Karlsruhe Institute of Technology

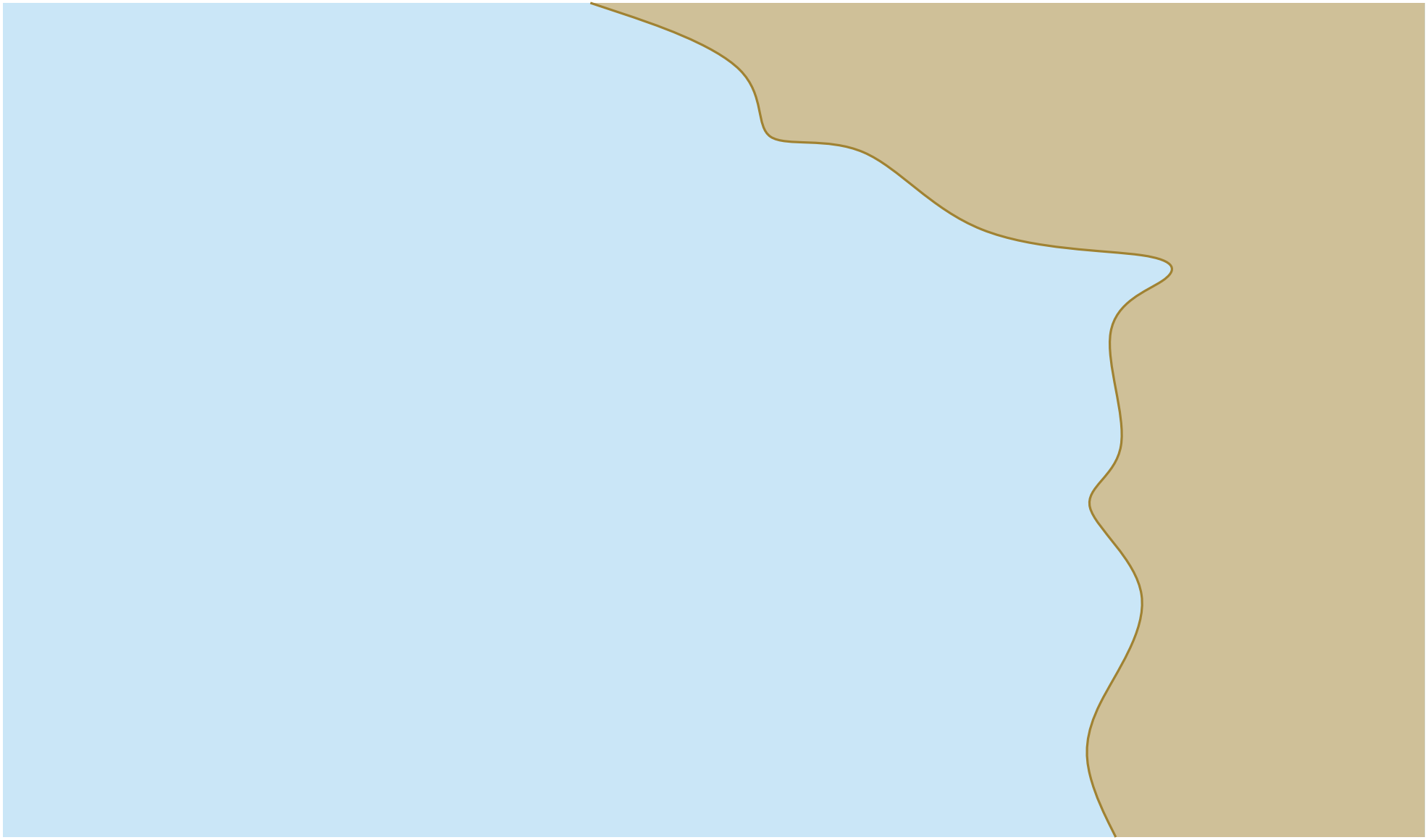
Cabling Optimization in a Wind Farm
Heuristics Based on Simulated Annealing

Master's Thesis · Final Presentation · May 31, 2016

Sebastian Lehmann

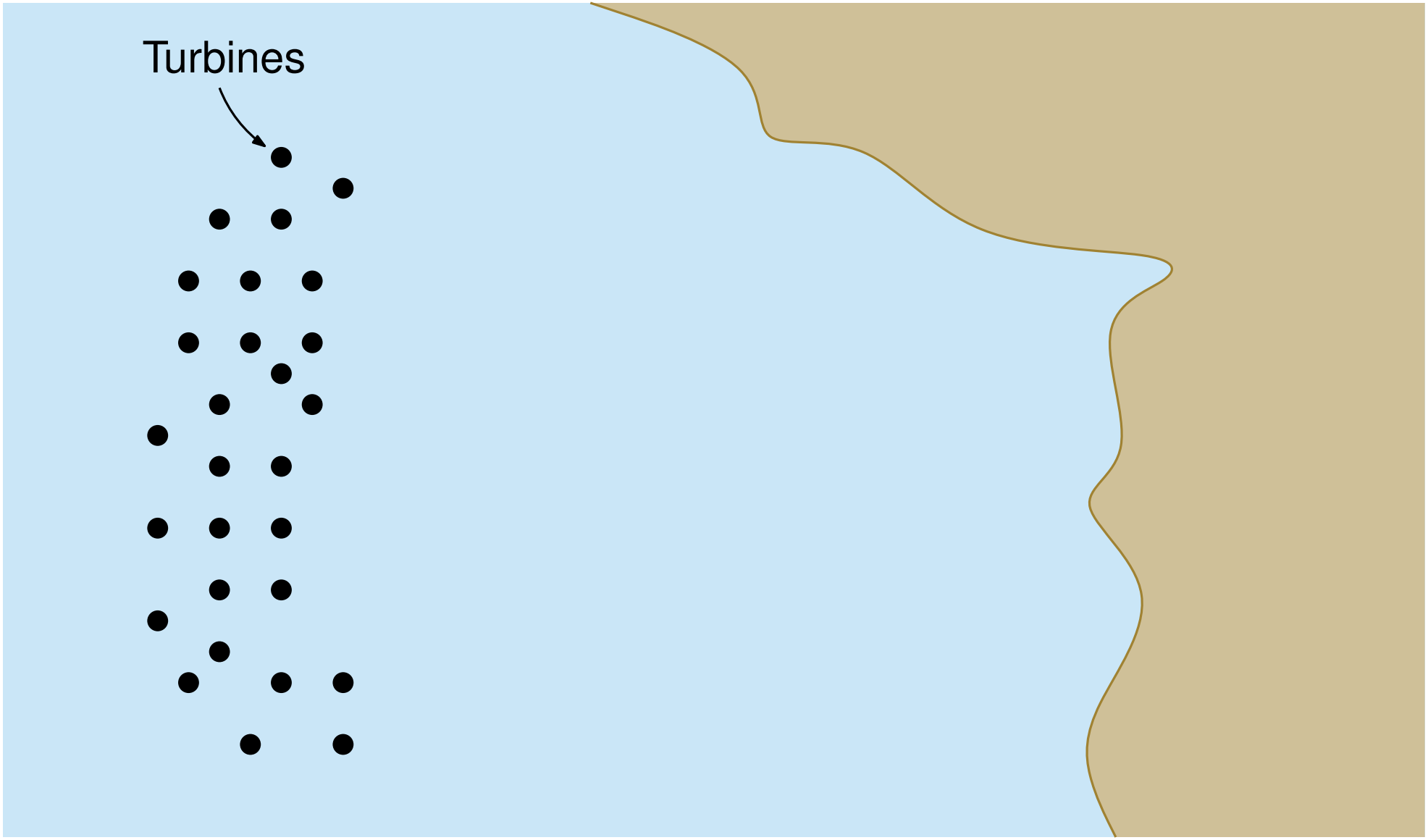
INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP

Components in a Wind Farm

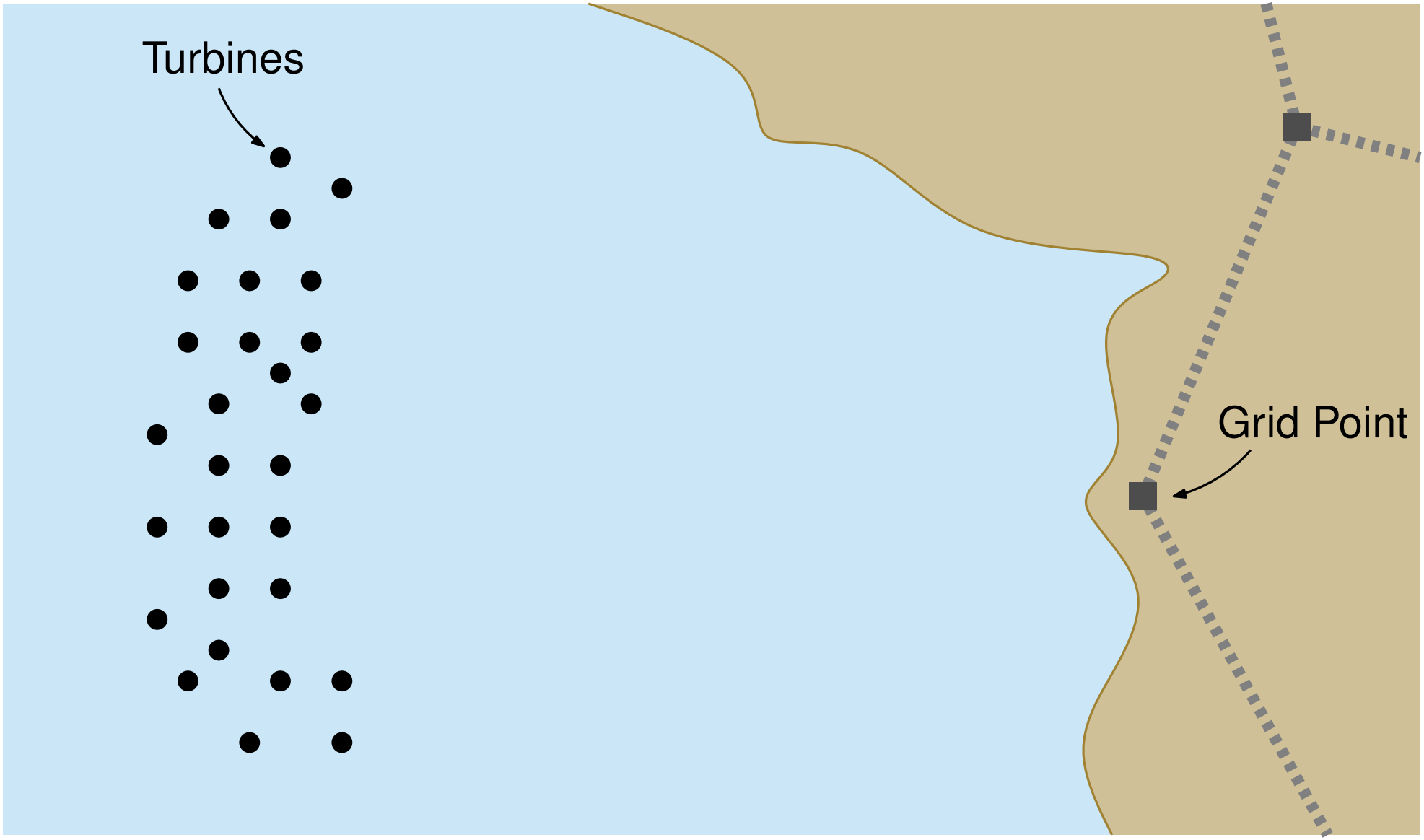


Institute of Theoretical Informatics
Algorithmics Group

Components in a Wind Farm

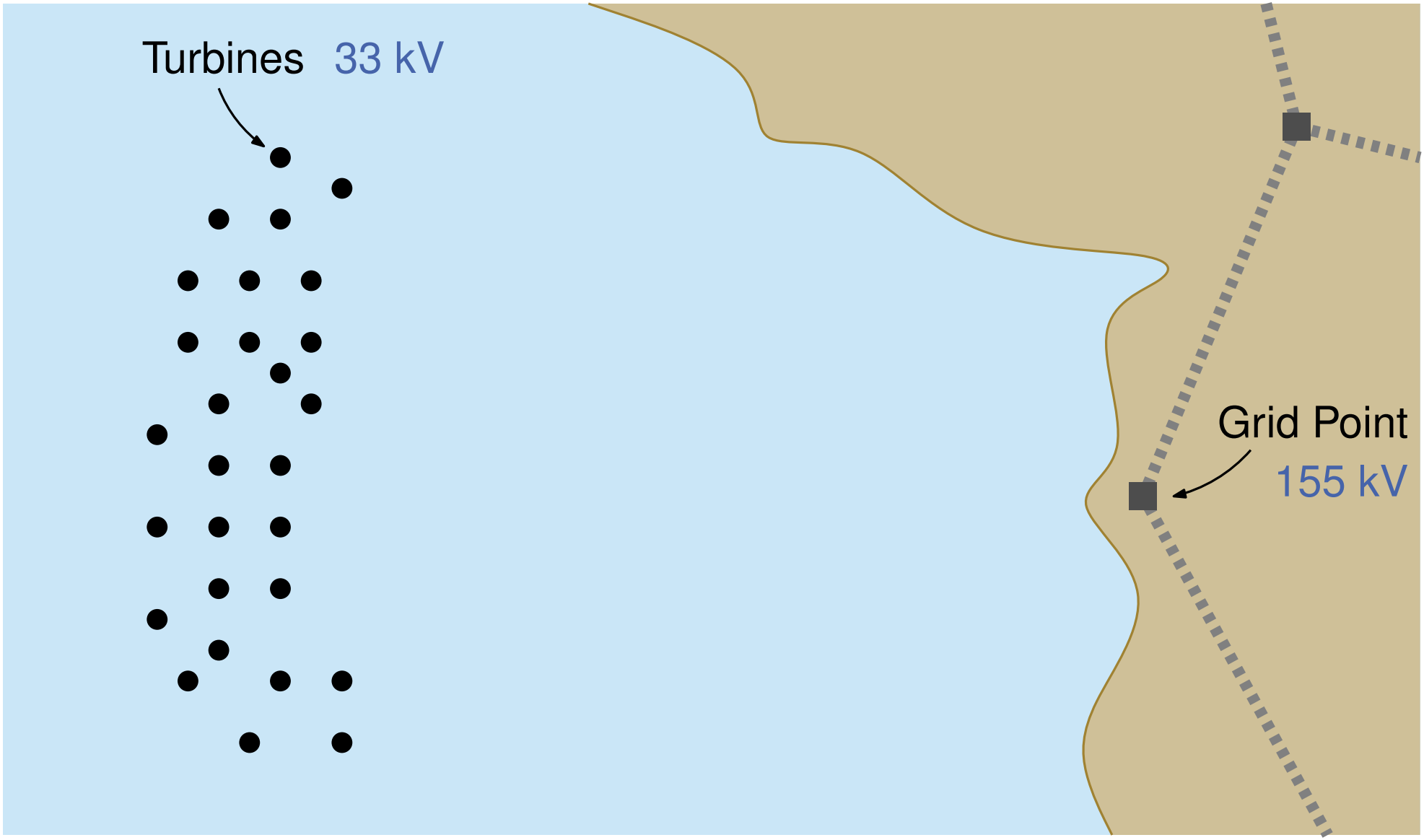


Components in a Wind Farm



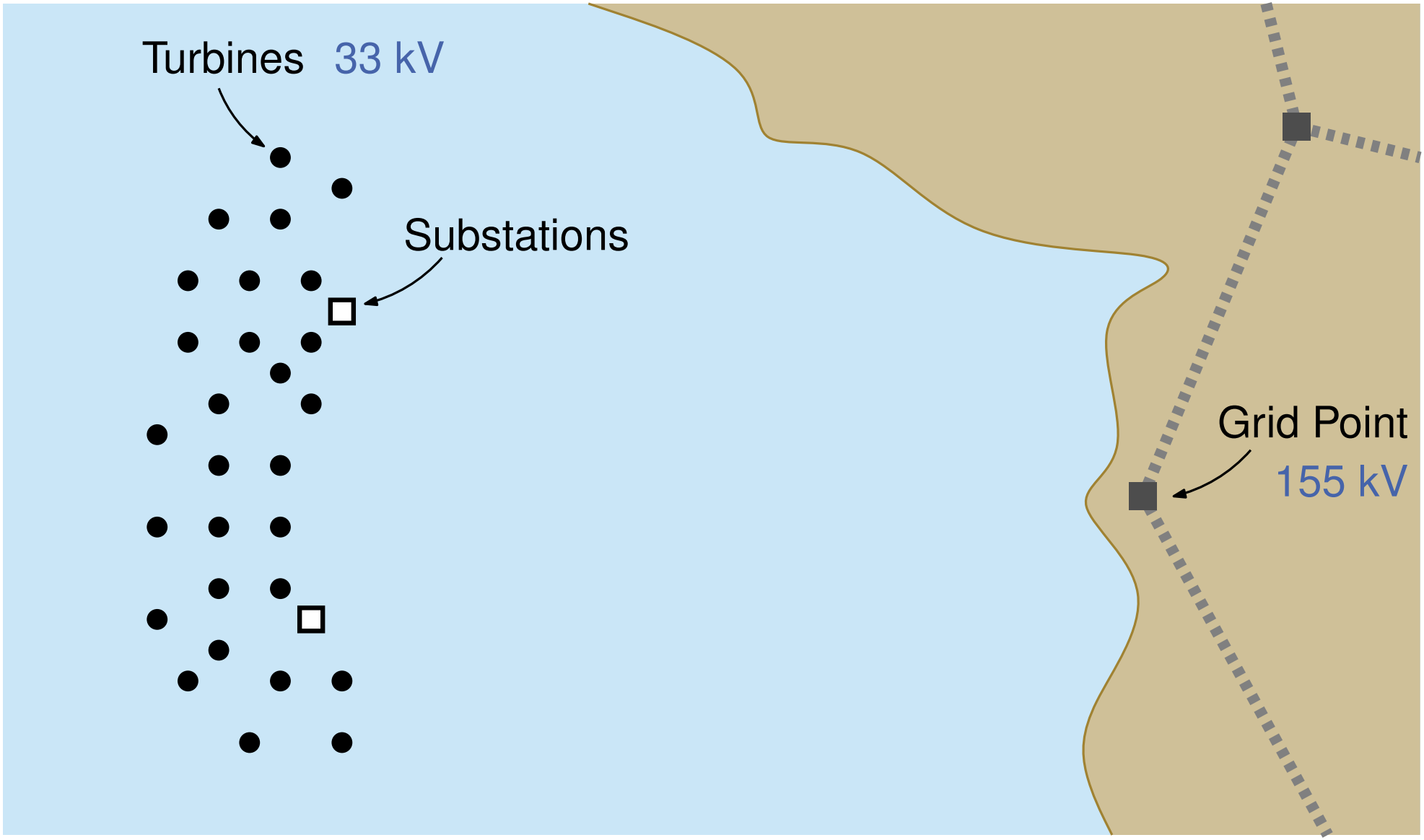
Institute of Theoretical Informatics
Algorithmics Group

Components in a Wind Farm



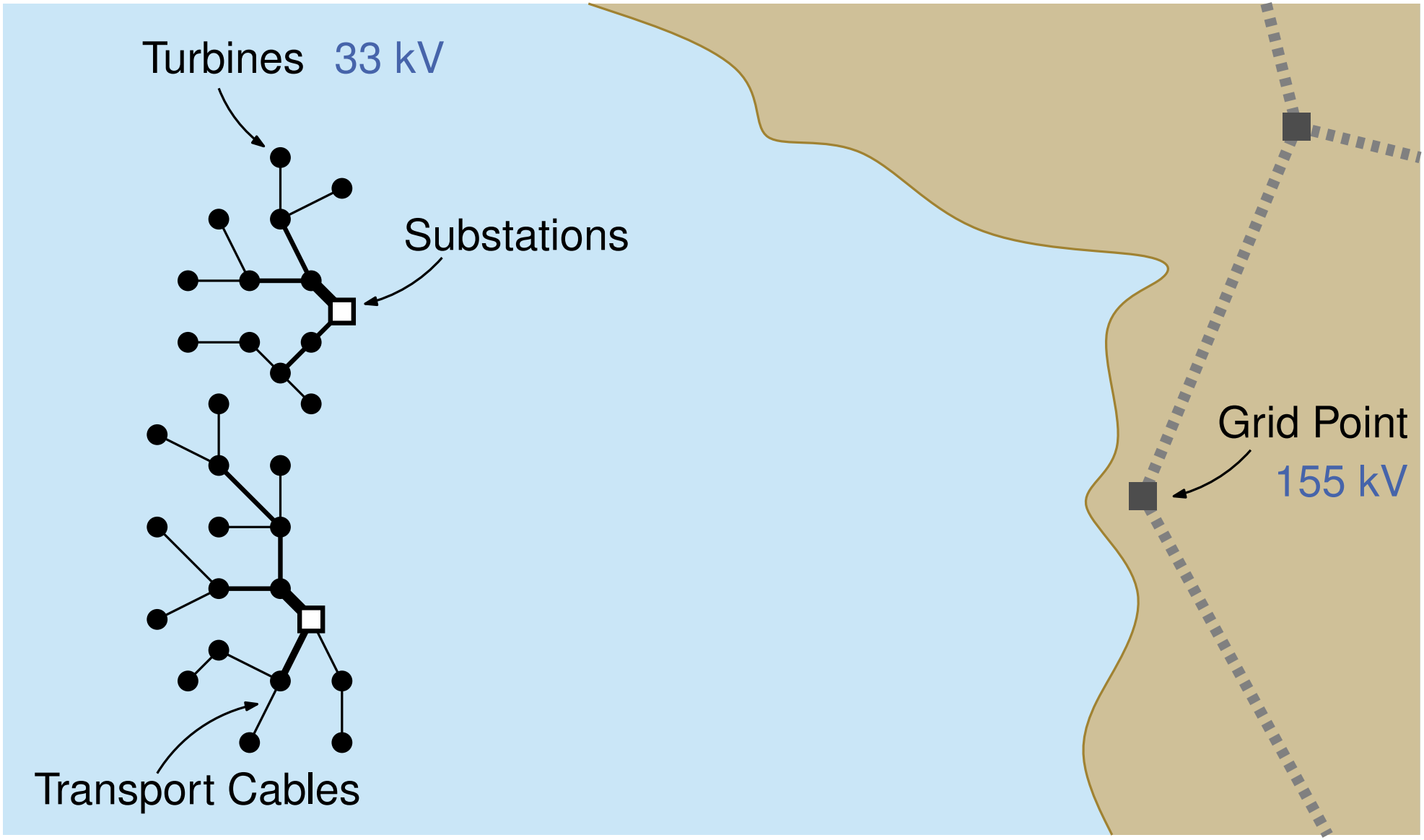
Institute of Theoretical Informatics
Algorithmics Group

Components in a Wind Farm



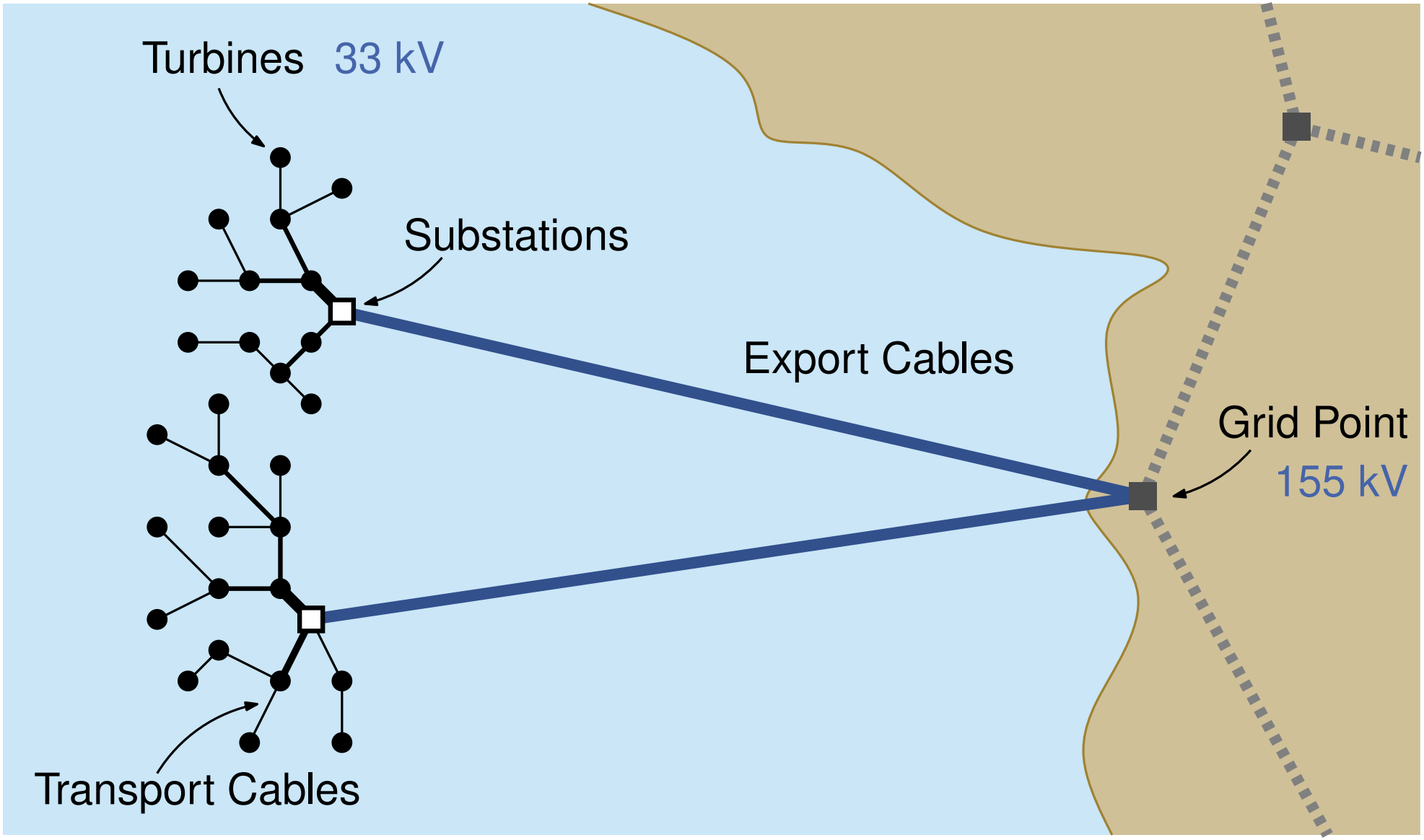
Institute of Theoretical Informatics
Algorithmics Group

