From Symmetry to Asymmetry: Generalizing TSP Approximations by Parametrization

Lukas Behrend[1], Katrin Casel[1], Tobias Friedrich[1], Gregor Lagondzinski[1], Andreas Löser[1], Marcus Wilhelm[2]

[1] HPI Hasso Plattner Institut Digital Engineering · Universität Potsdam

[2] KIT Karlsruhe Institute of Technology

From Symmetry to Asymmetry: Generalizing TSP Approximations by Parametrization

Lukas Behrend[1], Katrin Casel[1], Tobias Friedrich[1], Gregor Lagondzinski[1], Andreas Löser[1], Marcus Wilhelm[2]

[1] Hasso Plattner Institut, Digital Engineering · Universität Potsdam

[2] Karlsruhe Institute of Technology

From Symmetry to Asymmetry: Generalizing TSP Approximations by Parametrization

Lukas Behrend[1], Katrin Casel[1], Tobias Friedrich[1], Gregor Lagondzinski[1], Andreas Löser[1], Marcus Wilhelm[2]

Hasso Plattner Institut
Digital Engineering · Universität Potsdam [1]

Karlsruhe Institute of Technology [2]

The Traveling Salesman Problem

Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

The Traveling Salesman Problem

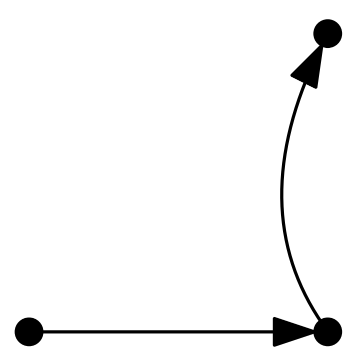
Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.

The Traveling Salesman Problem

Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

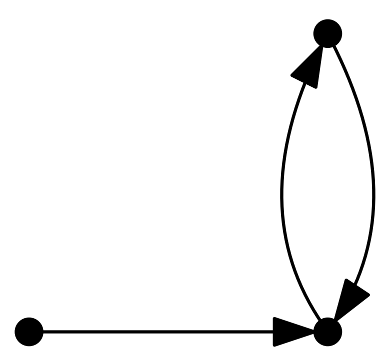
Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.



The Traveling Salesman Problem

Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

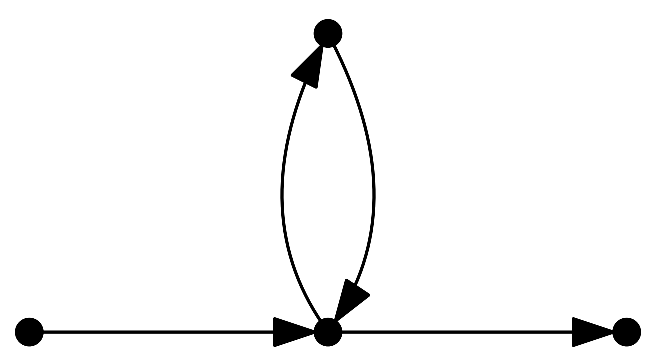
Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.



The Traveling Salesman Problem

Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

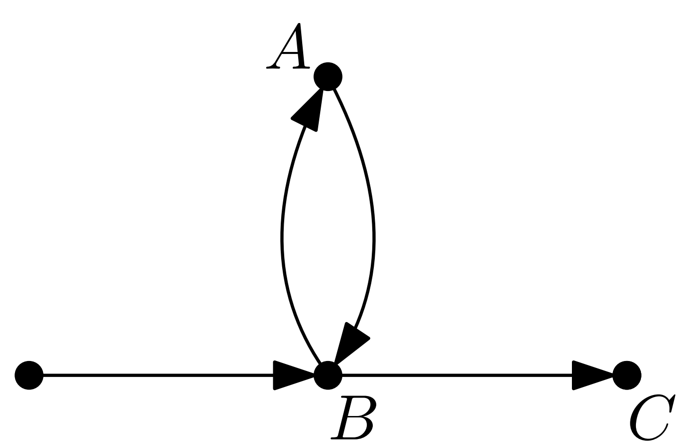
Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.



The Traveling Salesman Problem

Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

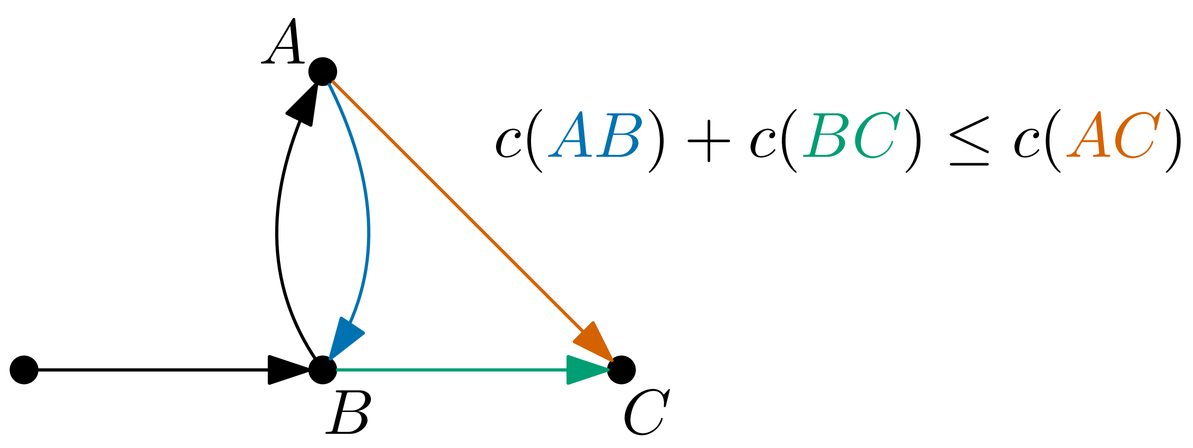
Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.



The Traveling Salesman Problem

Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

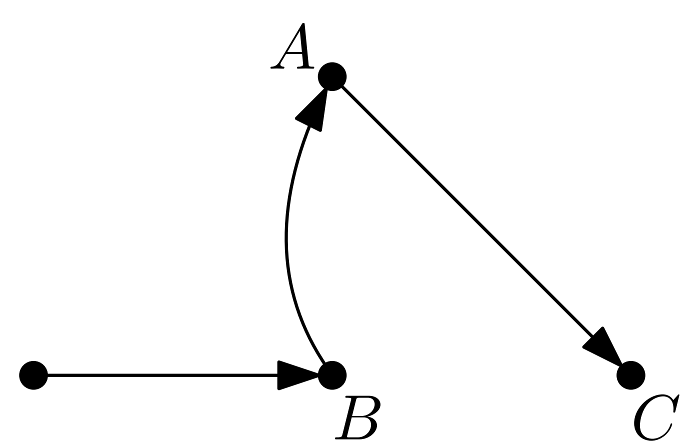
Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.



The Traveling Salesman Problem

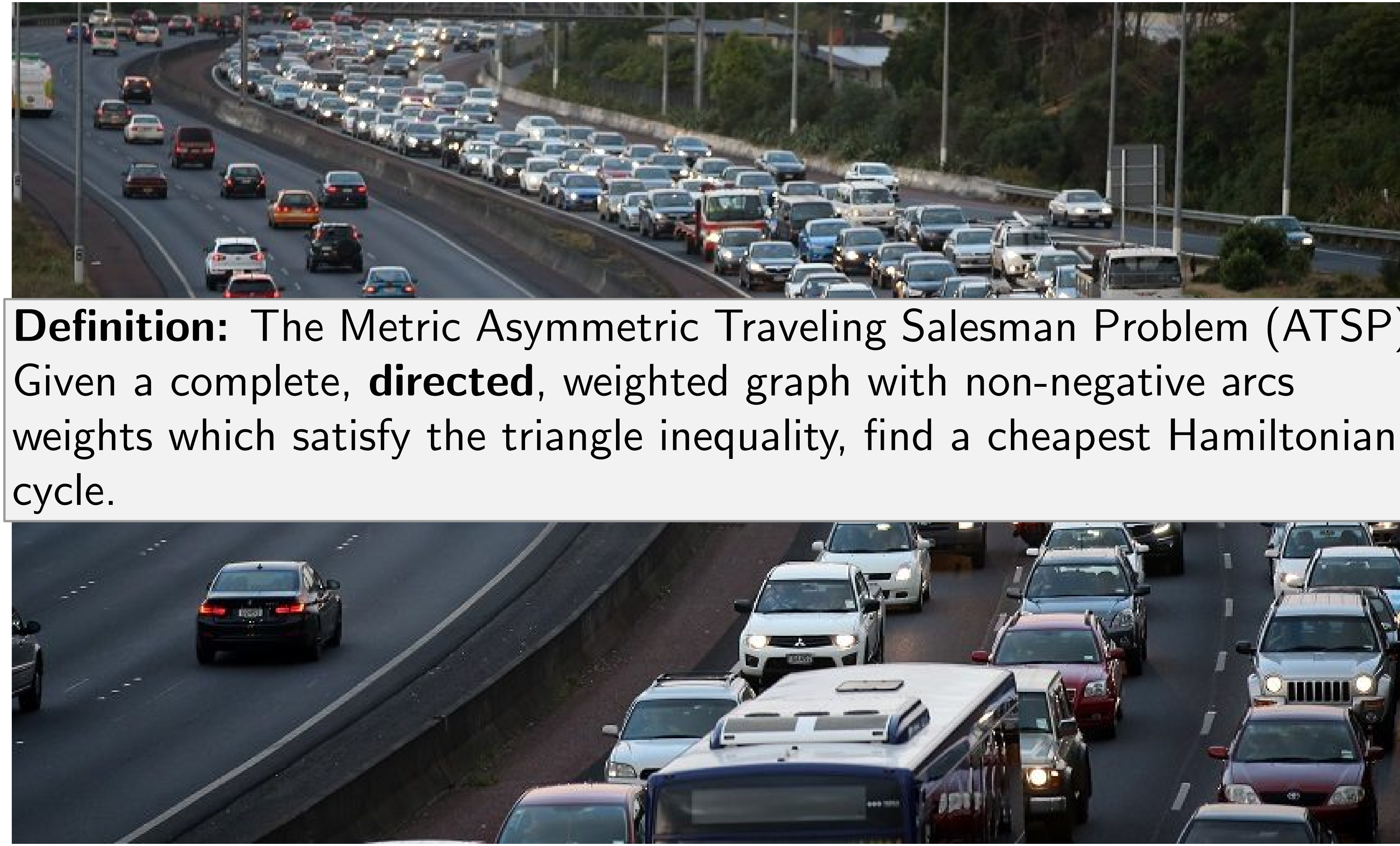
Definition: The Traveling Salesman Problem
Given a complete undirected graph with non-negative edge weights, find a cheapest Hamiltonian cycle.

Definition: The Metric Traveling Salesman Problem (TSP)
Given a complete, undirected, graph with non-negative edge weights **which satisfy the triangle inequality**, find a cheapest Hamiltonian cycle.



