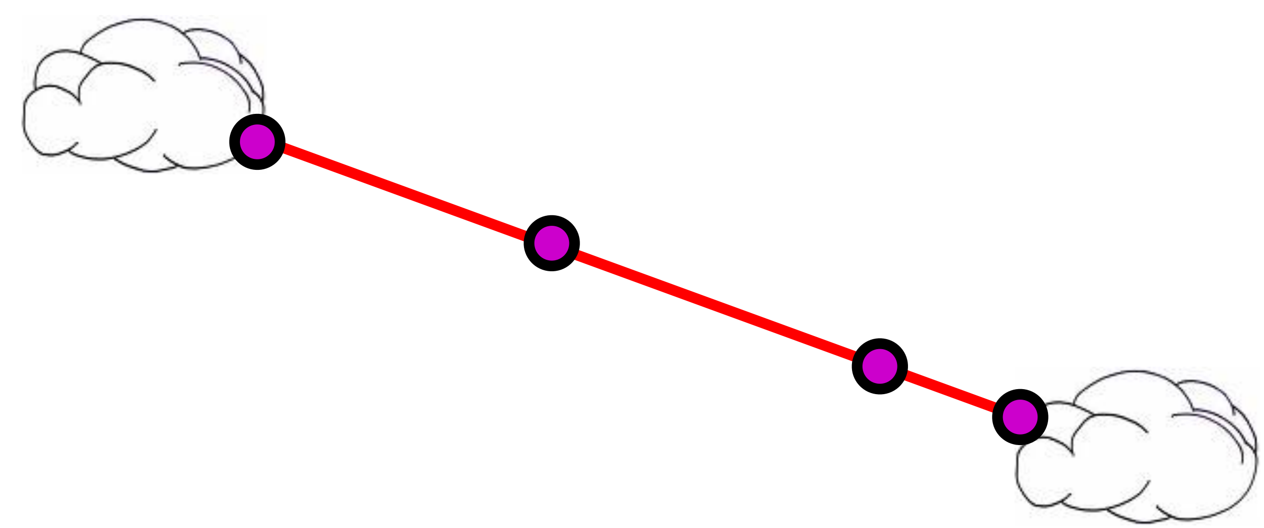
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



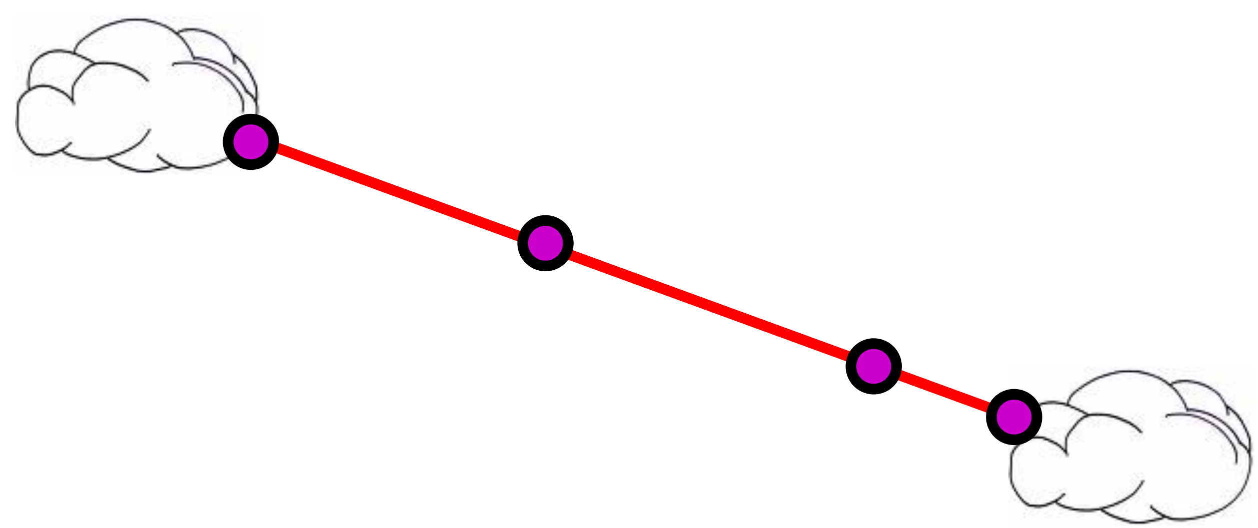
Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



Simple 6.73-Approximation (III)

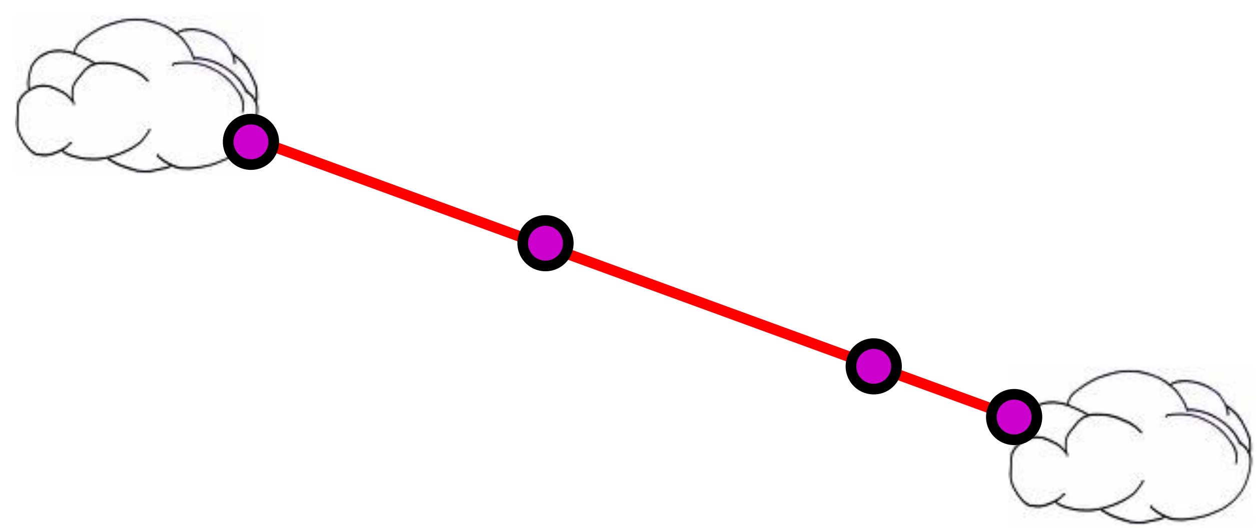
- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



$$|R''| \leq 2(|\mathrm{C}|-1) + \sum_e \left\lfloor \frac{|e|}{r} \right\rfloor$$

Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e

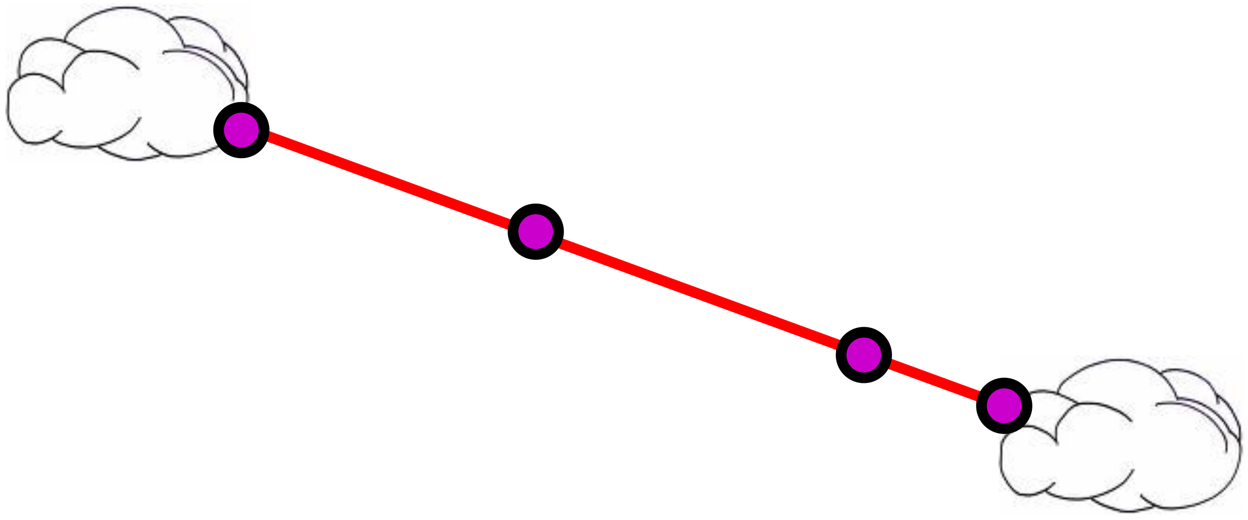


$$|R''| \leq 2(|\mathrm{C}|-1) + \sum_e \left\lfloor \frac{|e|}{r} \right\rfloor$$

$$\leq 2|\mathrm{C}| + \frac{|\mathrm{MSFN(C)}|}{r}$$

Simple 6.73-Approximation (III)

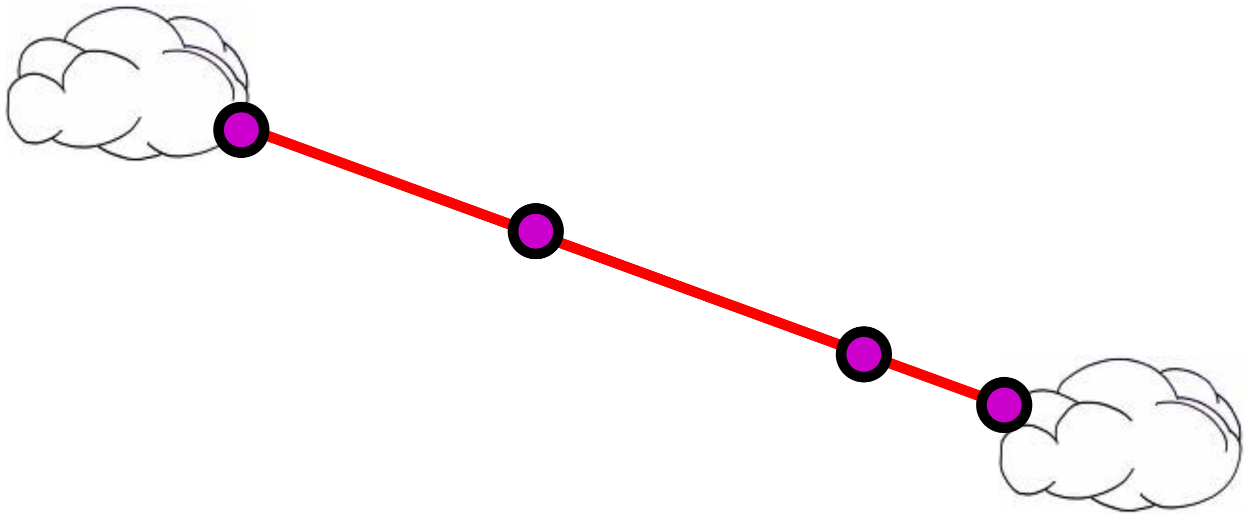
- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



$$|R''| \le 2(|\mathrm{C}|-1) + \sum_e \left\lfloor \frac{|e|}{r} \right\rfloor$$
$$\le 2|\mathrm{C}| + \frac{|\mathrm{MSFN(C)}|}{r}$$
$$\le 2|\mathrm{C}| + \frac{2}{\sqrt{3}} \cdot \frac{|\mathrm{MStFN(C)}|}{r}$$

Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



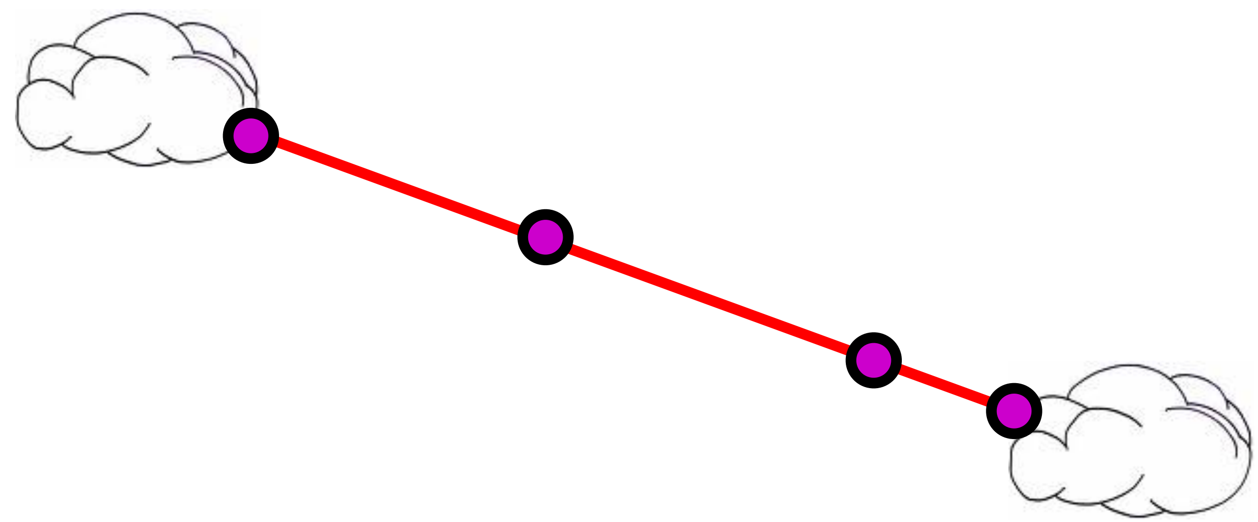
$$|R''| \le 2(|\mathrm{C}|-1) + \sum_e \left\lfloor \frac{|e|}{r} \right\rfloor$$

$$\le 2|\mathrm{C}| + \frac{|\mathrm{MSFN(C)}|}{r}$$

$$\le 2|\mathrm{C}| + \frac{2}{\sqrt{3}} \cdot \frac{|\mathrm{MStFN(C)}|}{r}$$

Simple 6.73-Approximation (III)

- Compute MSFN($\mathcal{C}$).
- Put relays R'' along edges:
  - 2 relays at end of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ along each edge e



$$|R''| \leq 2(|\mathrm{C}|-1) + \sum_e \left\lfloor \frac{|e|}{r} \right\rfloor$$

$$\leq 2|\mathrm{C}| + \frac{|\mathrm{MSFN(C)}|}{r}$$

$$\leq 2|\mathrm{C}| + \frac{2}{\sqrt{3}} \cdot \frac{|\mathrm{MStFN(C)}|}{r}$$

Simple 6.73-Approximation (IV)

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \leq$$

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \leq$$

Simple 6.73-Approximation (IV)

$|R'| + |R''| \leq$

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \leq$$

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \le$$

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \le \frac{137}{30}|\mathrm{Stab(B)}| - |\mathrm{C}| + \; 2|\mathrm{C}| + \frac{2}{\sqrt{3}} \cdot \frac{|\mathrm{MStFN(C)}|}{r}$$

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \le \frac{137}{30}|\text{Stab(B)}| - |\text{C}| + 2|\text{C}| + \frac{2}{\sqrt{3}} \cdot \frac{|\text{MStFN(C)}|}{r}$$

$$\le \left(\frac{137}{30} + 1 + \frac{2}{\sqrt{3}}\right)|R^*|$$

Simple 6.73-Approximation (IV)

$$|R'| + |R''| \le \frac{137}{30}|\text{Stab(B)}| - |\text{C}| + 2|\text{C}| + \frac{2}{\sqrt{3}} \cdot \frac{|\text{MStFN(C)}|}{r}$$

$$\le \left(\frac{137}{30} + 1 + \frac{2}{\sqrt{3}}\right)|R^*|$$

$$< 6.73|R^*|$$

Simple 6.73-Approximation (V)

Simple 6.73-Approximation (V)

- Running time:
  - Build Delaunay of n sensors $O(n \log n)$
  - Identify blobs and clouds $O(n)$
  - Greedy to stab blobs (≤5 per relay) $O(n)$
  - MSFN(C) is subgraph of Delaunay $O(n \log n)$
  - Total: $O(n \log n)$

A 3.11-Approximation

A 3.11-Approximation

Overview:

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays

    Complete connectivity of these clouds (Lemma 1)

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays

  Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
    - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

Relays of 3 types: RED, GREEN, YELLOW : A_r , A_g , A_y

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

Relays of 3 types: RED, GREEN, YELLOW : A_r , A_g , A_y

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

Relays of 3 types: RED, GREEN, YELLOW : A_r , A_g , A_y

A 3.11-Approximation

Overview:

- Compute *optimal* stabbings for clouds of blobs that can be stabbed with few (≤ k, constant) relays
  - Complete connectivity of these clouds (Lemma 1)
- For other clouds, *greedily* connect blobs
- Greedily connect clouds into *clusters*, adding 2 additional relays per cloud
- Connect the clusters with spanning forest

Relays of 3 types: RED, GREEN, YELLOW : A_r , A_g , A_y

A 3.11-Approximation II

A 3.11-Approximation II

Approximation bound:

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \dot{\cup} R_l^*$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \dot{\cup} R_l^*$$

dark: within clouds

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot{\cup}\, R_l^*$$

dark: within clouds

light: outside of clouds

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot{\cup}\, R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \mathbin{\dot\cup} R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \dot{\cup} R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \le (3.084 + 1/k)\,|R_d^*|$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \le (3.084 + 1/k)\,|R_d^*|$$

$$|A_y| \le 3.11\,|R_l^*|$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

$$|A_y| \leq 3.11\,|R_l^*|$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

$$|A_y| \leq 3.11\,|R_l^*|$$

Consequence: Total # of relays we compute:

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \cup R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \leq (3.084 + 1/k)|R_d^*|$$

$$|A_y| \leq 3.11|R_l^*|$$

Consequence: Total # of relays we compute:

$$|A| = |A_r| + |A_g| + |A_y| \leq 3.11(|R_d^*| + |R_l^*|) = 3.11|R^*|$$

A 3.11-Approximation II

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark: within clouds

light: outside of clouds

Will show:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

$$|A_y| \leq 3.11|R_l^*|$$

Consequence: Total # of relays we compute:

$$|A| = |A_r| + |A_g| + |A_y| \leq 3.11(|R_d^*| + |R_l^*|) = 3.11|R^*|$$

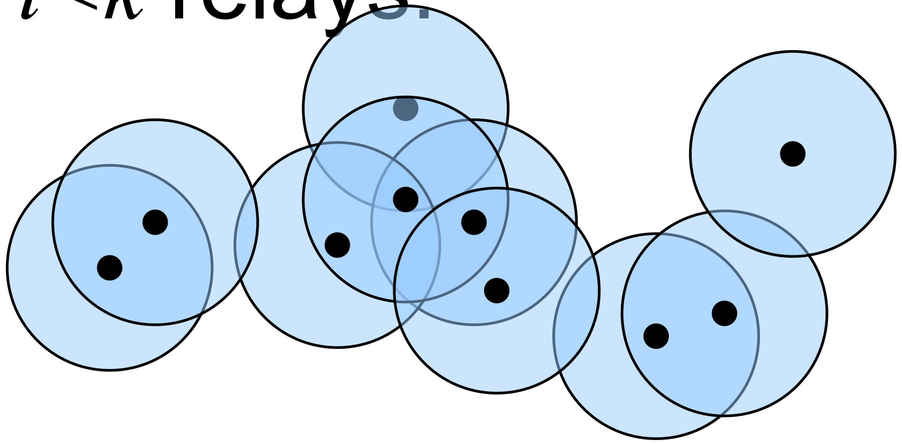
Clouds with Few Stabs

Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:

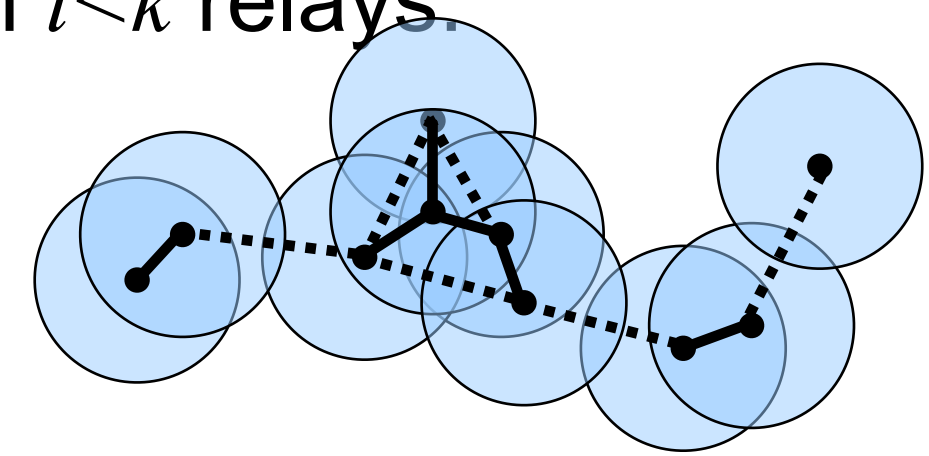
Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:



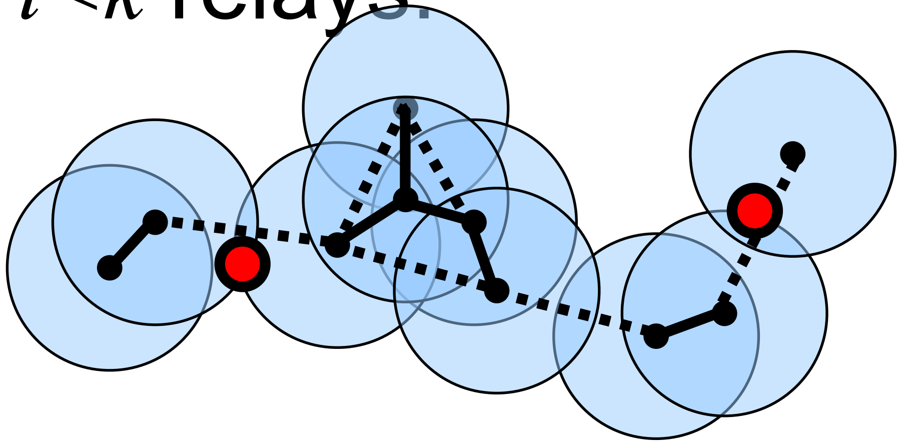
Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i < k$ relays:



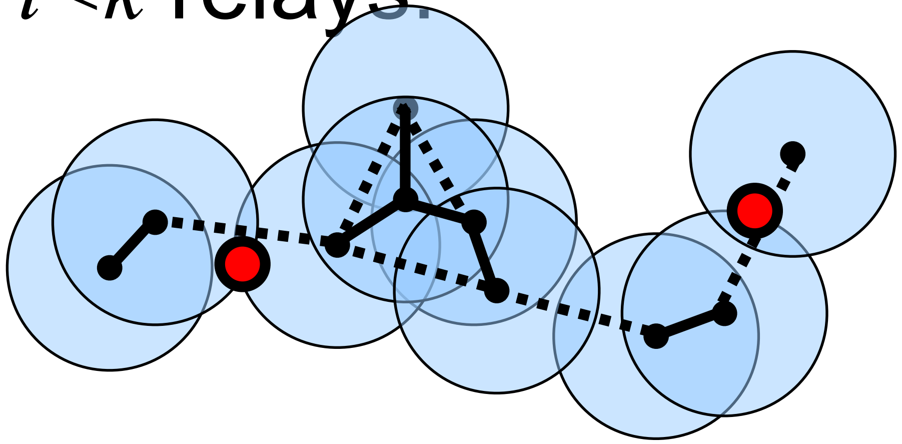
Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:



Clouds with Few Stabs

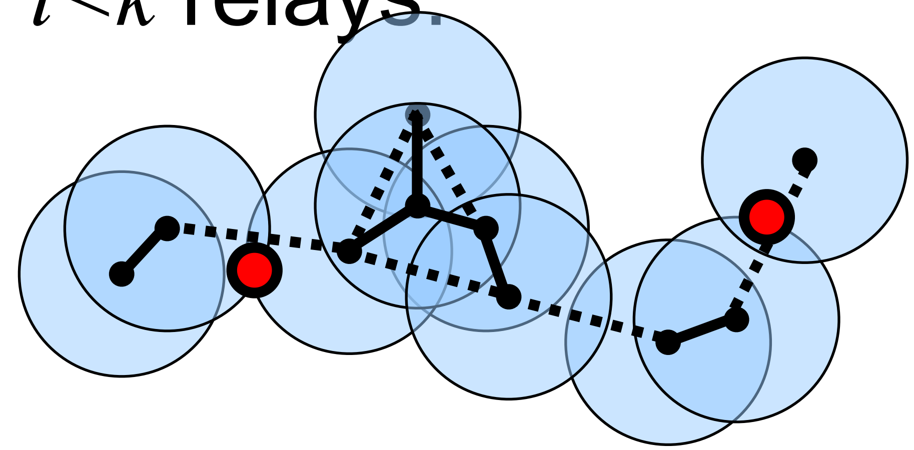
For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:



Then use Lemma 1 to complete connection:

Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:

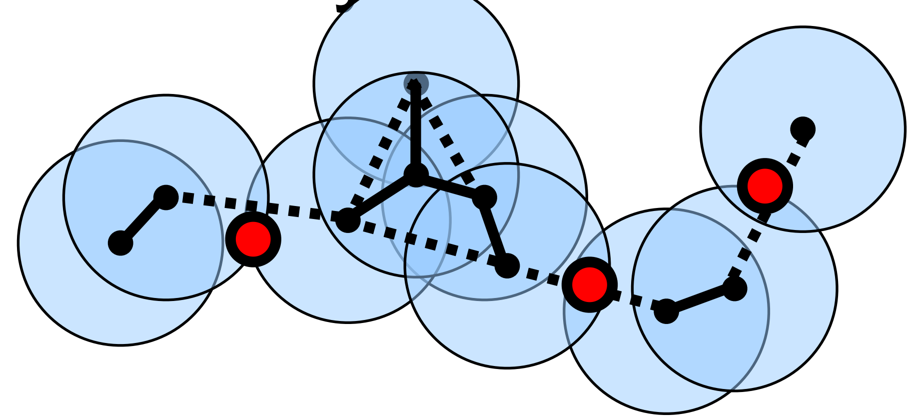


Then use Lemma 1 to complete connection:

$\leq 2i-1$ red relays to interconnect each $C \in \mathcal{C}_i$

Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:

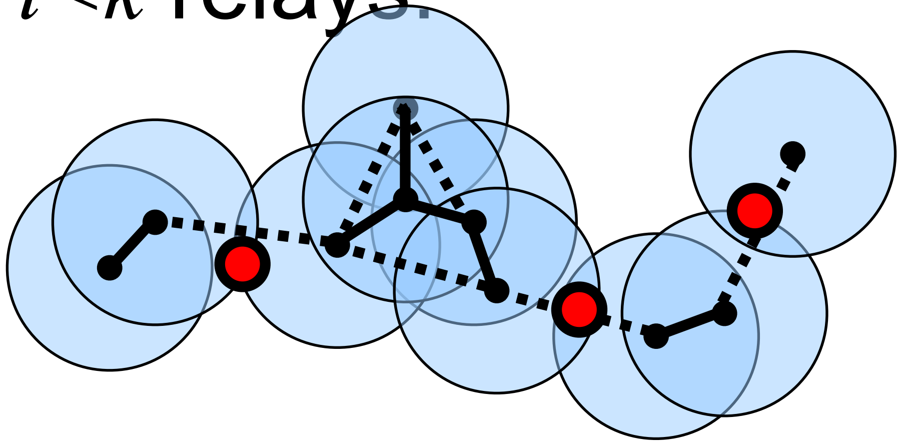


Then use Lemma 1 to complete connection:

$\leq 2i\text{-}1$ red relays to interconnect each $C \in \mathcal{C}_i$

Clouds with Few Stabs

For any constant k, determine those clouds $C \in \mathcal{C}_i$ whose blobs can be stabbed with $i<k$ relays:



Then use Lemma 1 to complete connection:

≤ $2i-1$ red relays to interconnect each $C \in \mathcal{C}_i$

Let $\mathcal{C}_{k+}$ be the set of clouds requiring ≥ k stabs

Stitching a Cloud From $\mathcal{C}_{k+}$

Stitching a Cloud From $\mathcal{C}_{k+}$

Consider clouds $C \in \mathcal{C}_{k+}$ that need more than k stabs.

Stitching a Cloud From $\mathcal{C}_{k+}$

Consider clouds $C \in \mathcal{C}_{k+}$ that need more than k stabs.

Use greedy set cover for those clouds.

Stitching a Cloud From $\mathcal{C}_{k+}$

Consider clouds $C \in \mathcal{C}_{k+}$ that need more than k stabs.

Use greedy set cover for those clouds.

Lemma 3: For each cloud C,

Stitching a Cloud From $\mathcal{C}_{k+}$

Consider clouds $C \in \mathcal{C}_{k+}$ that need more than k stabs.

Use greedy set cover for those clouds.

Lemma 3: For each cloud C,

$$|A_r^*| \leq \frac{37}{12} |R_d^* \cap C| - 1$$

Stitching a Cloud From $\mathcal{C}_{k+}$

Consider clouds $C \in \mathcal{C}_{k+}$ that need more than k stabs.

Use greedy set cover for those clouds.

Lemma 3: For each cloud C,

$$|A_r^*| \leq \frac{37}{12} |R_d^* \cap C| - 1$$

Proof: Omitted. Note that

Stitching a Cloud From $\mathcal{C}_{k+}$

Consider clouds $C \in \mathcal{C}_{k+}$ that need more than k stabs.

Use greedy set cover for those clouds.

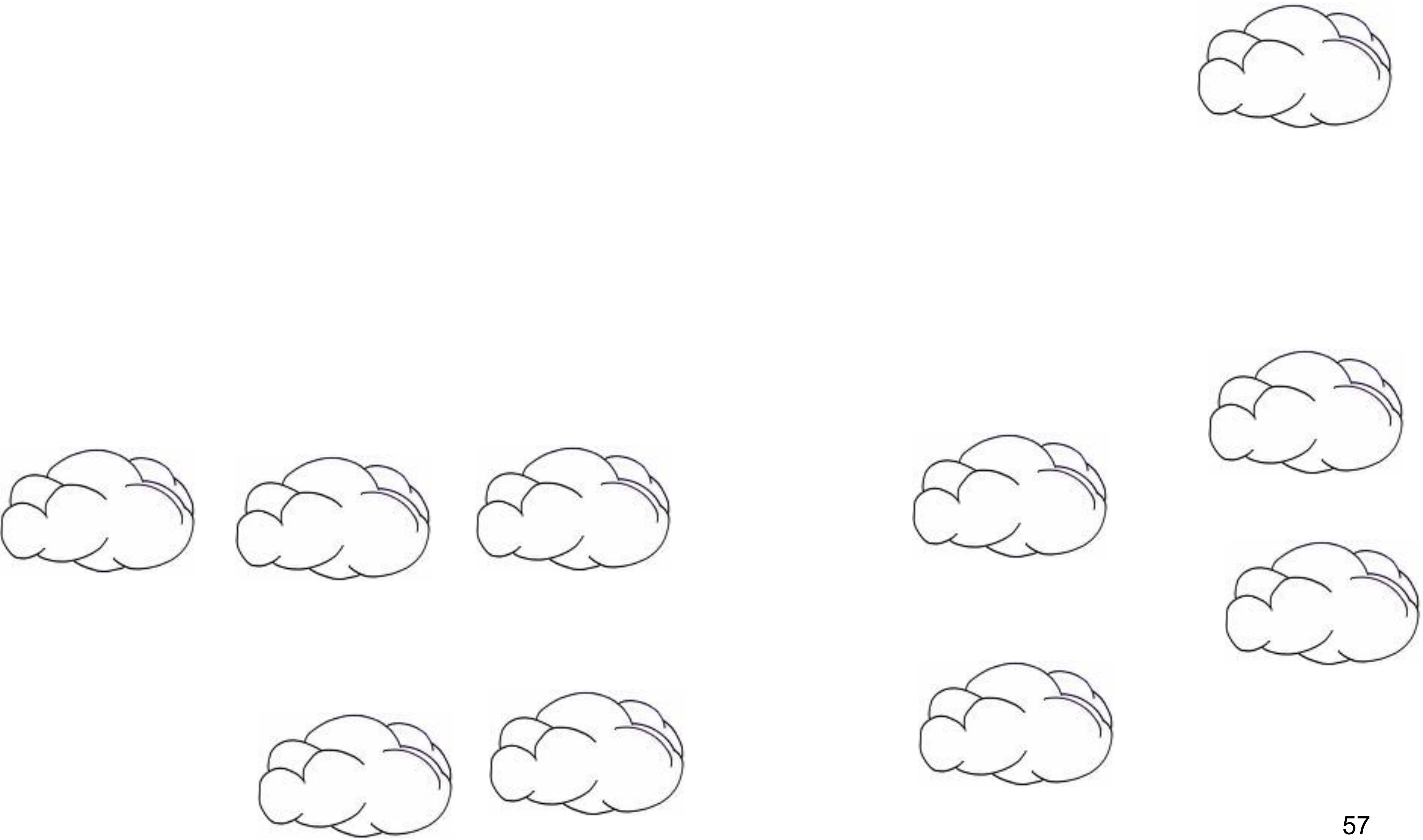
Lemma 3: For each cloud C,

$$|A_r^*| \leq \frac{37}{12} |R_d^* \cap C| - 1$$

Proof: Omitted. Note that $1 + \frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{1} = \frac{37}{12}.$

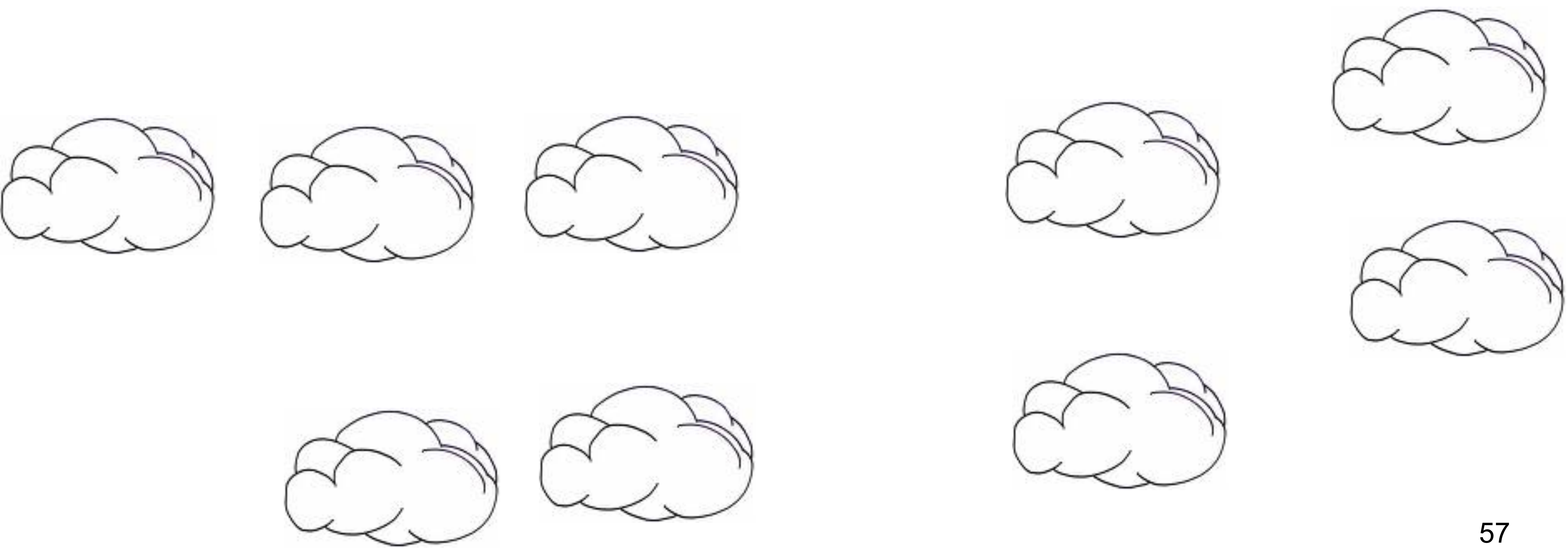
Cloud Clusters: Green Relays

Cloud Clusters: Green Relays



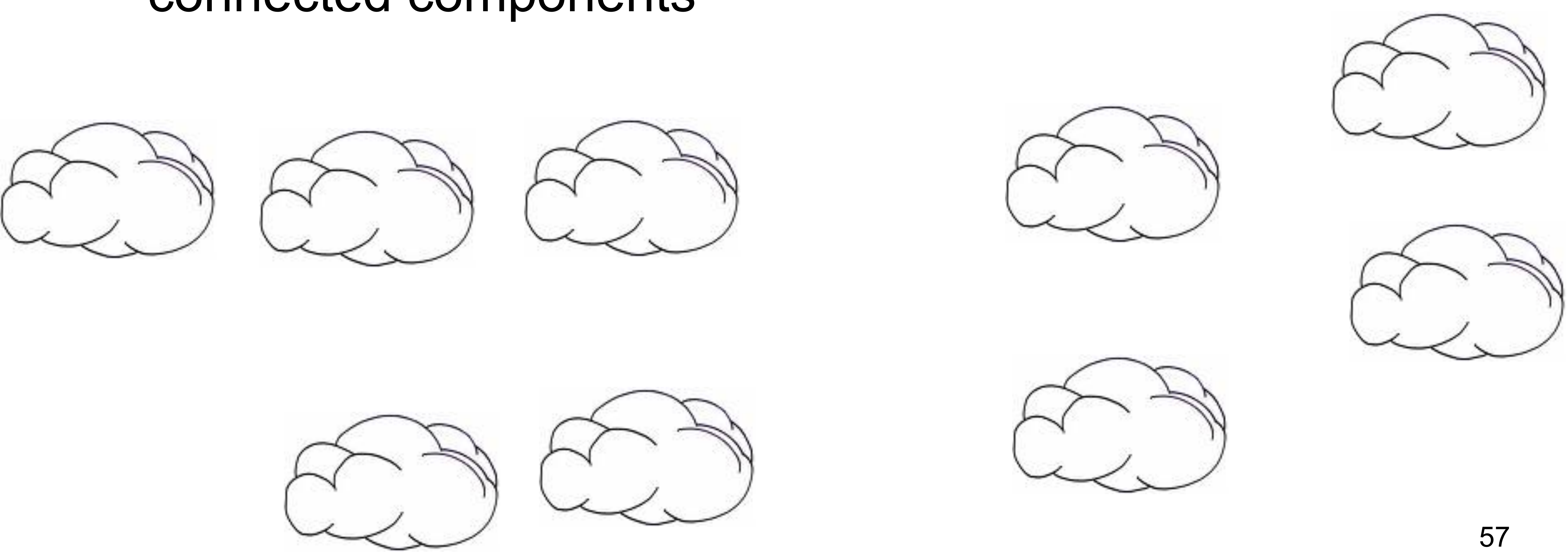
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  - Place 2 relays to connect 2 clusters; repeat
  - Then, place 4 relays to interconnect 3 clusters; repeat
  - Then, place 6 relays to interconnect 4 clusters; repeat