Components in a Wind Farm



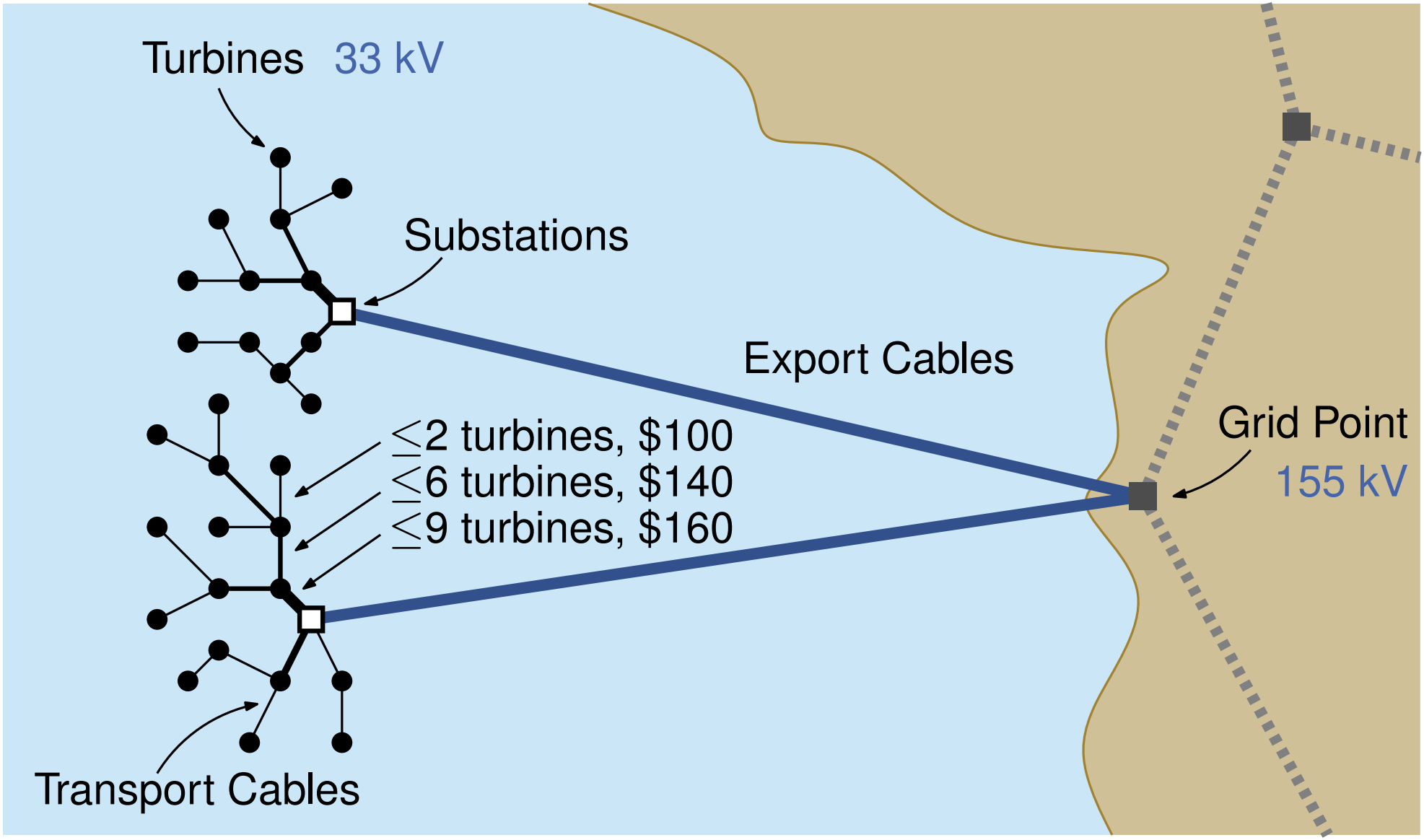
Institute of Theoretical Informatics
Algorithmics Group

Components in a Wind Farm



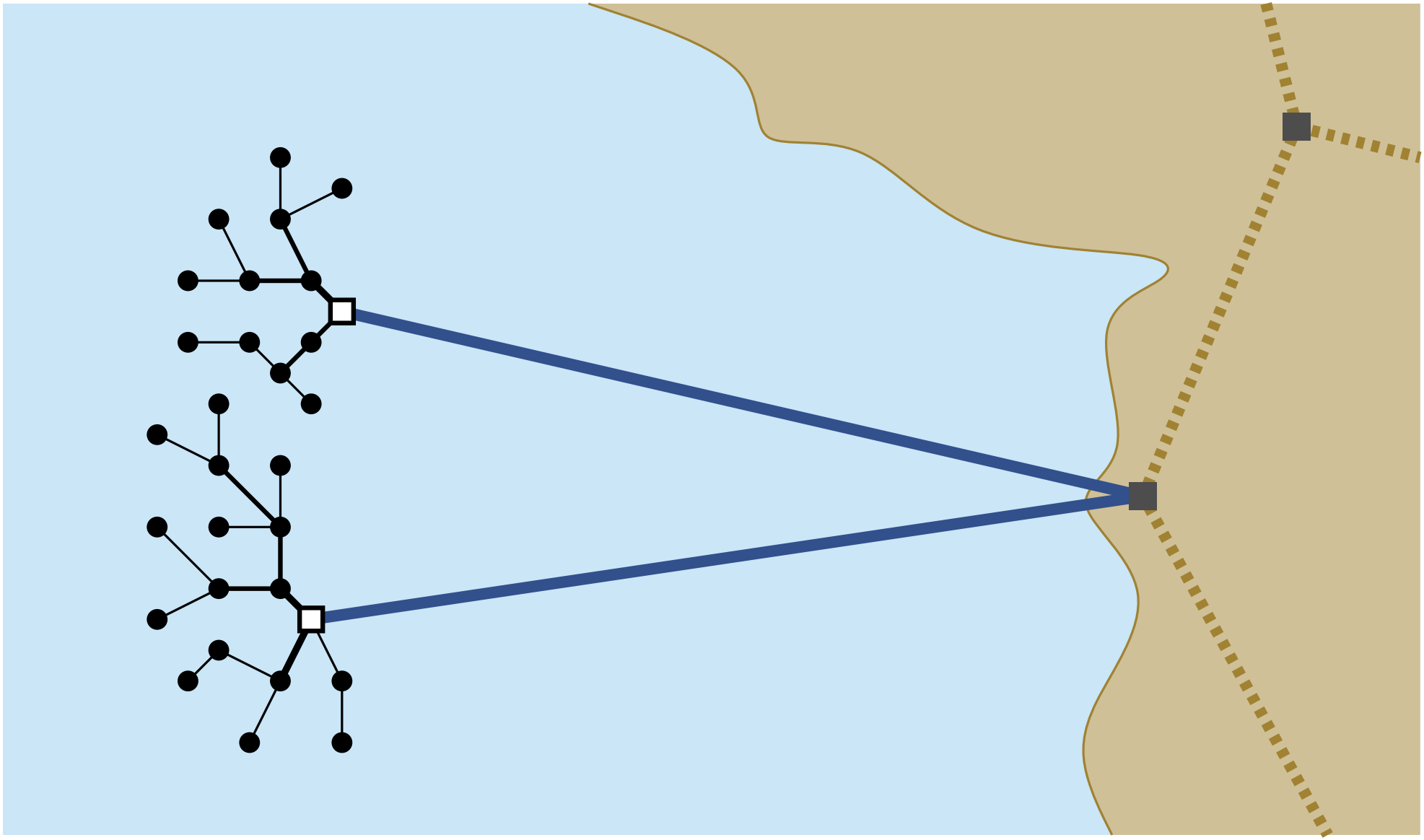
Institute of Theoretical Informatics
Algorithmics Group

Components in a Wind Farm

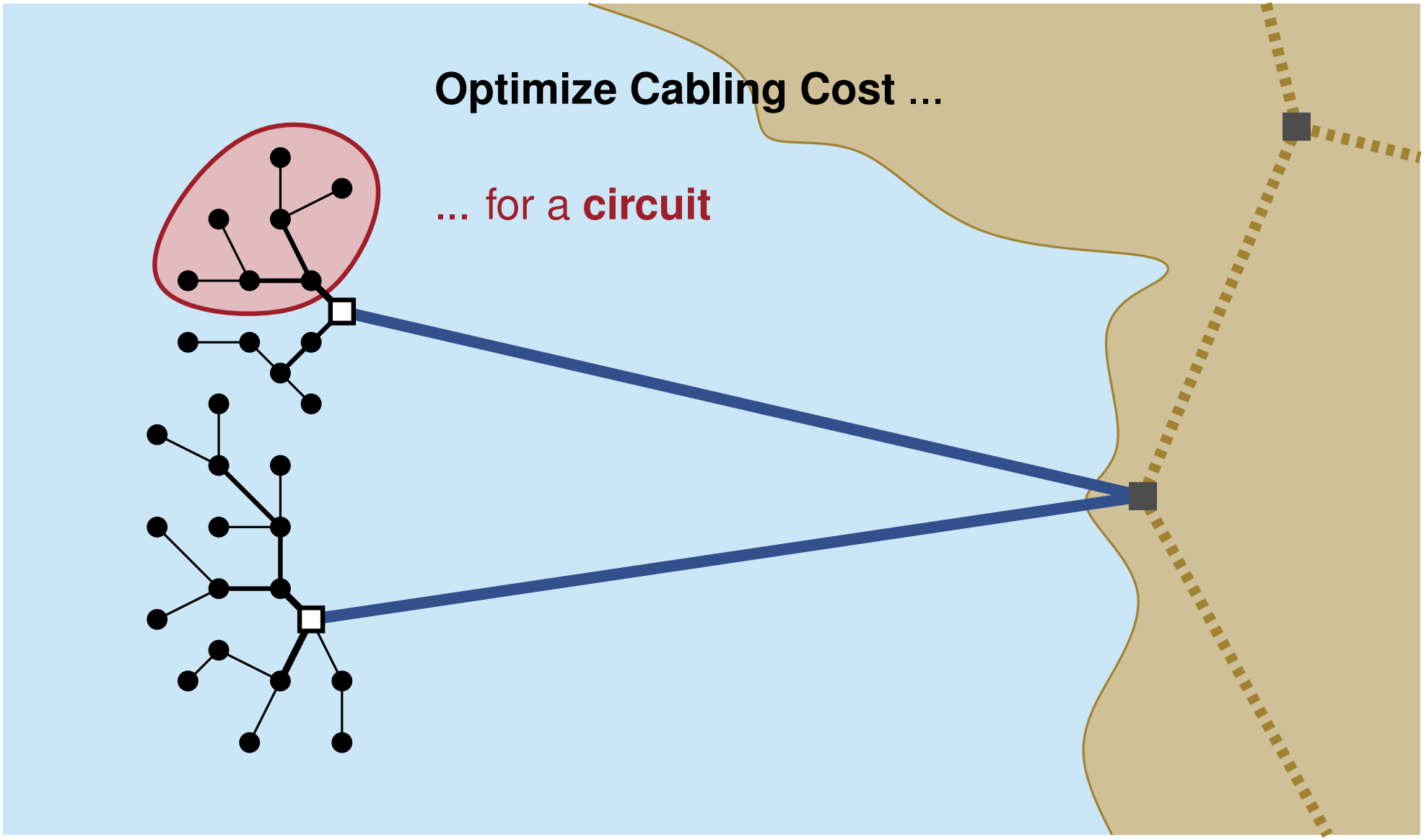


Institute of Theoretical Informatics
Algorithmics Group

Transmission Network Expansion Planning

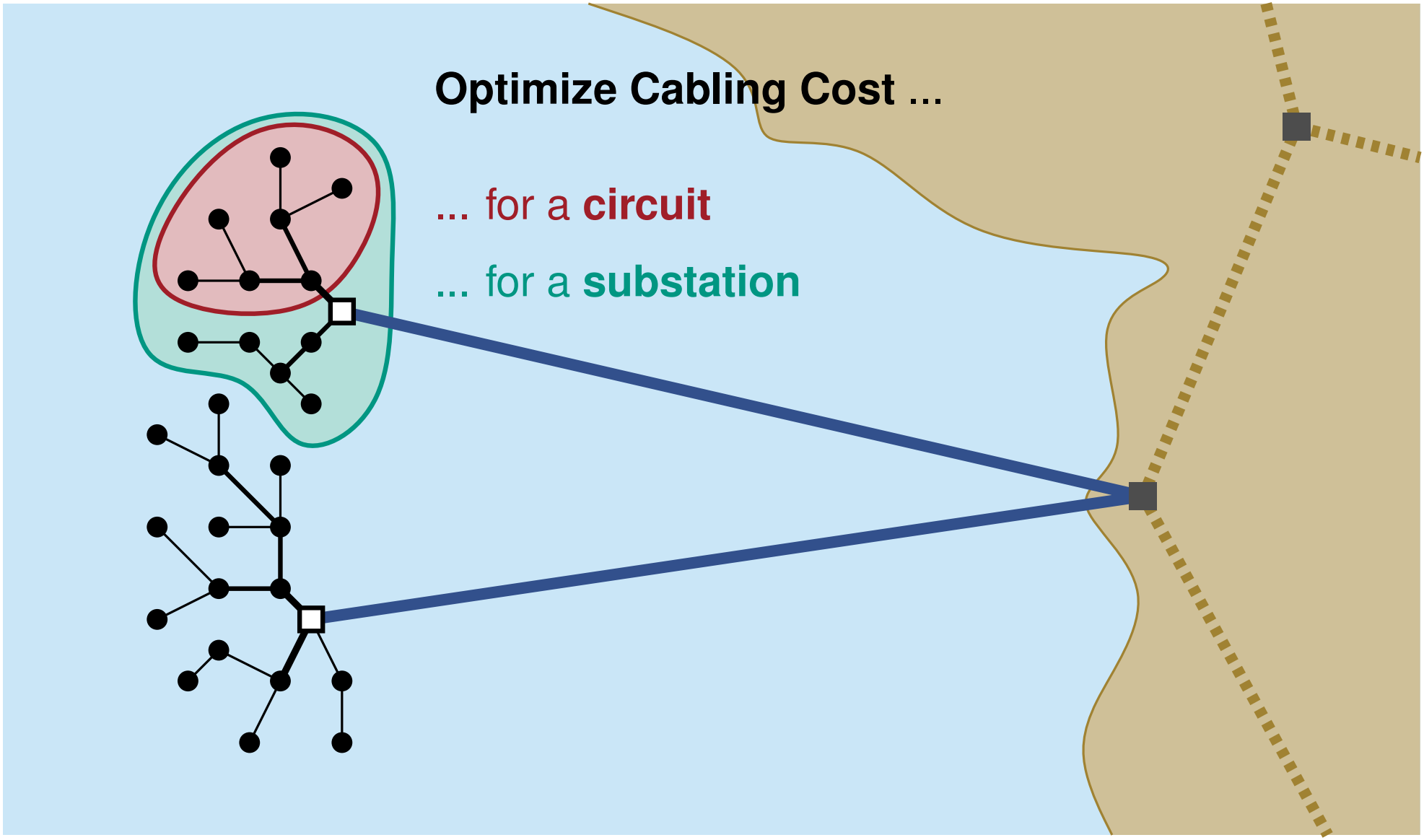


Transmission Network Expansion Planning



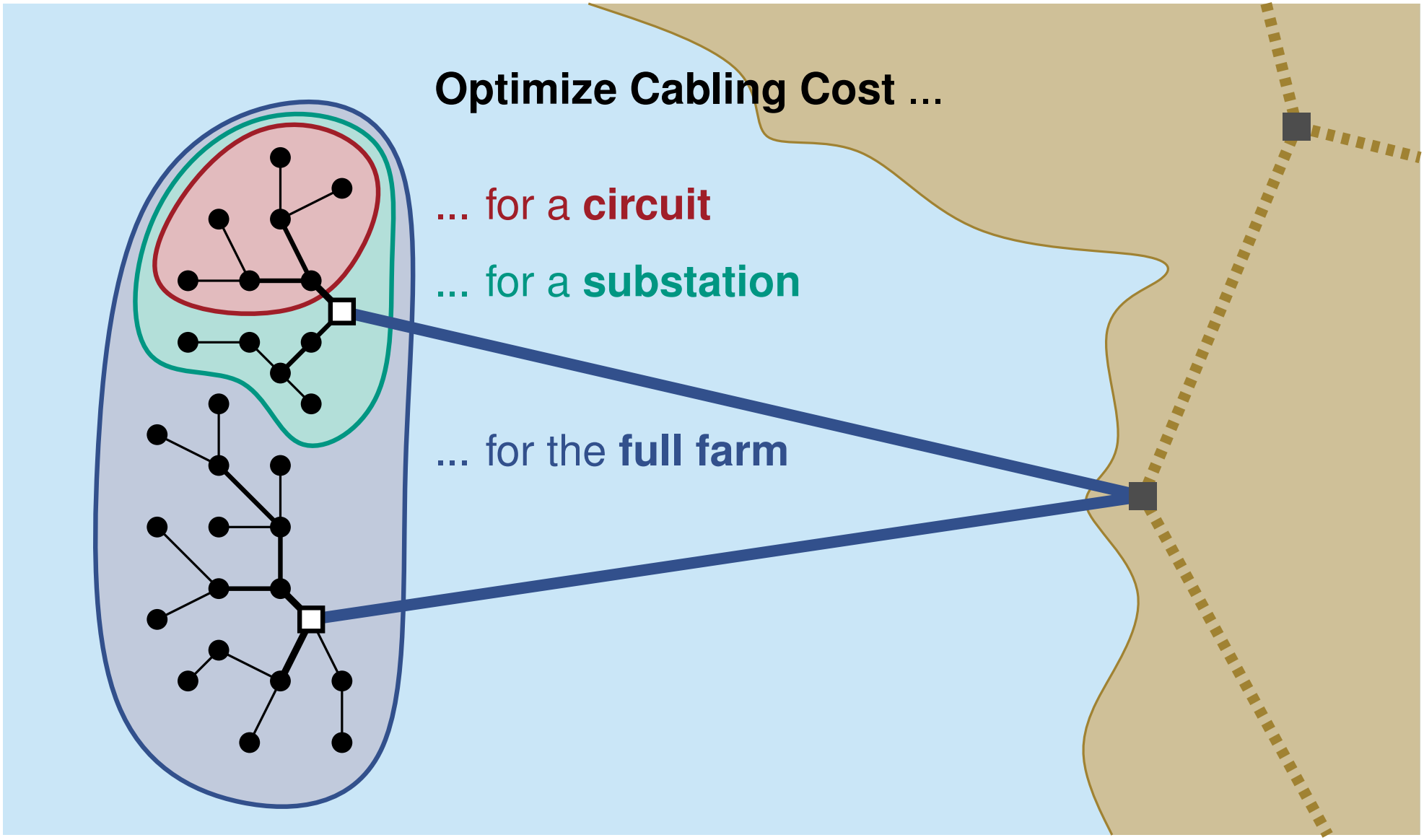
Institute of Theoretical Informatics
Algorithmics Group

Transmission Network Expansion Planning



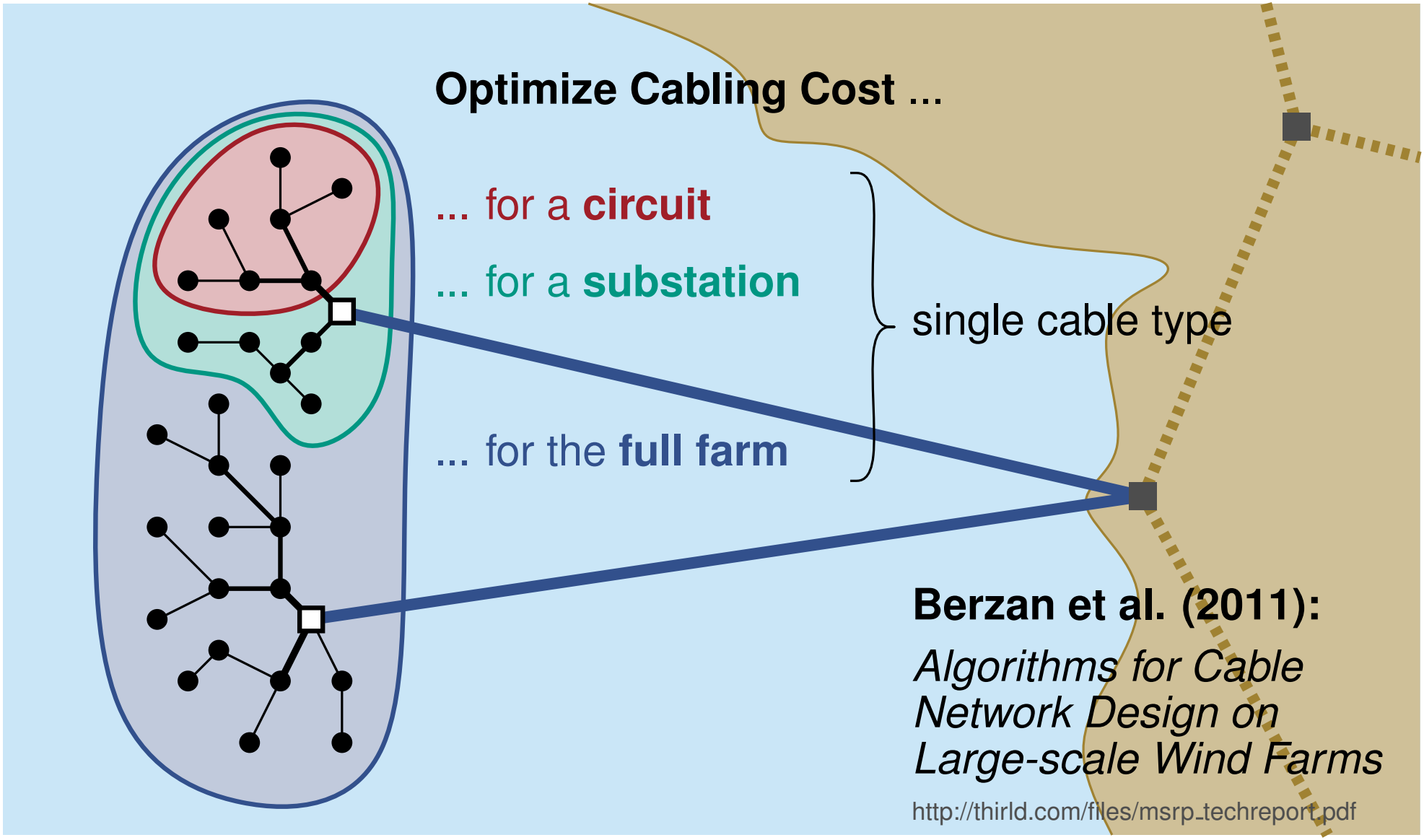
Institute of Theoretical Informatics
Algorithmics Group

Transmission Network Expansion Planning

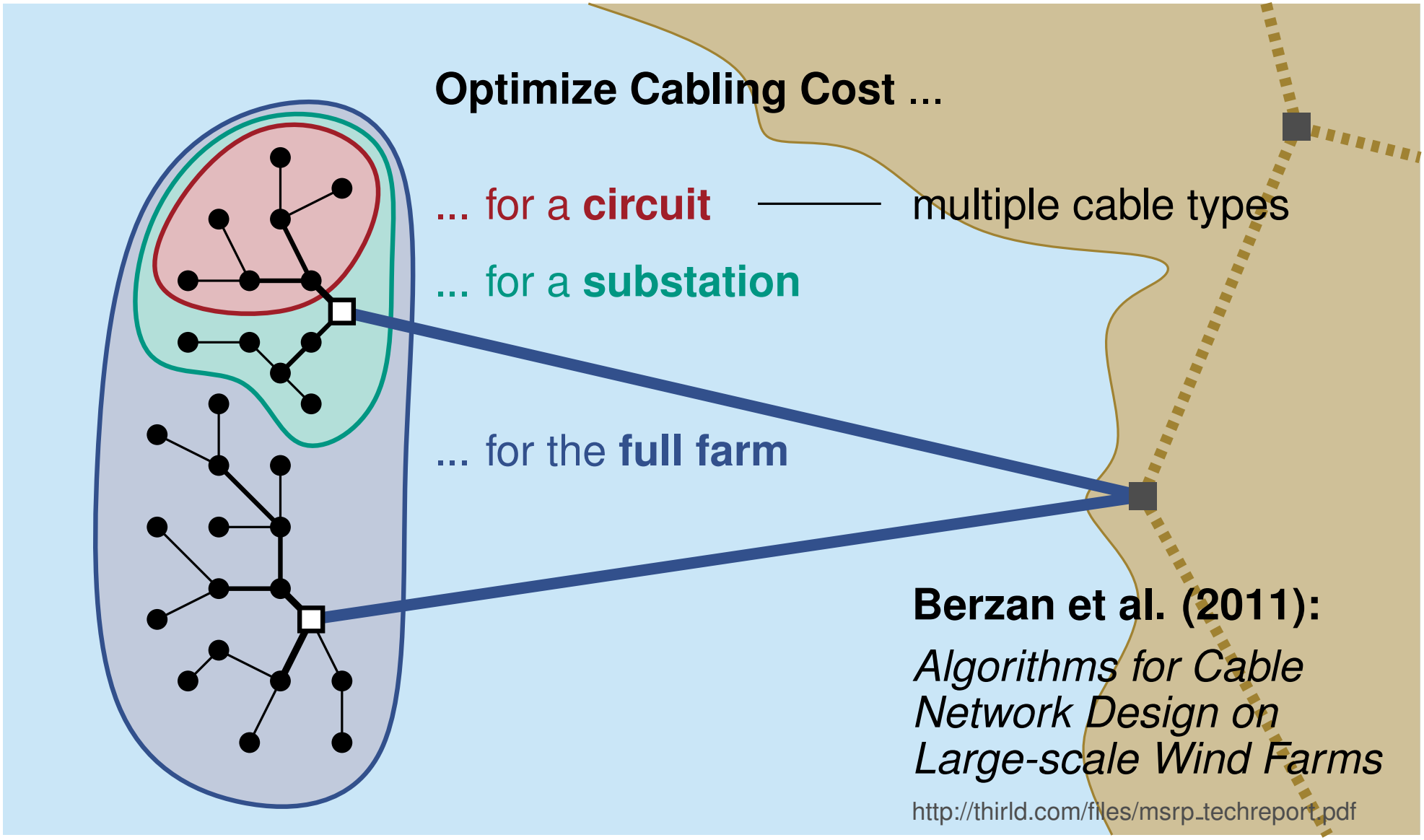


Institute of Theoretical Informatics
Algorithmics Group

Transmission Network Expansion Planning



Transmission Network Expansion Planning



Wind Farm Cable Layout Problem

Given **t** turbines, **s** substations (each with **capacity**),
for each edge: cable types (each with **cost** and **capacity**)