Asymmetries

Asymmetries



Definition: The Metric Asymmetric Traveling Salesman Problem (ATSP)
Given a complete, **directed**, weighted graph with non-negative arcs weights which satisfy the triangle inequality, find a cheapest Hamiltonian cycle.

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)

State of the Art

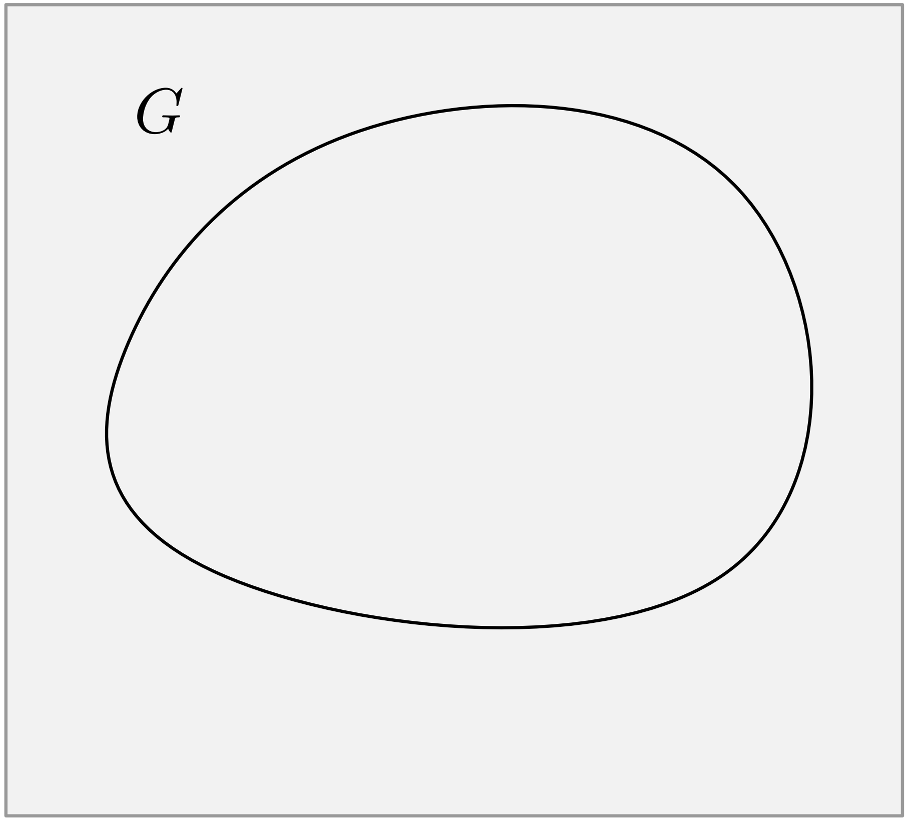
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

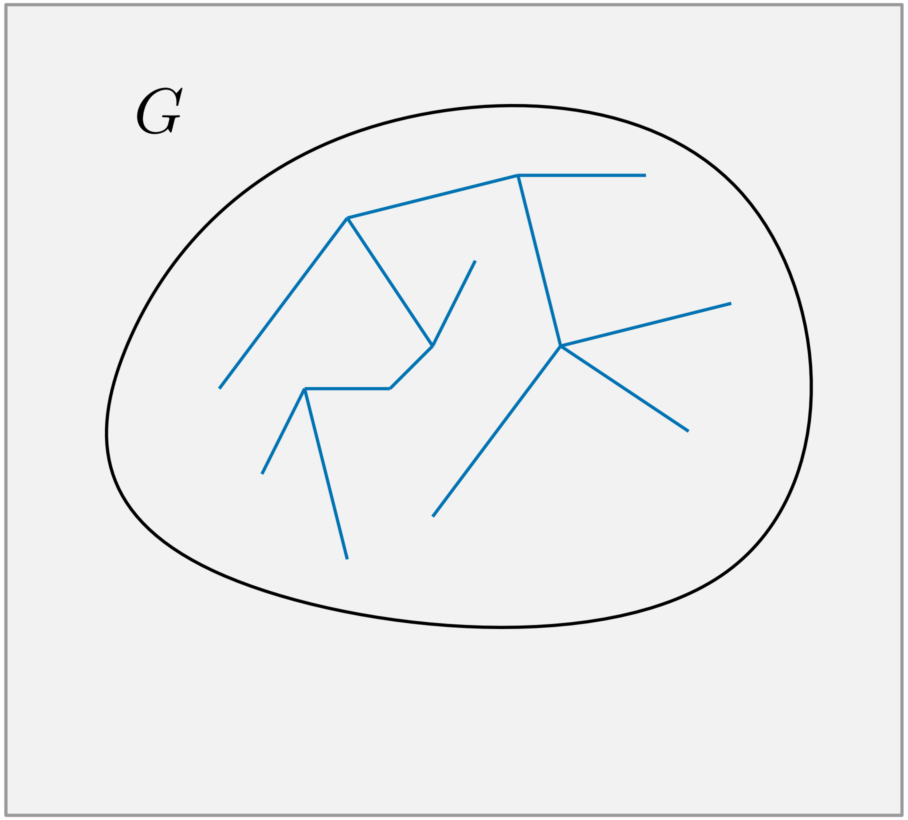
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

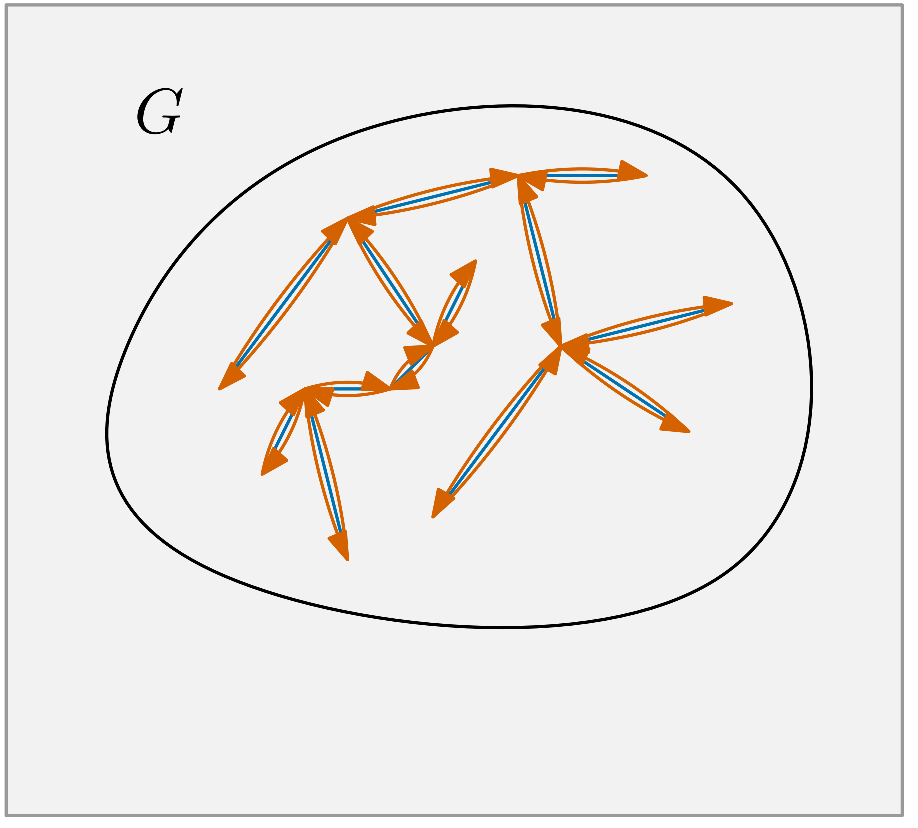
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

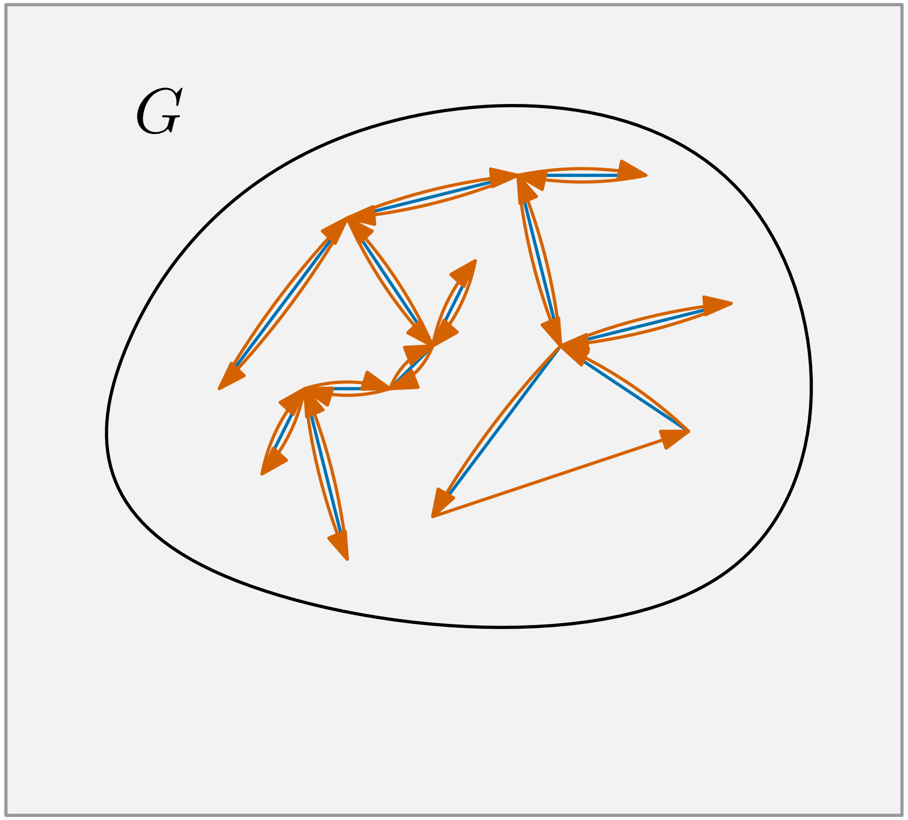
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