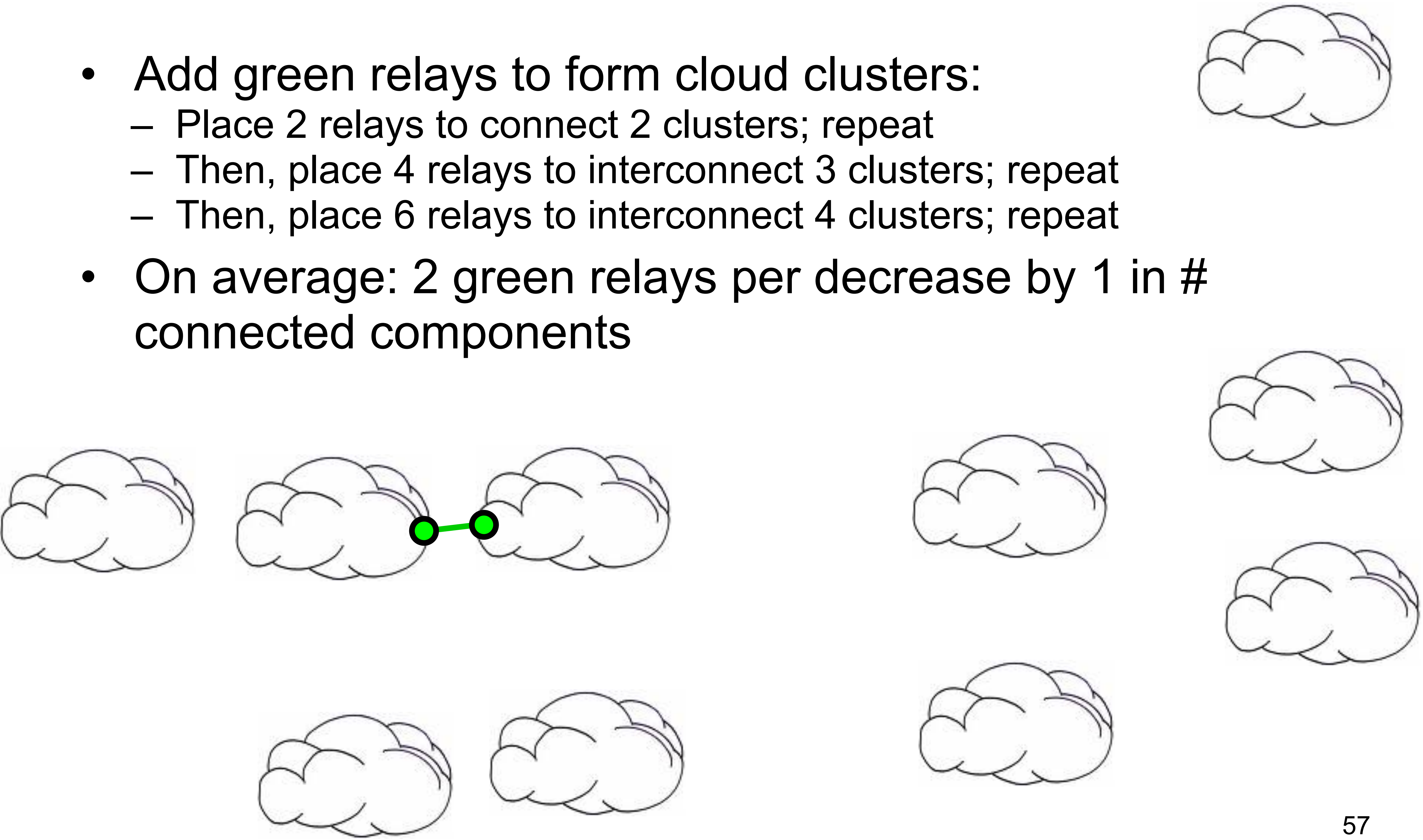
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  - Place 2 relays to connect 2 clusters; repeat
  - Then, place 4 relays to interconnect 3 clusters; repeat
  - Then, place 6 relays to interconnect 4 clusters; repeat
- On average: 2 green relays per decrease by 1 in # connected components



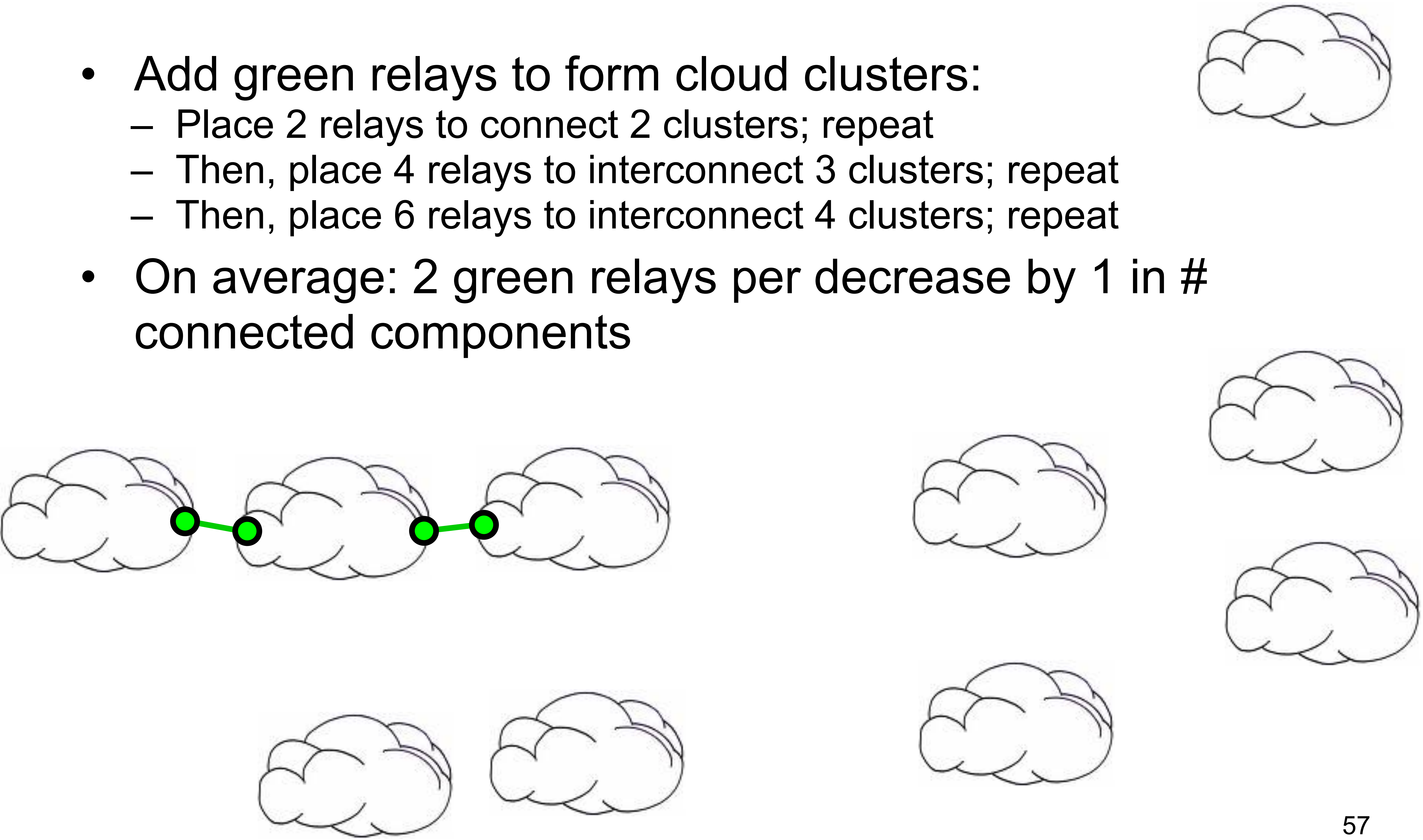
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  - Place 2 relays to connect 2 clusters; repeat
  - Then, place 4 relays to interconnect 3 clusters; repeat
  - Then, place 6 relays to interconnect 4 clusters; repeat
- On average: 2 green relays per decrease by 1 in # connected components



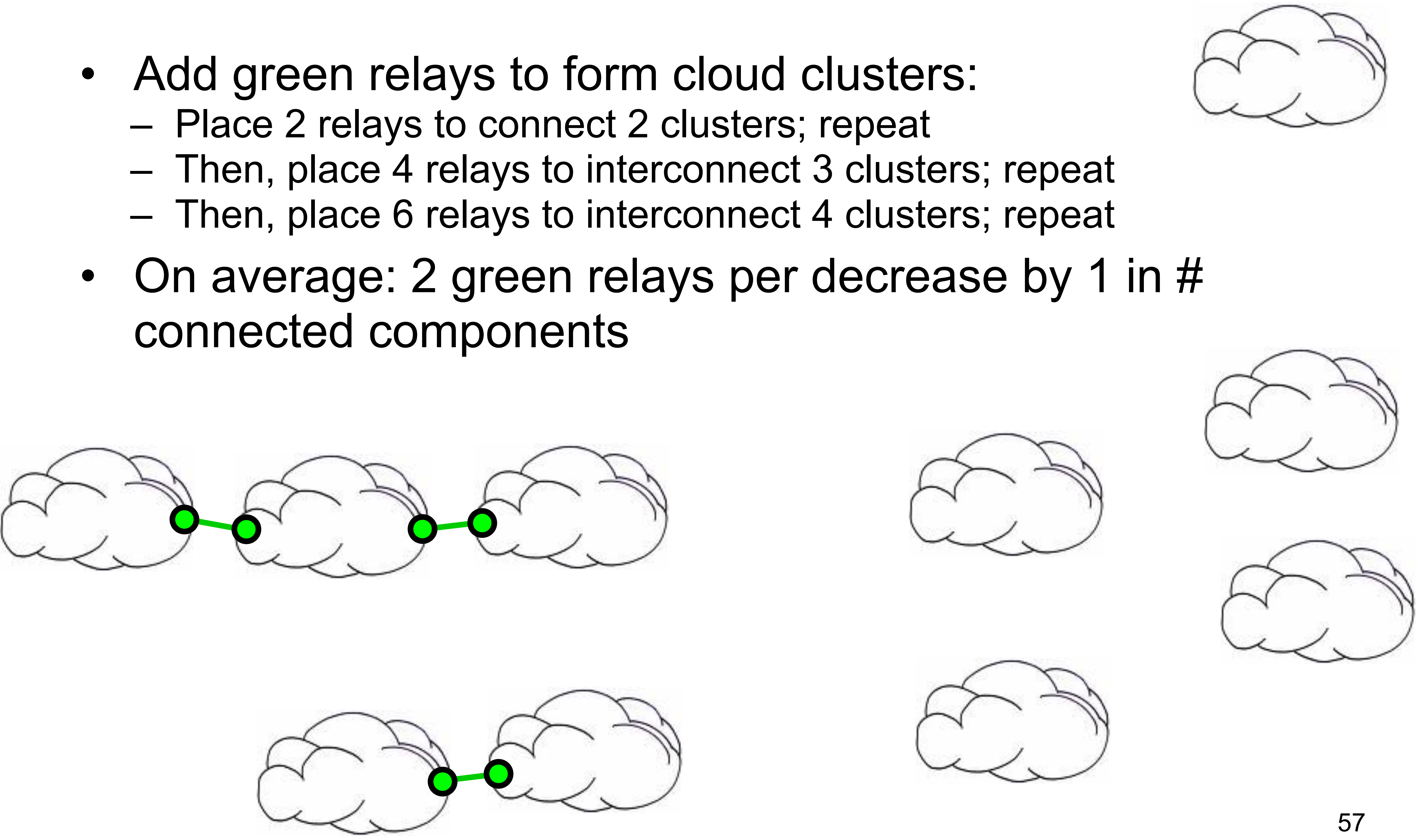
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  - Place 2 relays to connect 2 clusters; repeat
  - Then, place 4 relays to interconnect 3 clusters; repeat
  - Then, place 6 relays to interconnect 4 clusters; repeat
- On average: 2 green relays per decrease by 1 in # connected components



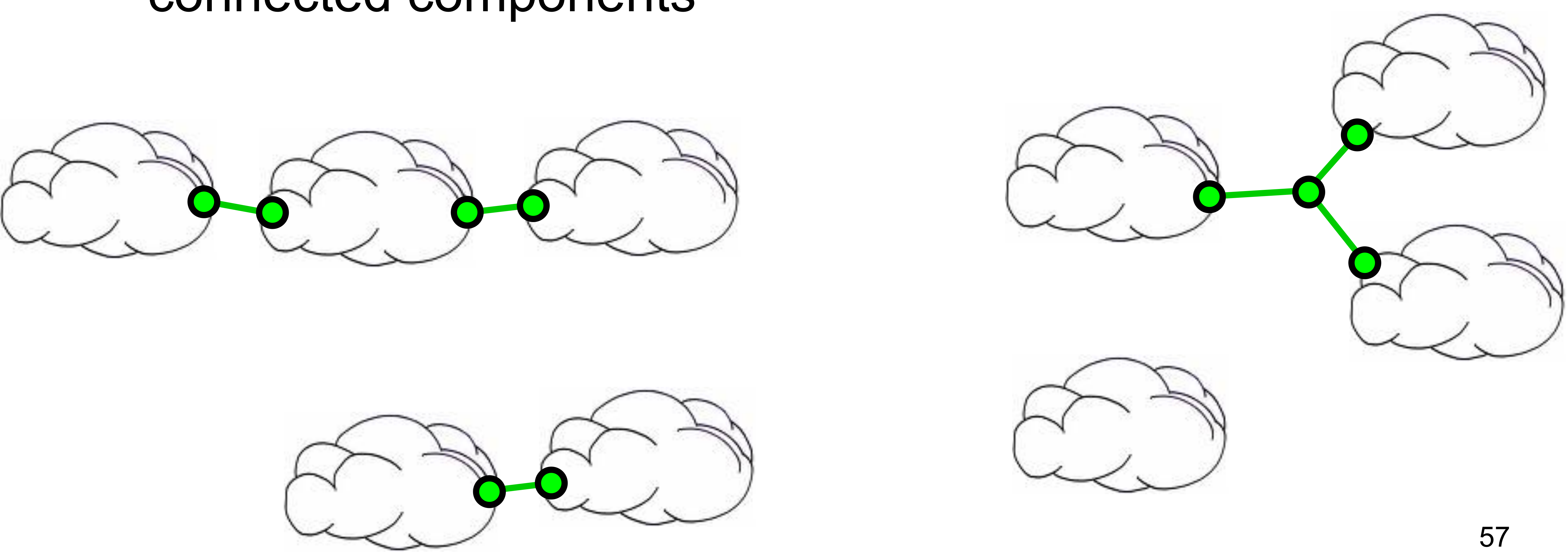
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  - Place 2 relays to connect 2 clusters; repeat
  - Then, place 4 relays to interconnect 3 clusters; repeat
  - Then, place 6 relays to interconnect 4 clusters; repeat
- On average: 2 green relays per decrease by 1 in # connected components



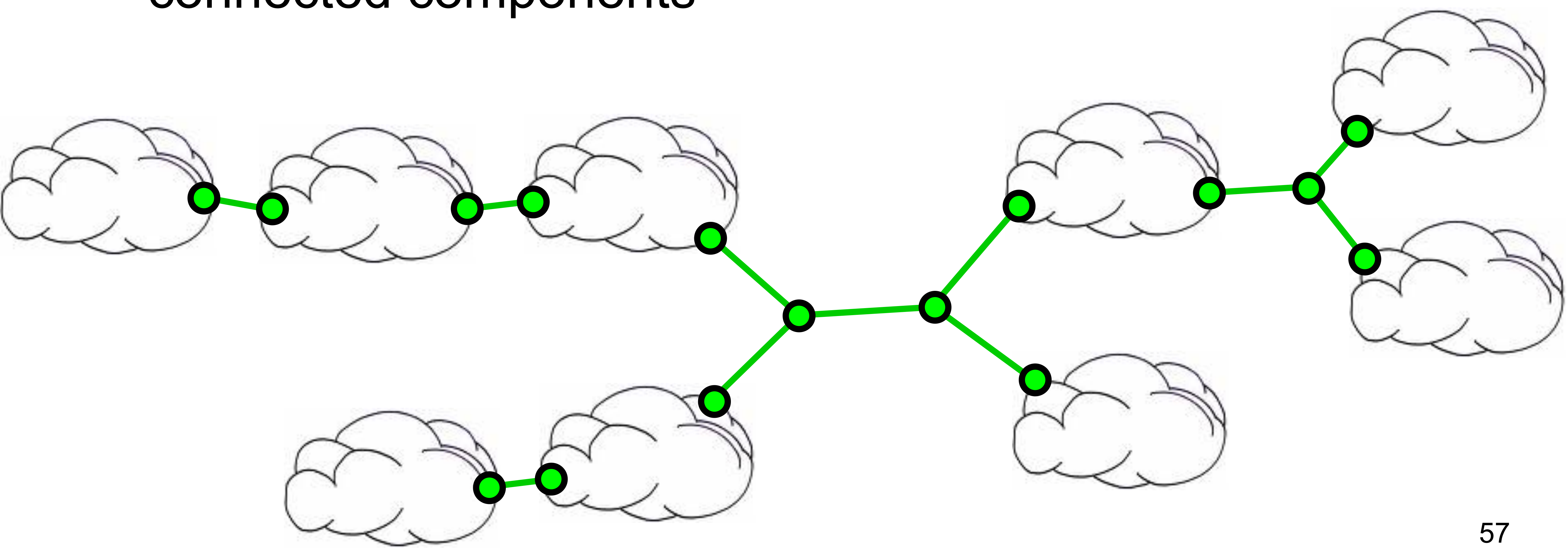
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  – Place 2 relays to connect 2 clusters; repeat
  – Then, place 4 relays to interconnect 3 clusters; repeat
  – Then, place 6 relays to interconnect 4 clusters; repeat
- On average: 2 green relays per decrease by 1 in # connected components



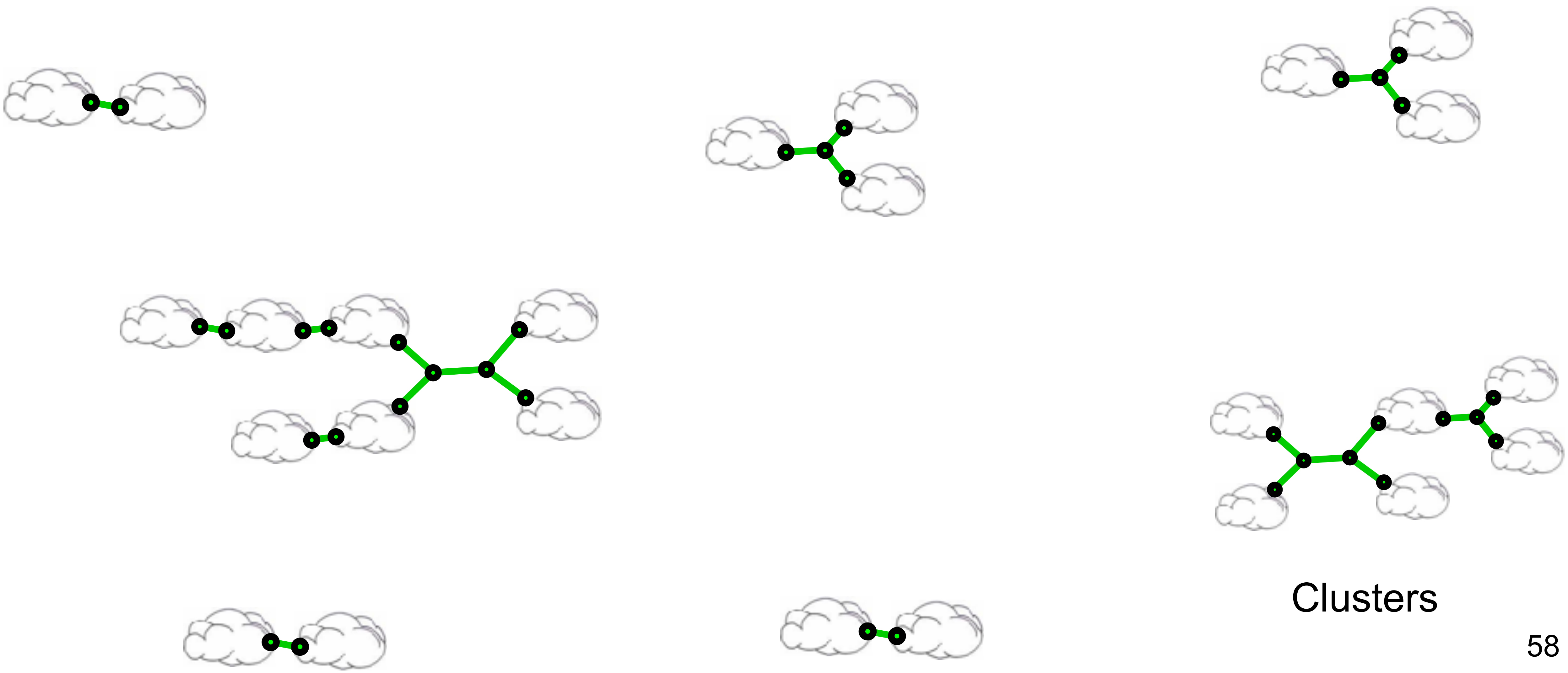
Cloud Clusters: Green Relays

- Add green relays to form cloud clusters:
  - Place 2 relays to connect 2 clusters; repeat
  - Then, place 4 relays to interconnect 3 clusters; repeat
  - Then, place 6 relays to interconnect 4 clusters; repeat
- On average: 2 green relays per decrease by 1 in # connected components