Institute of Theoretical Informatics
Algorithmics Group

Wind Farm Cable Layout Problem

Given **t** turbines, **s** substations (each with **capacity**),
for each edge: cable types (each with **cost** and **capacity**)

find for each edge: the **cable type**

Institute of Theoretical Informatics
Algorithmics Group

Wind Farm Cable Layout Problem

Karlsruhe Institute of Technology

Given **t** turbines, **s** substations (each with **capacity**),
for each edge: cable types (each with **cost** and **capacity**)

find for each edge: the **cable type**

minimizing their **total cost**

Institute of Theoretical Informatics
Algorithmics Group

Wind Farm Cable Layout Problem

Given	**t** turbines, **s** substations (each with **capacity**), for each edge: cable types (each with **cost** and **capacity**)
find	for each edge: the **cable type**
minimizing	their **total cost**
subject to	cable capacity constraints substation capacity constraints flow conservation constraints

Integer Linear Programming

Given	**t** turbines, **s** substations (each with **capacity**), for each edge: cable types (each with **cost** and **capacity**)
find	for each edge: the **cable type**
minimizing	their **total cost**
subject to	cable capacity constraints substation capacity constraints flow conservation constraints

Integer Linear Programming

Given	**t** turbines, **s** substations (each with **capacity**), for each edge: cable types (each with **cost** and **capacity**)
find	for each edge: the **cable type**
minimizing	their **total cost**
subject to	cable capacity constraints substation capacity constraints flow conservation constraints

inputs

Institute of Theoretical Informatics
Algorithmics Group

Integer Linear Programming

Given **t** turbines, **s** substations (each with **capacity**),
for each edge: cable types (each with **cost** and **capacity**)

find for each edge: the **cable type**

binary variables

minimizing their **total cost**

variables

subject to cable capacity constraints
substation capacity constraints
flow conservation constraints

Institute of Theoretical Informatics
Algorithmics Group

Integer Linear Programming

Given **t** turbines, **s** substations (each with **capacity**),
for each edge: cable types (each with **cost** and **capacity**)

find for each edge: the **cable type**

minimizing their **total cost** $= \sum_{\text{edges}} \sum_{\text{cable types}}$ **cable chosen?** $\times$ **cable cost**

objective

subject to cable capacity constraints
substation capacity constraints
flow conservation constraints

Institute of Theoretical Informatics
Algorithmics Group

Meta-Heuristic Optimization Techniques

Greedy

Simulated Annealing

Hill Climbing

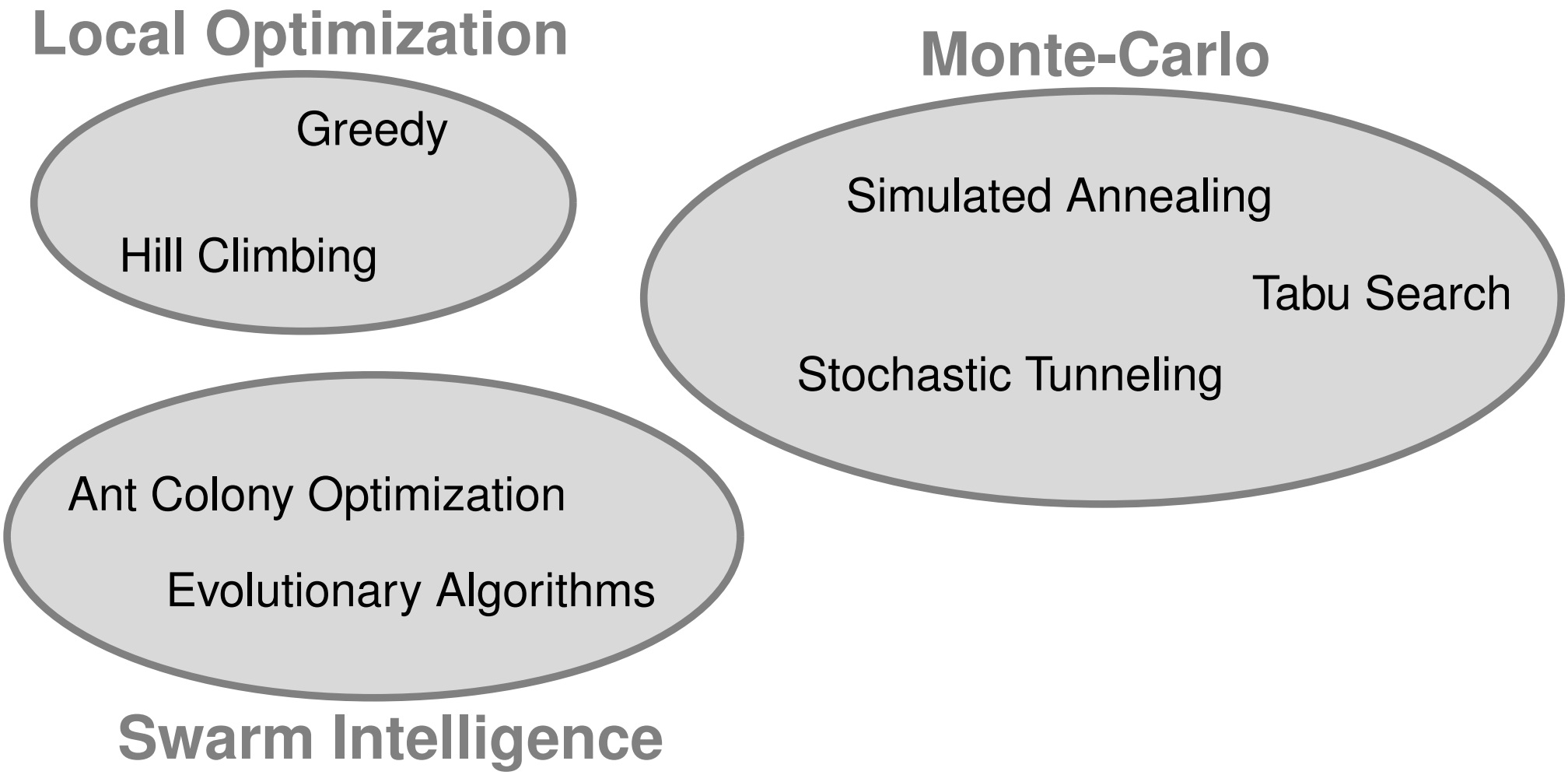
Tabu Search

Stochastic Tunneling

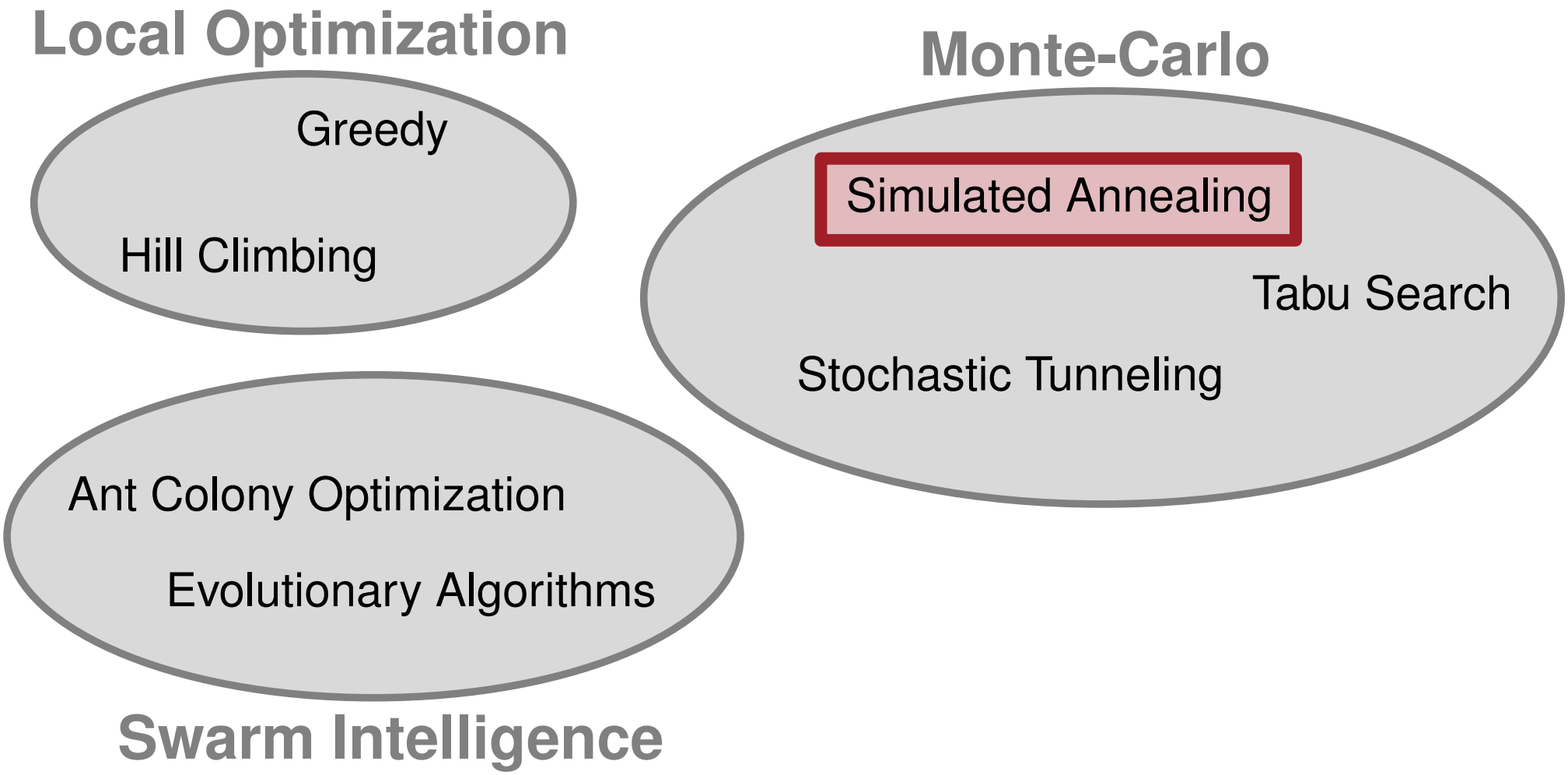
Ant Colony Optimization

Evolutionary Algorithms

Meta-Heuristic Optimization Techniques



Meta-Heuristic Optimization Techniques

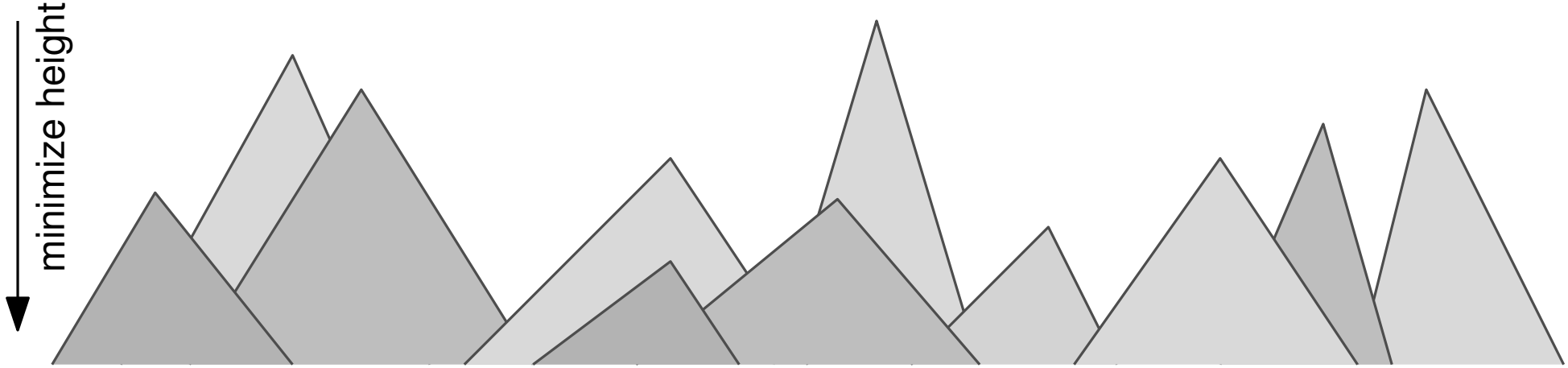


Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily

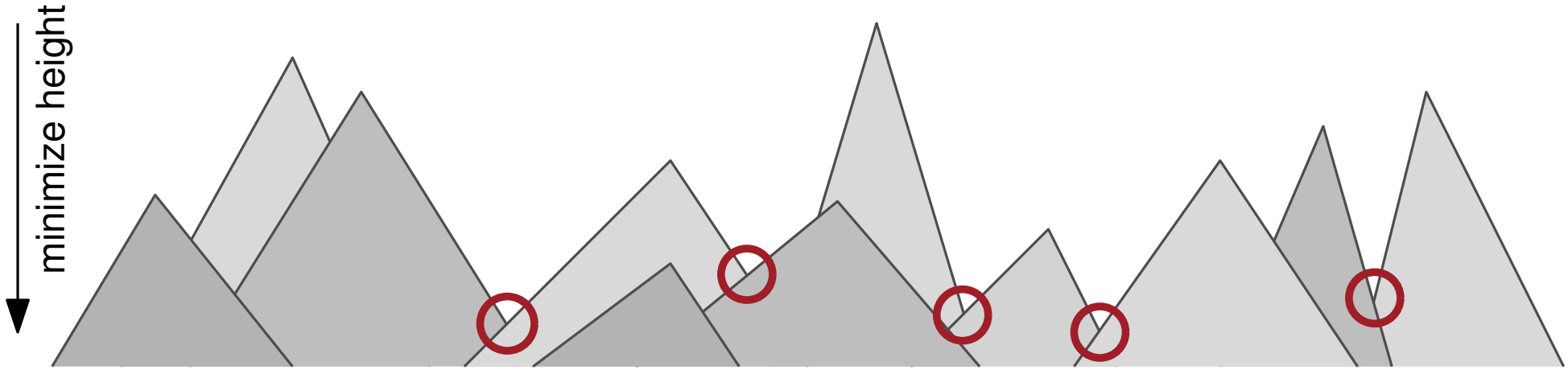
Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily



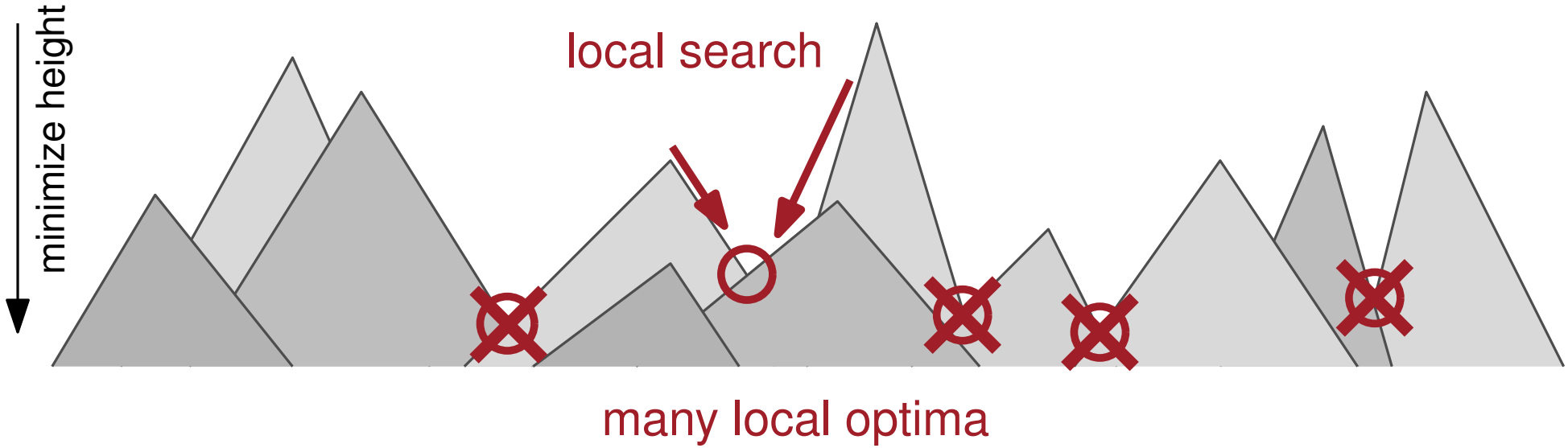
Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily



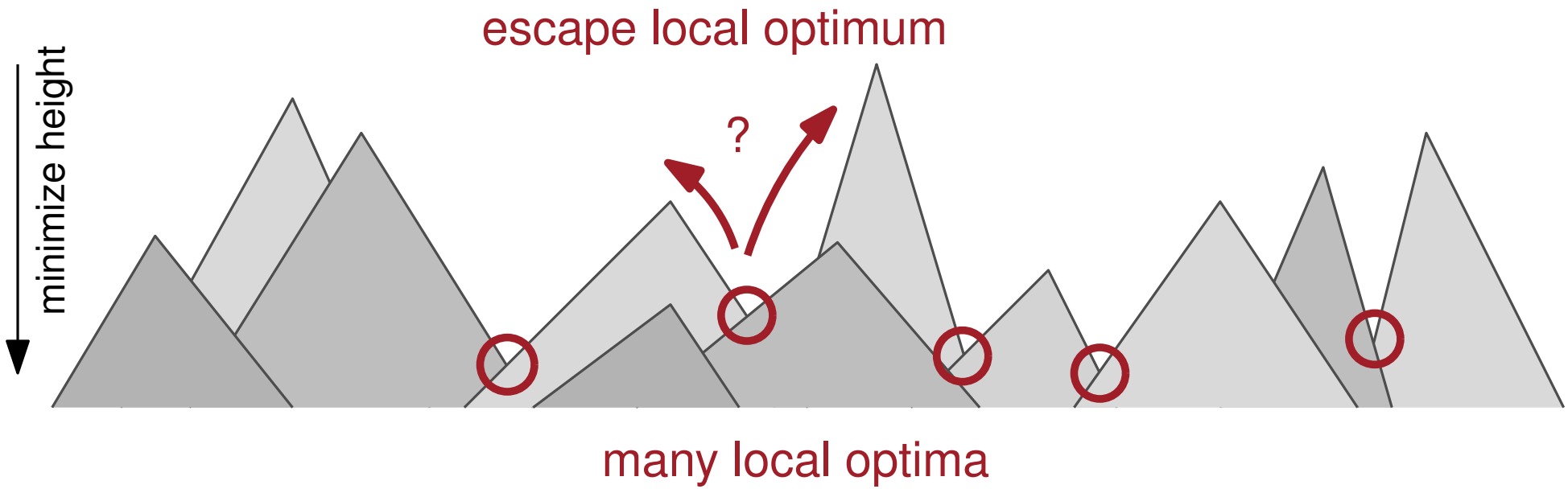
Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily



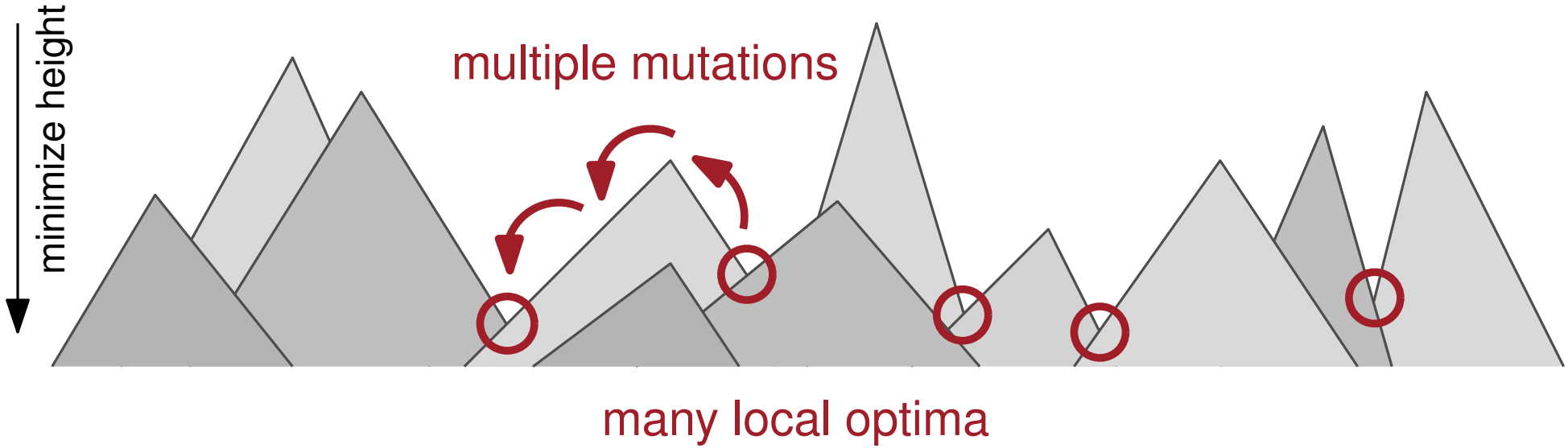
Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily



Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily

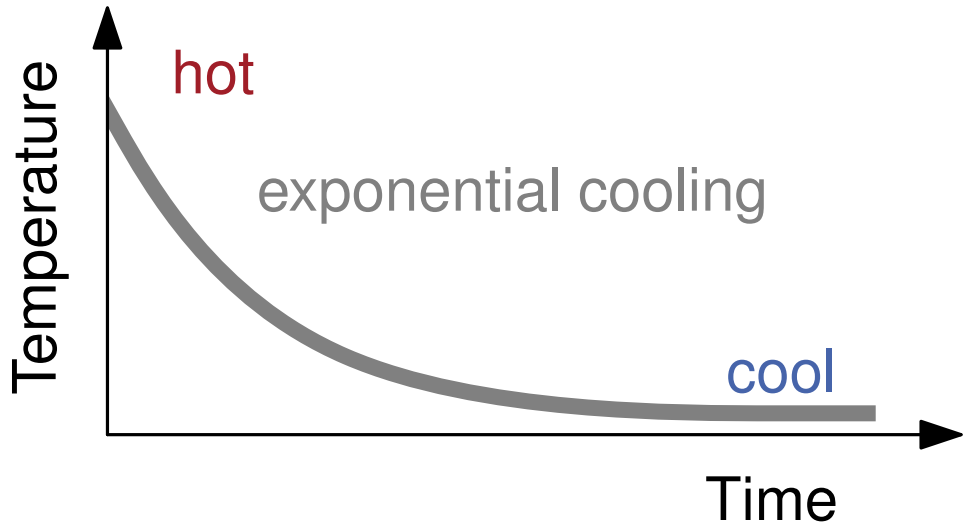


Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily
- **temperature** controls acceptance of worse solutions

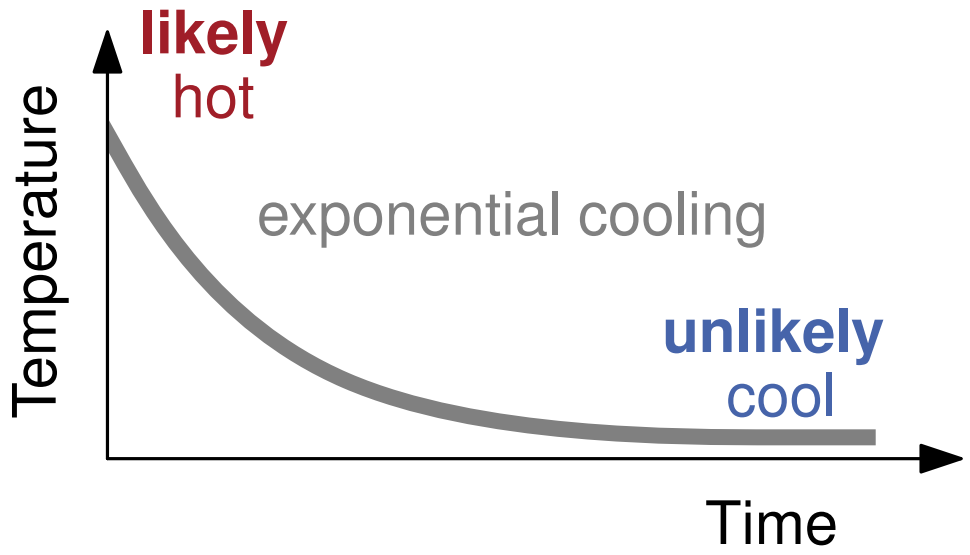
Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily
- **temperature** controls acceptance of worse solutions