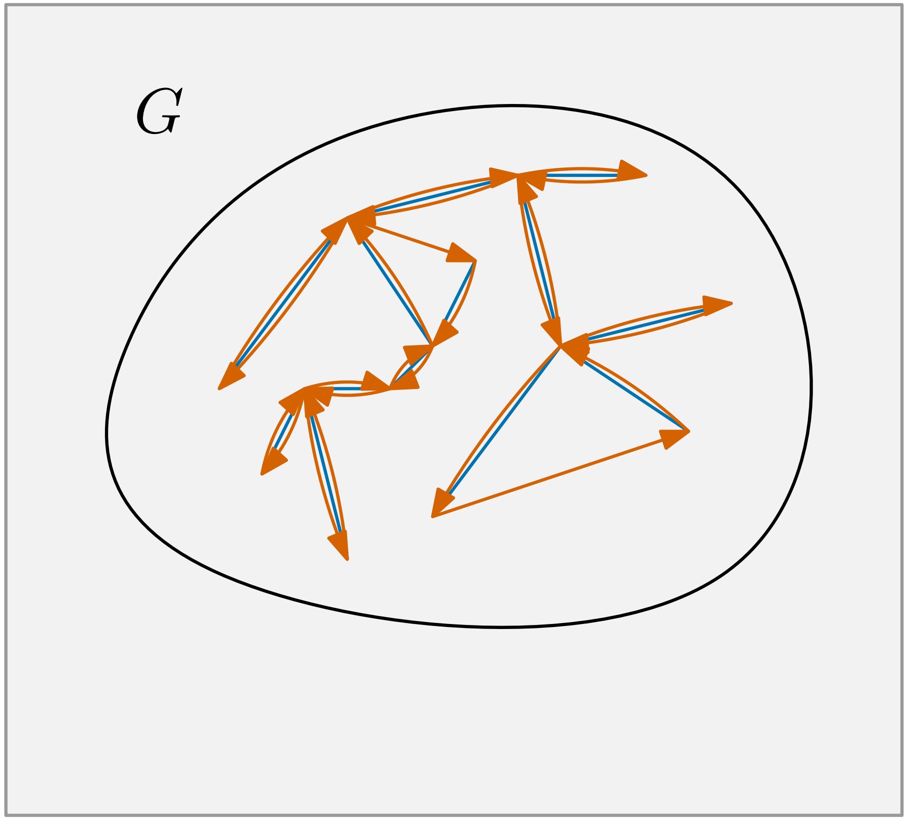
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

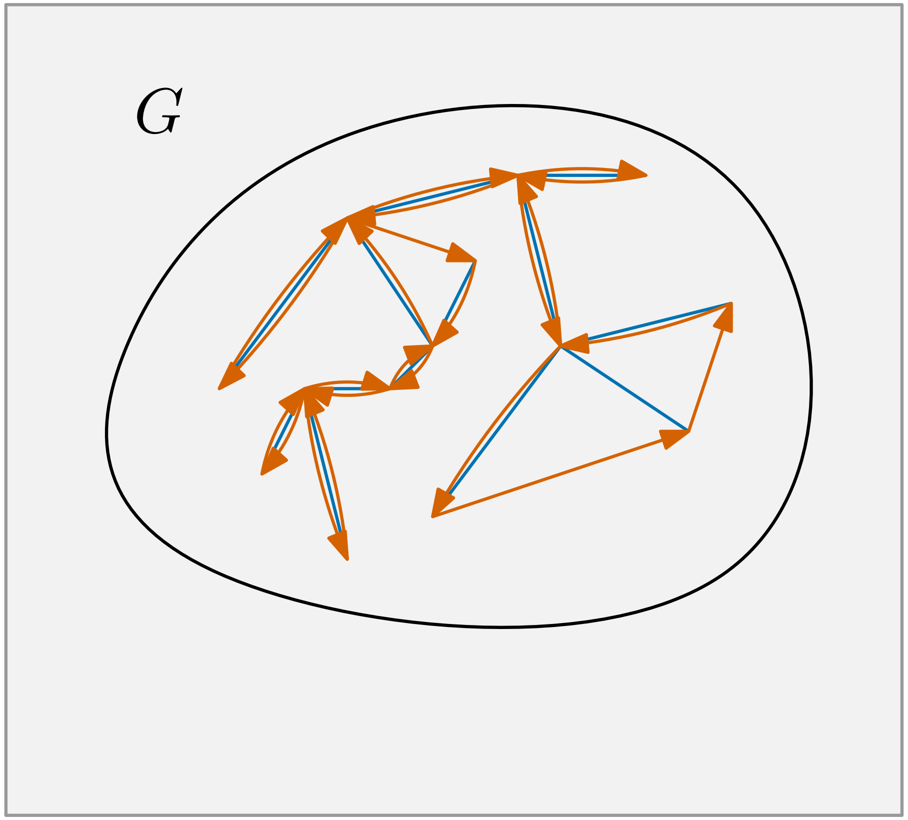
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

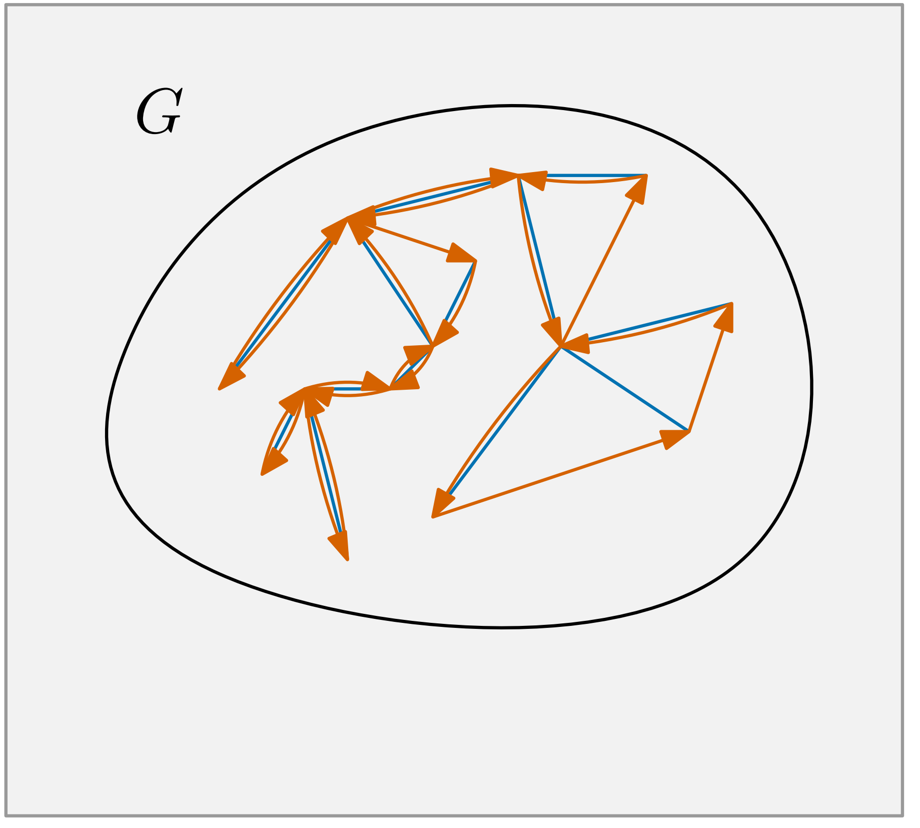
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

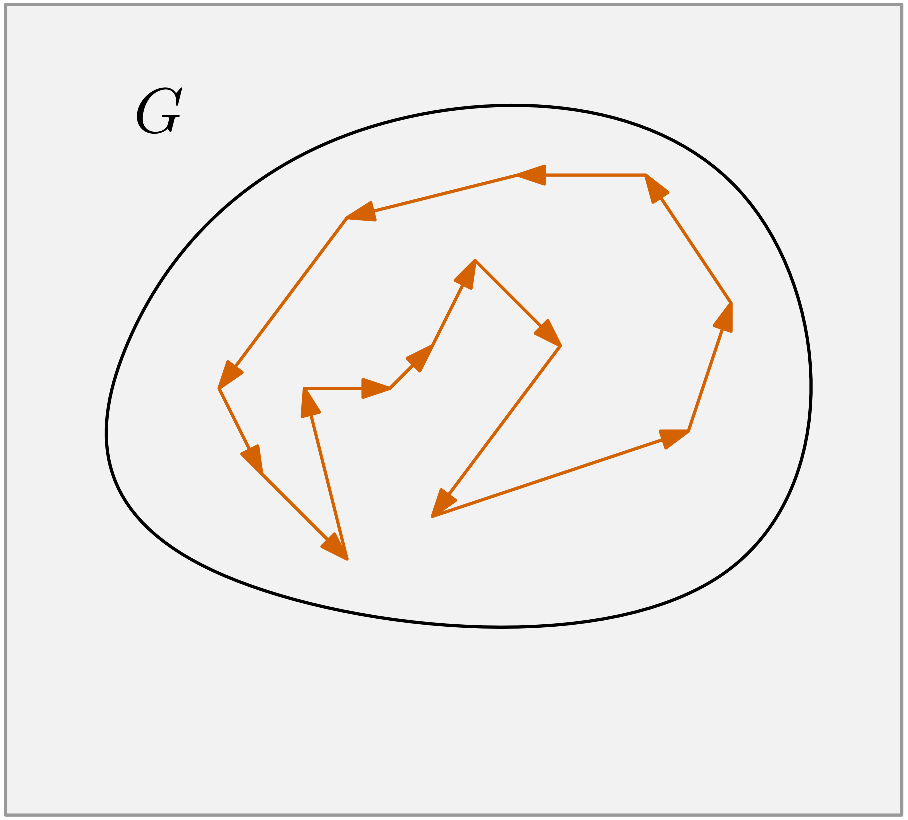
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)



State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)

State of the Art

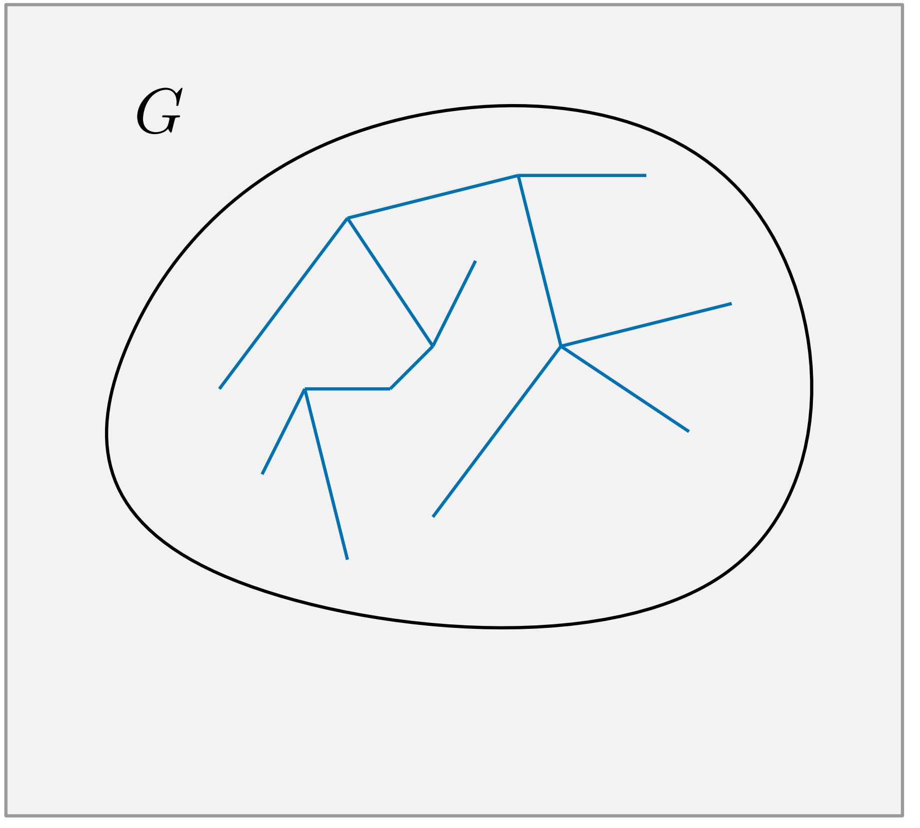
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)