Interconnecting the Clusters

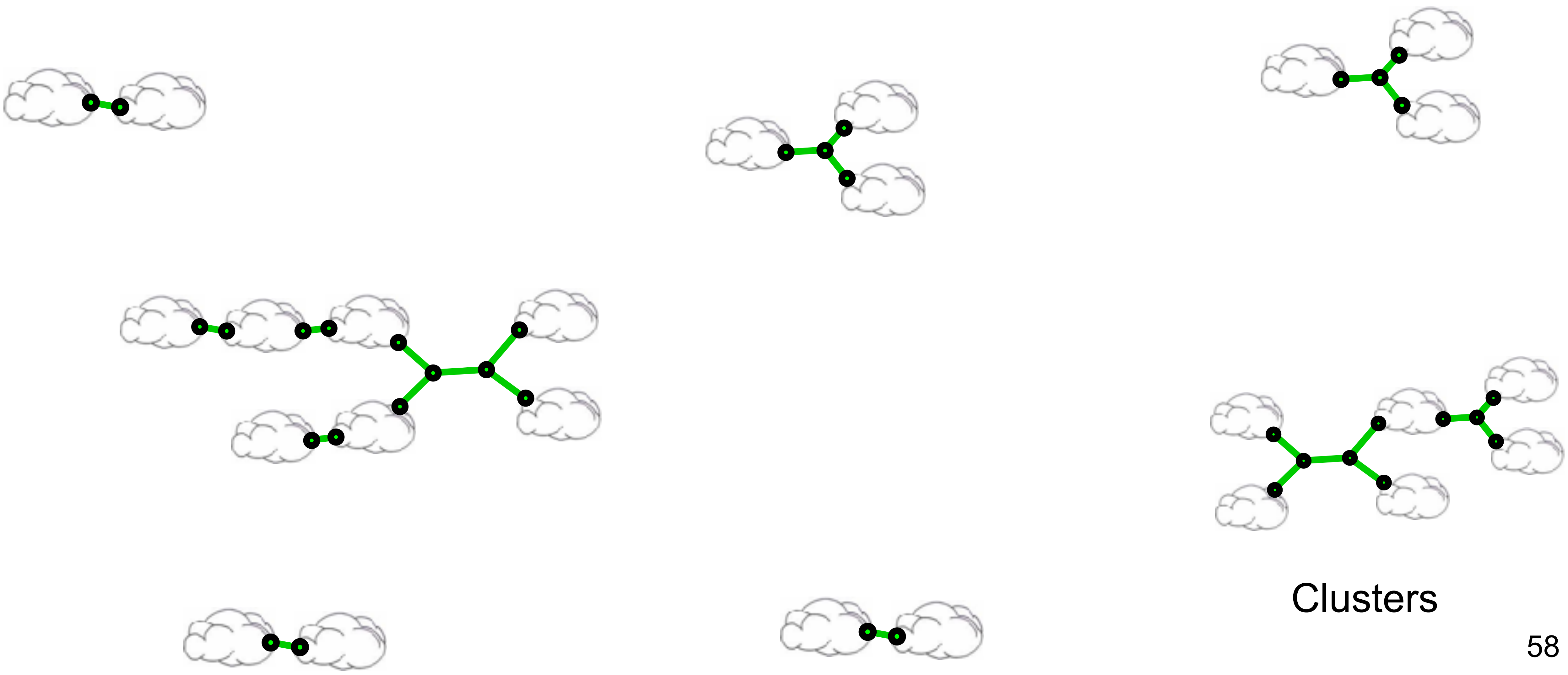
Interconnecting the Clusters



Clusters

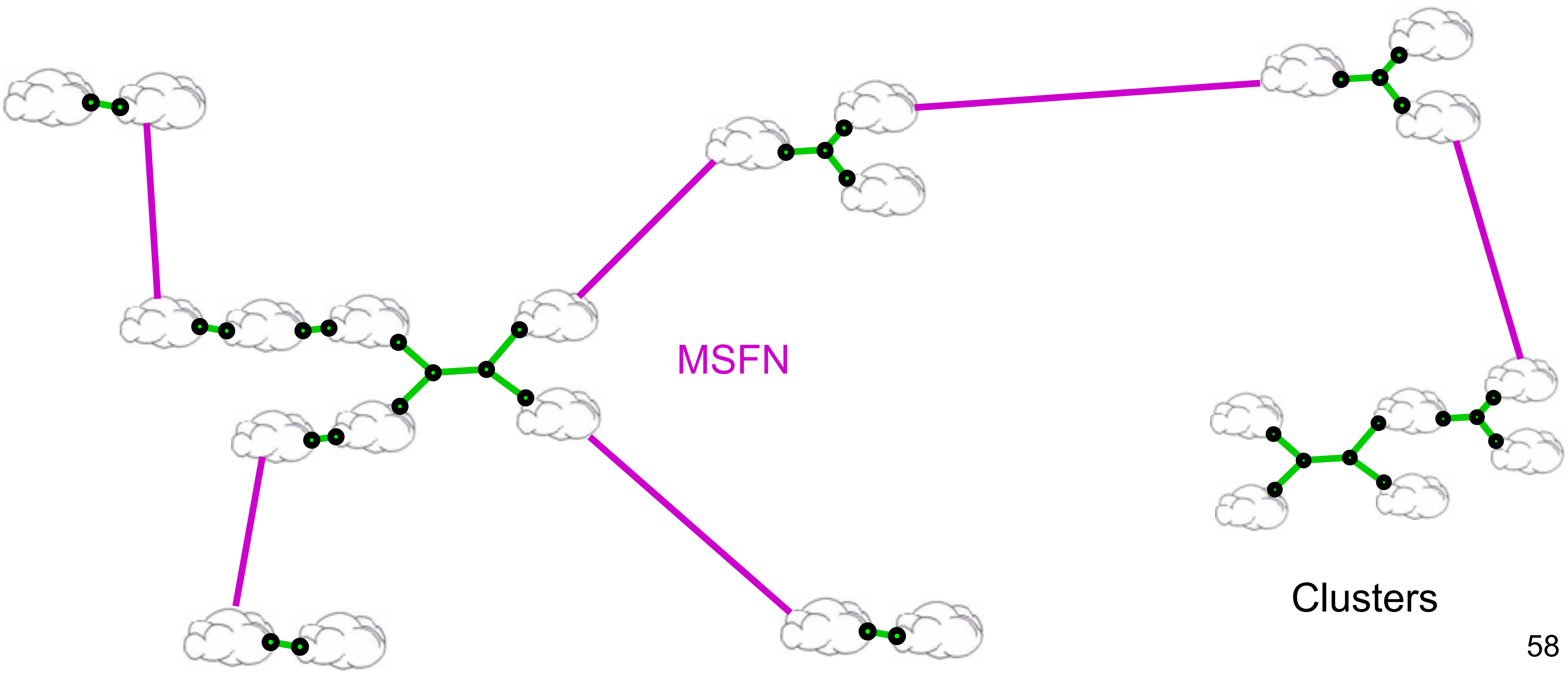
Interconnecting the Clusters

- Compute MSFN on set of *clusters*



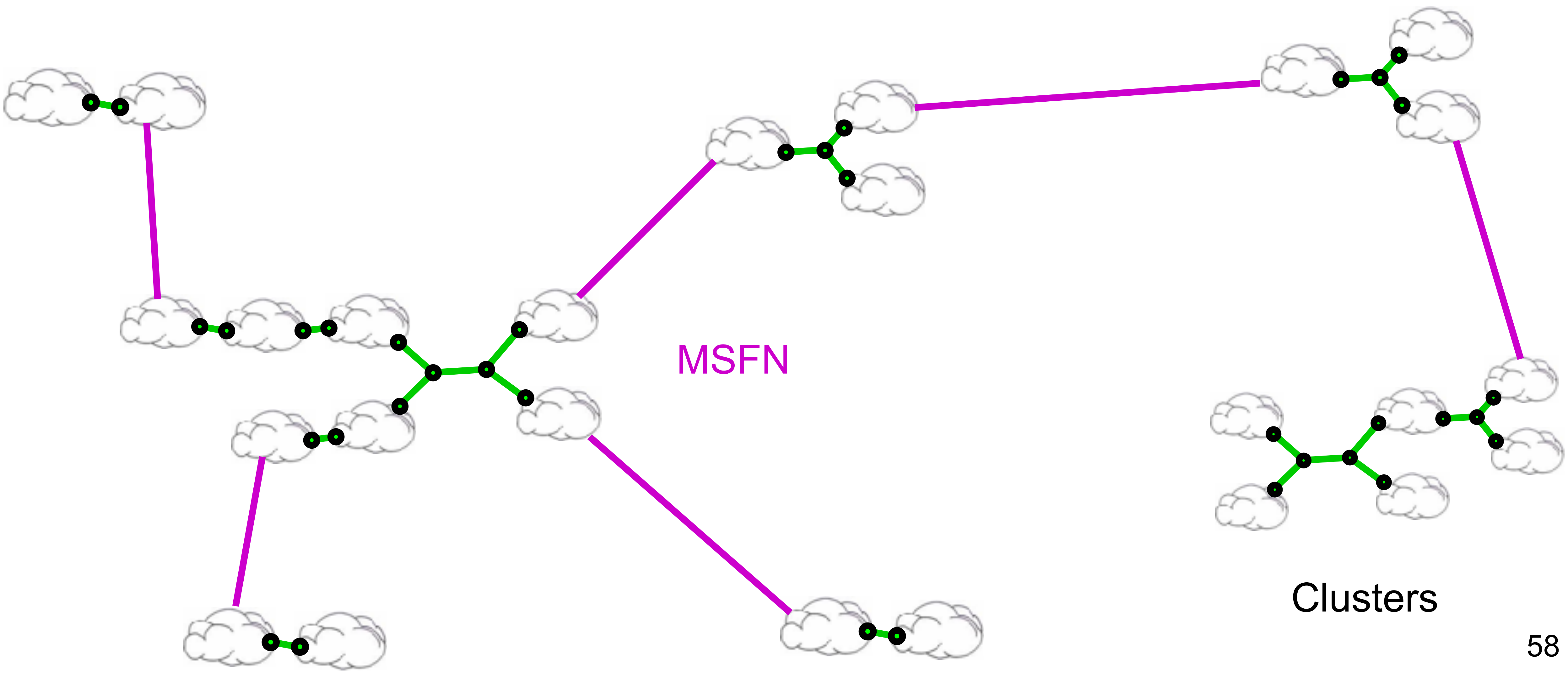
Interconnecting the Clusters

- Compute MSFN on set of *clusters*



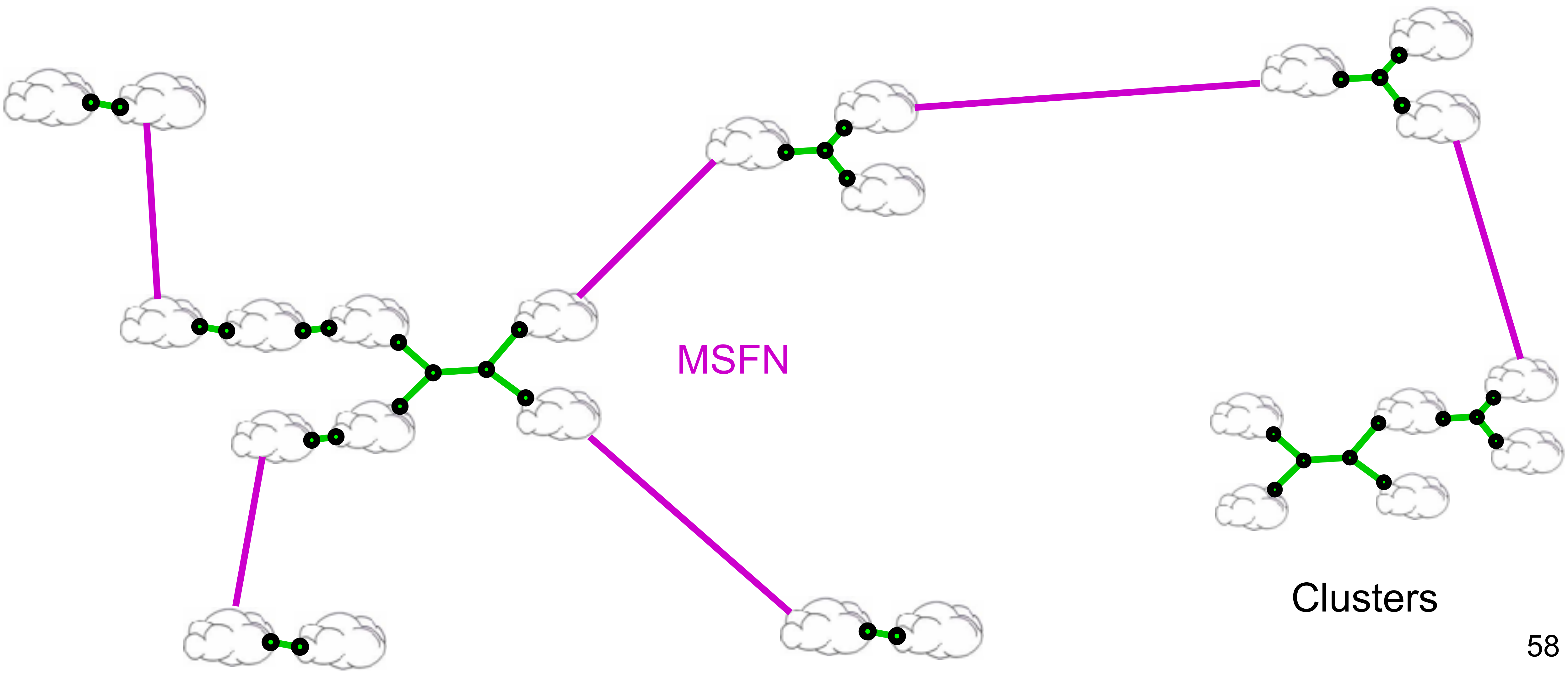
Interconnecting the Clusters

- Compute MSFN on set of *clusters*
- Put relays along edges:



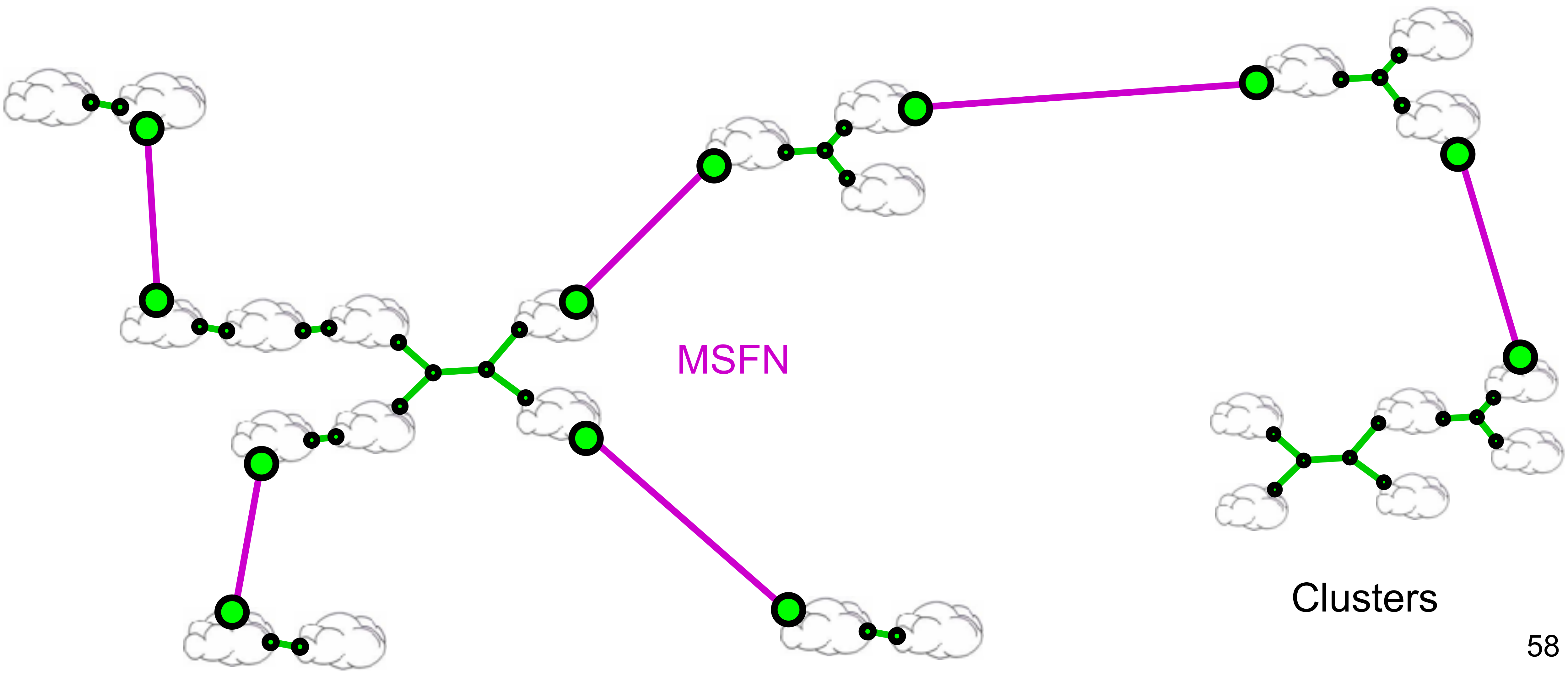
Interconnecting the Clusters

- Compute MSFN on set of *clusters*
- Put relays along edges:
  - 2 green relays at ends of each edge



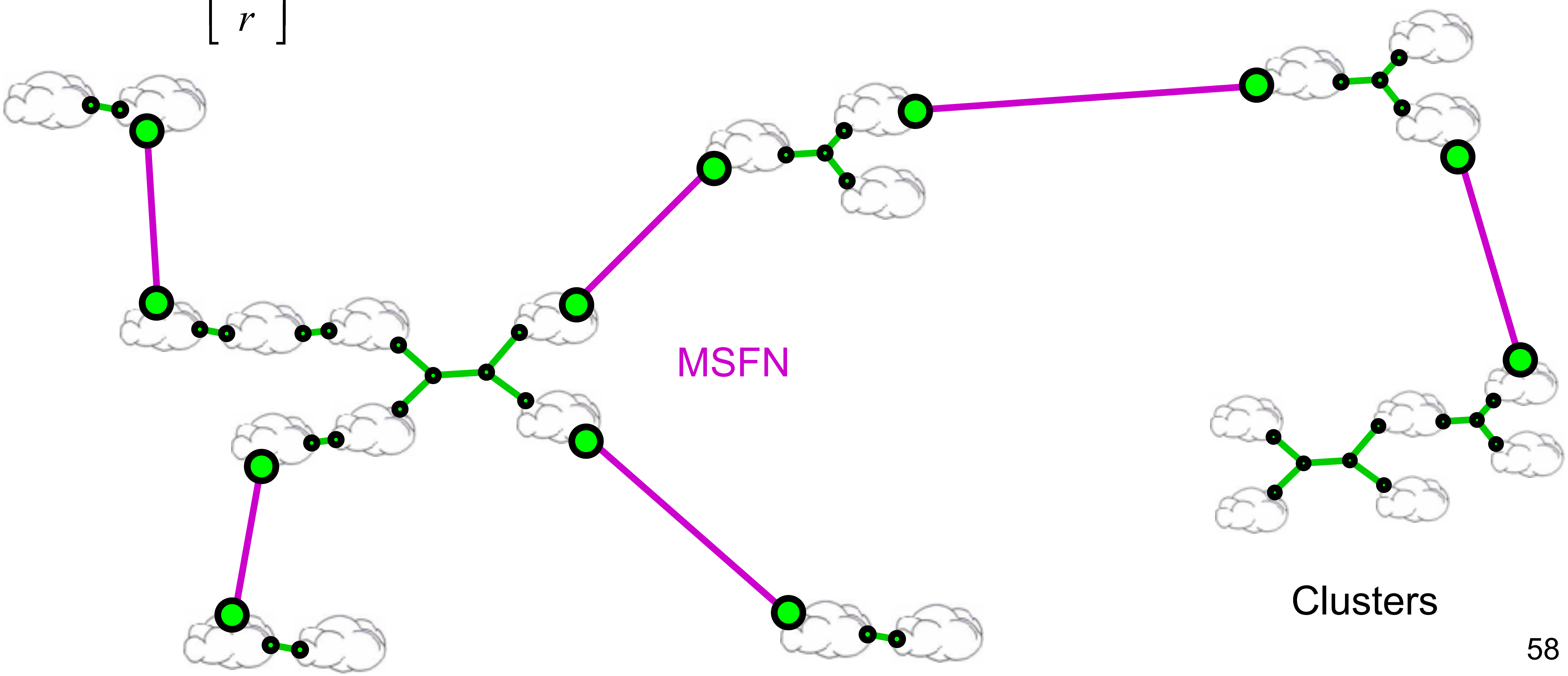
Interconnecting the Clusters

- Compute MSFN on set of *clusters*
- Put relays along edges:
  - 2 green relays at ends of each edge