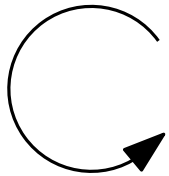


Simulated Annealing

- mutate solution candidates
- idea: allow worse solutions temporarily
- **temperature** controls acceptance of worse solutions

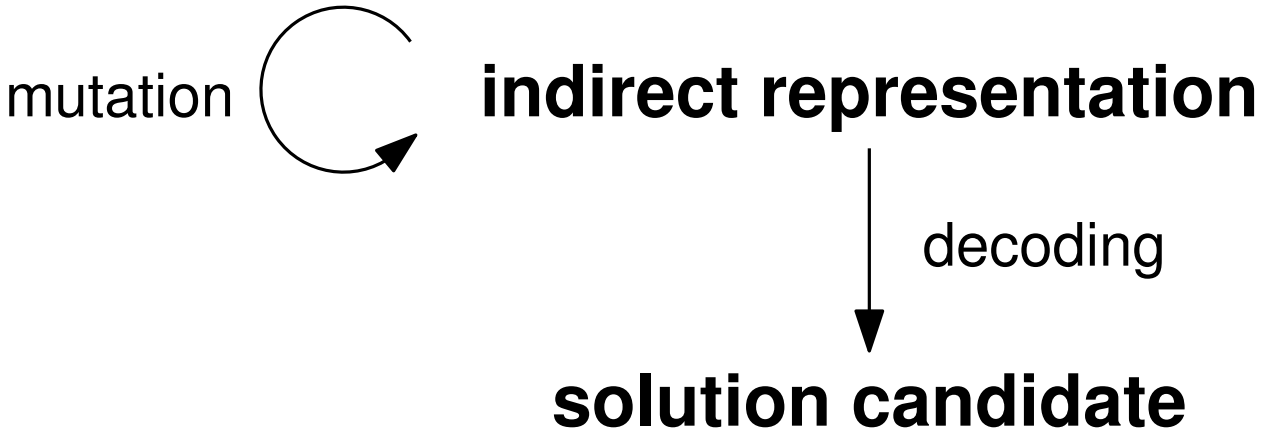


Simulated Annealing

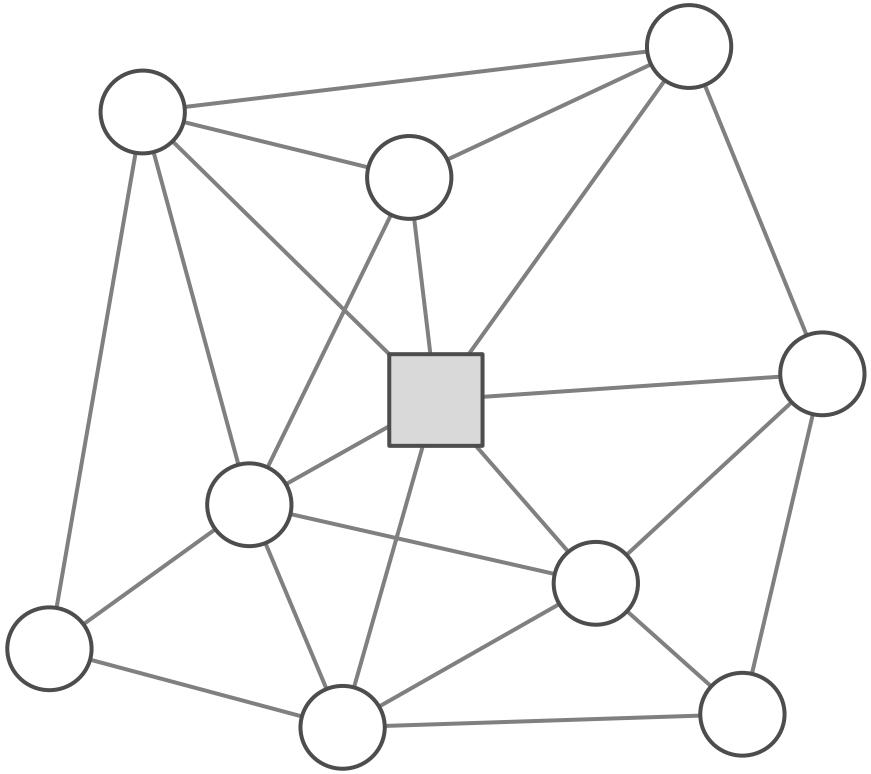
mutation  **solution candidate**

Institute of Theoretical Informatics
Algorithmics Group

Simulated Annealing

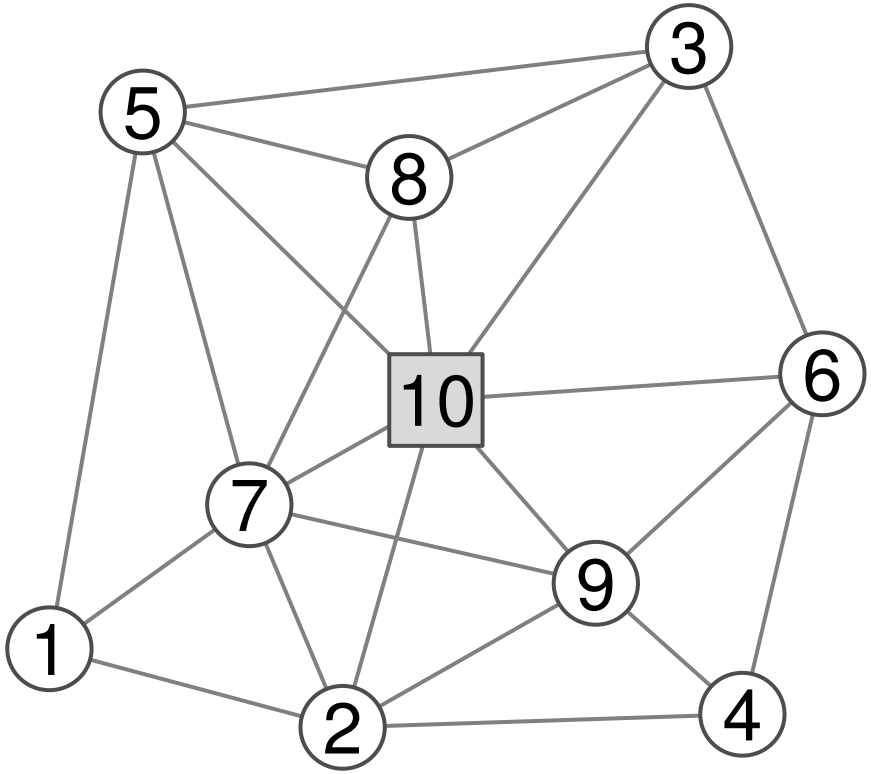


Our Representation



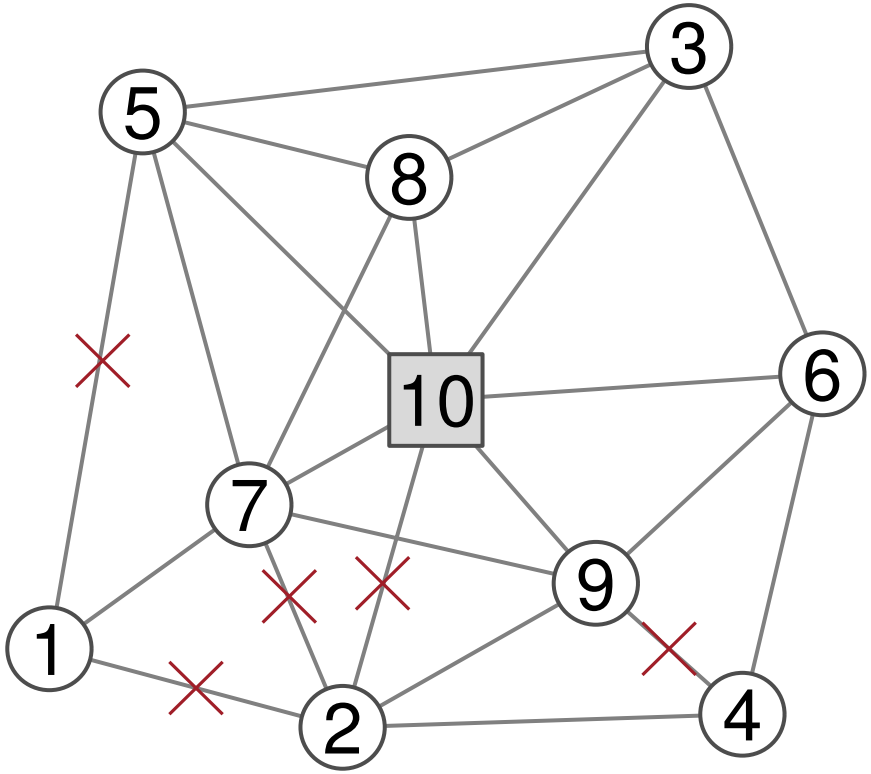
Our Representation

- nodes: potential values (permutation of indices)



Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

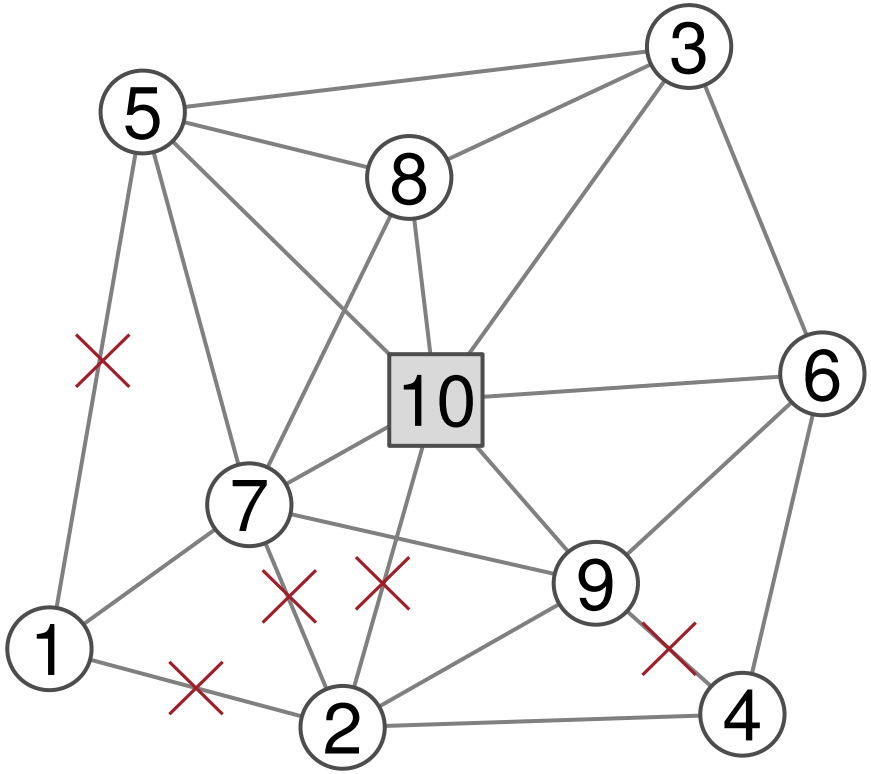


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Decoding

- each turbine: construct path

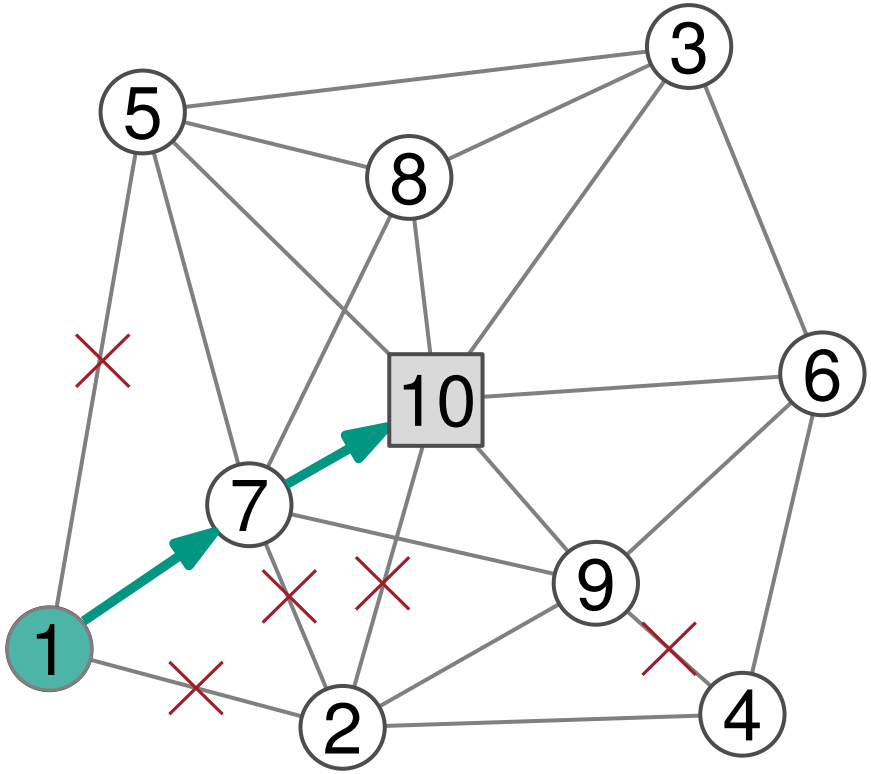


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Decoding

- each turbine: construct path

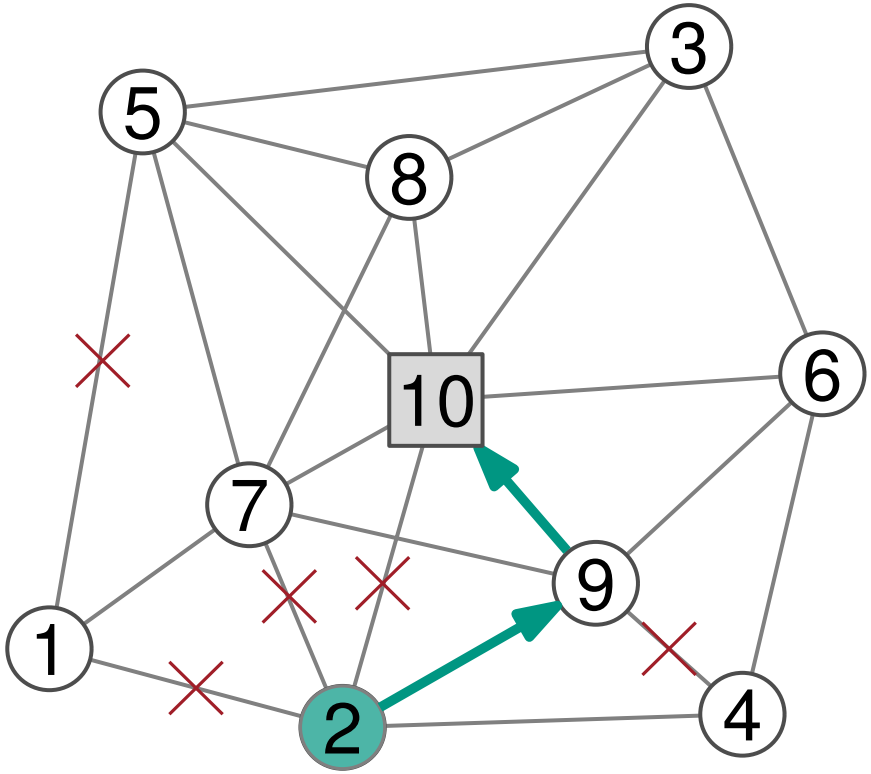


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Decoding

- each turbine: construct path

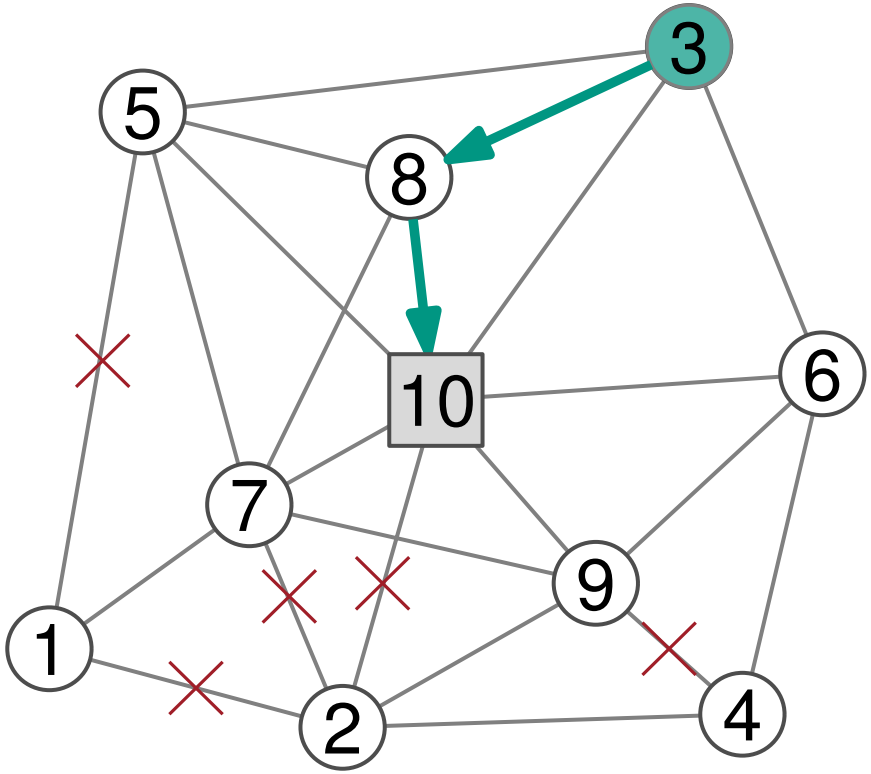


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Decoding

- each turbine: construct path

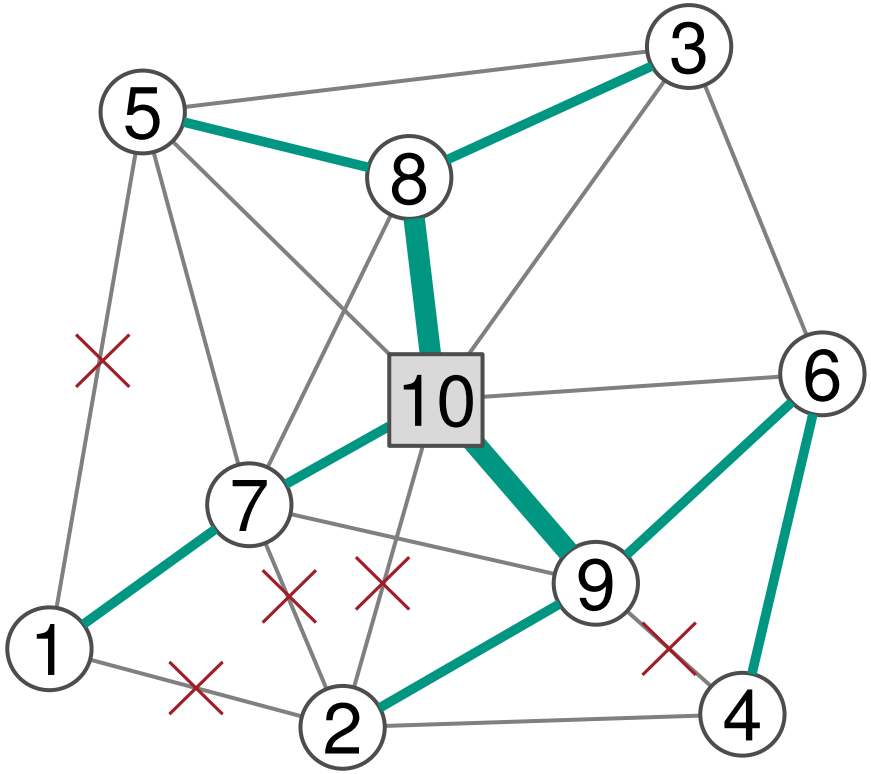


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Decoding

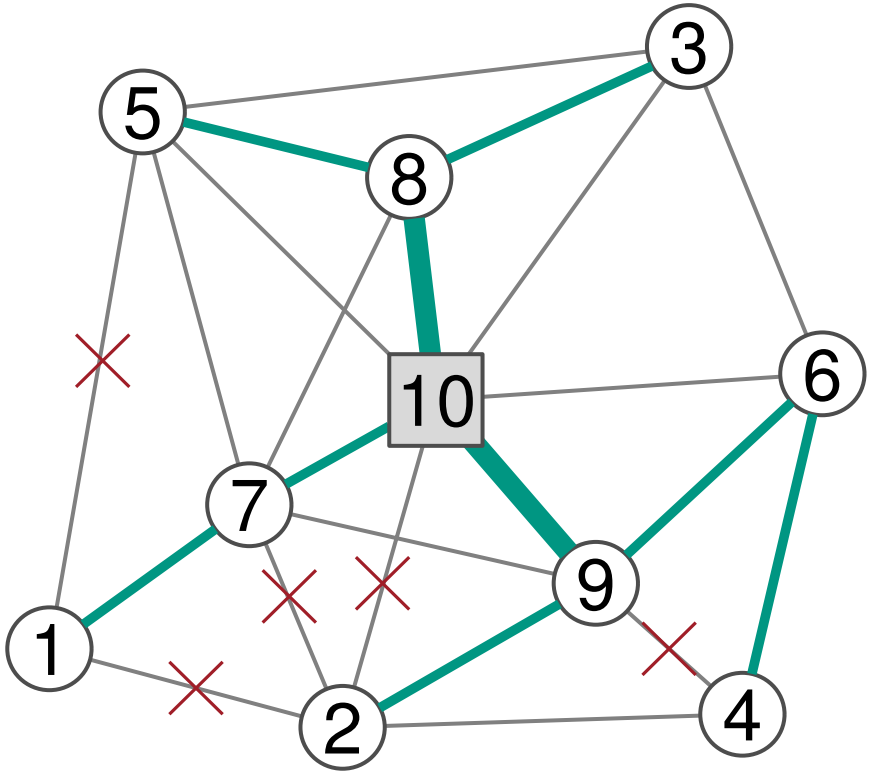
- each turbine: construct path
- each edge: find suited cable



Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

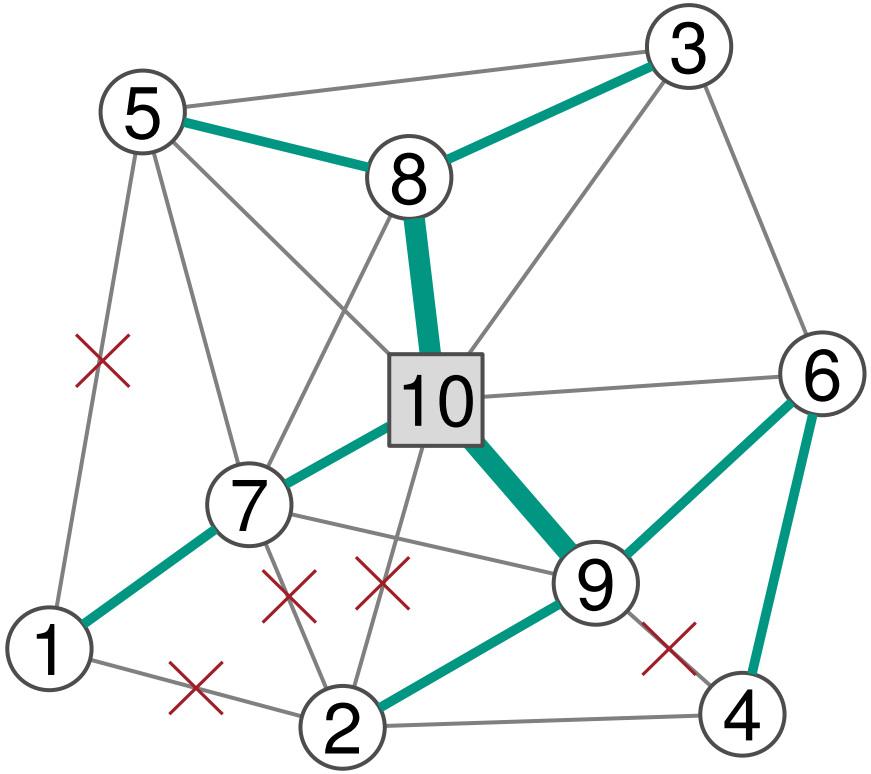


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

- swap node potentials

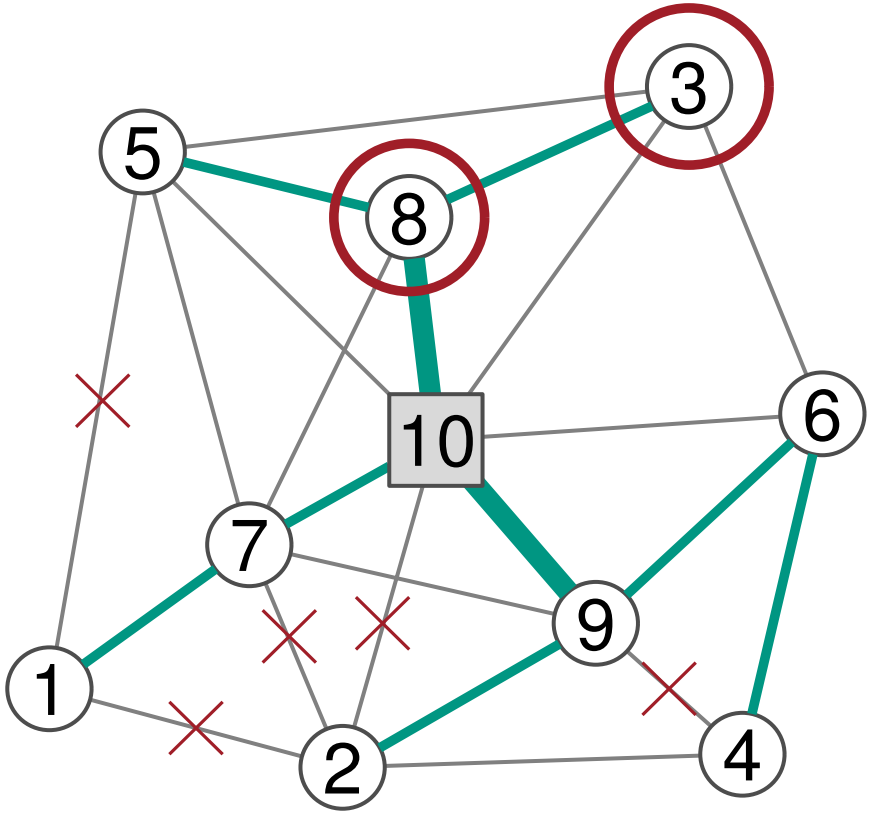


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

- swap node potentials

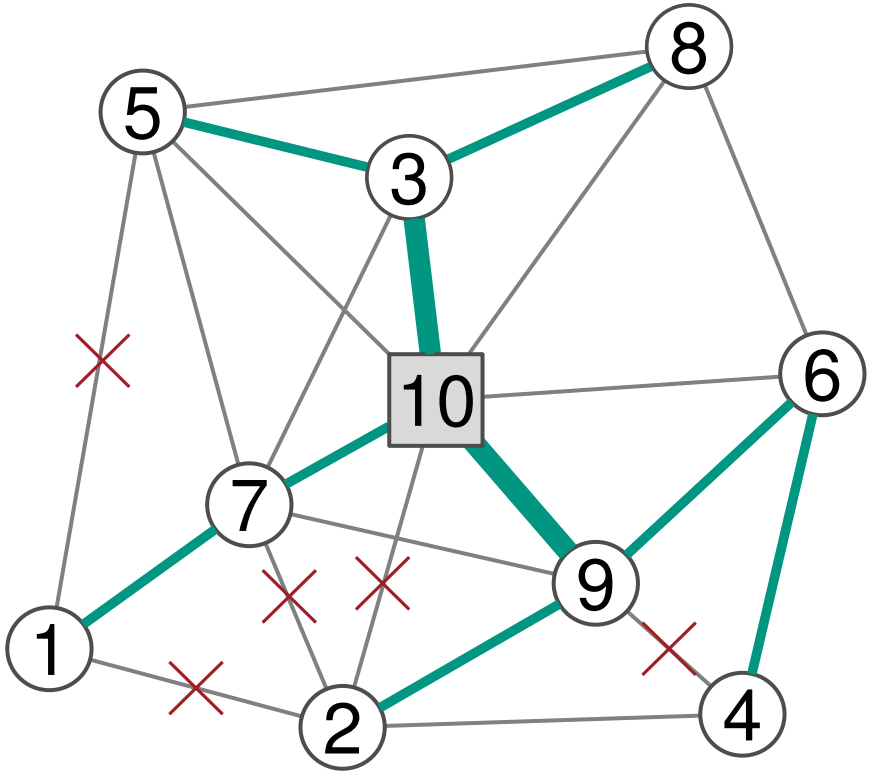


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

- swap node potentials

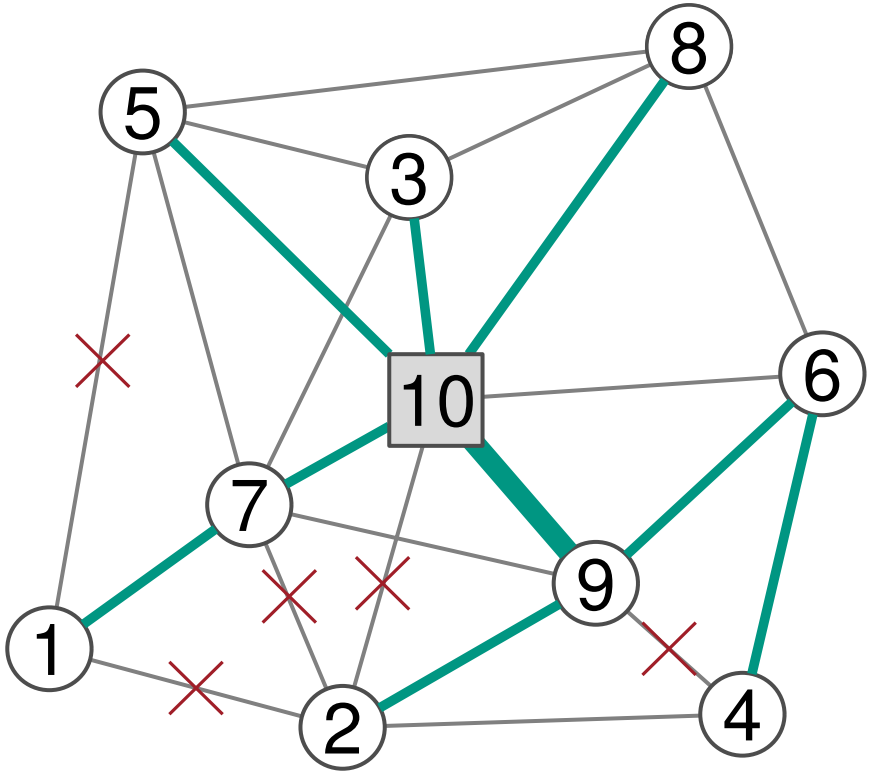


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

- swap node potentials

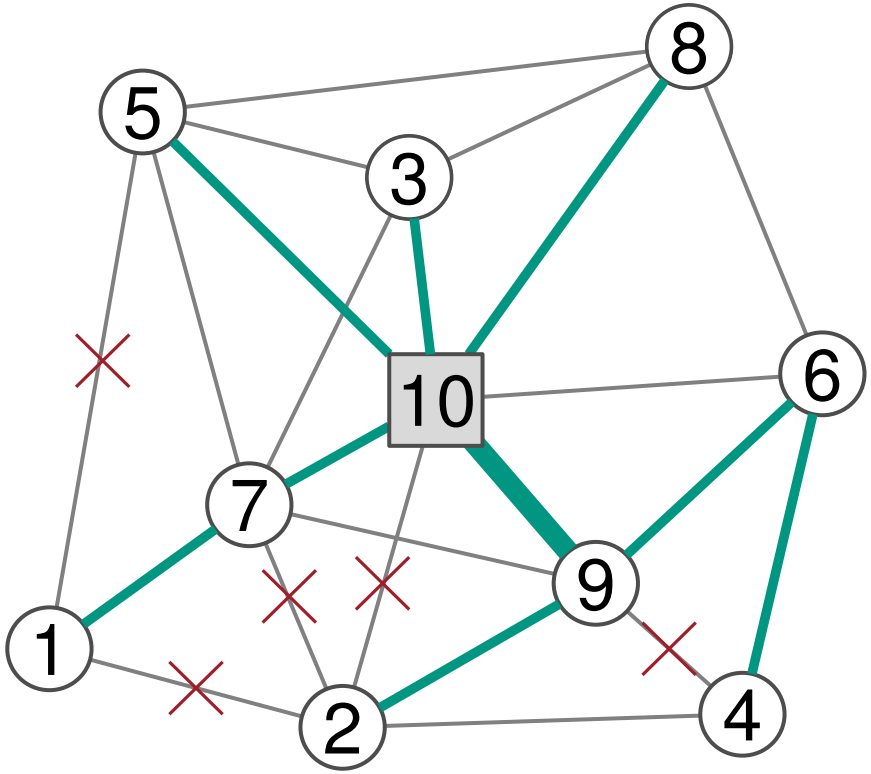


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

- swap node potentials
- forbid / allow an edge

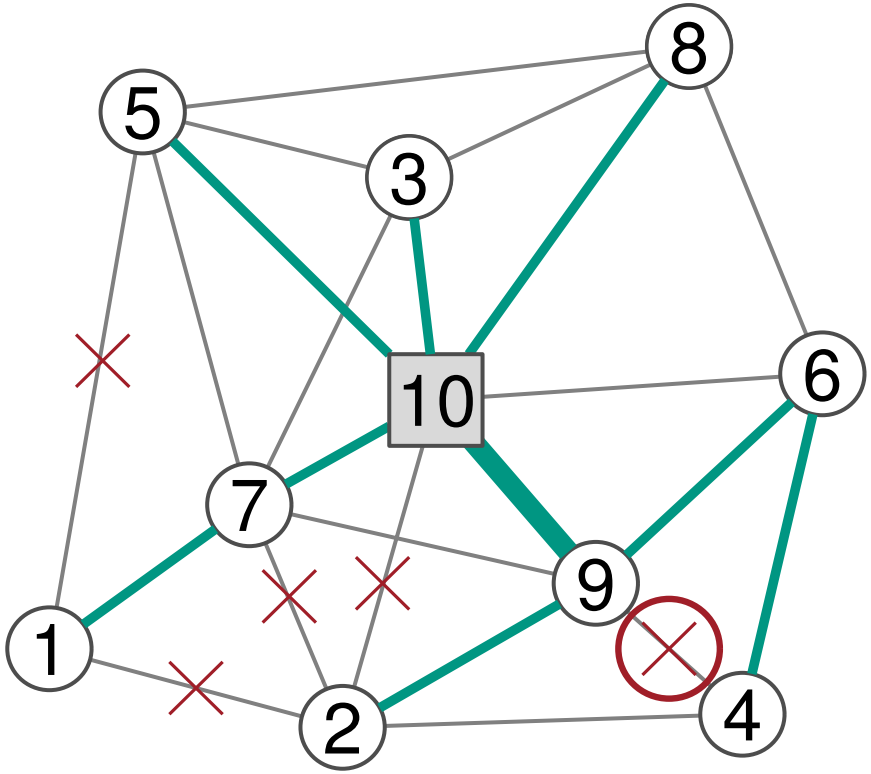


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

Mutation 

- swap node potentials
- forbid / allow an edge

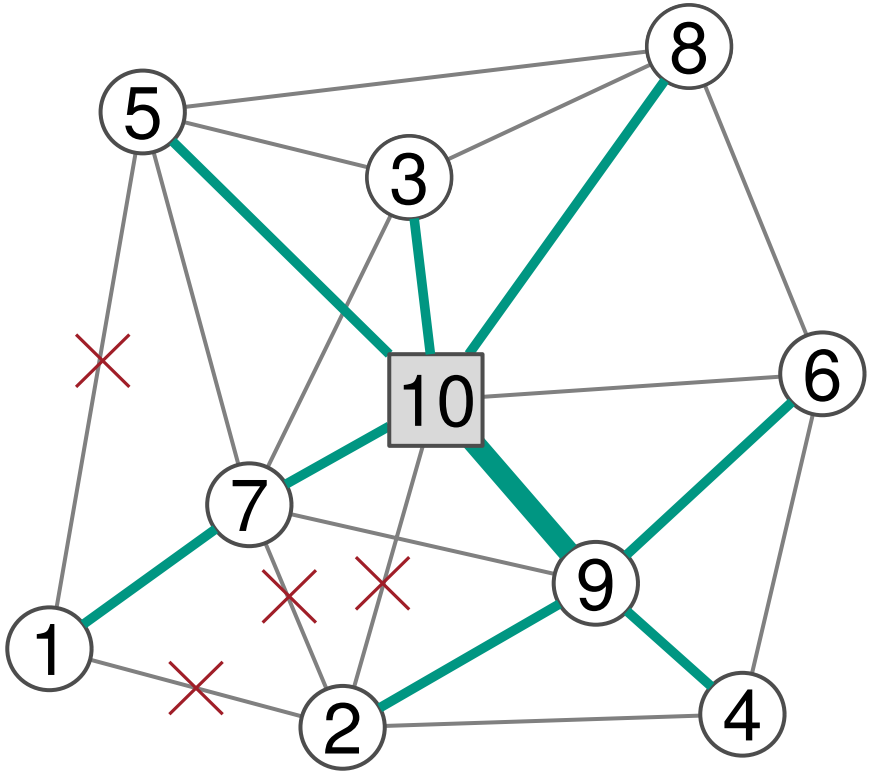


Our Representation

- nodes: potential values (permutation of indices)
- forbid some edges

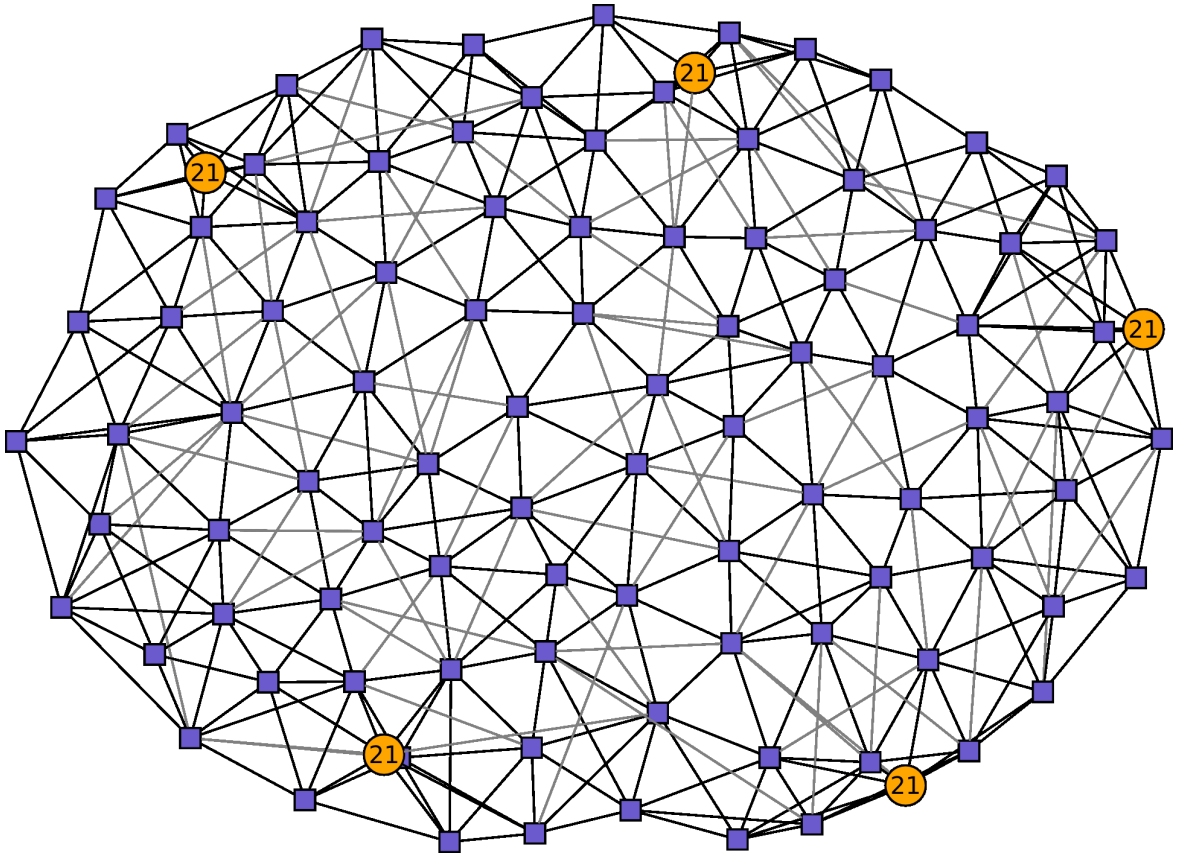
Mutation 

- swap node potentials
- forbid / allow an edge



Generating Instances

- Turbines & substations evenly distributed (Poisson Disk Sampling)
- Edges: 6 nearest neighbors + shortcuts
- Substation capacities: tight vs. loose



Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

reference solution:
run Gurobi for 1h

Institute of Theoretical Informatics
Algorithmics Group

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

better results after 30 min

reference solution: run Gurobi for 1h

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

converges slower than Gurobi

better results after 30 min

reference solution: run Gurobi for 1h

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

- **temperature** curve: parameter tuning difficult

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

- **temperature** curve: parameter tuning difficult
- result depends on **random seed**

Institute of Theoretical Informatics
Algorithmics Group

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

- **temperature** curve: parameter tuning difficult
- result depends on **random seed**
- bad results for **tight** substation capacities

Observations & Problems

good results for medium-sized farms ($\mathbf{t} < 350$: faster than Gurobi)

long running time required for large farms

- **temperature** curve: parameter tuning difficult
- result depends on **random seed**
- bad results for **tight** substation capacities
- bad results for **many** substations

Dynamic Temperature Curve

temperature curve: parameter tuning difficult

Institute of Theoretical Informatics
Algorithmics Group

Dynamic Temperature Curve

temperature curve: parameter tuning difficult

Observation:

- escaping deep local optimum takes **long time**

Dynamic Temperature Curve

temperature curve: parameter tuning difficult

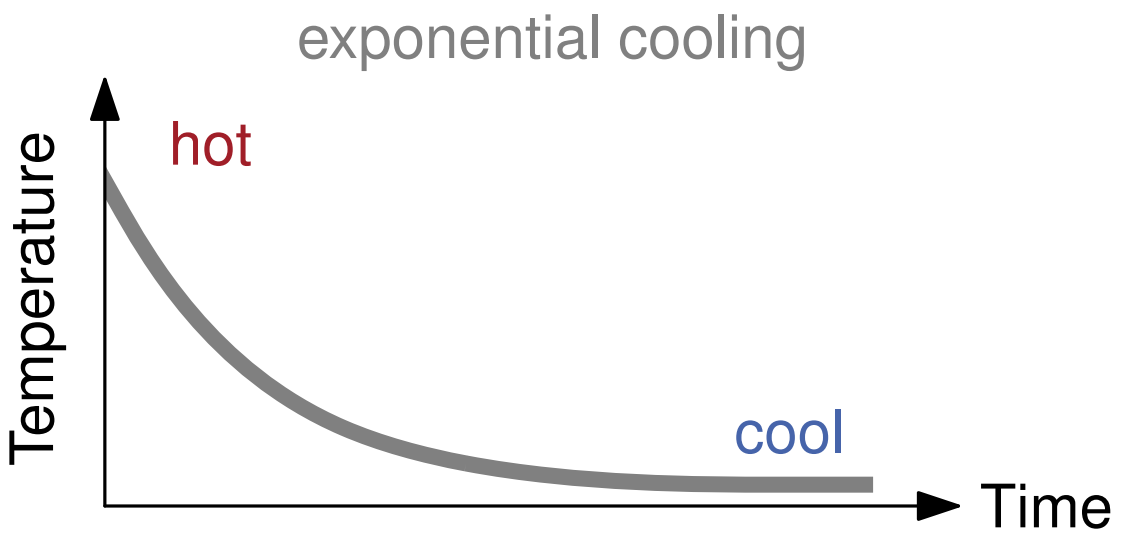
Observation:
- escaping deep local optimum takes **long time**

Dynamic Temperature Curve

temperature curve: parameter tuning difficult

Observation:

- escaping deep local optimum takes **long time**

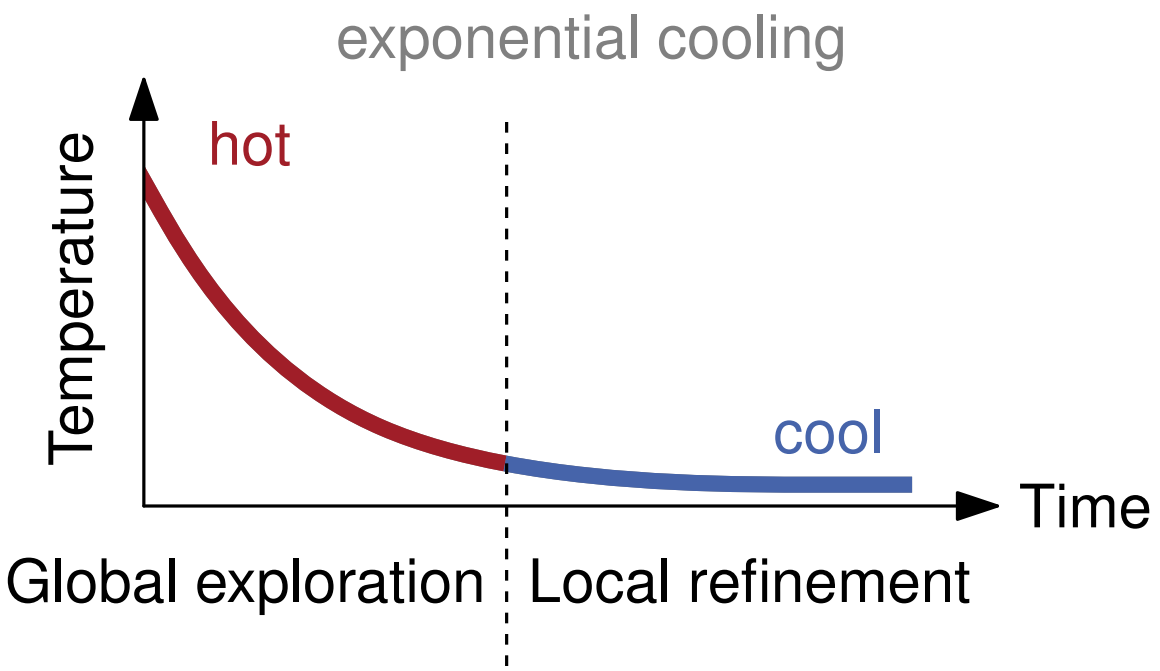


Dynamic Temperature Curve

temperature curve: parameter tuning difficult

Observation:

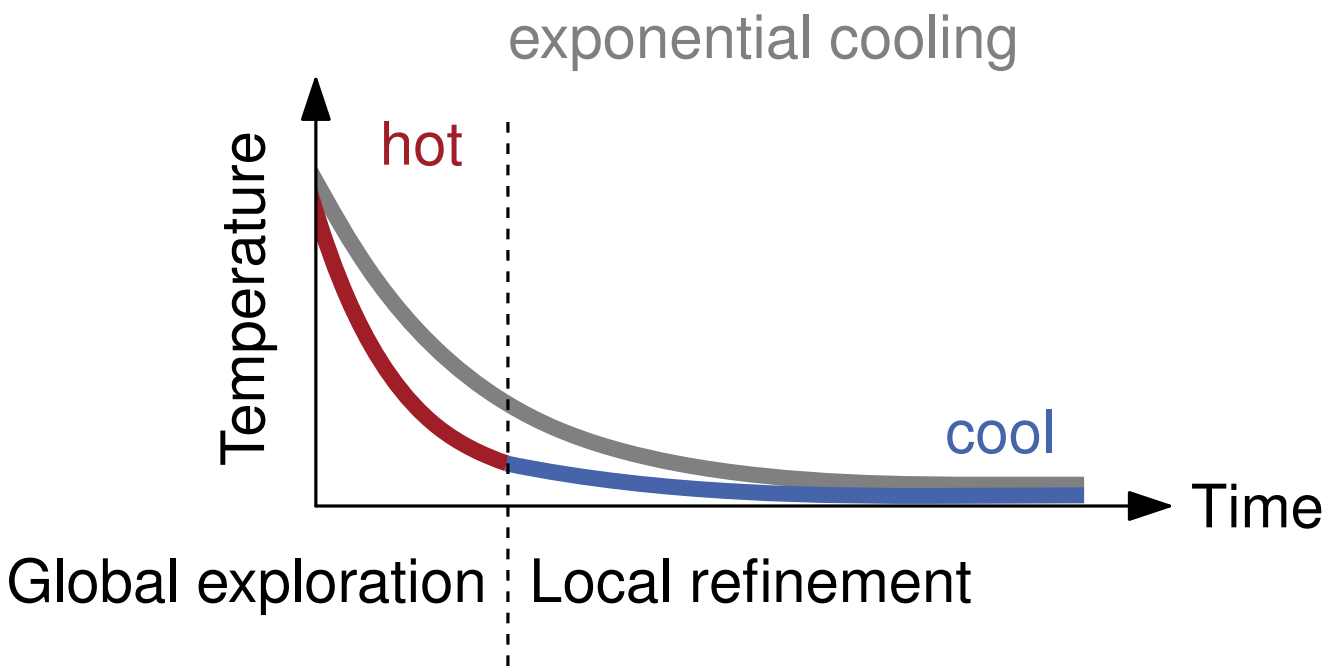
- escaping deep local optimum takes **long time**



Dynamic Temperature Curve

temperature curve: parameter tuning difficult

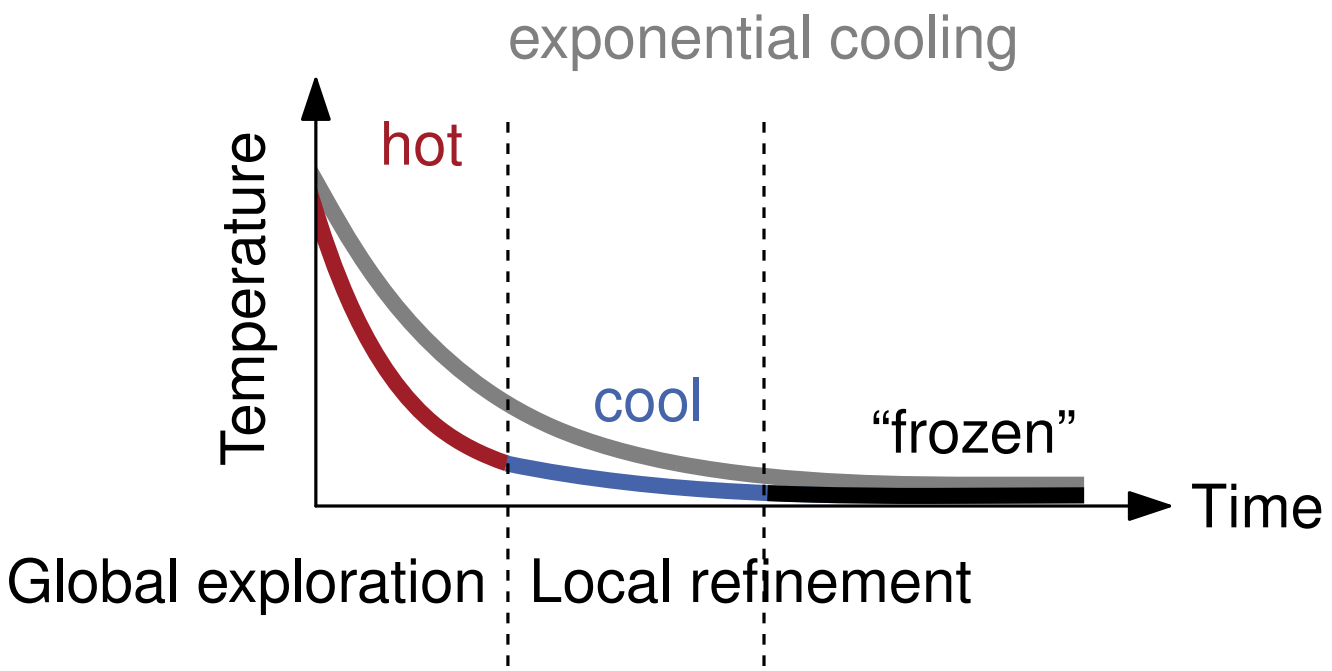
Observation:
- escaping deep local optimum takes **long time**