State of the Art

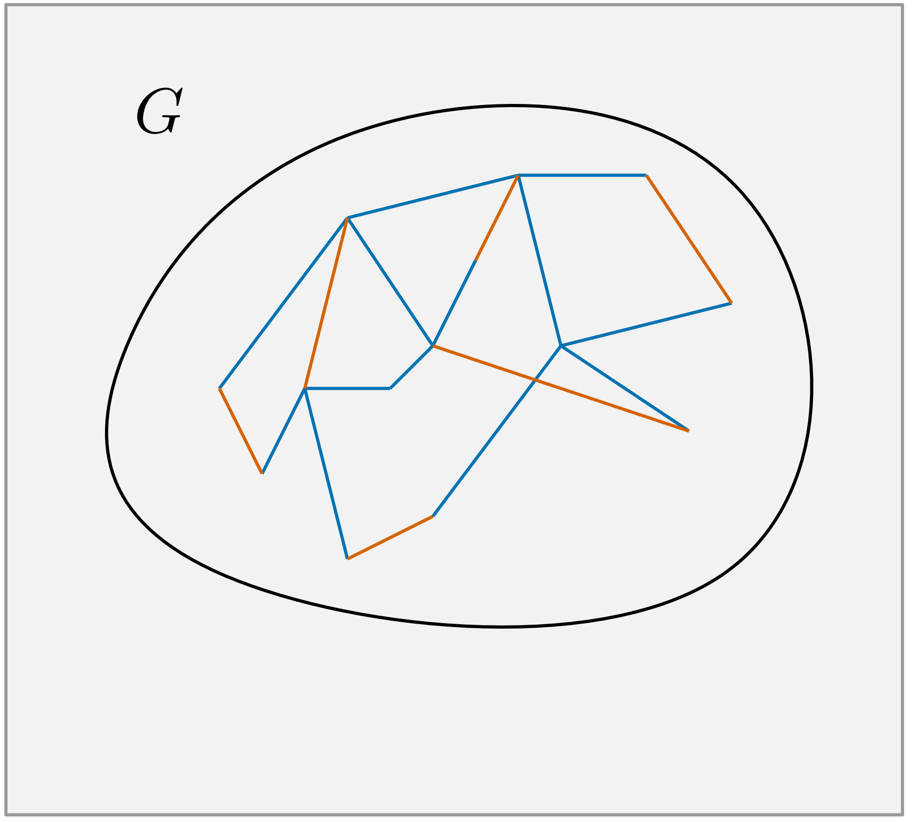
Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)



State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

- $\frac{2}{3}\log_2 n$-approx. (Feige and Singh)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

- $\frac{2}{3} \log_2 n$-approx. (Feige and Singh)
- $\mathcal{O}(\log n / \log \log n)$-approx. (Asadpour et al.)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

- $\frac{2}{3} \log_2 n$-approx. (Feige and Singh)
- $\mathcal{O}(\log n / \log \log n)$-approx. (Asadpour et al.)
- 5500-approx (Svensson et al.)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

- $\frac{2}{3}\log_2 n$-approx. (Feige and Singh)
- $\mathcal{O}(\log n / \log\log n)$-approx. (Asadpour et al.)
- 506-approx (Svensson et al.)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

- $\frac{2}{3}\log_2 n$-approx. (Feige and Singh)
- $\mathcal{O}(\log n / \log\log n)$-approx. (Asadpour et al.)
- $22 + \varepsilon$-approx. (Traub and Vygen)

State of the Art

Exact Solutions

- $\mathcal{O}(2^n n^2)$ by Held and Karp
- practical solvers without worst case guarantees

Polynomial approximations

TSP

- 2-approx. (tree doubling)
- $\frac{3}{2}$-approx. (Christofides / Serdyukov)
- $\frac{3}{2} - \varepsilon$-approx. (Karlin et al.)

ATSP

- $\frac{2}{3}\log_2 n$-approx. (Feige and Singh)
- $\mathcal{O}(\log n / \log\log n)$-approx. (Asadpour et al.)
- $22 + \varepsilon$-approx. (Traub and Vygen)

best lower bounds: $\frac{123}{122}$ vs. $\frac{75}{74}$ (Karpinski et al.)

Our contributions

Goal: FPT runtime

$$f(k) \cdot n^{\mathcal{O}(1)}$$

Our contributions

Goal: FPT runtime
$f(k) \cdot n^{\mathcal{O}(1)}$

Generalized Christofides algorithm

- 2.5-approximation
- k: size of a minimum vertex cover of the graph induced by the asymmetric links

Our contributions

Goal: FPT runtime
$f(k) \cdot n^{\mathcal{O}(1)}$

Generalized Christofides algorithm

- 2.5-approximation
- k: size of a minimum vertex cover of the graph induced by the asymmetric links

Generalized tree doubling algorithm

- 3-approximation
- k: number of one-way arcs in a minimum spanning arborescence of the graph

Our contributions

Goal: FPT runtime
$f(k) \cdot n^{\mathcal{O}(1)}$

Generalized Christofides algorithm

- 2.5-approximation
- k: size of a minimum vertex cover of the graph induced by the asymmetric links

Generalized tree doubling algorithm

- 3-approximation
- k: number of one-way arcs in a minimum spanning arborescence of the graph

Additionally:

- adapted algorithms for β-asymmetry

Our contributions

Goal: FPT runtime
$f(k) \cdot n^{\mathcal{O}(1)}$

Generalized Christofides algorithm

- 2.5-approximation
- k: size of a minimum vertex cover of the graph induced by the asymmetric links

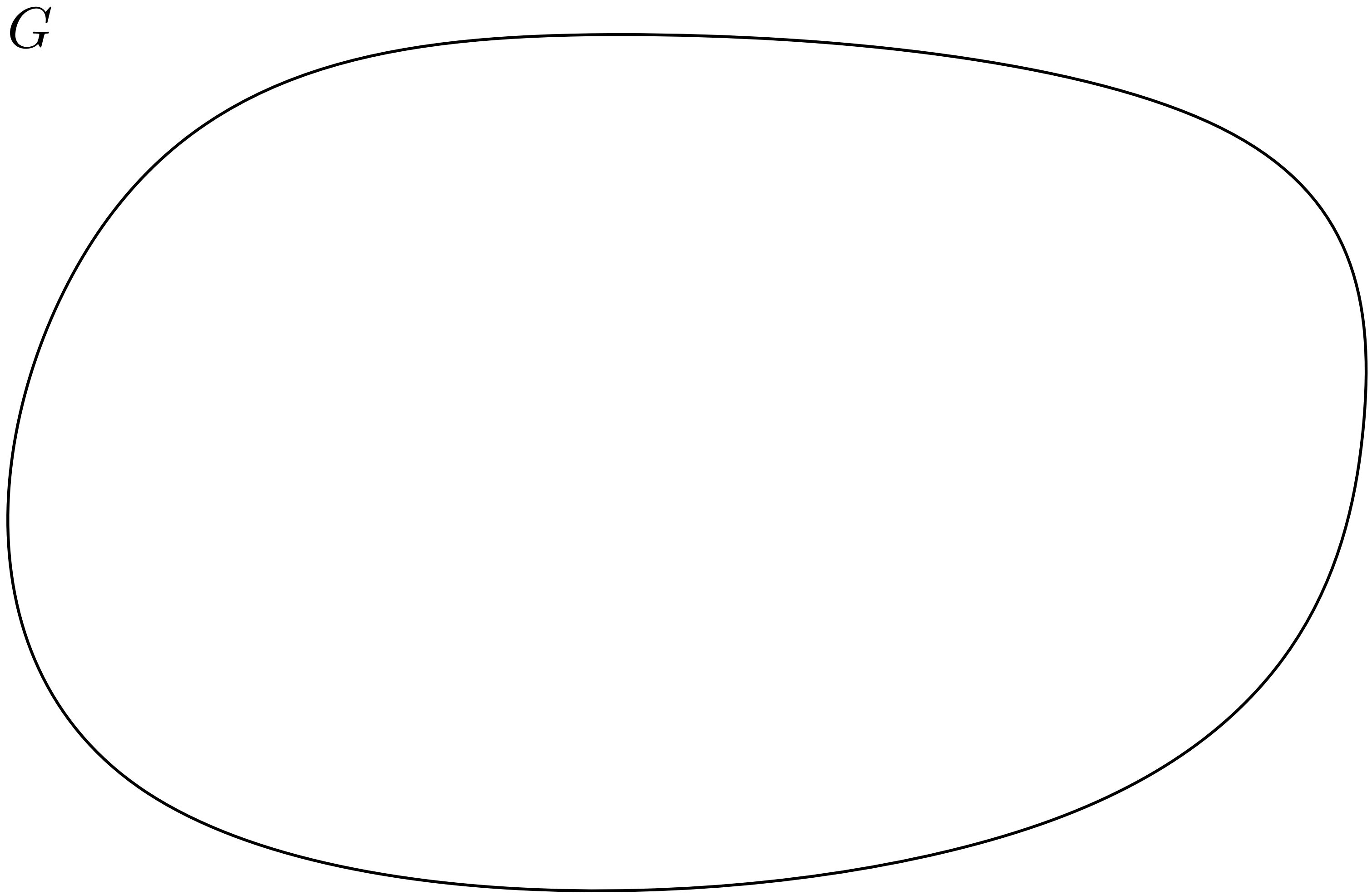
Generalized tree doubling algorithm

- 3-approximation
- k: number of one-way arcs in a minimum spanning arborescence of the graph