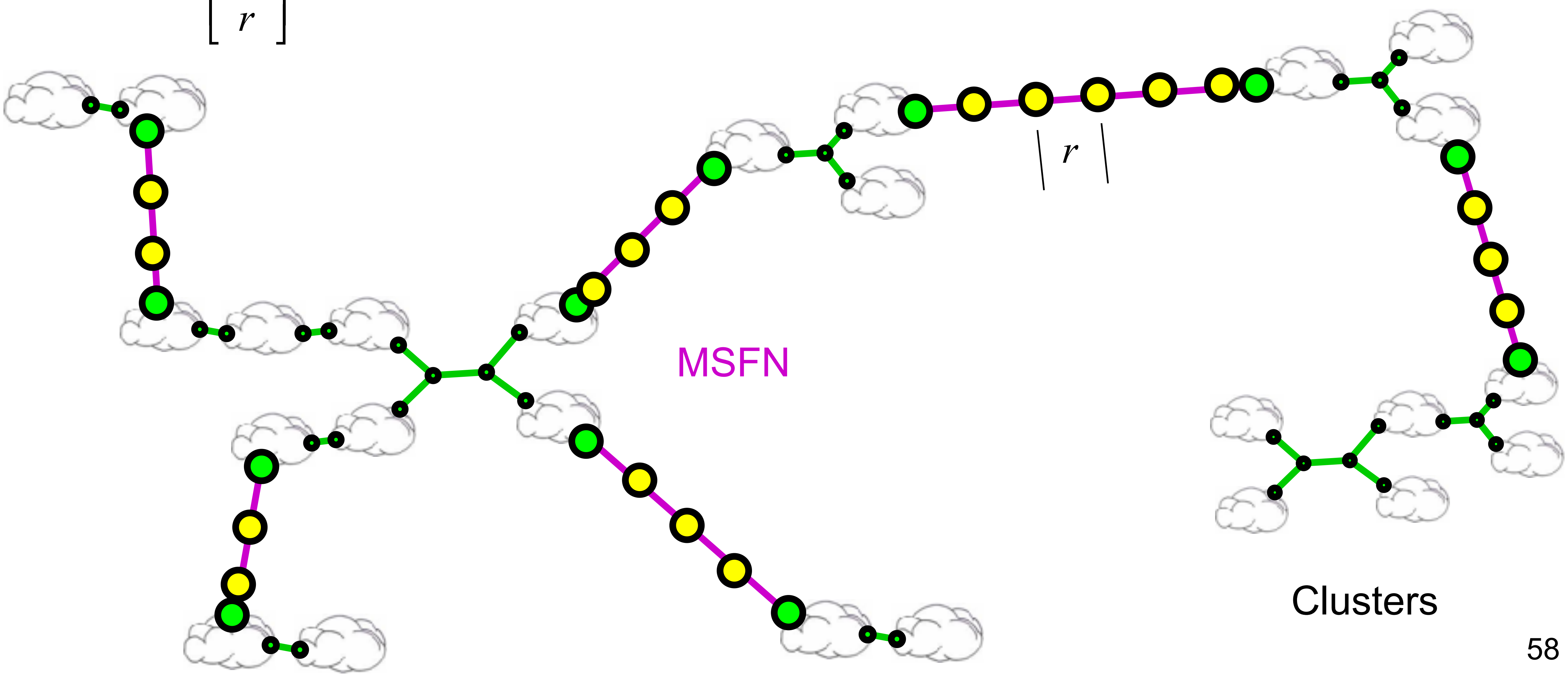
Interconnecting the Clusters

- Compute MSFN on set of *clusters*
- Put relays along edges:
  – 2 green relays at ends of each edge
  – $\left\lfloor \frac{|e|}{r} \right\rfloor$ yellow relays along each edge e



Interconnecting the Clusters

- Compute MSFN on set of *clusters*
- Put relays along edges:
  - 2 green relays at ends of each edge
  - $\left\lfloor \frac{|e|}{r} \right\rfloor$ yellow relays along each edge e



Analysis: Red and Green Relays

Analysis: Red and Green Relays

- An opt solution R^* has at least

Analysis: Red and Green Relays

- An opt solution R^* has at least

$$|R_d^*| \geq k\,|C^{k+}| + \sum_{i=1}^{k-1} i\,|C^i|$$

Analysis: Red and Green Relays

- An opt solution R* has at least
  $$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$
  dark relays (inside clouds)

Analysis: Red and Green Relays

- An opt solution R* has at least

$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$

  dark relays (inside clouds)
- Also,

Analysis: Red and Green Relays

- An opt solution R* has at least
$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$
dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$

Analysis: Red and Green Relays

- An opt solution R* has at least
$$|R_d^*| \ge k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$
dark relays (inside clouds)
- Also, $|R_d^* \cap C| \ge 1, \quad \forall C$

$$|A_r| + |A_g| \le$$

Analysis: Red and Green Relays

- An opt solution R* has at least
$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$
dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$

$|A_r| + |A_g| \leq$

Analysis: Red and Green Relays

- An opt solution R* has at least

$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$

  dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$

$$|A_r| + |A_g| \leq$$

Analysis: Red and Green Relays

- An opt solution R^* has at least
$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^{i}|$$
dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$

$$|A_r| + |A_g| \leq$$

Analysis: Red and Green Relays

- An opt solution R* has at least

$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$

  dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$
- We place ≤ *2i-1* red relays per cloud of $\mathcal{C}_i$,

$$|A_r| + |A_g| \leq$$

Analysis: Red and Green Relays

- An opt solution R* has at least
$$|R_d^*| \geq k\,|C^{k+}| + \sum_{i=1}^{k-1} i\,|C^i|$$
dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$
- We place ≤ *2i-1* red relays per cloud of $\mathcal{C}_i$,

$$|A_r| + |A_g| \leq \qquad + \sum_{i=1}^{k-1} (2i-1)\,|C^i|$$

Analysis: Red and Green Relays

- An opt solution R* has at least

$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$

  dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall\, C$
- We place ≤ $2i\text{-}1$ red relays per cloud of $\mathcal{C}^i$,

  and ≤ 37/12$|R^*_d \cap C|$-1 red relays per cloud of $\mathcal{C}^{k+}$

$$|A_r| + |A_g| \leq \qquad + \sum_{i=1}^{k-1} (2i-1)\,|\mathrm{C}^i|$$

Analysis: Red and Green Relays

- An opt solution R* has at least
$$|R_d^*| \geq k\,|\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathrm{C}^i|$$
dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$
- We place ≤ *2i-1* red relays per cloud of $\mathcal{C}^i$,

and ≤ 37/12$|R^*_d \cap C|$-1 red relays per cloud of $\mathcal{C}^{k+}$

$$|A_r| + |A_g| \leq \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)\,|\mathrm{C}^i|$$

Analysis: Red and Green Relays

- An opt solution R* has at least

$$|R_d^*| \geq k\,|\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} i\,|\mathsf{C}^i|$$

  dark relays (inside clouds)

- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$
- We place ≤ $2i-1$ red relays per cloud of $\mathcal{C}^i$,

  and ≤ 37/12$|R^*{}_d \cap C|$-1 red relays per cloud of $\mathcal{C}^{k+}$

- We place ≤ 2$|\mathcal{C}|$ green relays for cloud interconn

$$|A_r| + |A_g| \leq \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)\,|\mathsf{C}^i|$$

Analysis: Red and Green Relays

- An opt solution R* has at least

$$|R_d^*| \geq k|\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} i|\mathsf{C}^i|$$

  dark relays (inside clouds)
- Also, $|R_d^* \cap C| \geq 1, \quad \forall C$
- We place ≤ *2i-1* red relays per cloud of $\mathcal{C}^i$,

  and ≤ 37/12$|R^*_d \cap C|$-1 red relays per cloud of $\mathcal{C}^{k+}$
- We place ≤ 2$|\mathcal{C}|$ green relays for cloud interconn

$$|A_r| + |A_g| \leq \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathsf{C}^i| + 2|C|$$

Analysis: Red and Green Relays

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1) |\mathsf{C}^i| + 2|\mathsf{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathcal{C}^i| + 2|\mathcal{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \leq \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathcal{C}^i| + 2|\mathcal{C}|$$

$$\leq \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathcal{C}^i| \right) - |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathcal{C}^i| - \sum_{i=1}^{k-1} |\mathcal{C}^i| + 2|\mathcal{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathcal{C}^i| + 2|\mathcal{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathcal{C}^i| \right) - |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathcal{C}^i| - \sum_{i=1}^{k-1} |\mathcal{C}^i| + 2|\mathcal{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathsf{C}^i| \right) - |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathsf{C}^i| - \sum_{i=1}^{k-1} |\mathsf{C}^i| + 2|\mathsf{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i - 1) |\mathcal{C}^i| + 2|\mathcal{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathcal{C}^i| \right) - |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathcal{C}^i| - \sum_{i=1}^{k-1} |\mathcal{C}^i| + 2|\mathcal{C}|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathrm{C}^i|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathcal{C}^i| + 2|\mathcal{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathcal{C}^i| \right) - |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathcal{C}^i| - \sum_{i=1}^{k-1} |\mathcal{C}^i| + 2|\mathcal{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathcal{C}^i| \right) + |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathcal{C}^i|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathbf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1) |\mathbf{C}^i| + 2|\mathbf{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathbf{C}^i| \right) - |\mathbf{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathbf{C}^i| - \sum_{i=1}^{k-1} |\mathbf{C}^i| + 2|\mathbf{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathbf{C}^i| \right) + |\mathbf{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1) |\mathbf{C}^i|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathcal{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathcal{C}^i| + 2|\mathcal{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathcal{C}^i| \right) - |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathcal{C}^i| - \sum_{i=1}^{k-1} |\mathcal{C}^i| + 2|\mathcal{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathcal{C}^i| \right) + |\mathcal{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathcal{C}^i|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \leq \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\leq \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathsf{C}^i| \right) - |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathsf{C}^i| - \sum_{i=1}^{k-1} |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathsf{C}^i| \right) + |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathsf{C}^i|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1) |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1) |\mathrm{C}^i|$$

$$\le \frac{37}{12} |R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1) |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathrm{C}^i| \right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1) |\mathrm{C}^i|$$

$$\le \frac{37}{12} |R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathrm{C}^i|$$

$$\le \frac{37}{12}|R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathsf{C}^i| \right) - |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathsf{C}^i| - \sum_{i=1}^{k-1} |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\frac{37}{12} i > (2i+1)$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathsf{C}^i| \right) + |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathsf{C}^i|$$

$$\le \frac{37}{12}|R_d^*| + |\mathsf{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathsf{C}^i| \right) - |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathsf{C}^i| - \sum_{i=1}^{k-1} |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$= \frac{37}{12} \left(|R_d^*| \qquad\qquad \right. + |\mathsf{C}^{k+}| + \qquad\qquad$$

$$\le \frac{37}{12} |R_d^*| + |\mathsf{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12}\Big(|R_d^*| \qquad\qquad + |\mathrm{C}^{k+}| + \qquad\qquad$$

$$\le \frac{37}{12}|R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k}\right)|R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathsf{C}^i| \right) - |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathsf{C}^i| - \sum_{i=1}^{k-1} |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$= \frac{37}{12} \left(|R_d^*| \qquad \qquad \right. + |\mathsf{C}^{k+}| + \qquad \qquad$$

$$\le \frac{37}{12} |R_d^*| + |\mathsf{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathsf{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1) |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathsf{C}^i| \right) - |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} 2i |\mathsf{C}^i| - \sum_{i=1}^{k-1} |\mathsf{C}^i| + 2|\mathsf{C}|$$

$$= \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i |\mathsf{C}^i| \right) + |\mathsf{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1) |\mathsf{C}^i|$$

$$\le \frac{37}{12} |R_d^*| + |\mathsf{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathrm{C}^i|$$

$$\le \frac{37}{12} |R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r| + |A_g| \le \sum_{C \in \mathrm{C}^{k+}} \left(\frac{37}{12} |R_d^* \cap C| - 1 \right) + \sum_{i=1}^{k-1} (2i-1)|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathrm{C}^i| - \sum_{i=1}^{k-1} |\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12} \left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i| \right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} (2i+1)|\mathrm{C}^i|$$

$$\le \frac{37}{12}|R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k} \right) |R_d^*|$$

Analysis: Red and Green Relays

$$|A_r|+|A_g| \le \sum_{C\in \mathrm{C}^{k+}} \left(\frac{37}{12}|R_d^* \cap C|-1\right) + \sum_{i=1}^{k-1}(2i-1)|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$\le \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i|\right) - |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1} 2i|\mathrm{C}^i| - \sum_{i=1}^{k-1}|\mathrm{C}^i| + 2|\mathrm{C}|$$

$$= \frac{37}{12}\left(|R_d^*| - \sum_{i=1}^{k-1} i|\mathrm{C}^i|\right) + |\mathrm{C}^{k+}| + \sum_{i=1}^{k-1}(2i+1)|\mathrm{C}^i|$$

$$\le \frac{37}{12}|R_d^*| + |\mathrm{C}^{k+}| < \left(3.084 + \frac{1}{k}\right)|R_d^*|$$

Analysis: Yellow Relays

Analysis: Yellow Relays

Lemma 4: ∃ spanning forest with neighborhoods on clusters requiring

Analysis: Yellow Relays

Lemma 4: ∃ spanning forest with neighborhoods on clusters requiring

$$|A_y| \le \left(\frac{4}{\sqrt{3}} + \frac{4}{5}\right)|R_w^*| < 3.11\,|R_w^*|$$

yellow relays.

Analysis: Yellow Relays

Lemma 4: ∃ spanning forest with neighborhoods on clusters requiring

$$|A_y| \le \left(\frac{4}{\sqrt{3}} + \frac{4}{5}\right)|R_w^*| < 3.11\,|R_w^*|$$

yellow relays.

Proof: Omitted.

(Combines Steiner tree ratio with degree arguments.).

Summary: 3.11-Approximation

Summary: 3.11-Approximation

Approximation bound:

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \dot{\cup} R_l^*$$

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \,\dot{\cup}\, R_l^*$$

dark

Within clouds

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \,\dot{\cup}\, R_l^*$$

dark light

Within clouds Outside clouds

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \dot{\cup} R_l^*$$

dark light

Within clouds Outside clouds

We showed:

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark — Within clouds

light — Outside clouds

We showed:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark — Within clouds

light — Outside clouds

We showed:

$$|A_r| + |A_g| \leq (3.084 + 1/k)\,|R_d^*|$$

$$|A_y| \leq 3.11\,|R_l^*|$$

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \,\dot\cup\, R_l^*$$

dark — Within clouds

light — Outside clouds

We showed:

$$|A_r| + |A_g| \le (3.084 + 1/k)\,|R_d^*|$$

$$|A_y| \le 3.11\,|R_l^*|$$

Theorem: Total # of relays we compute in poly time

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \cup R_l^*$$

dark — Within clouds

light — Outside clouds

We showed:

$$|A_r| + |A_g| \le (3.084 + 1/k)|R_d^*|$$

$$|A_y| \le 3.11|R_l^*|$$

Theorem: Total # of relays we compute in poly time

$$|A| = |A_r| + |A_g| + |A_y| \le 3.11(|R_d^*| + |R_l^*|) = 3.11|R^*|$$

Summary: 3.11-Approximation

Approximation bound:

$$R^* = R_d^* \dot{\cup} R_l^*$$

dark — Within clouds

light — Outside clouds

We showed:

$$|A_r| + |A_g| \leq (3.084 + 1/k)|R_d^*|$$

$$|A_y| \leq 3.11|R_l^*|$$

Theorem: Total # of relays we compute in poly time

$$|A| = |A_r| + |A_g| + |A_y| \leq 3.11(|R_d^*| + |R_l^*|) = 3.11|R^*|$$

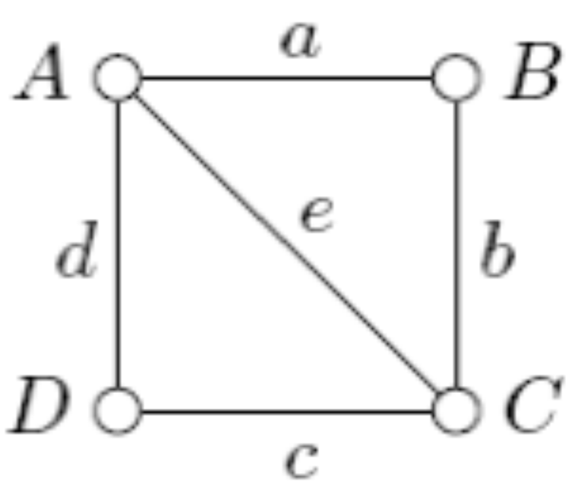
Inapproximability

Inapproximability

- From Vertex Cover in graphs of max-deg 3

Inapproximability

- From Vertex Cover in graphs of max-deg 3

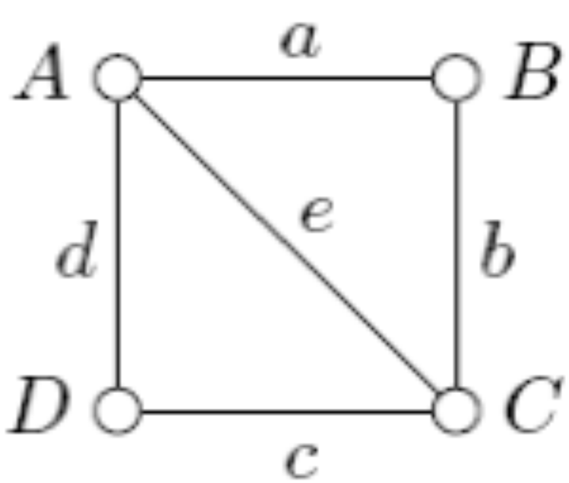


$n = 4$ vertices

$m = 5$ edges

Inapproximability

- From Vertex Cover in graphs of max-deg 3



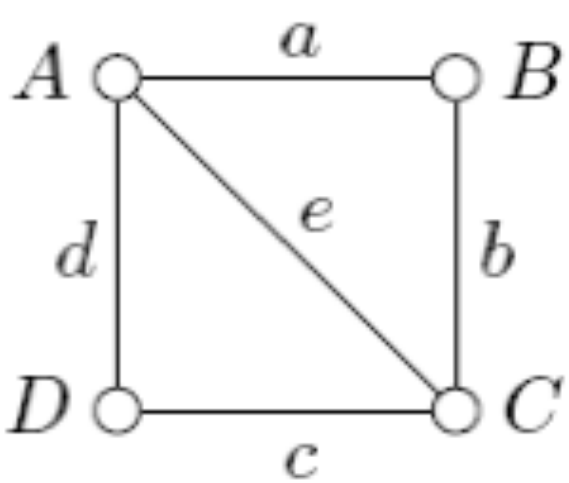
Note: r is part of the input!