Dynamic Temperature Curve

temperature curve: parameter tuning difficult

Observation:
- escaping deep local optimum takes **long time**

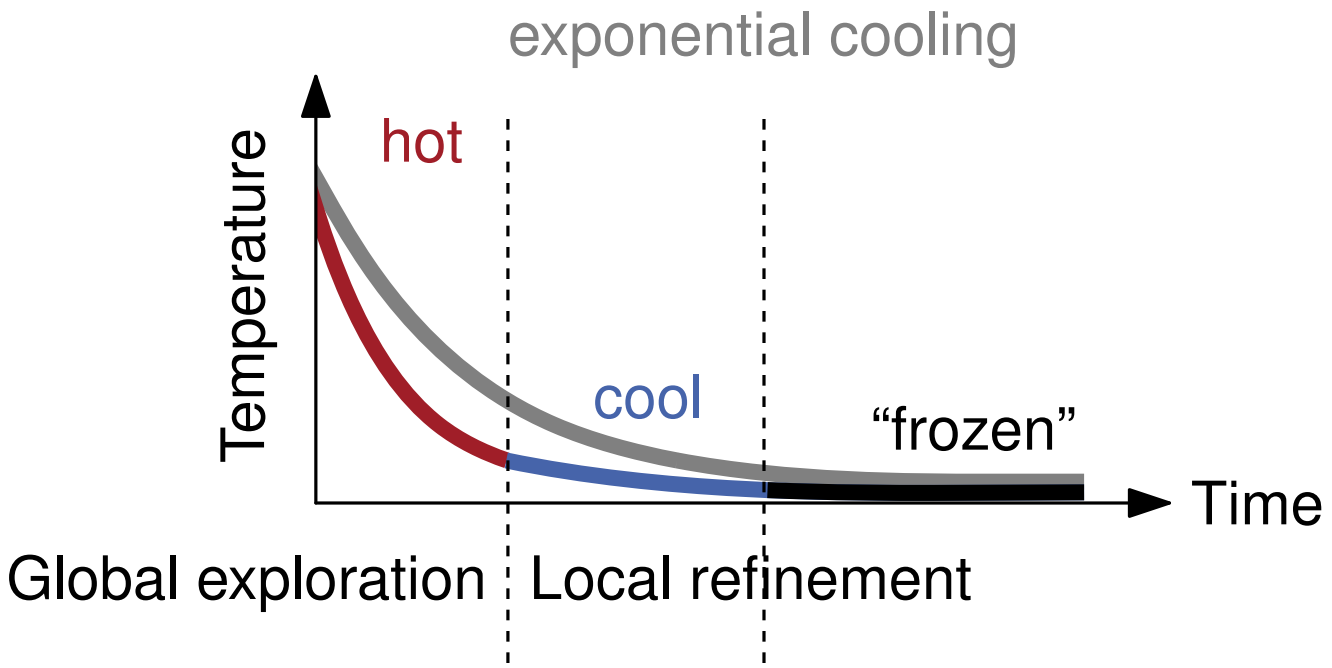


Dynamic Temperature Curve

temperature curve: parameter tuning difficult

Observation:
- escaping deep local optimum takes **long time**

Idea:
- adjust temperature drop velocity to **activity**
- **activity =** avg. probability for accepting worse solution

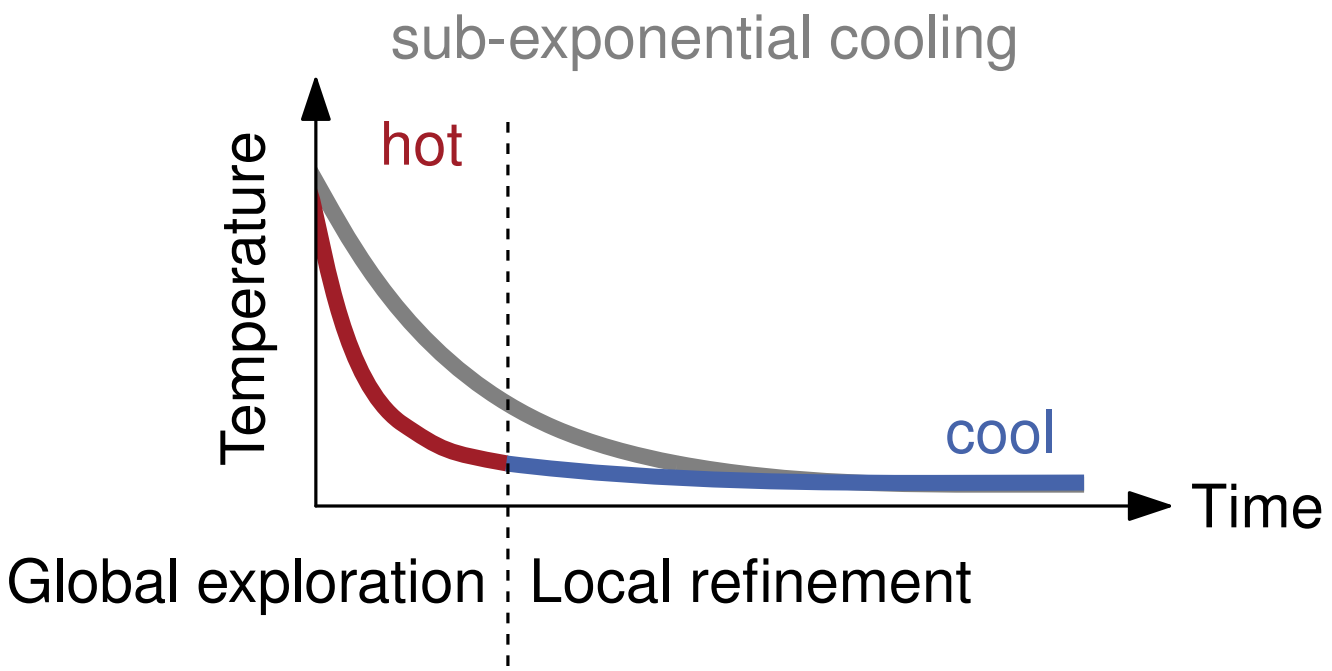


Dynamic Temperature Curve

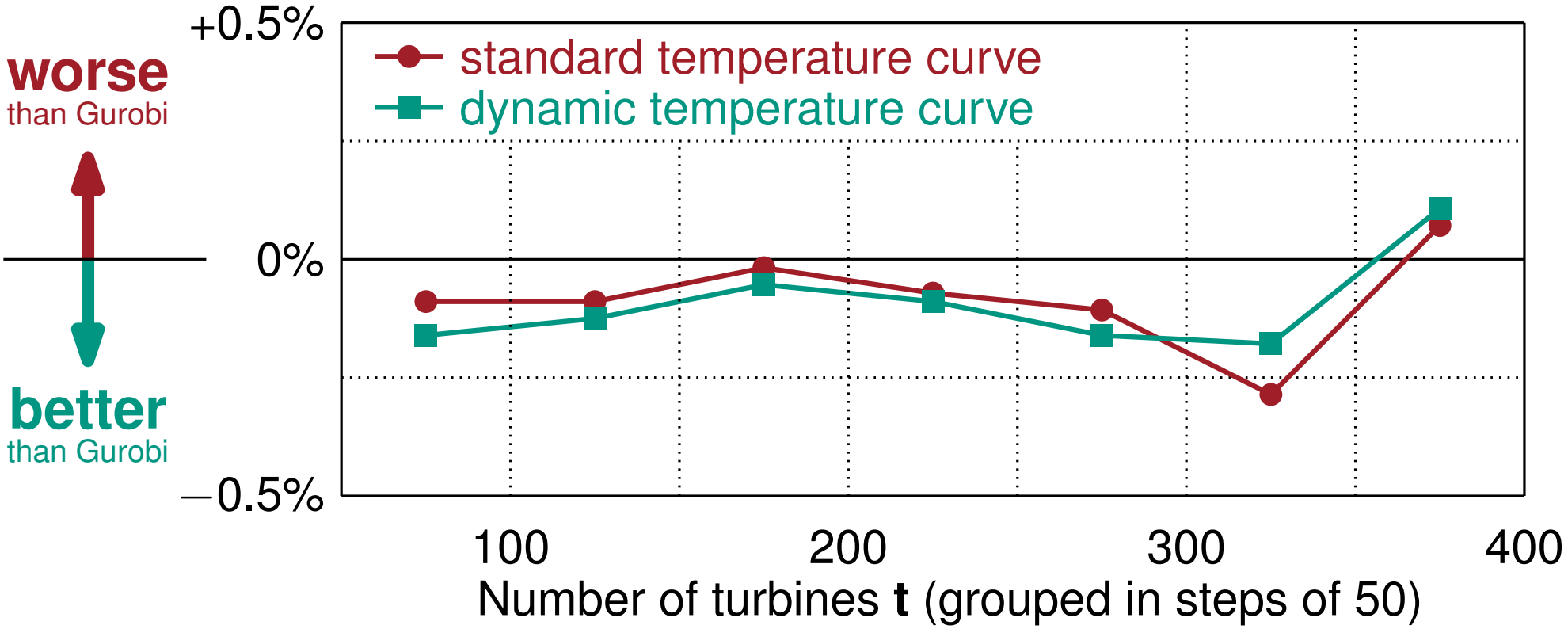
temperature curve: parameter tuning difficult

Observation:
- escaping deep local optimum takes **long time**

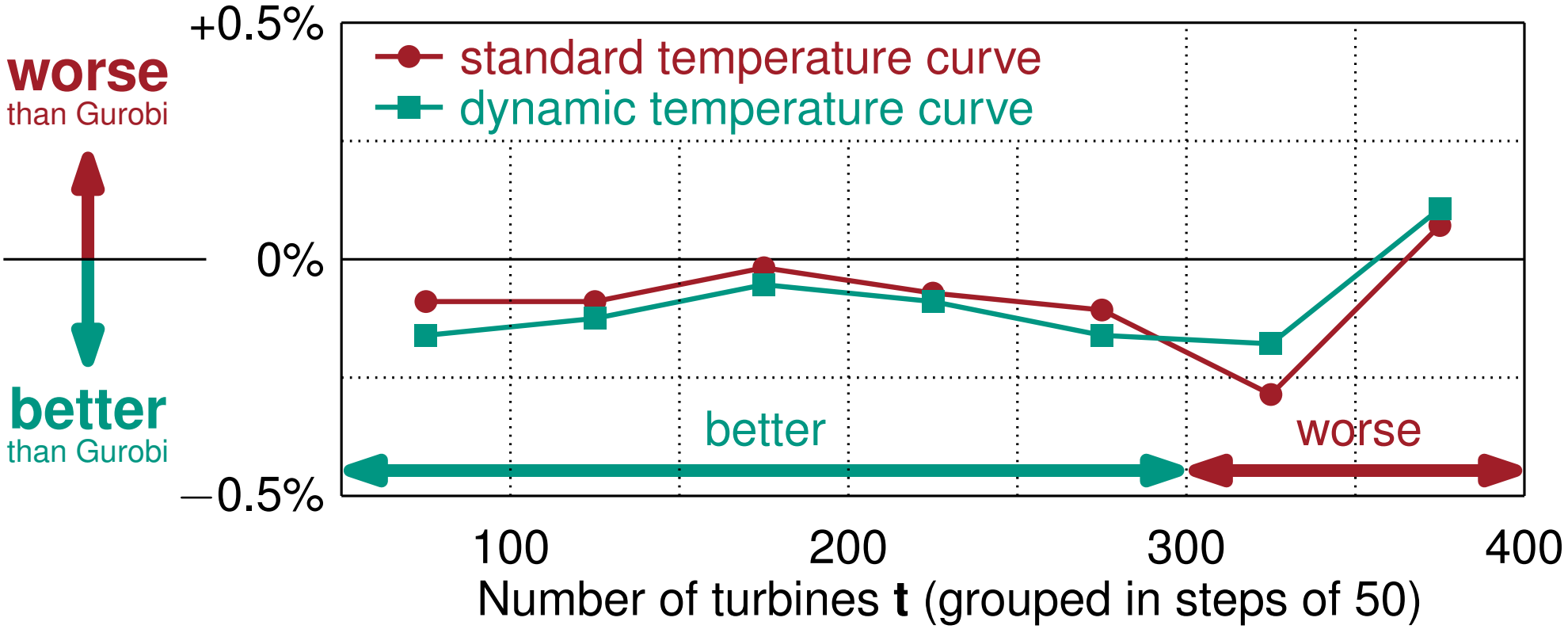
Idea:
- adjust temperature drop velocity to **activity**
- **activity =** avg. probability for accepting worse solution



Dynamic Temperature Curve



Dynamic Temperature Curve



Multiple Runs

result depends on **random seed**

Institute of Theoretical Informatics
Algorithmics Group

Multiple Runs

result depends on **random seed**

Idea:

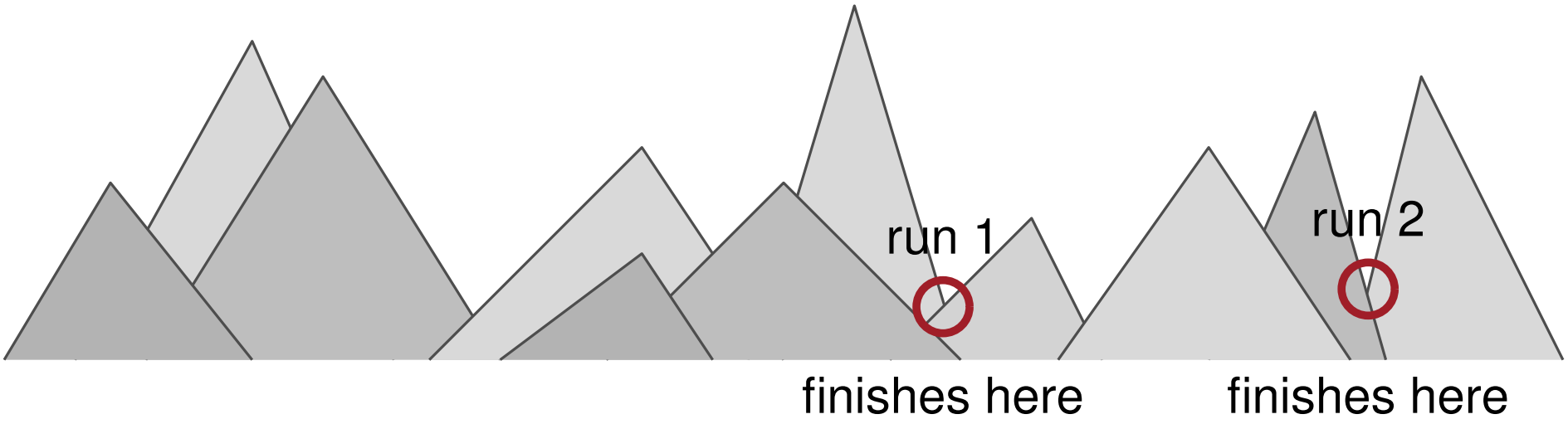
- start same algorithm **multiple times**
- each with different **random seed**

Multiple Runs

result depends on **random seed**

Idea:

- start same algorithm **multiple times**
- each with different **random seed**

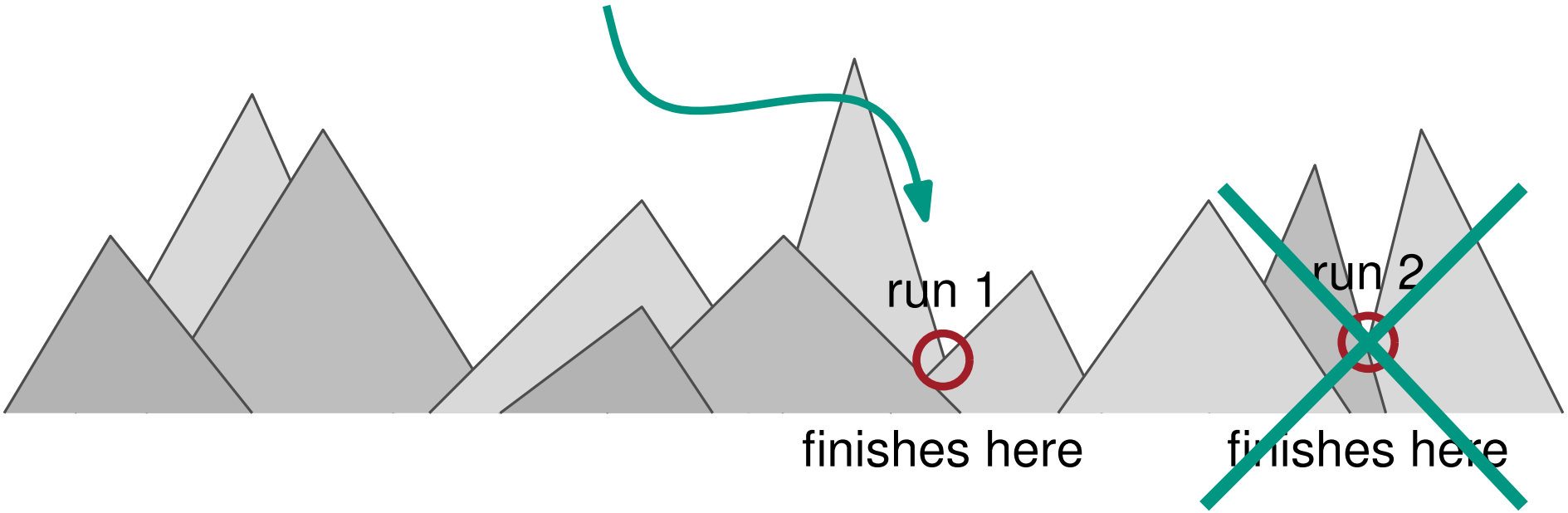


Multiple Runs

result depends on **random seed**

Idea:

- start same algorithm **multiple times**
- each with different **random seed**
- final result = **best** run



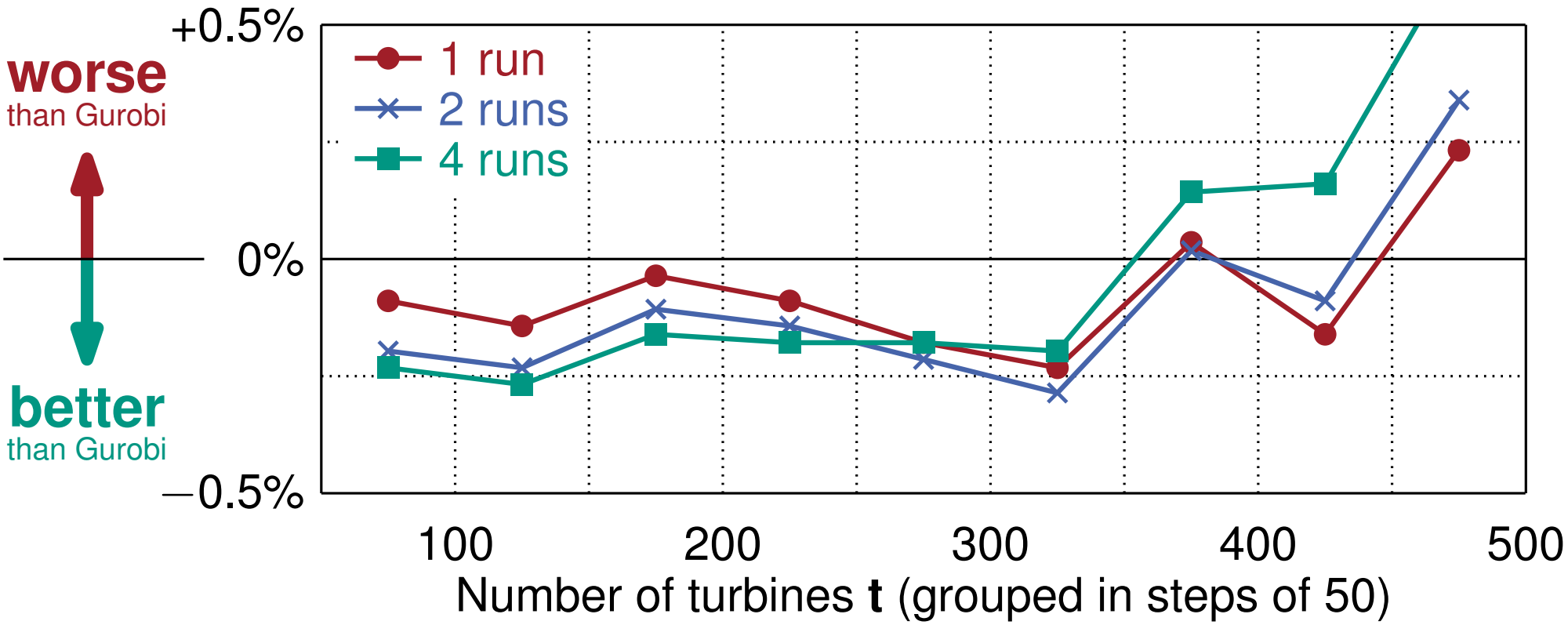
Multiple Runs

result depends on **random seed**

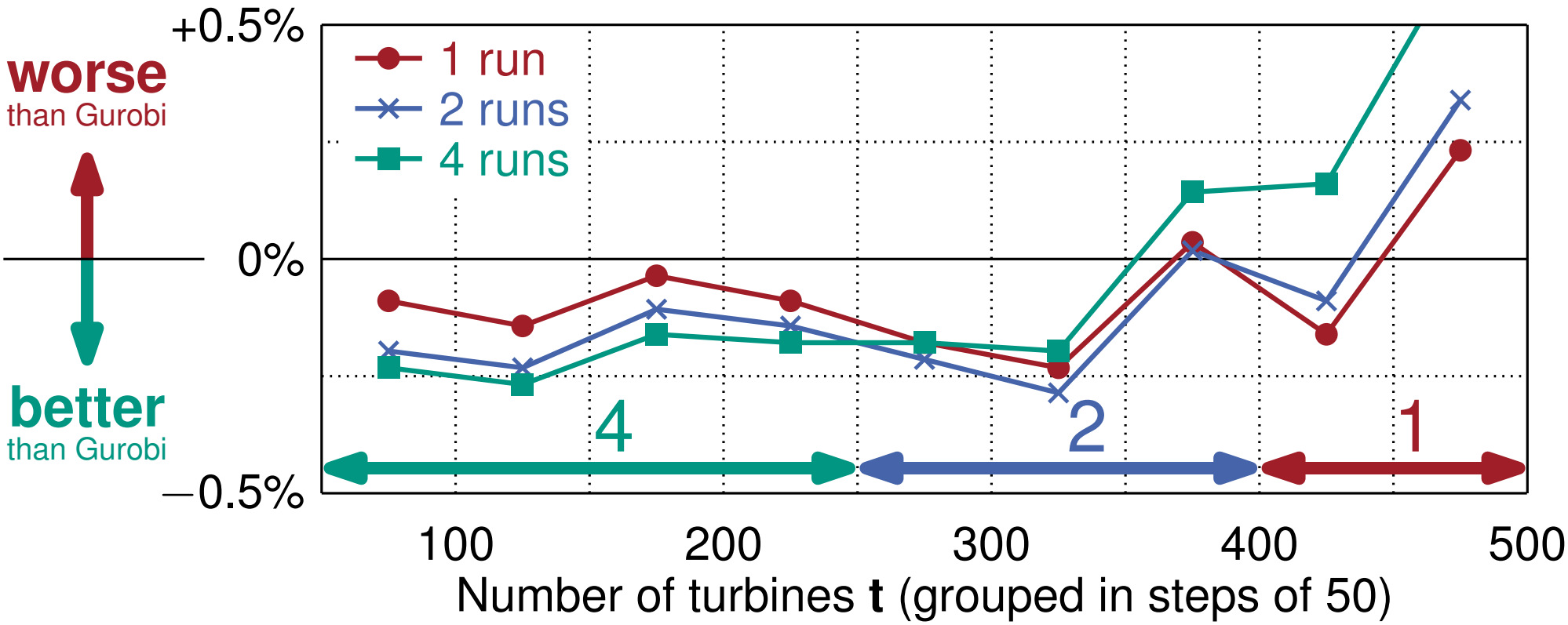
Idea:

- start same algorithm **multiple times**
- each with different **random seed**
- final result = **best** run
- distribute **available time** evenly

Multiple Runs



Multiple Runs



Two-Level Approach

bad results for **tight** substation capacities / **many** substations

Observation:

- has difficulties assigning turbines $\rightarrow$ substations

Two-Level Approach

bad results for **tight** substation capacities / **many** substations

Observation:
- has difficulties assigning turbines $\rightarrow$ substations

Idea: (1) **partition** into substation networks

(2) **optimize** each separately

Two-Level Approach

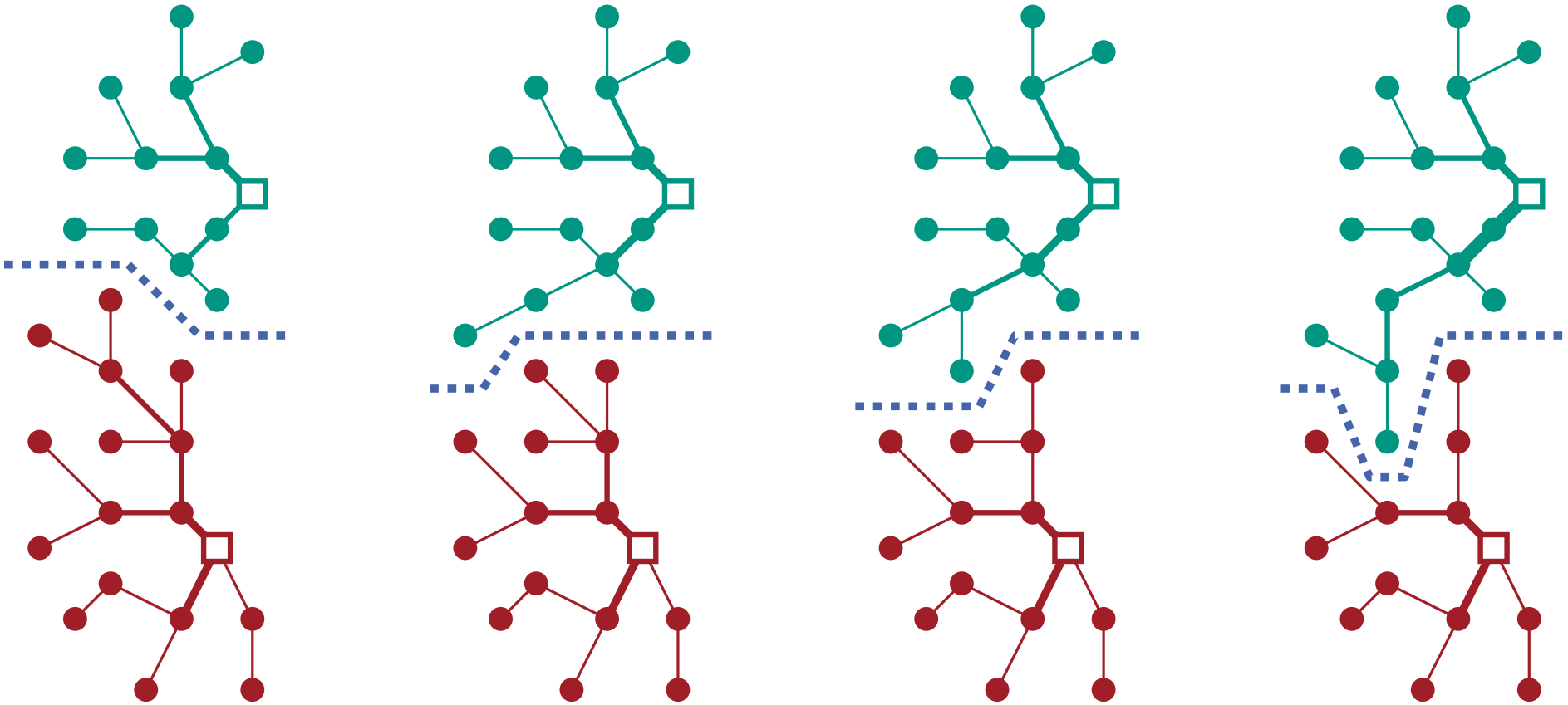
bad results for **tight** substation capacities / **many** substations

Observation:
- has difficulties assigning turbines $\rightarrow$ substations

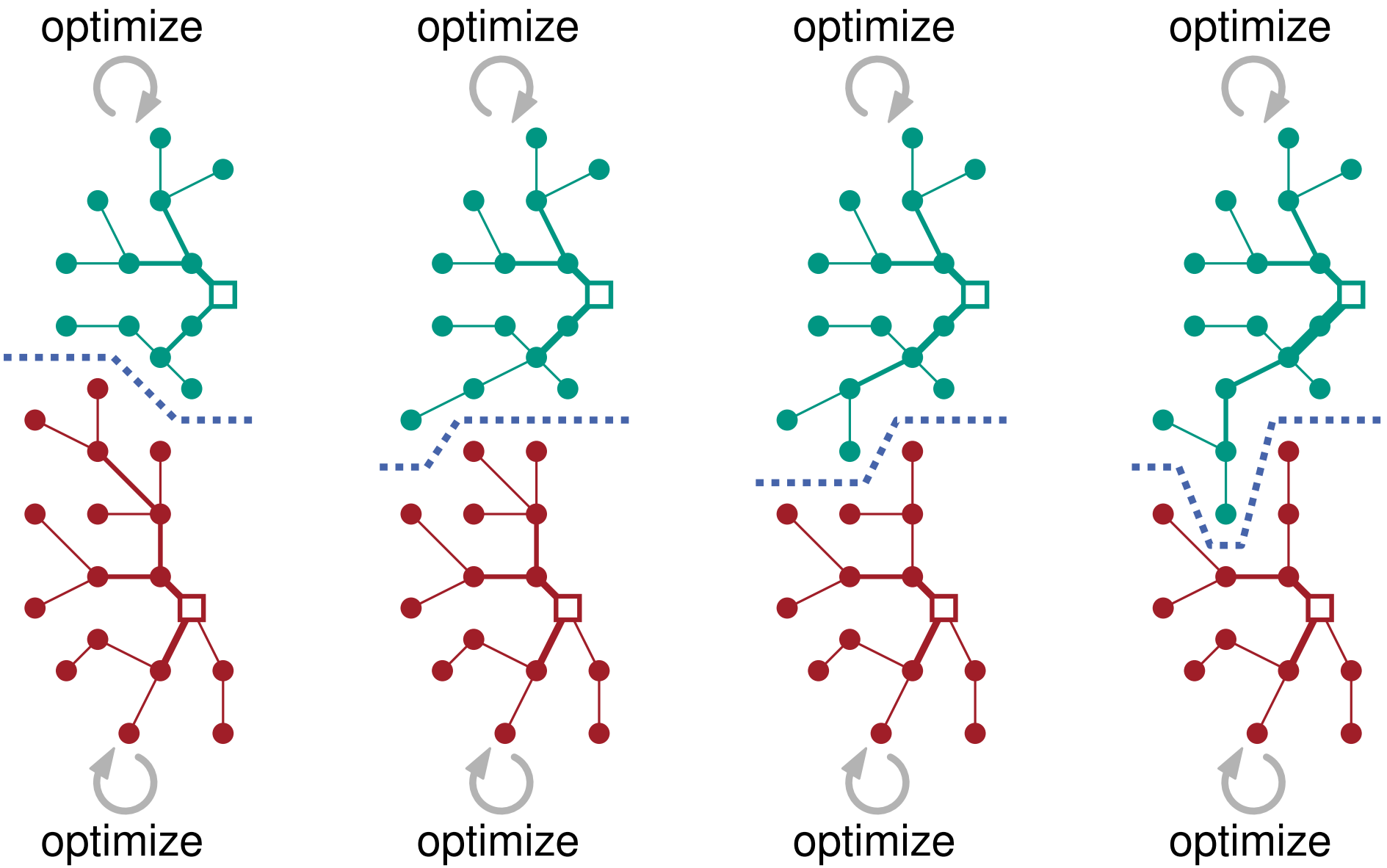
Idea: (1) **partition** into substation networks *in different ways*

(2) **optimize** each separately *for each partitioning*

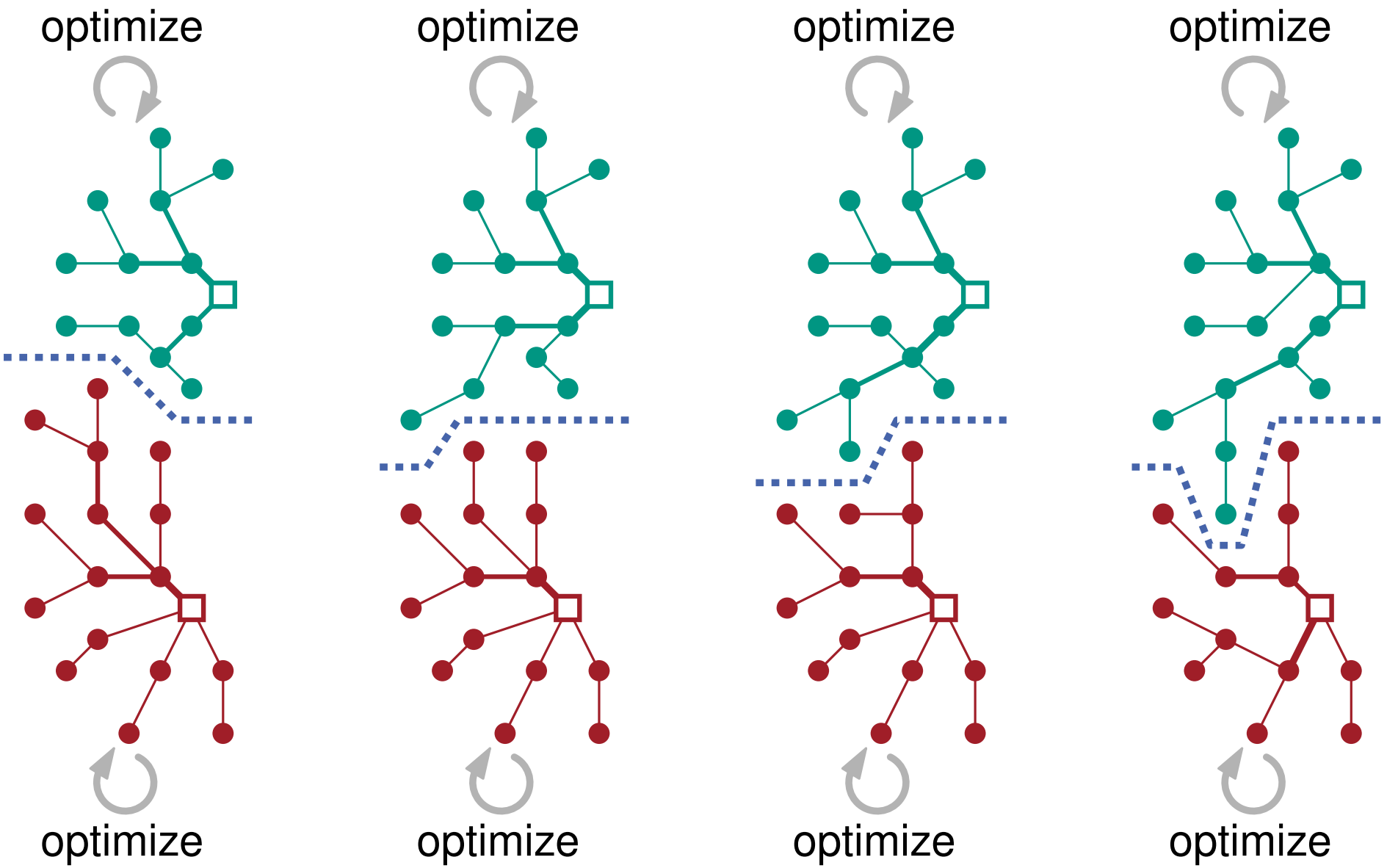
Two-Level Approach



Two-Level Approach

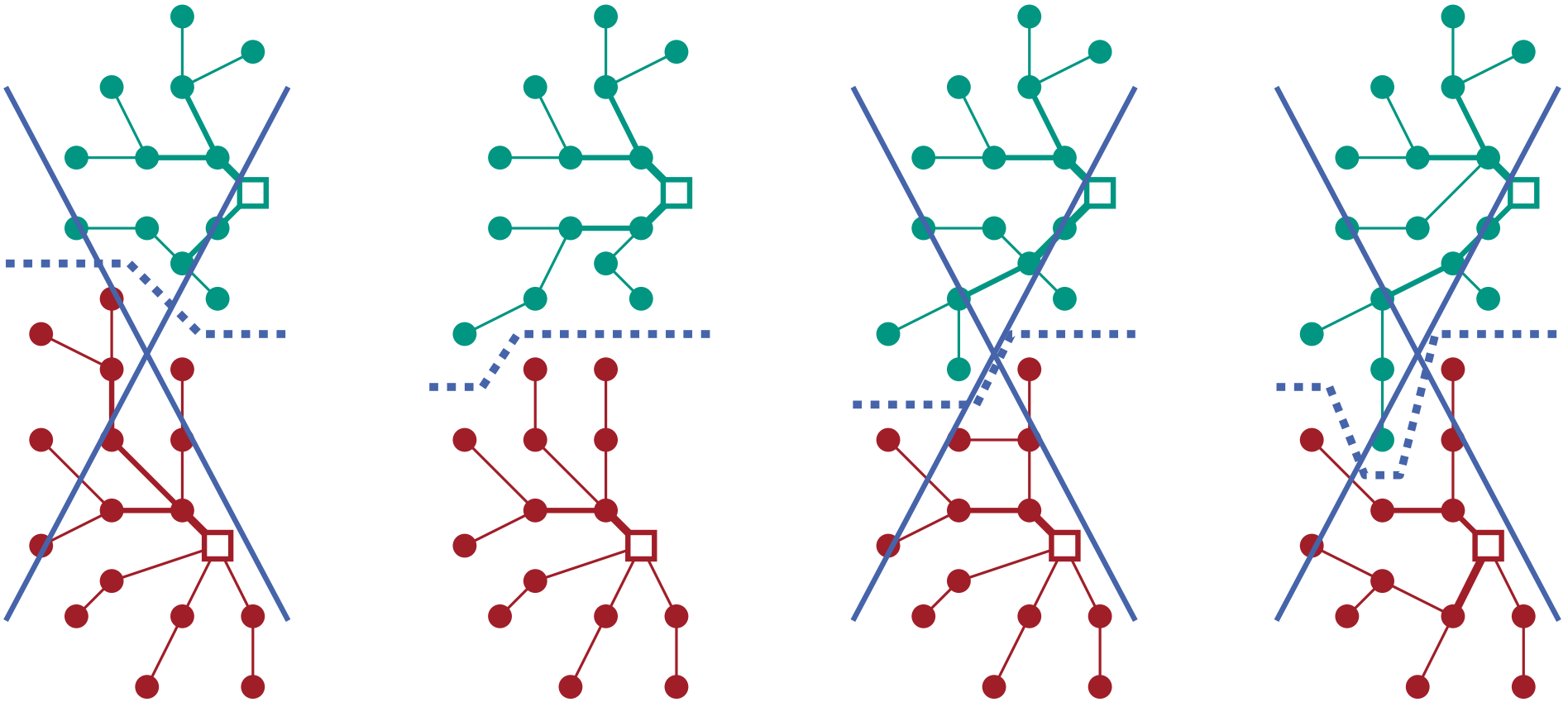


Two-Level Approach



Two-Level Approach

best final result



Two-Level Approach — Partitioning

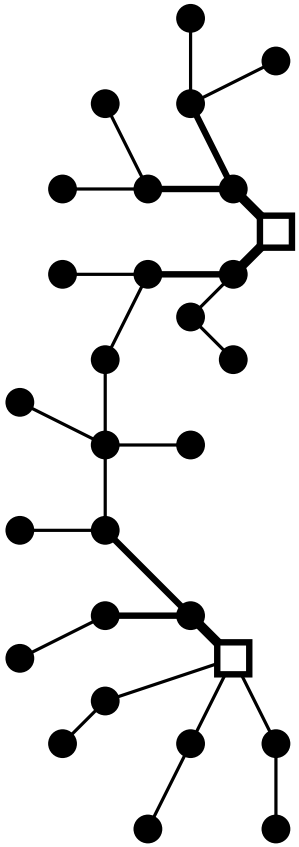
Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

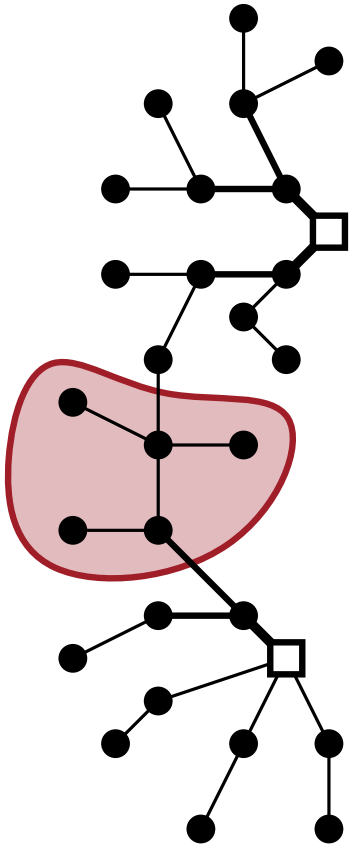
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

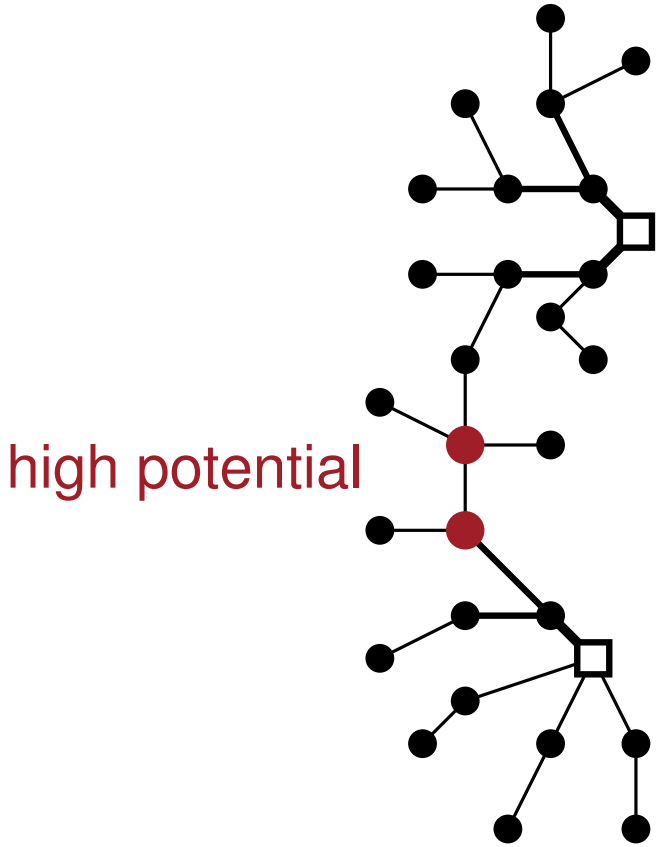
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

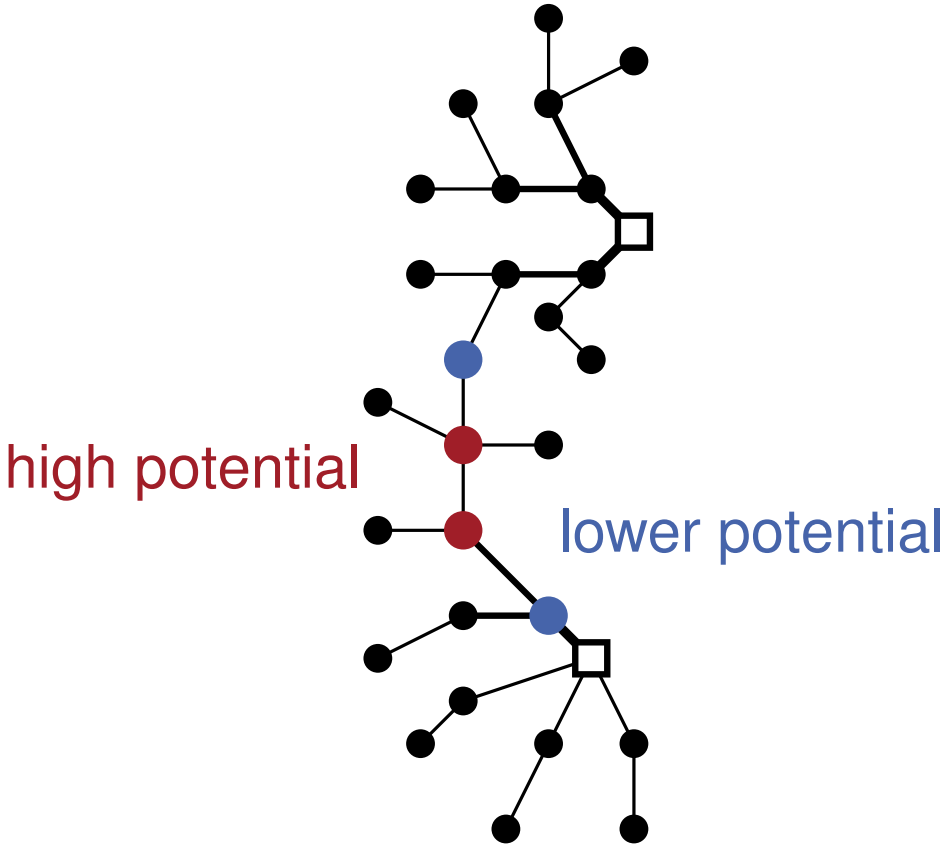
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

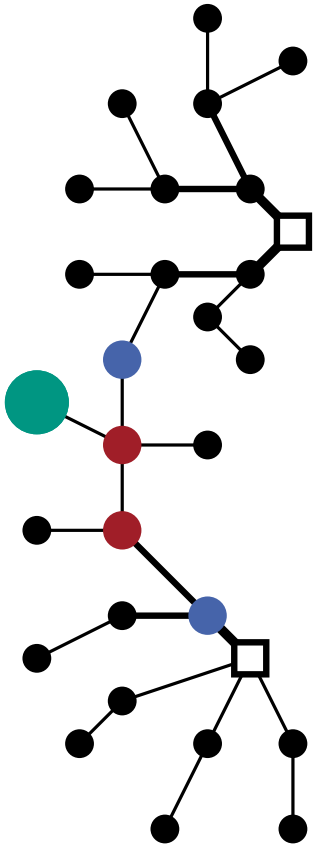
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

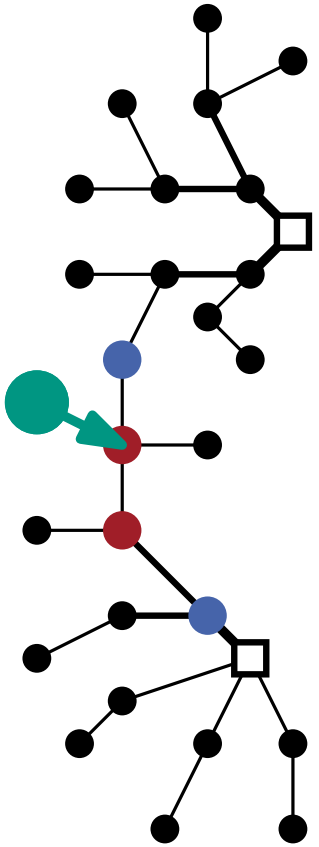
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

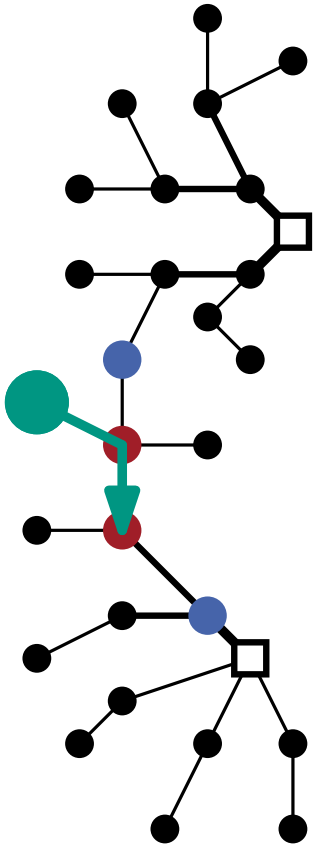
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

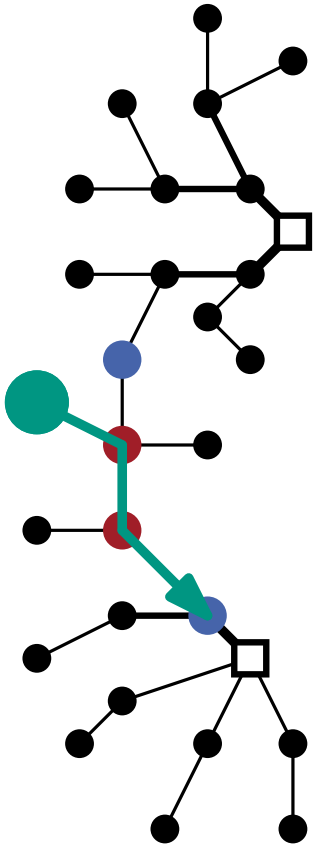
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

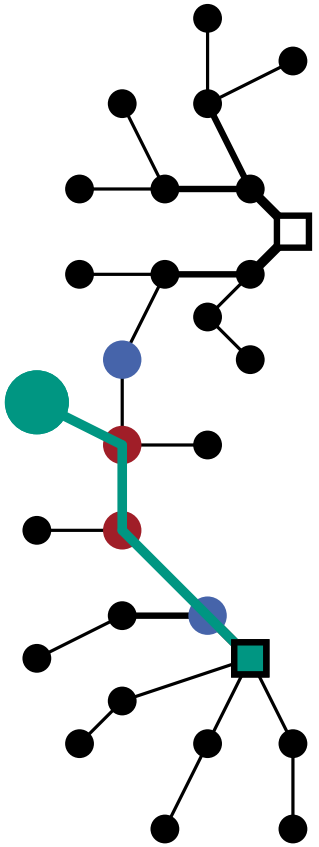
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

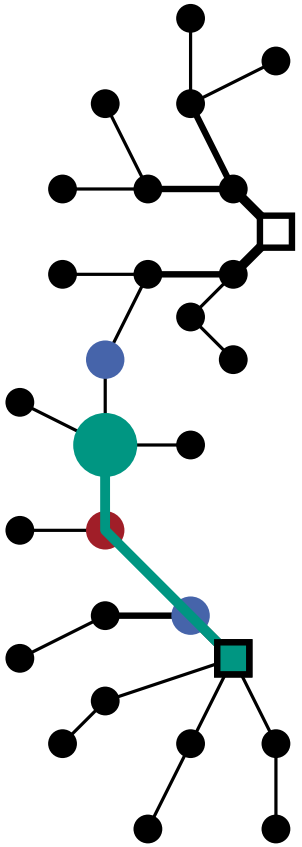
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