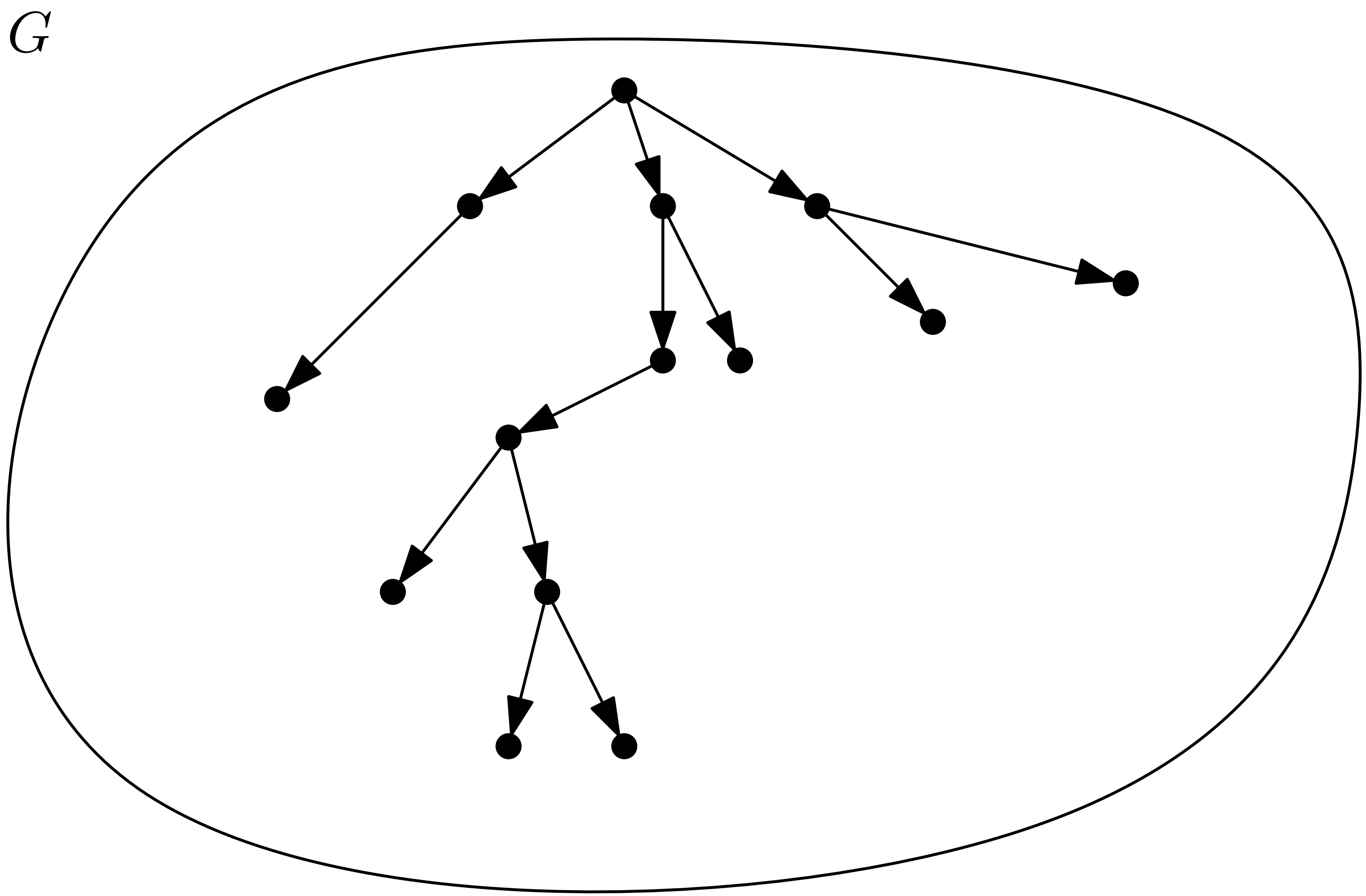
Additionally:

- adapted algorithms for β-asymmetry
- experimental evaluation

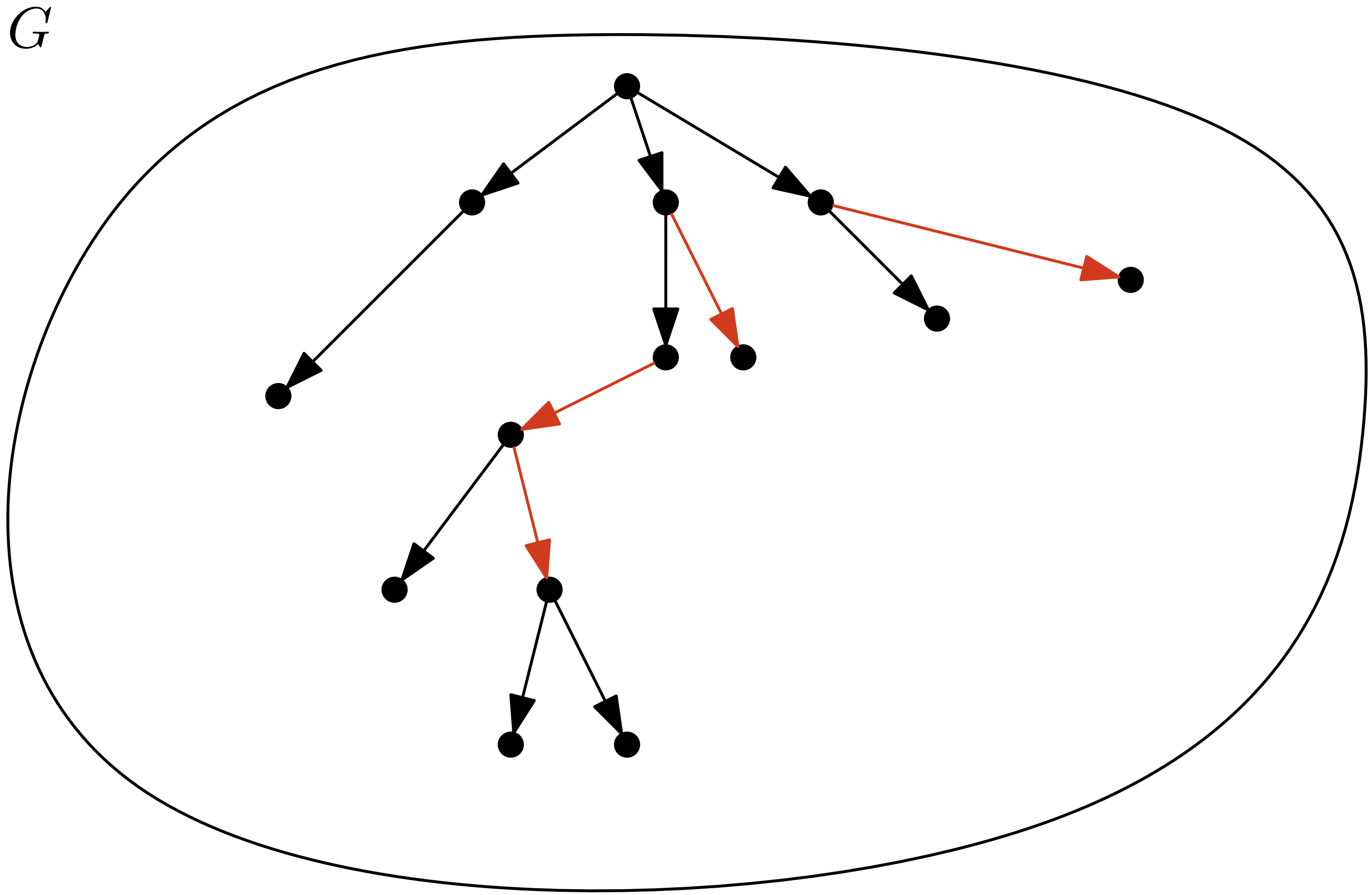
A tree based approach?



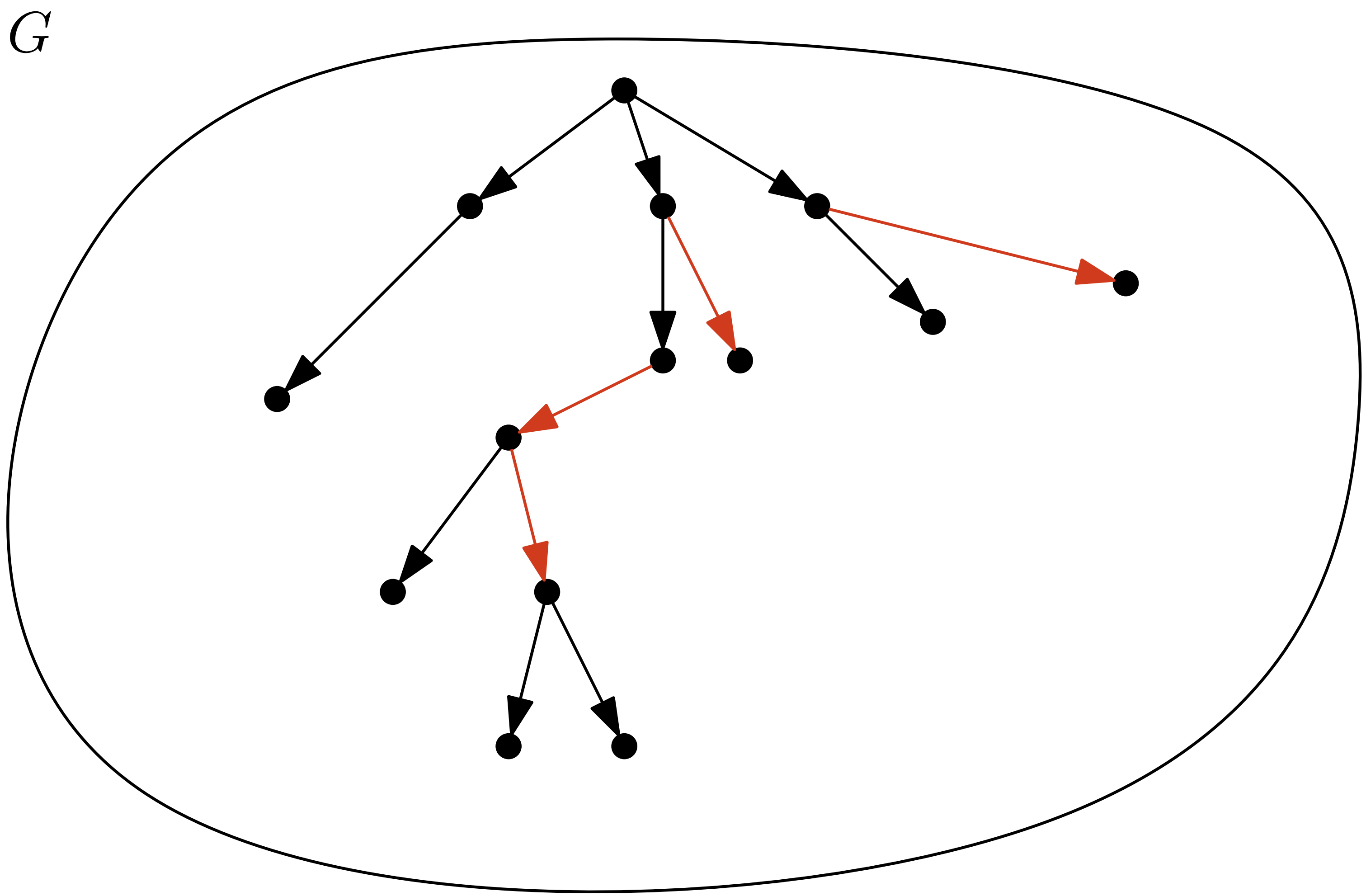
A tree based approach?



A tree based approach?