$n = 4$ vertices

$m = 5$ edges

Inapproximability

- From Vertex Cover in graphs of max-deg 3

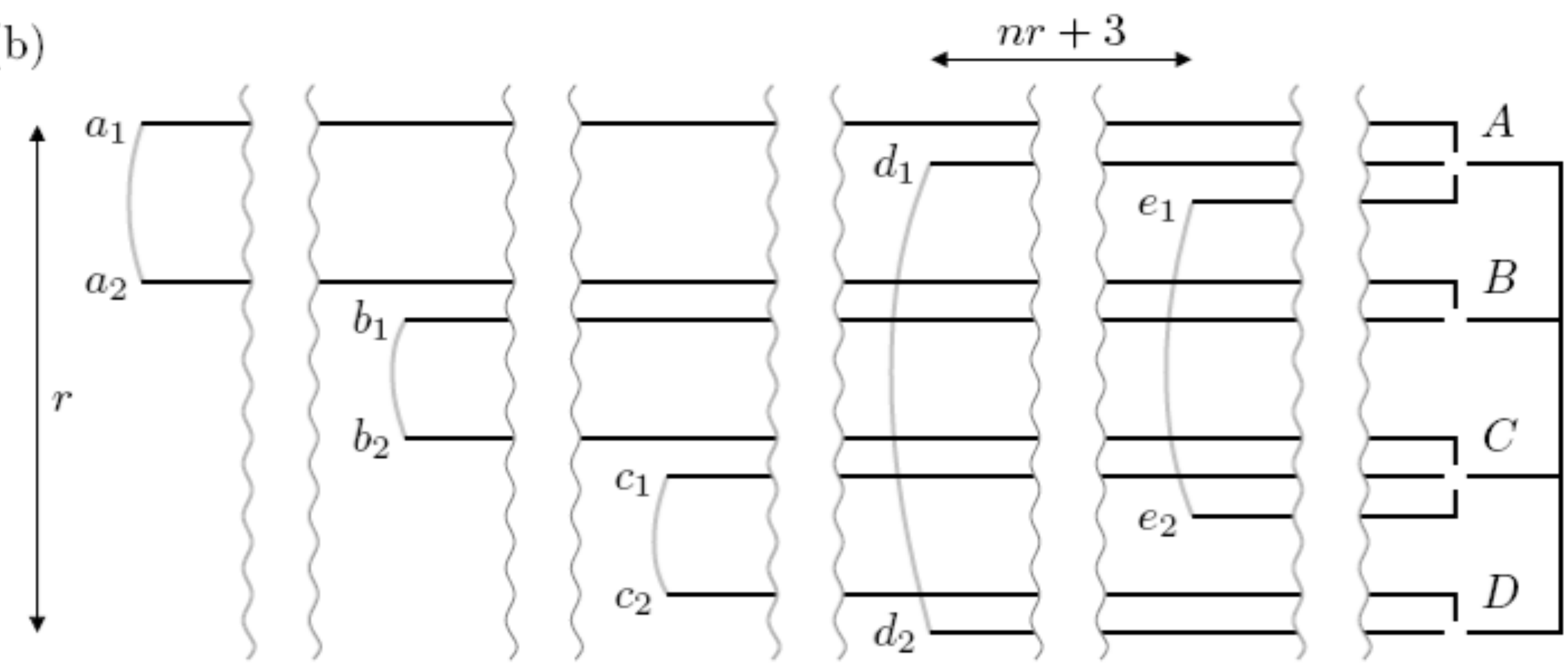


Note: r is part of the input!

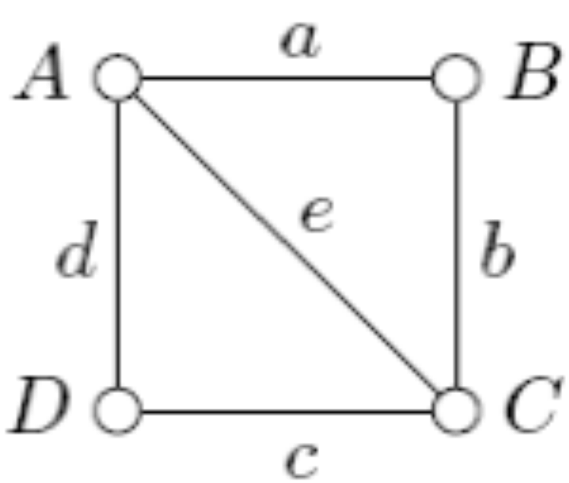
$n = 4$ vertices

$m = 5$ edges

(b)



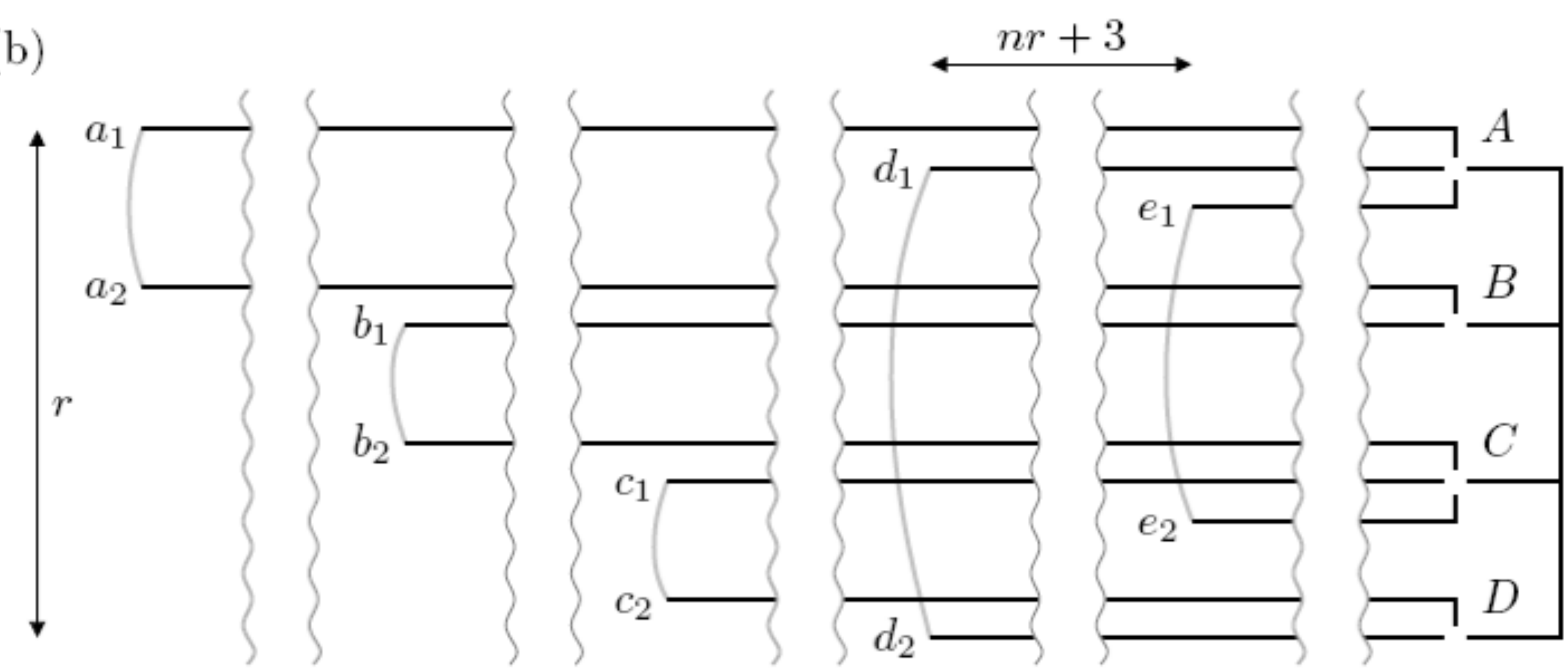
Inapproximability



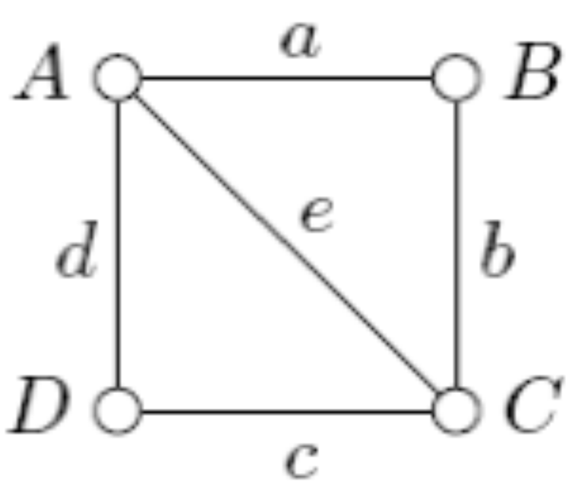
$n = 4$ vertices

$m = 5$ edges

(b)



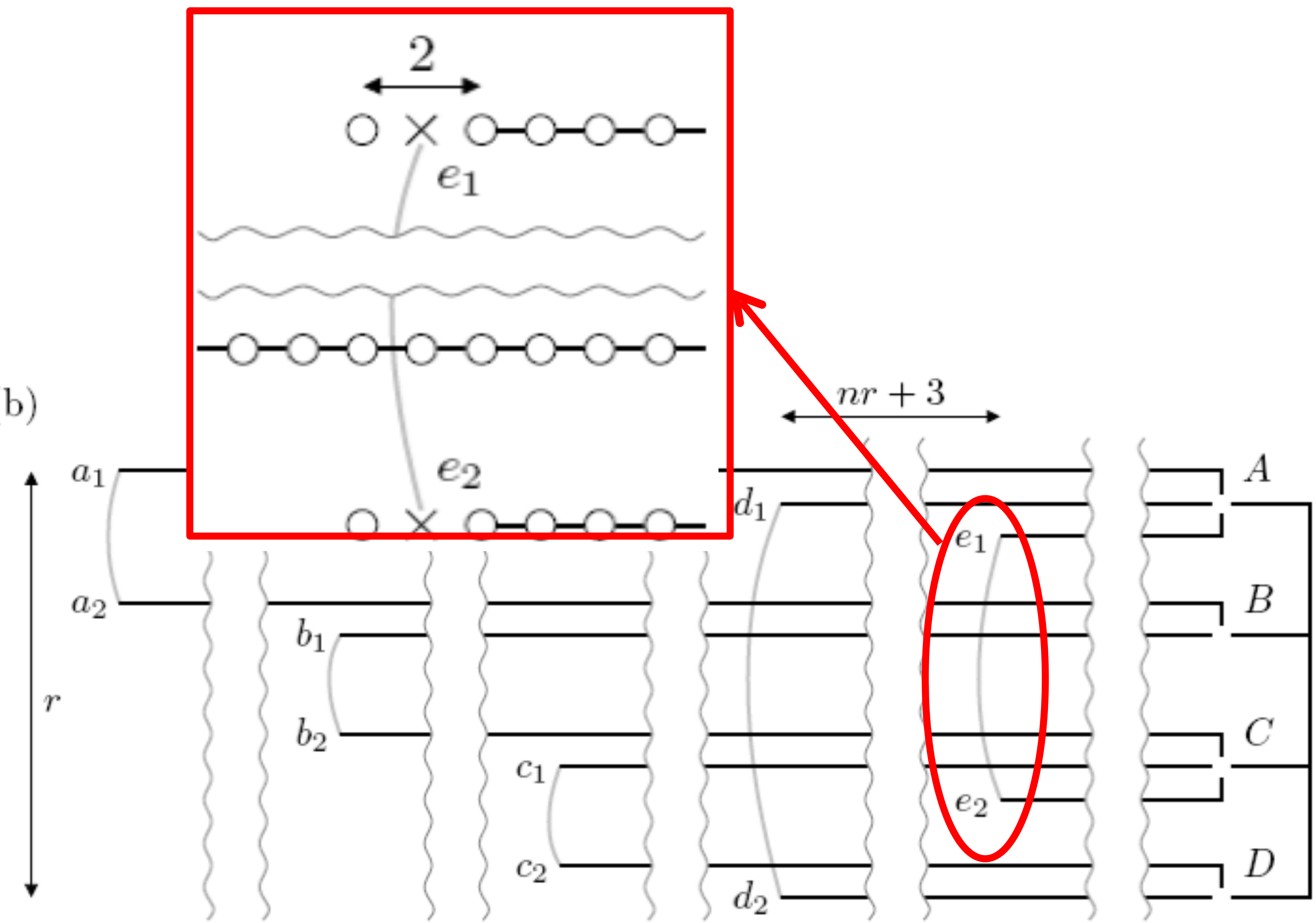
Inapproximability



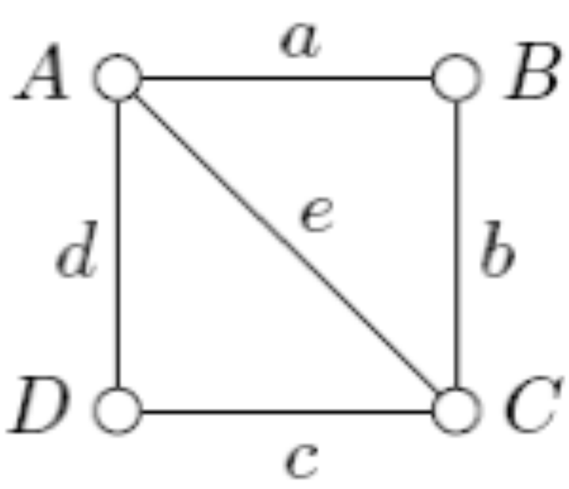
$n = 4$ vertices

$m = 5$ edges

(b)