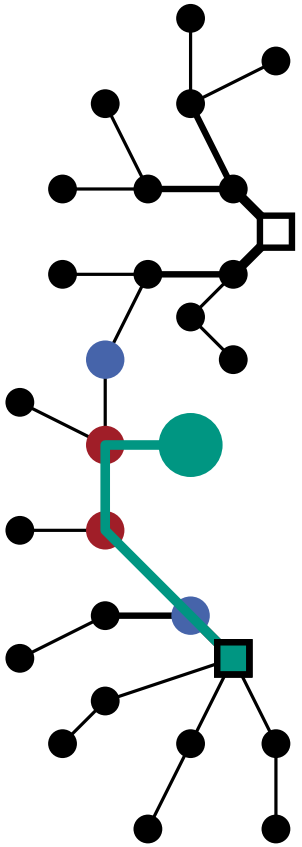
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

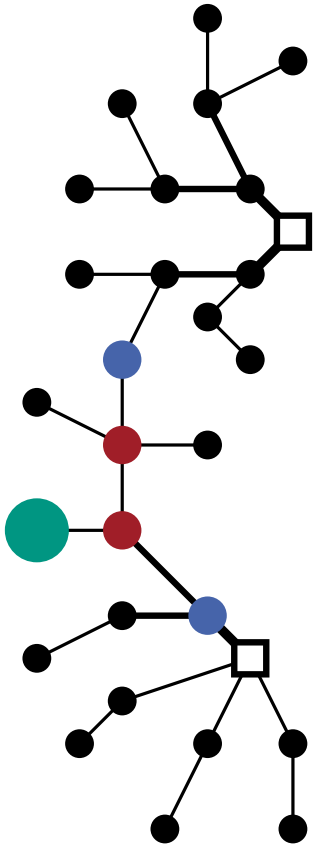
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

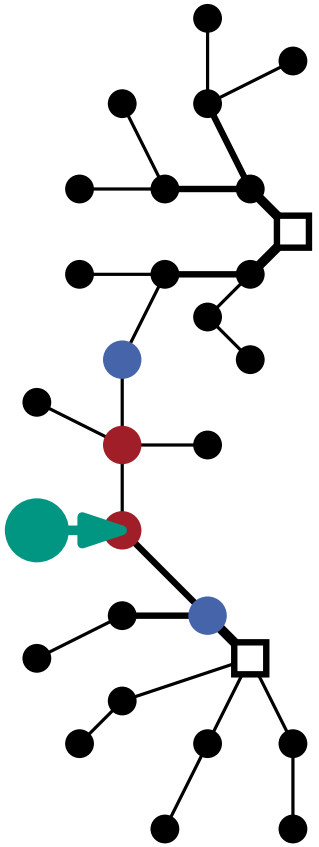
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

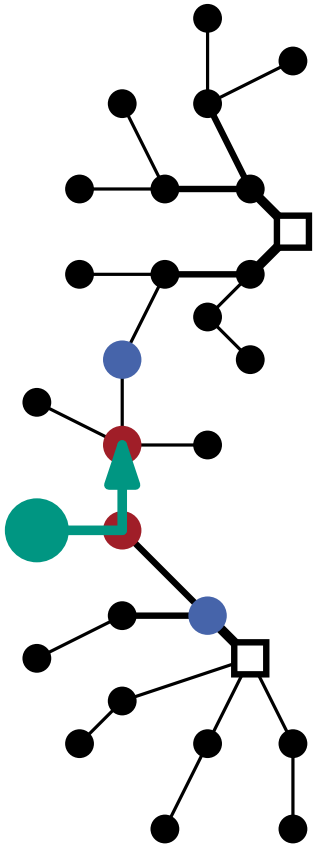
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

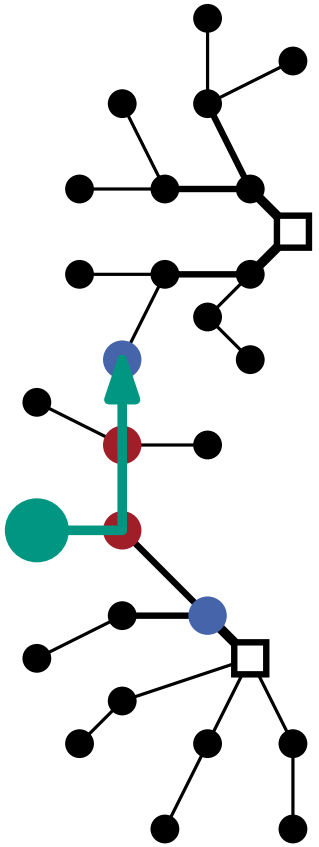
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

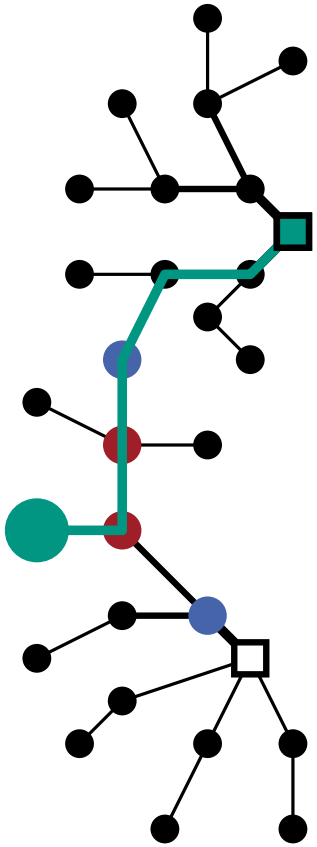
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

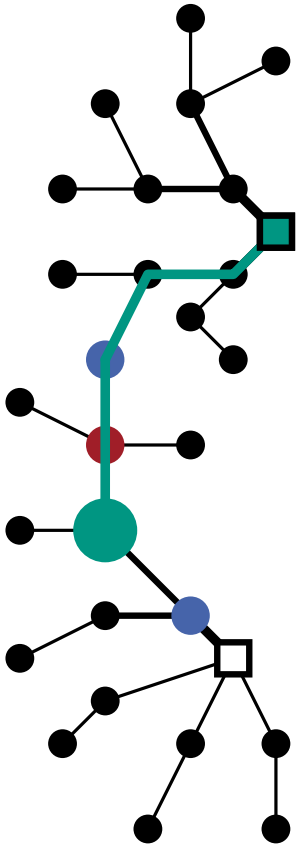
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

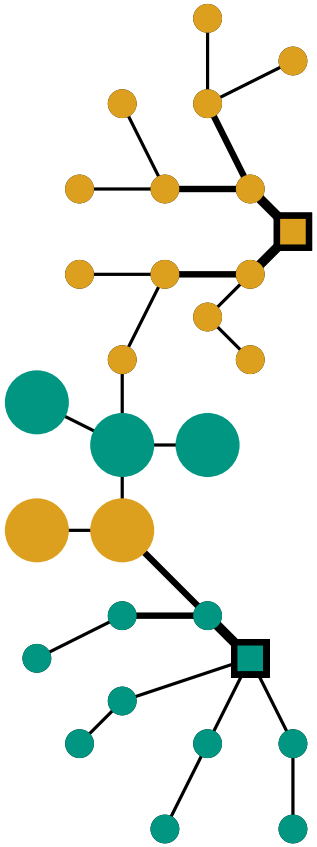
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

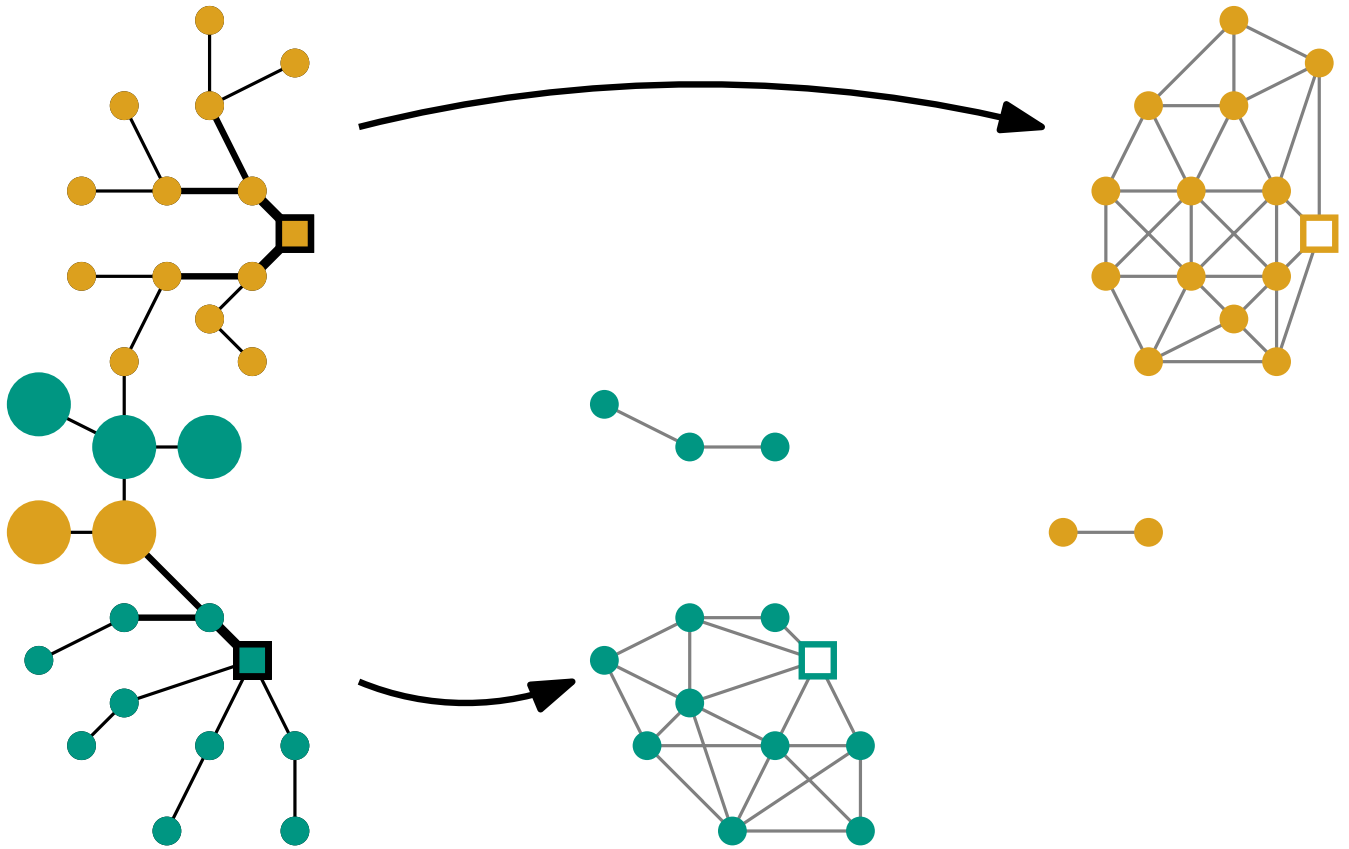
But:
- subgraphs often not connected!



Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

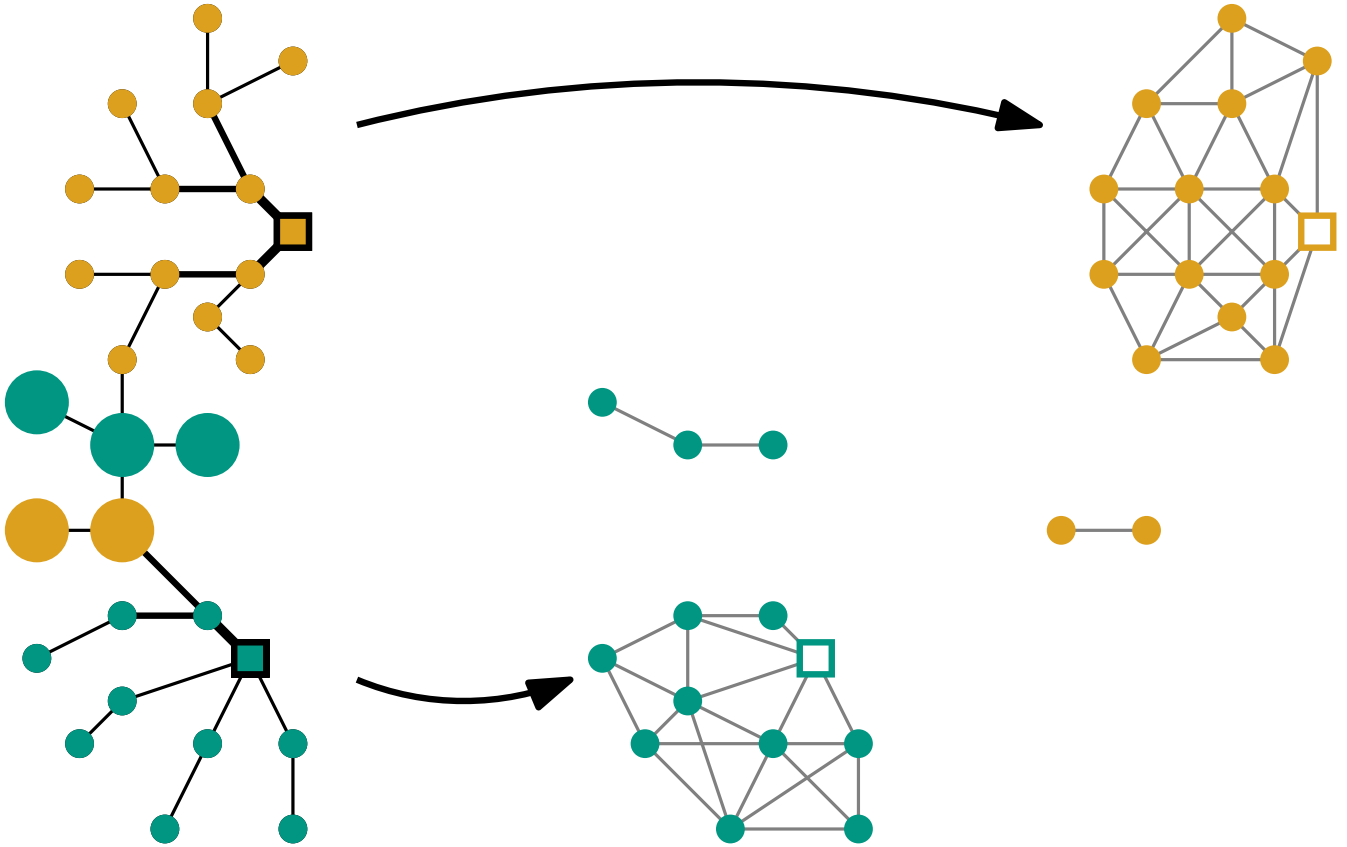


Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using
turbine paths

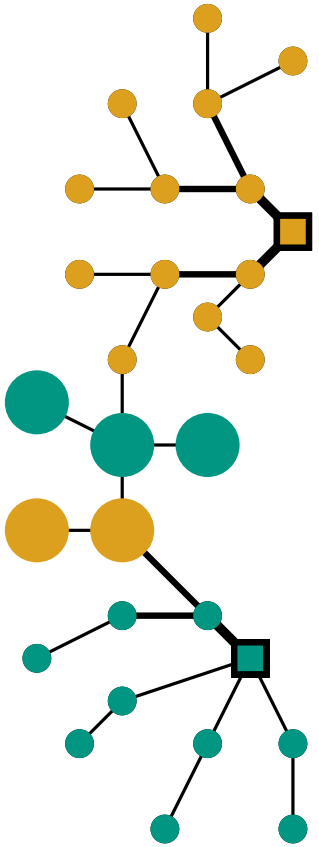


Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables

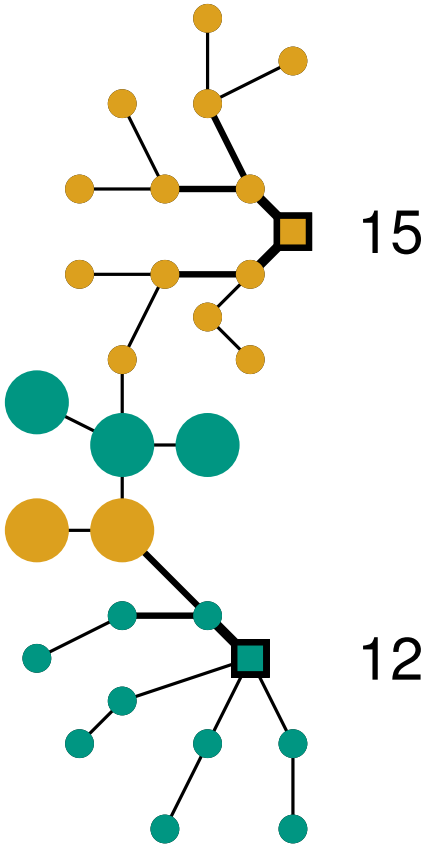


Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables



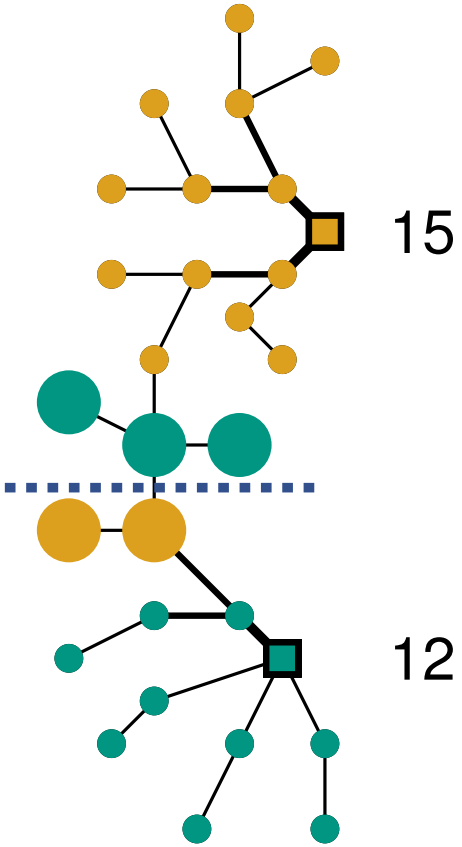
substation cap.: 15

Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables



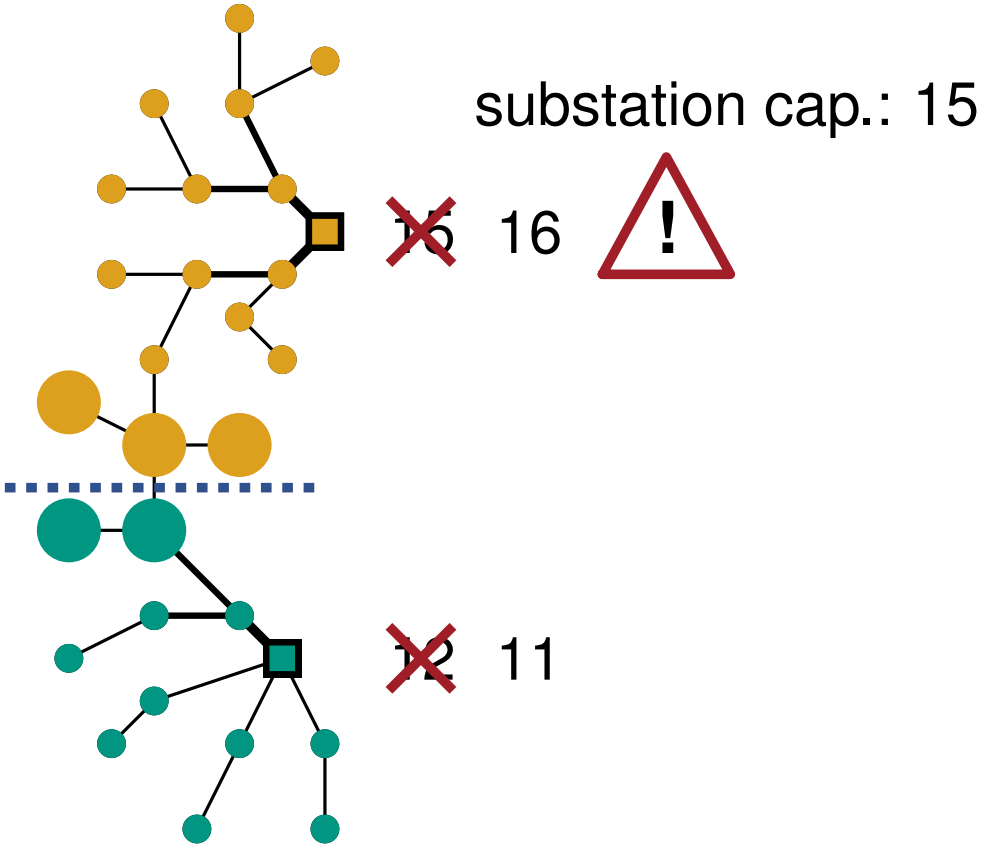
substation cap.: 15

Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables

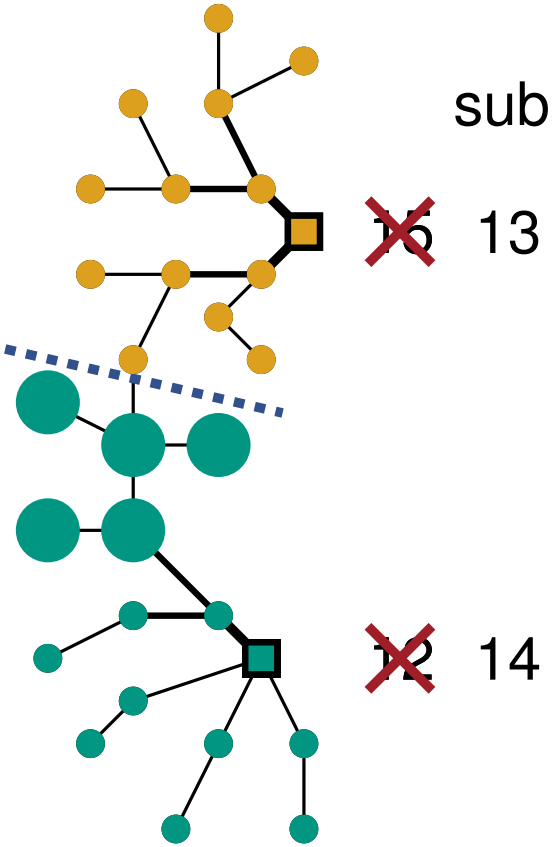


Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables



substation cap.: 15

~~16~~ 13

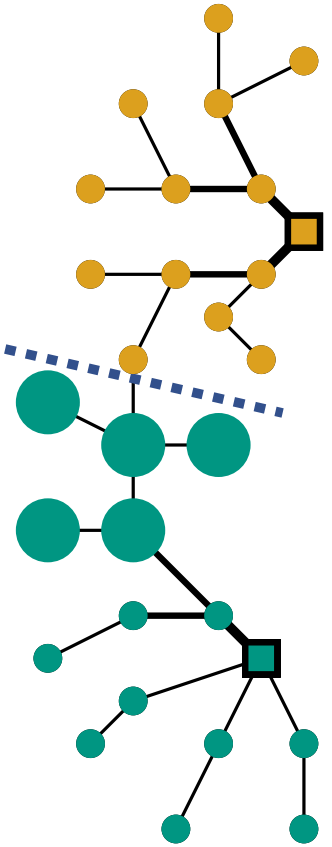
~~12~~ 14

Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables



substation cap.: 15

~~16~~ 13

... changed original assignment ...

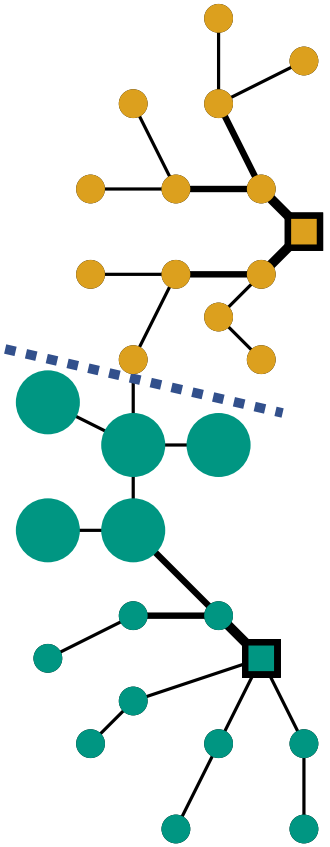
~~12~~ 14

Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

using cables



substation cap.: 15

~~16~~ 13

... changed original assignment ...

~~12~~ 14

Two-Level Approach — Partitioning

Karlsruhe Institute of Technology

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

Alternatives:
- Fuzzy C-means using cables as metric
- use partitioning as representation in top level
  (decoding = optimize single substation networks)

Two-Level Approach — Partitioning

Our idea:
- use substation assignment based on **turbine paths**

But:
- subgraphs often not connected!

Alternatives:
- Fuzzy C-means using cables as metric
- use partitioning as representation in top level
  (decoding = optimize single substation networks)

Future Work

Other Tings We've Tried

- Recover when caught in local optimum
  Problem: Recover to which state?

Institute of Theoretical Informatics
Algorithmics Group

Other Tings We've Tried

- Recover when caught in local optimum
  Problem: Recover to which state?
- Cancel runs when caught in local optimum
  Problem: How detect this?

Institute of Theoretical Informatics
Algorithmics Group

Other Tings We've Tried

- Recover when caught in local optimum
  Problem: Recover to which state?
- Cancel runs when caught in local optimum
  Problem: How detect this?
- Hybrid Simulated Annealing & Evolutionary Algorithm
  Problem: How cross two solutions?

Related Problems (Future Work)

- Optimize locations of substations
- Optimize types of substations (similar to types of cables)
- Allow adding merge points ($\Rightarrow$ Steiner Tree)
- Avoid intersections
- Redundant cables for failure safety