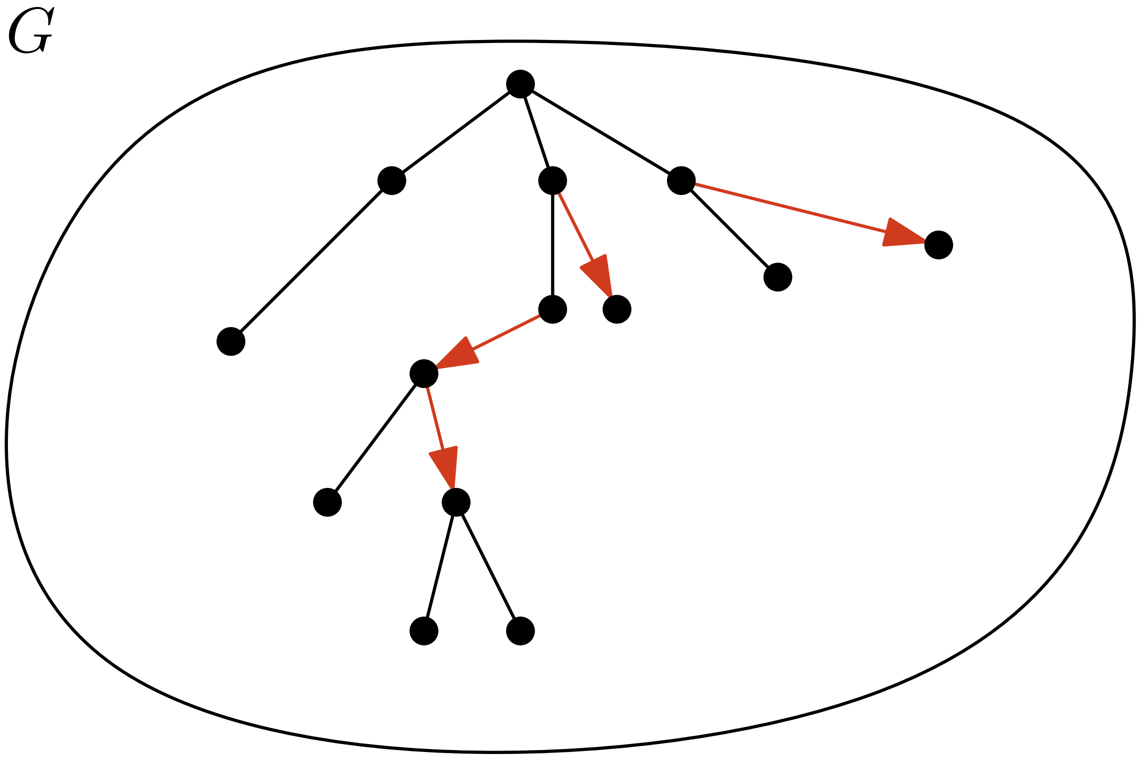


A tree based approach?