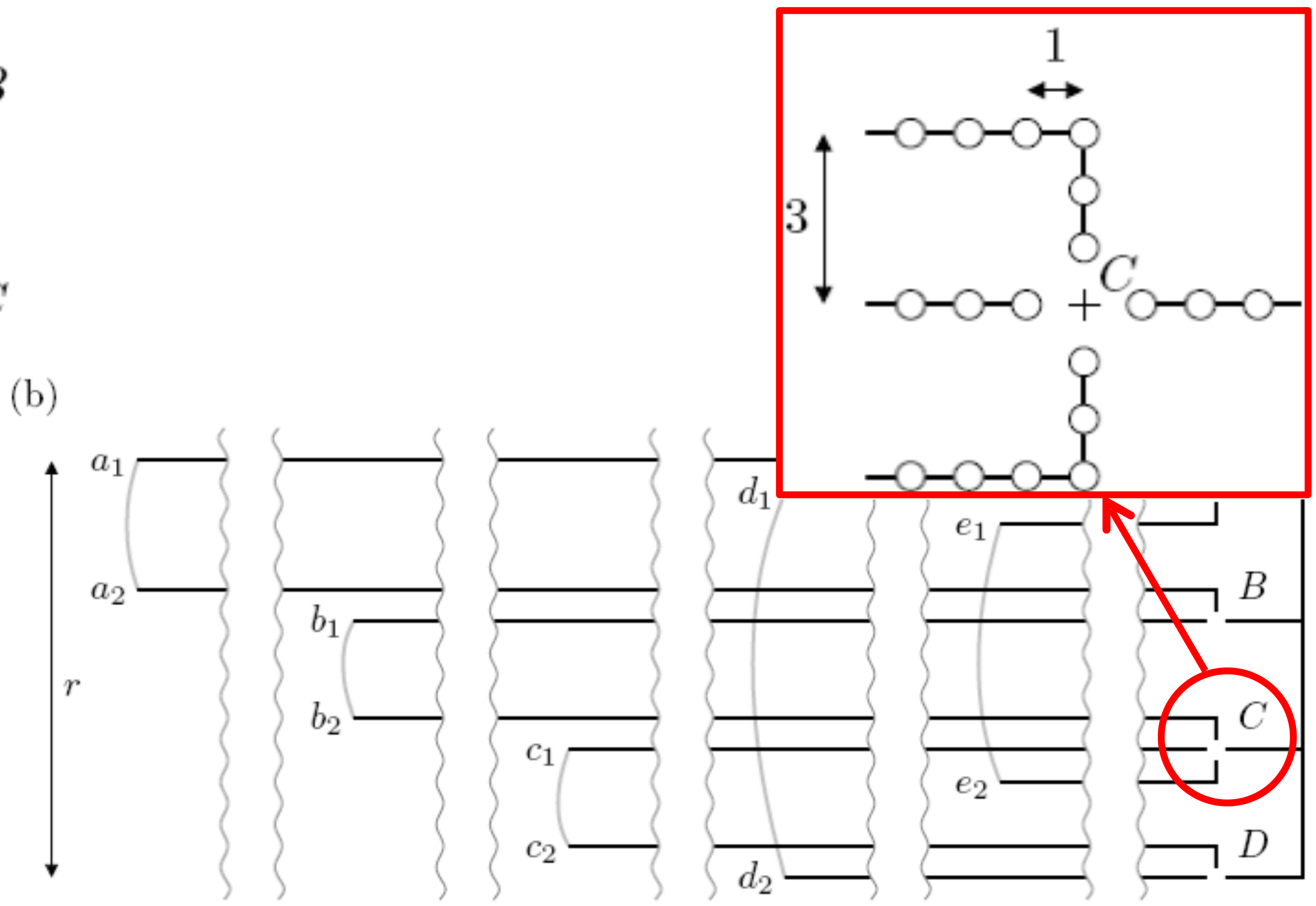


Inapproximability



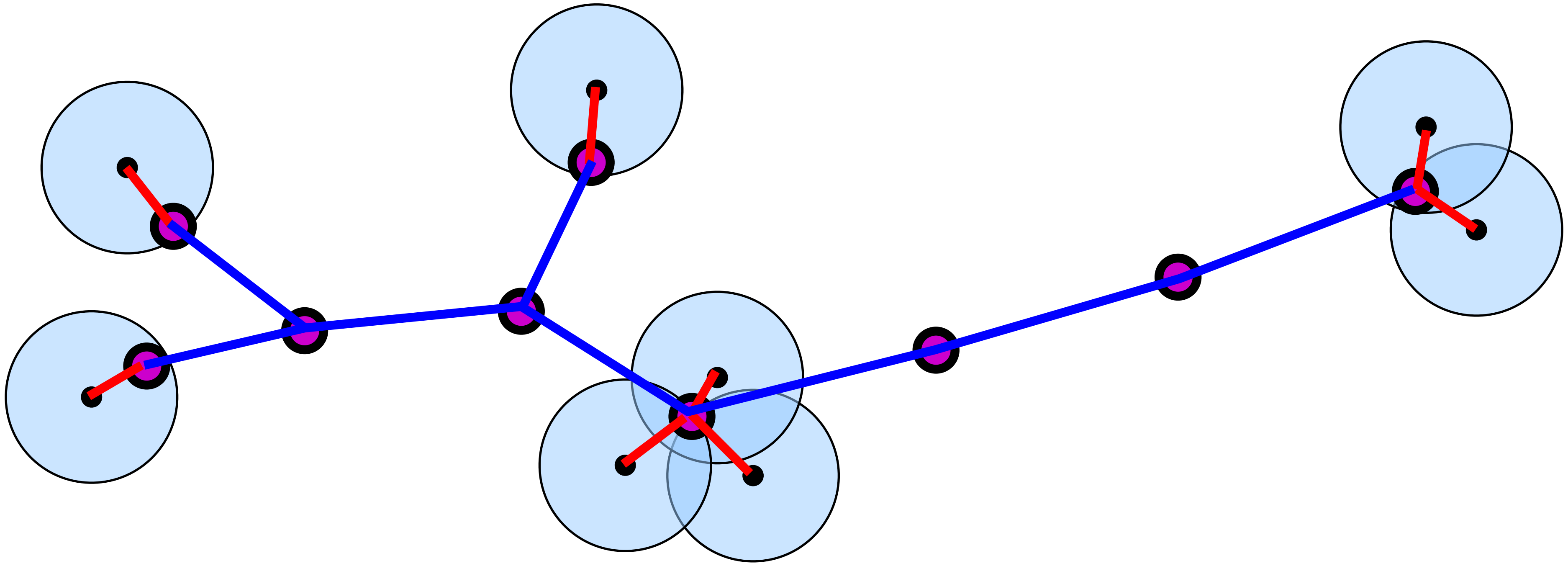
$n = 4$ vertices

$m = 5$ edges



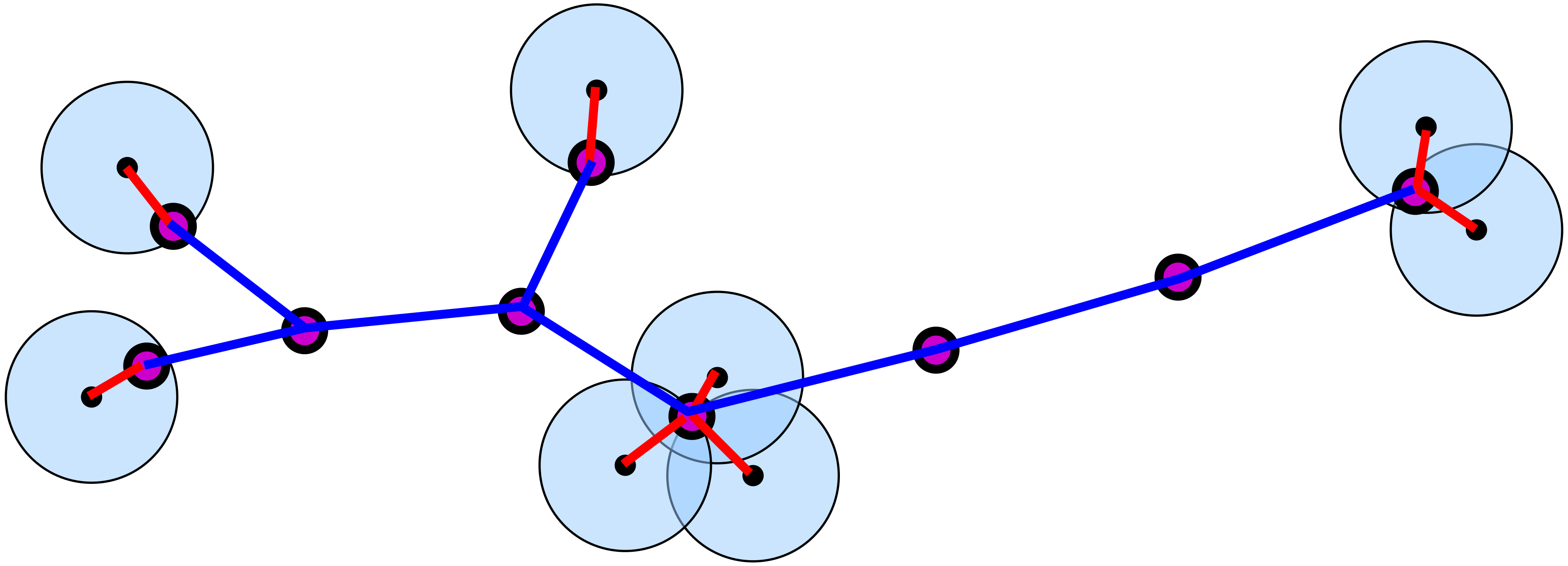
PTAS: Two-Tier Relay Placement

PTAS: Two-Tier Relay Placement



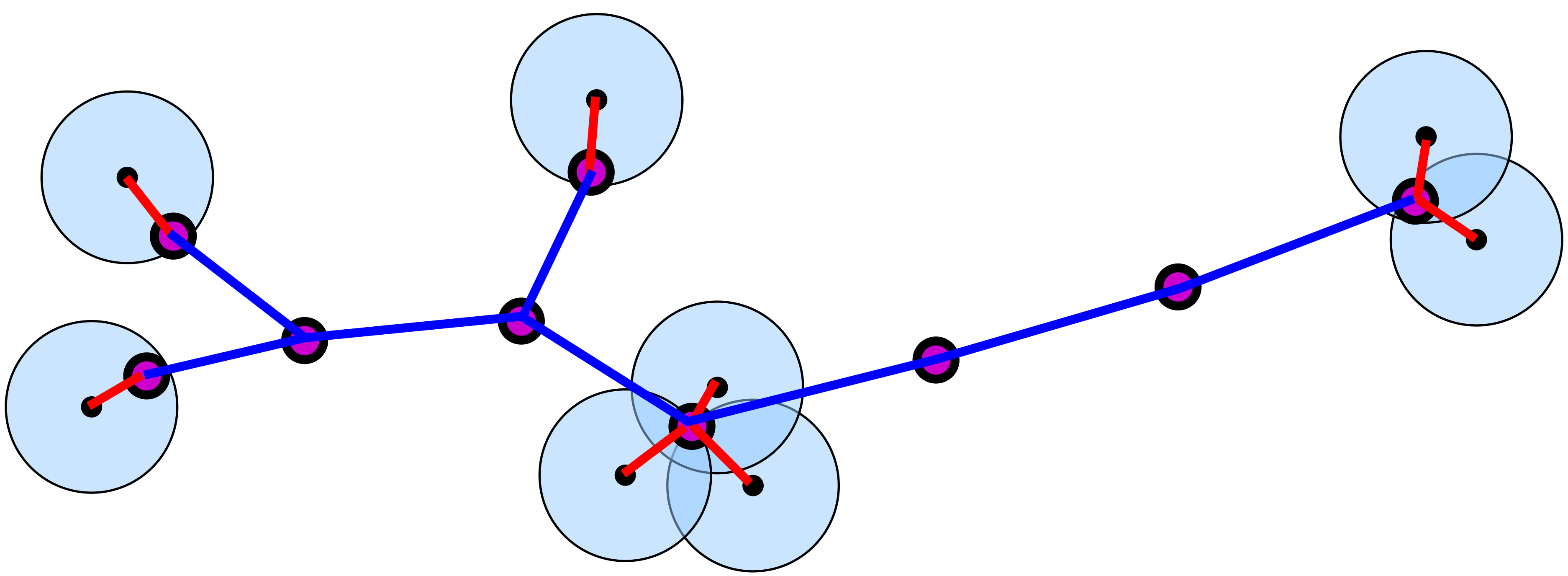
PTAS: Two-Tier Relay Placement

- Steiner spanning tree



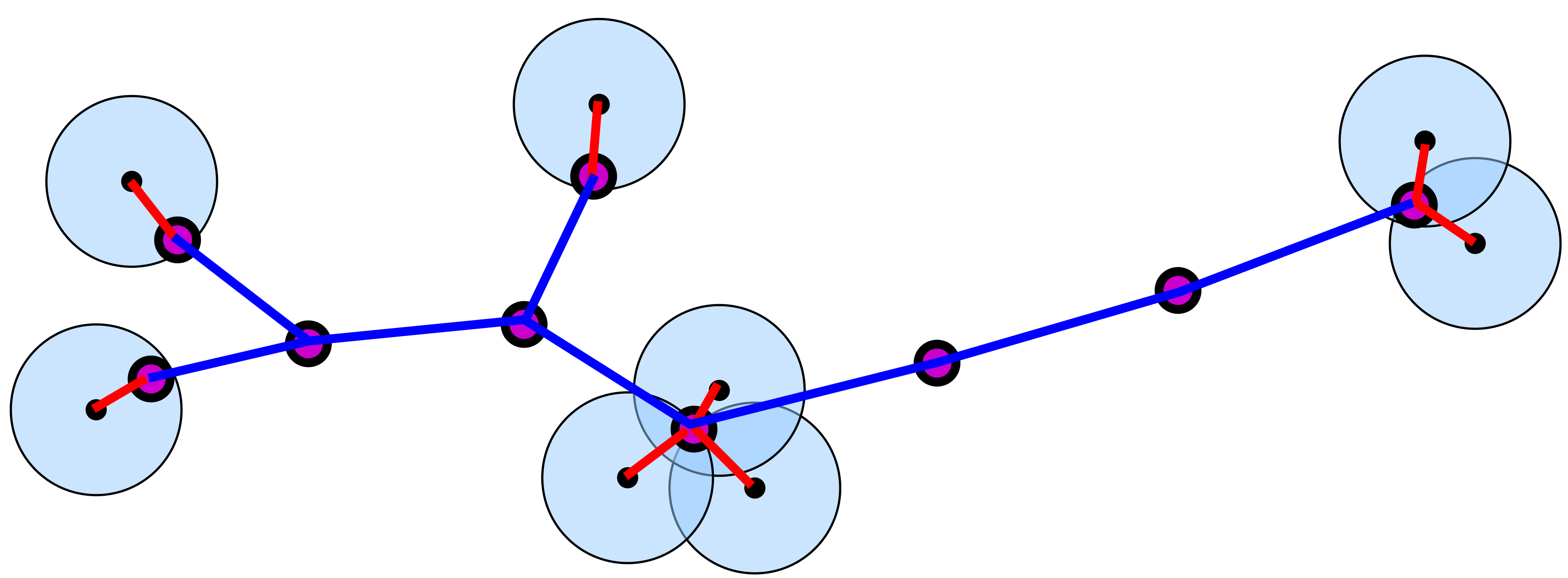
PTAS: Two-Tier Relay Placement

- Steiner spanning tree
- Edges of length ≤ r between relays, ≤ 1 between a relay and a sensor



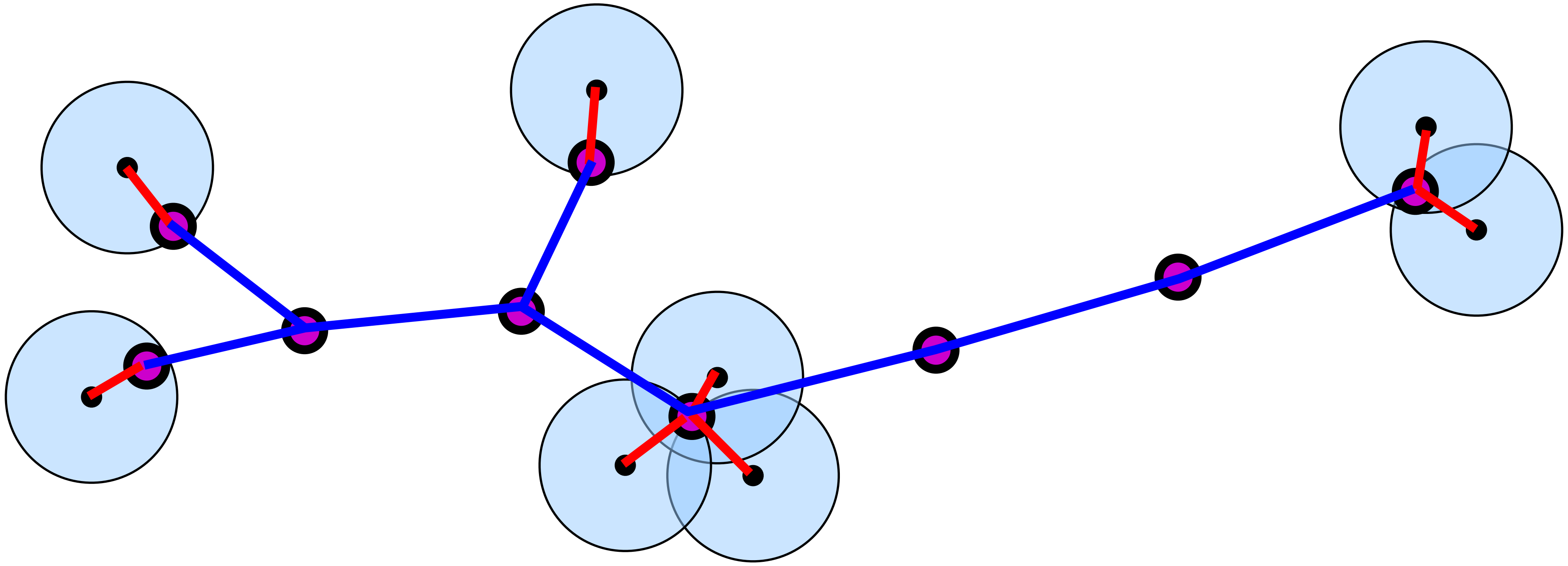
PTAS: Two-Tier Relay Placement

- Steiner spanning tree
- Edges of length ≤ r between relays, ≤ 1 between a relay and a sensor
- A sensor has degree 1 (cannot relay data)



PTAS: Two-Tier Relay Placement

- Steiner spanning tree
- Edges of length ≤ r between relays, ≤ 1 between a relay and a sensor
- A sensor has degree 1 (cannot relay data)
- Goal: Min # Steiner points (relays)



PTAS: Outline

PTAS: Outline

- Spanning tree has red edges (incident on sensors) and blue edges (between relays)

PTAS: Outline

- Spanning tree has red edges (incident on sensors) and blue edges (between relays)
- Round OPT to a poly-size grid, G, of candidate locations for relays

PTAS: Outline

- Spanning tree has red edges (incident on sensors) and blue edges (between relays)
- Round OPT to a poly-size grid, G, of candidate locations for relays
- Set of blue edges of OPT can be made m-guillotine, increasing length by only factor $(1+\varepsilon)$ by the added bridges

PTAS: Outline

- Spanning tree has red edges (incident on sensors) and blue edges (between relays)
- Round OPT to a poly-size grid, G, of candidate locations for relays
- Set of blue edges of OPT can be made m-guillotine, increasing length by only factor $(1+\varepsilon)$ by the added bridges
- Bridges can be replaced by a set of relays

PTAS: Outline

- Spanning tree has red edges (incident on sensors) and blue edges (between relays)
- Round OPT to a poly-size grid, G, of candidate locations for relays
- Set of blue edges of OPT can be made m-guillotine, increasing length by only factor $(1+\varepsilon)$ by the added bridges
- Bridges can be replaced by a set of relays
- Optimize over all m-guillotine spanning trees using dynamic programming

Summary

Summary

- Simple O($n\ log\ n$)-time 6.73-approx for one-tier relay placement

Summary

- Simple O($n\ log\ n$)-time 6.73-approx for one-tier relay placement
- Poly-time 3.11-approx for one-tier, any $r \geq 1$

Summary

- Simple O($n\ log\ n$)-time 6.73-approx for one-tier relay placement
- Poly-time 3.11-approx for one-tier, any $r \geq 1$
- No PTAS for one-tier (r is part of input), assuming P $\neq$ NP

Summary

- Simple $O(n \log n)$-time 6.73-approx for one-tier relay placement
- Poly-time 3.11-approx for one-tier, any $r \geq 1$
- No PTAS for one-tier (r is part of input), assuming P $\neq$ NP
- PTAS for two-tier relay placement

Additional Remarks

Additional Remarks

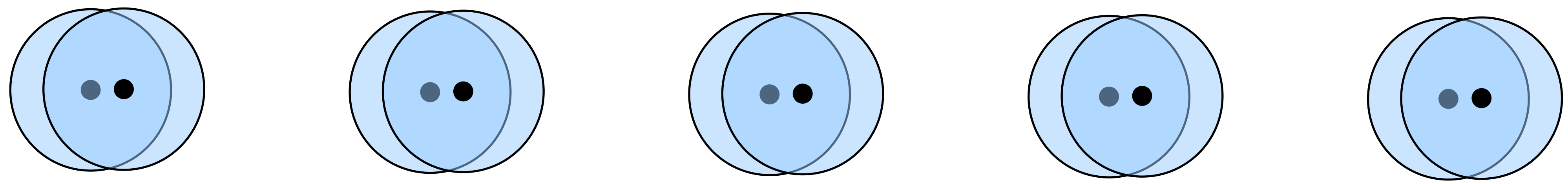
- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16

Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:

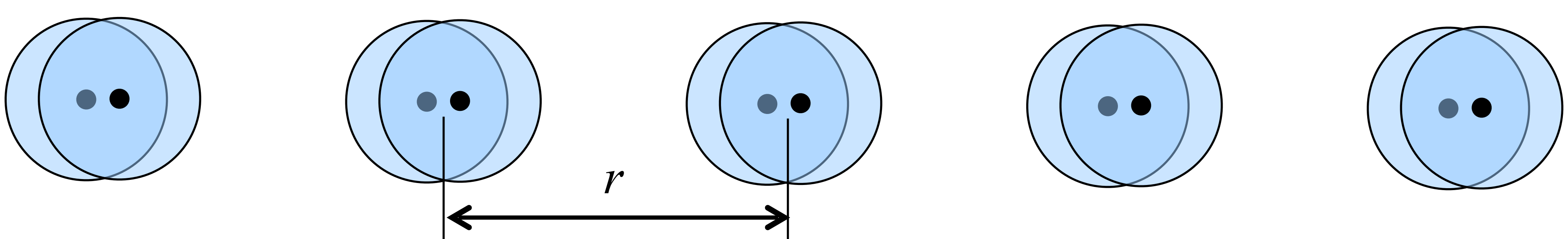
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



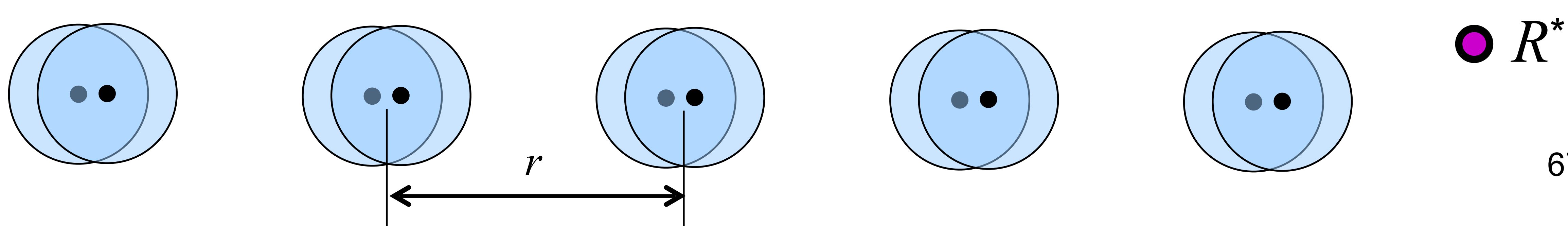
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



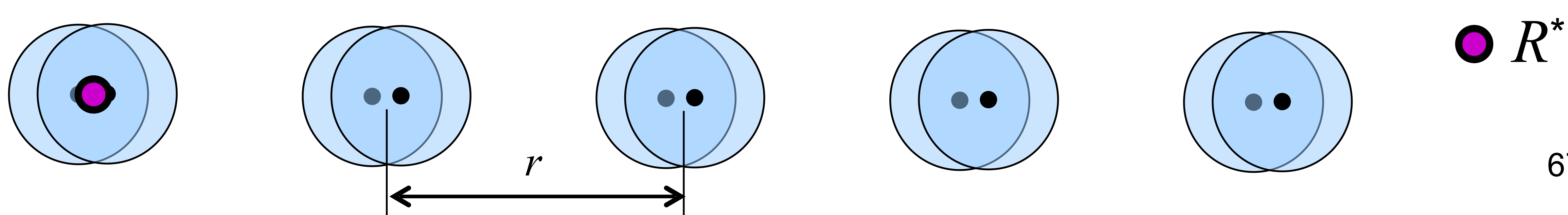
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



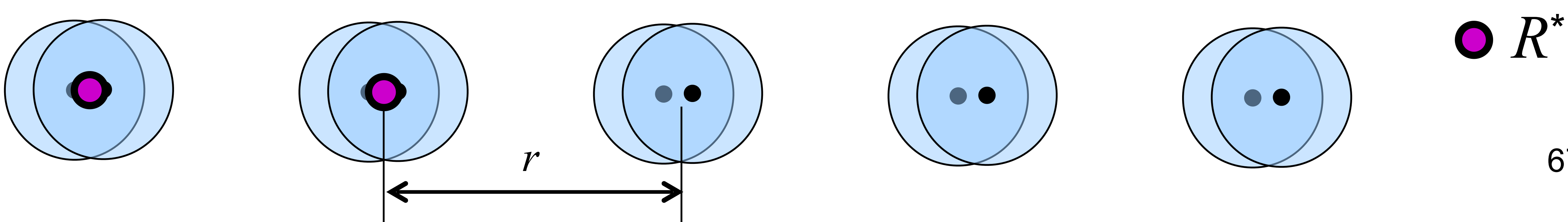
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