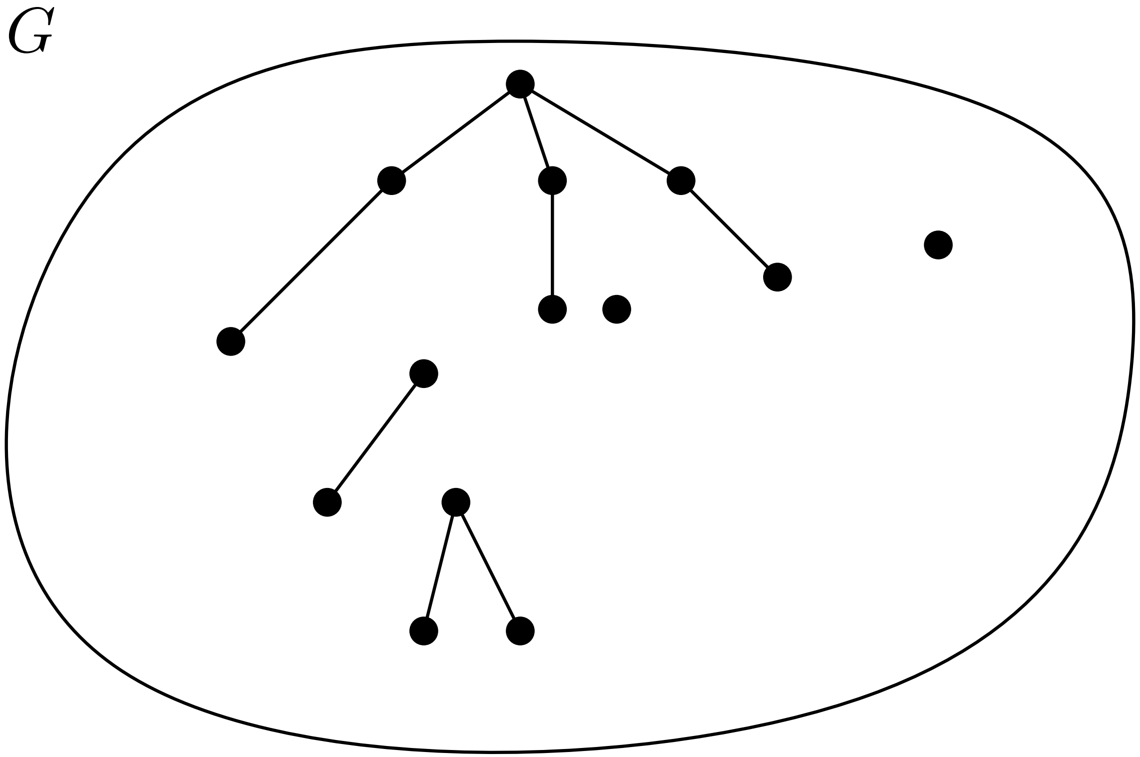


Def. One-way arc: arc (u, v) in tree with $c(u, v) < c(v, u)$

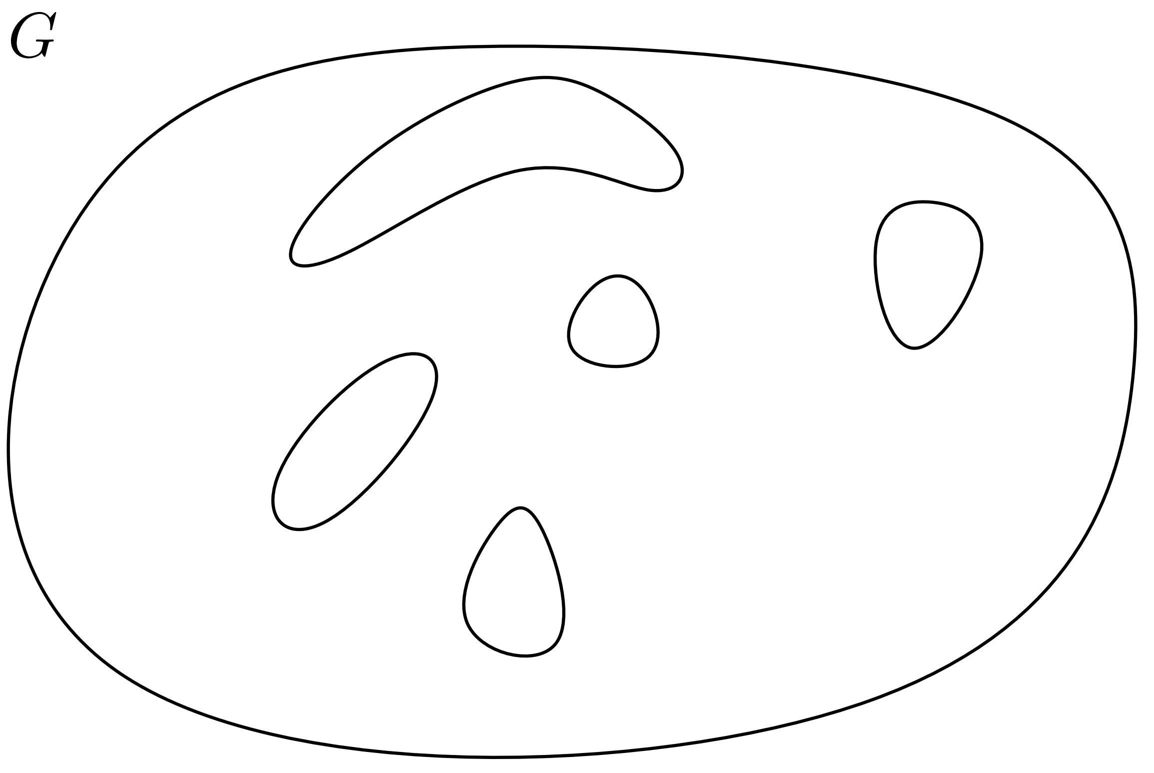
Generalized Tree Doubling Algorithm



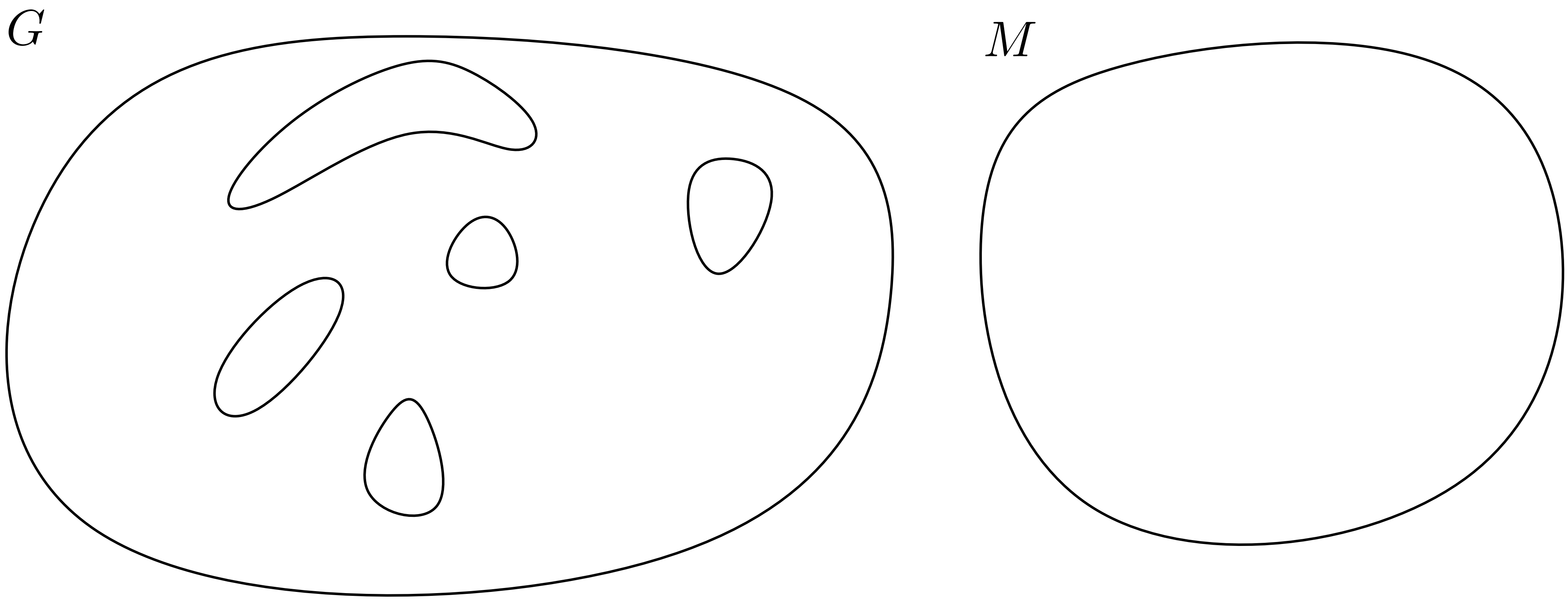
Generalized Tree Doubling Algorithm



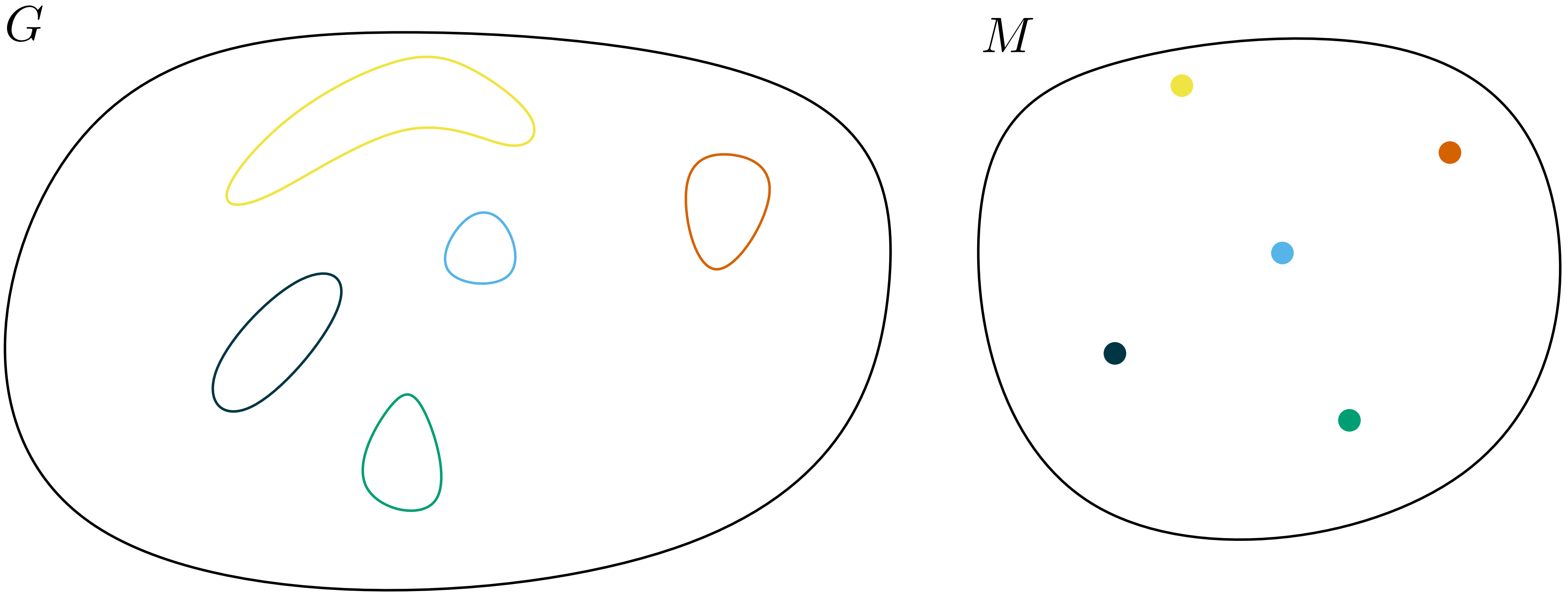
Generalized Tree Doubling Algorithm



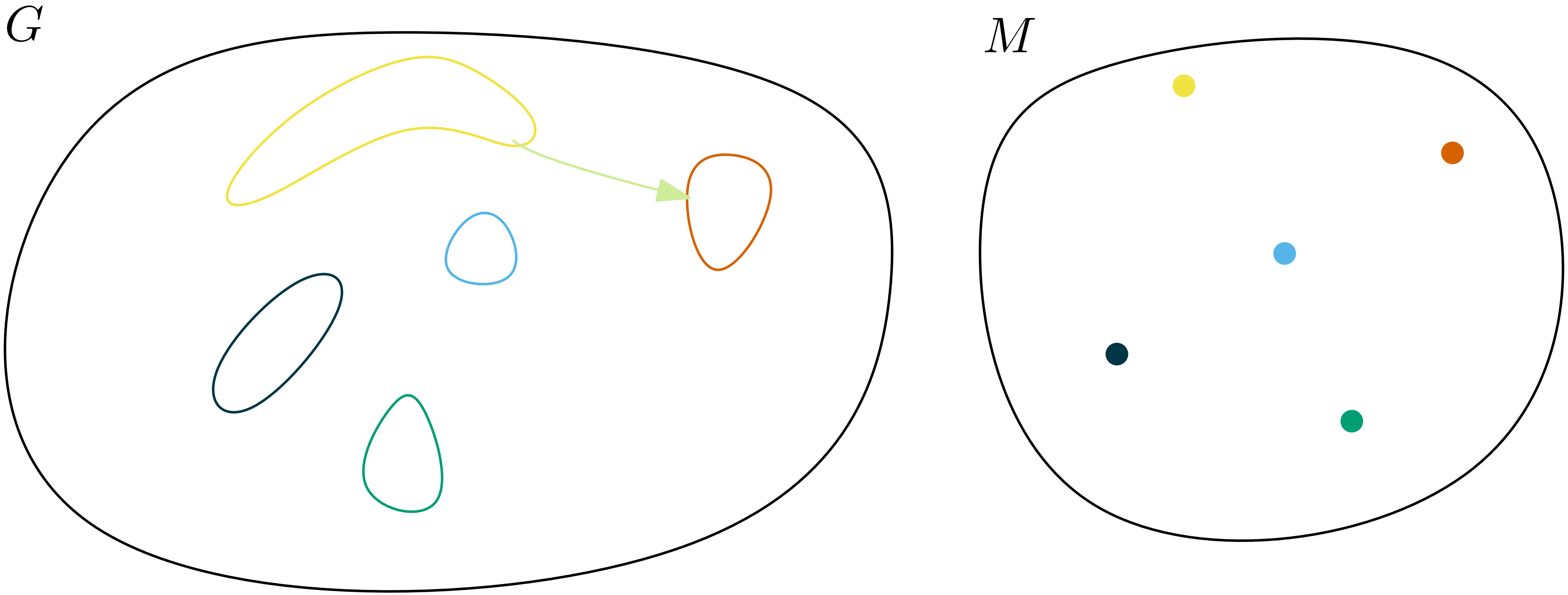
Generalized Tree Doubling Algorithm