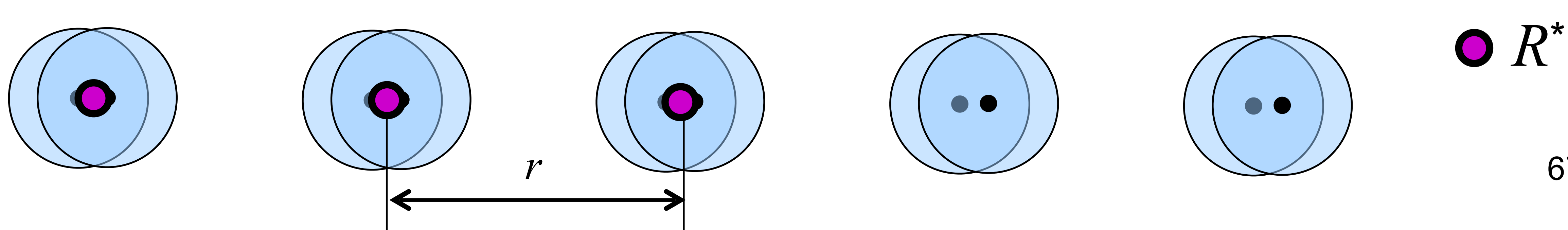
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



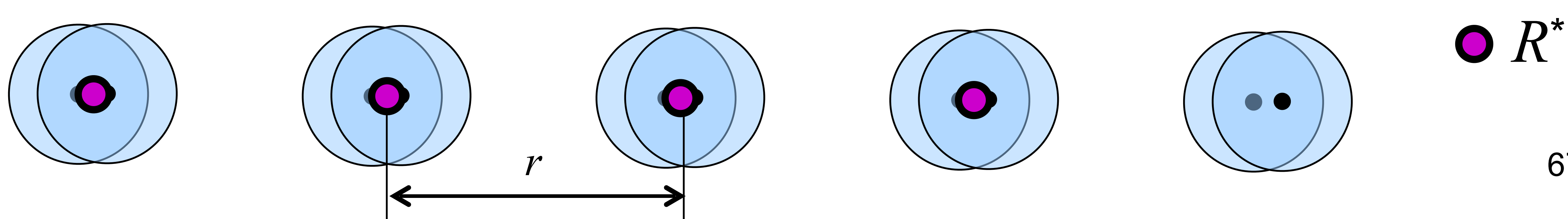
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



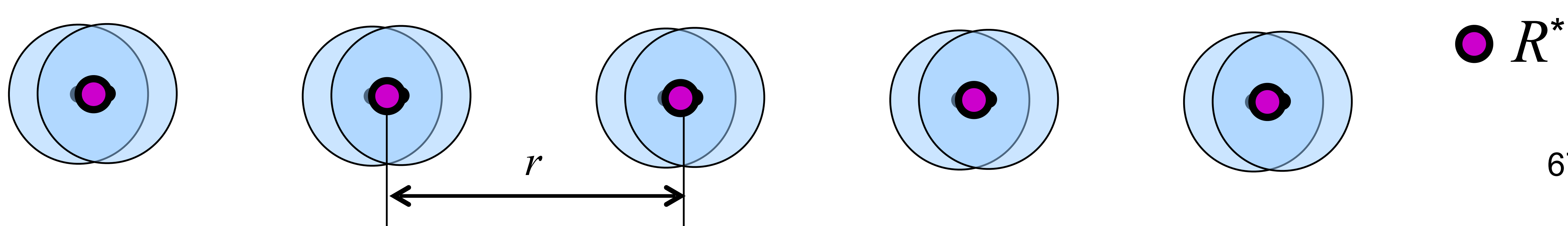
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



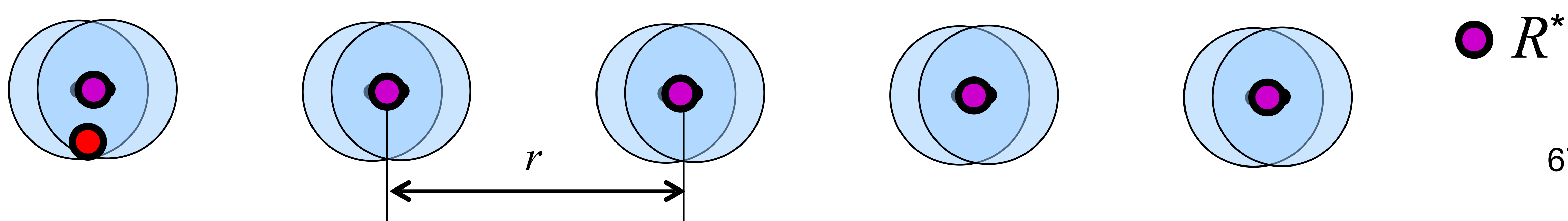
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



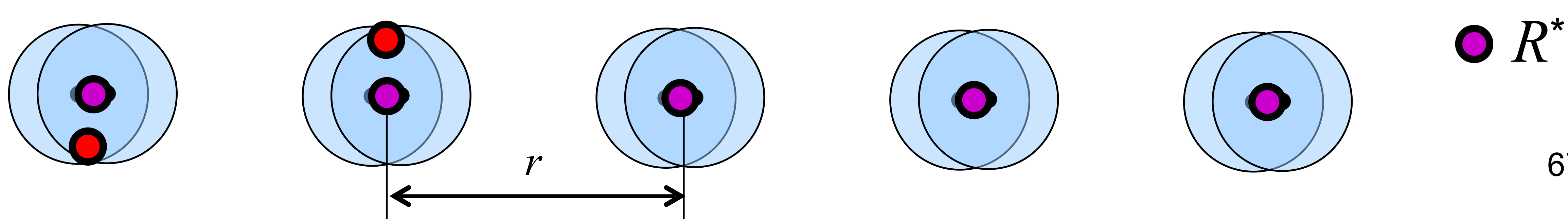
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



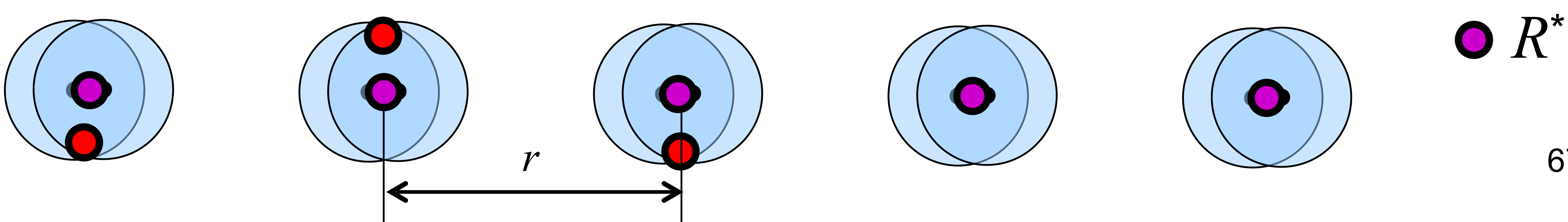
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



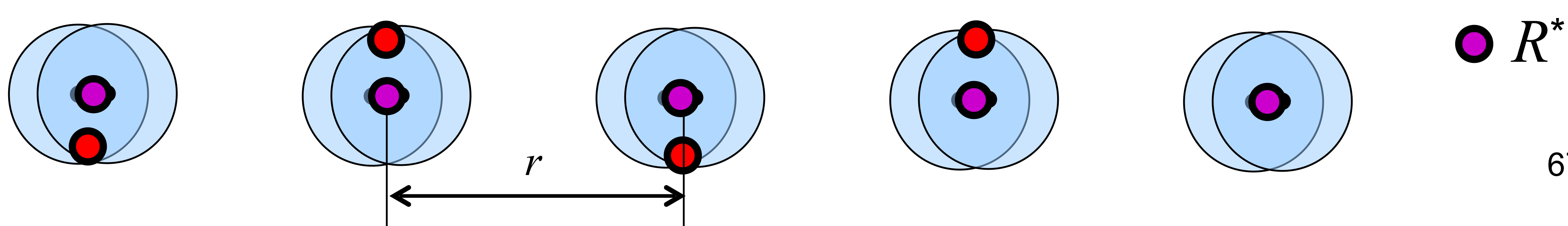
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



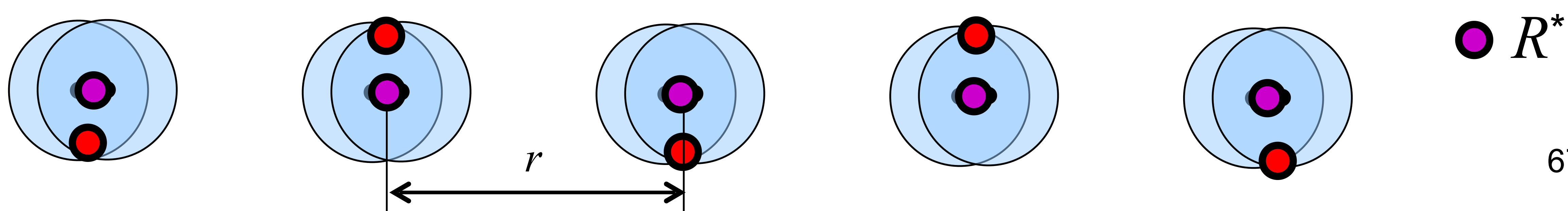
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



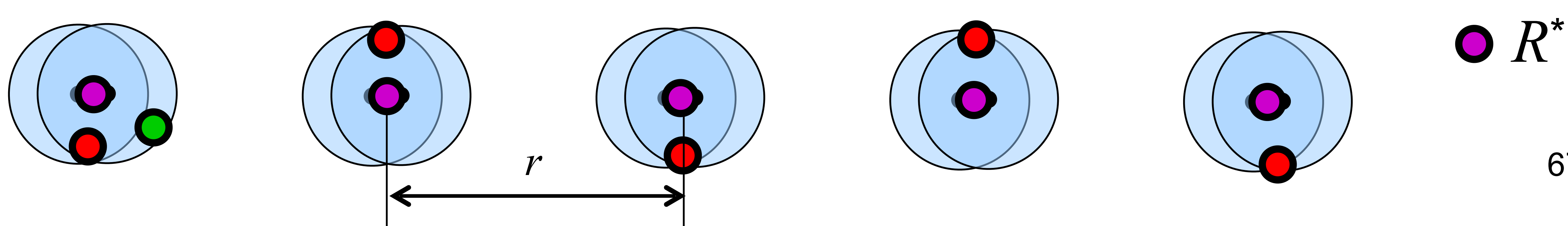
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  – Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  – With a PTAS for PRBDSP we get 3+ε.
  – 3-approx is best possible analysis for our method:



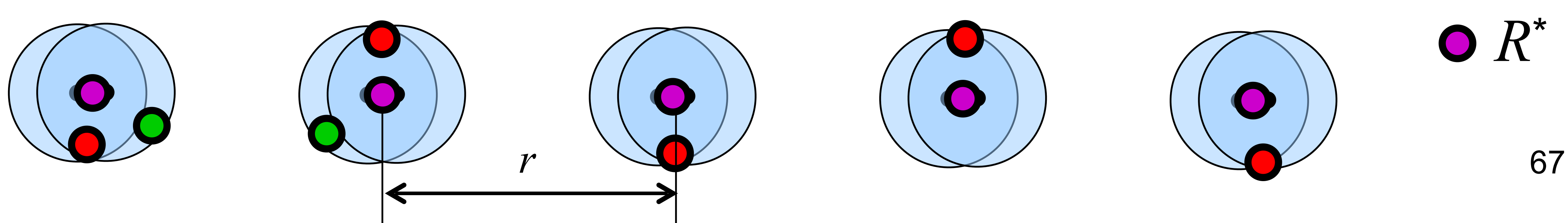
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



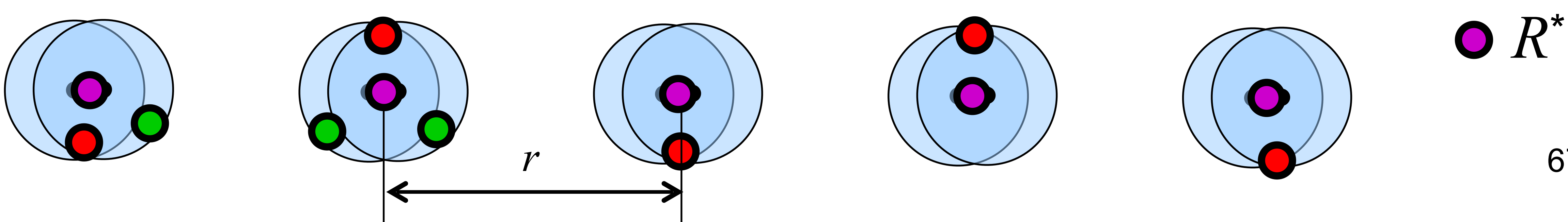
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



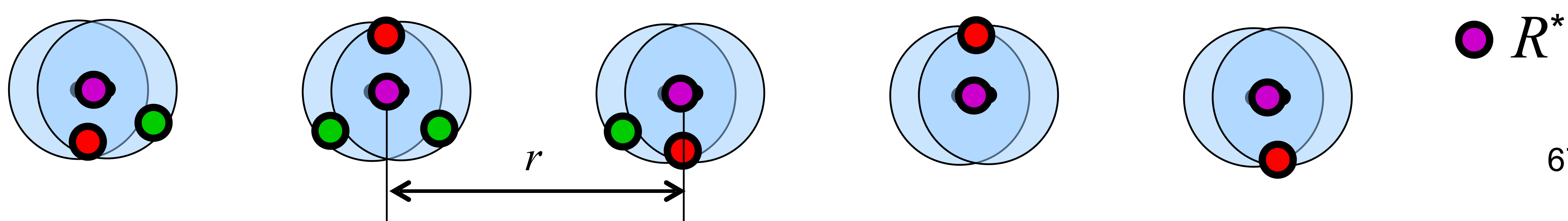
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



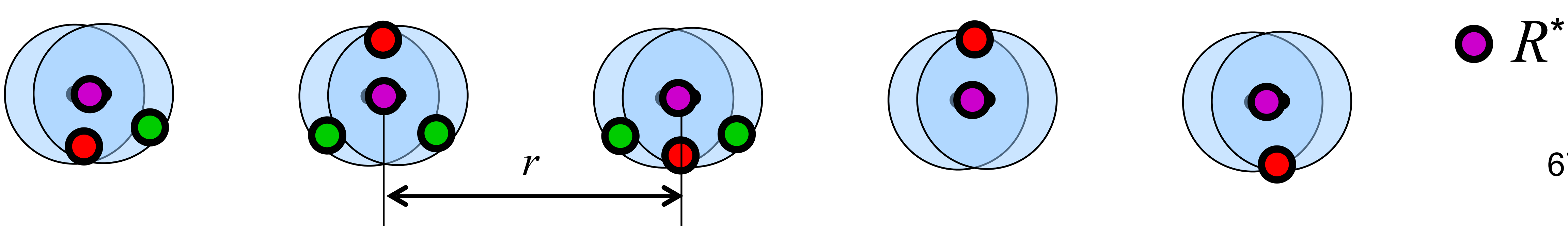
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



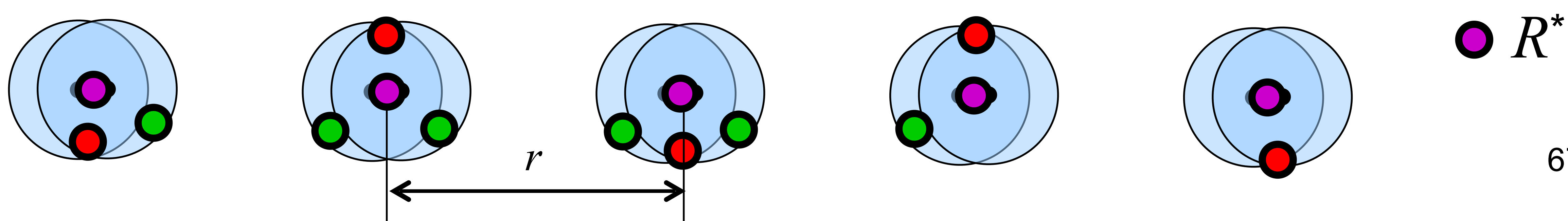
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



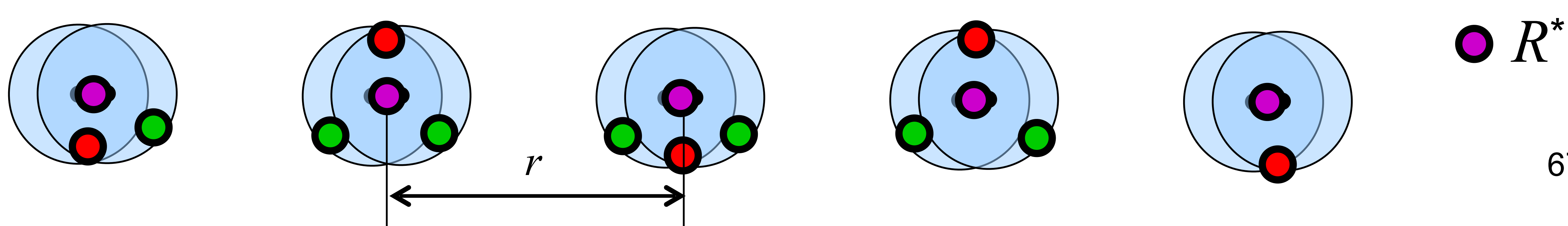
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



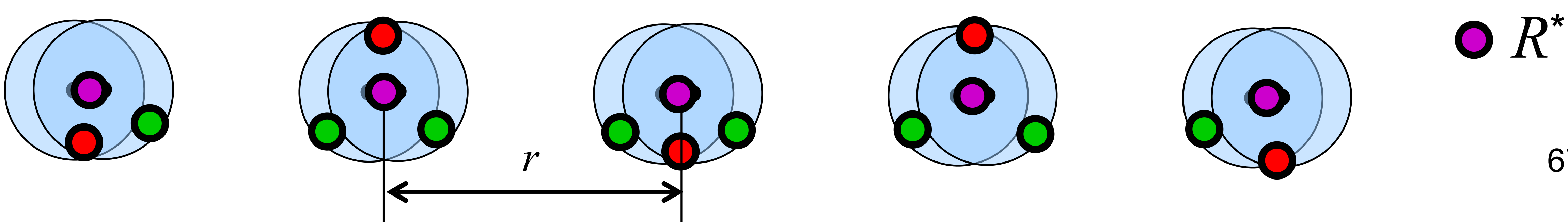
Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



Additional Remarks

- We believe we can give a PTAS for PRBDSP (Planar Red/Blue Dominating Set Problem)
  - Then 6.73 becomes 4.16
- More involved geometric argument: Lemma 4 improves (3 instead of 3.11), resulting in 3.09-approx
  - With a PTAS for PRBDSP we get 3+ε.
  - 3-approx is best possible analysis for our method:



Concluding Remarks

Concluding Remarks

- Is there a PTAS for *constant* values of r ?
  - Our hardness of approximation relies on large r.
  - Problem is related to TSP/MST with Neighborhoods.

Concluding Remarks

- Is there a PTAS for *constant* values of r ?
  - Our hardness of approximation relies on large r.
  - Problem is related to TSP/MST with Neighborhoods.
- Fault tolerance: k-connectivity
  - [Bredin, Demaine, Hajiaghayi, Rus, MobiHoc'05, Zhang, Xue, Misra]

1. Introduction
2. Review
3. Extra Packing: Dispersion
4. Extra Tours: Lawn Mowing
5. Relay Placement
6. Coordinated Motion Planning

Technische Universität Braunschweig

Video!

Technische Universität Braunschweig

Thank you!