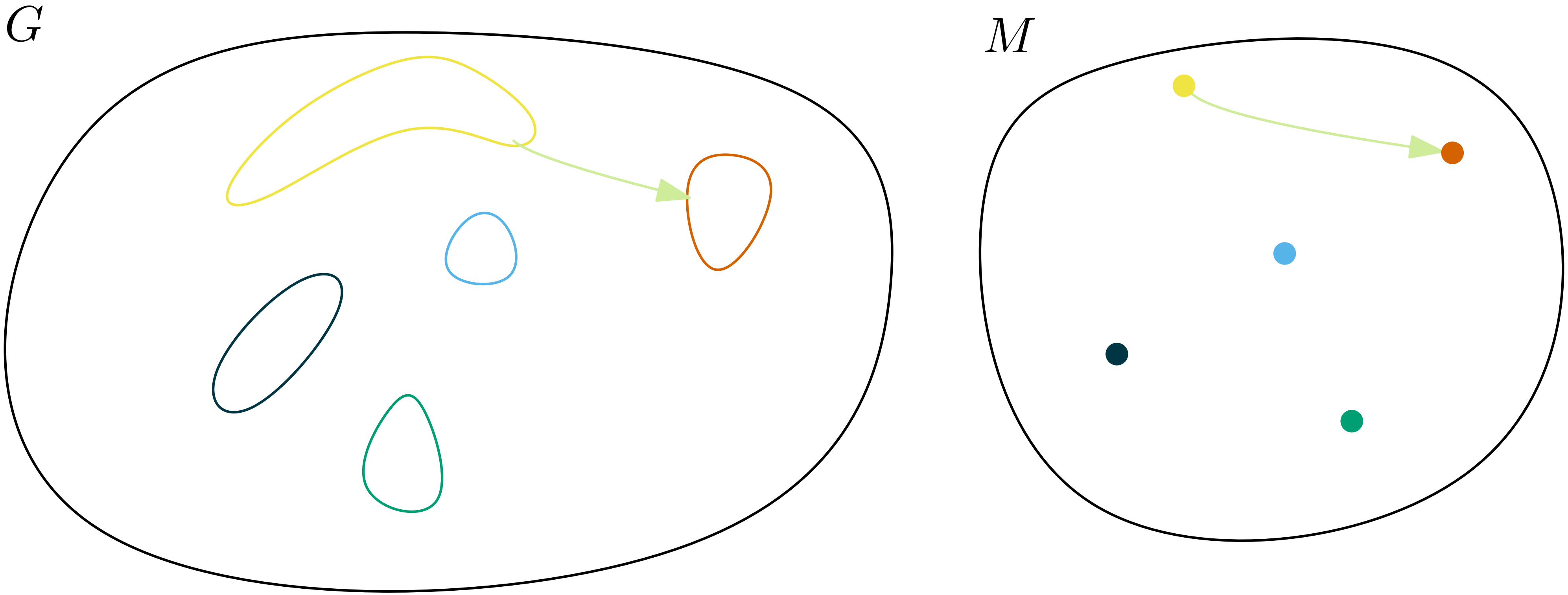
Generalized Tree Doubling Algorithm



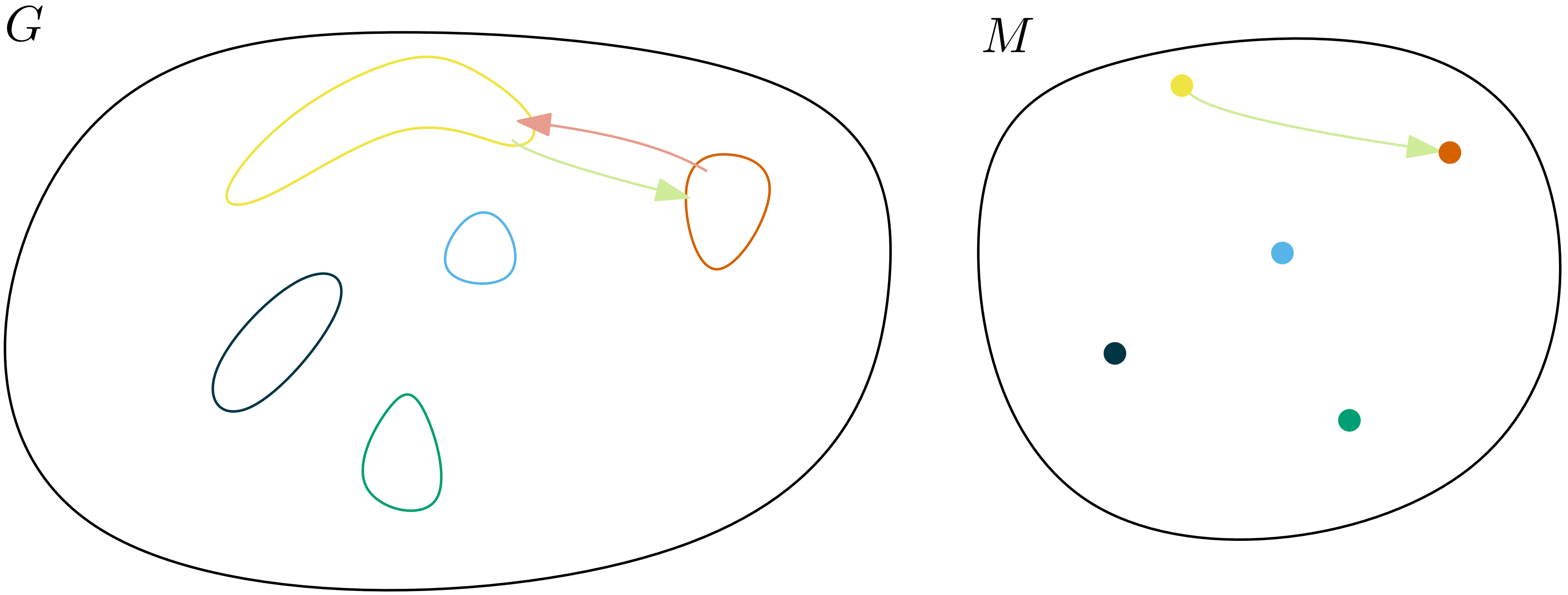
Generalized Tree Doubling Algorithm



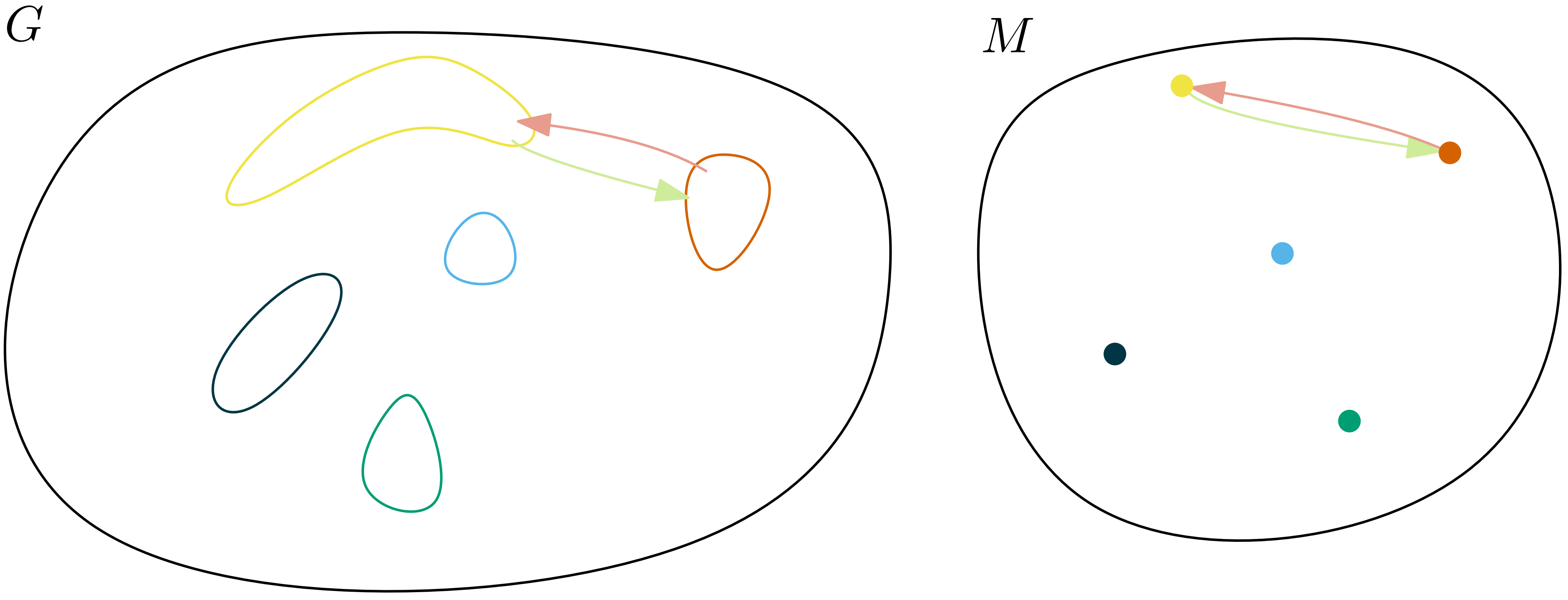
Generalized Tree Doubling Algorithm



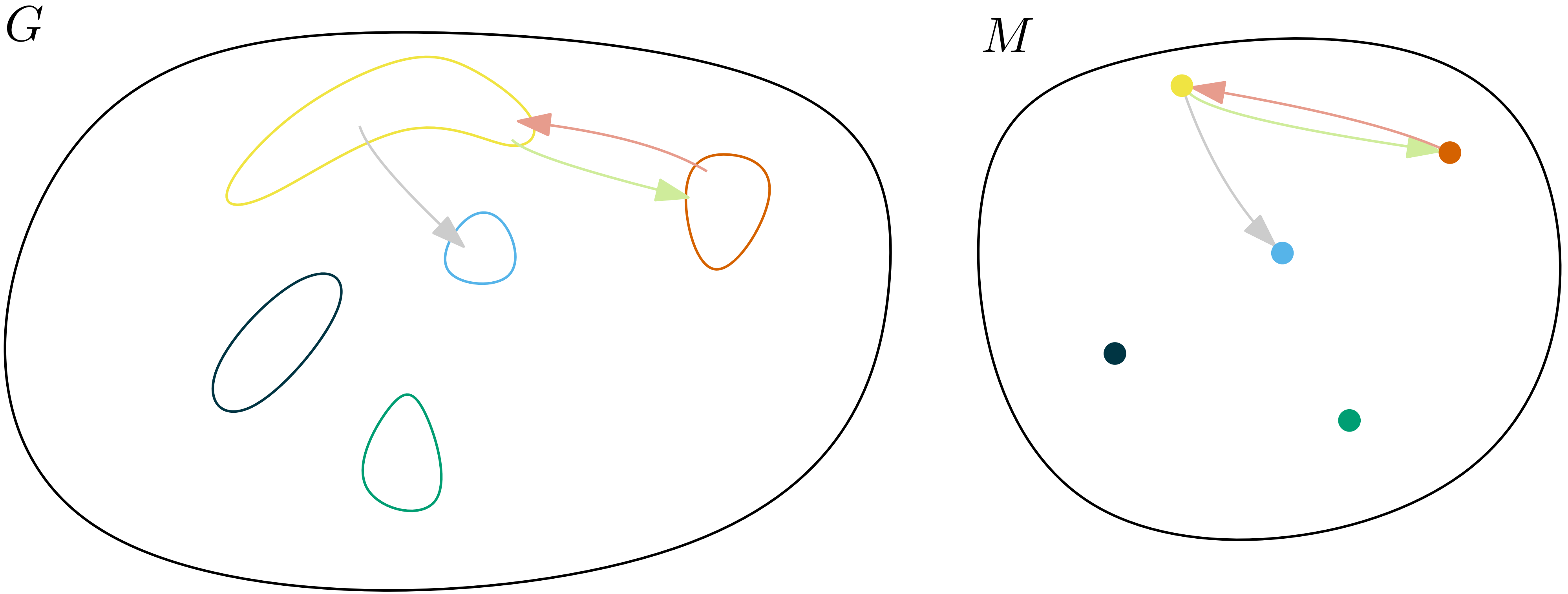
Generalized Tree Doubling Algorithm



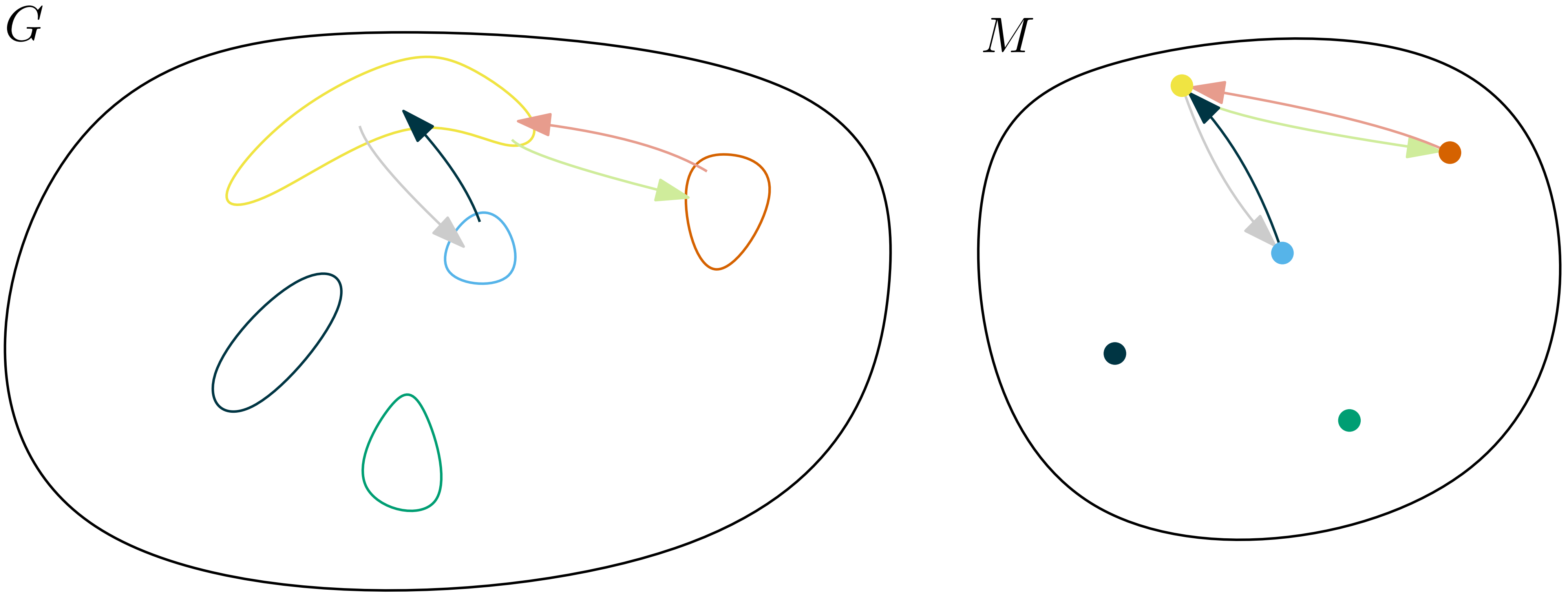
Generalized Tree Doubling Algorithm



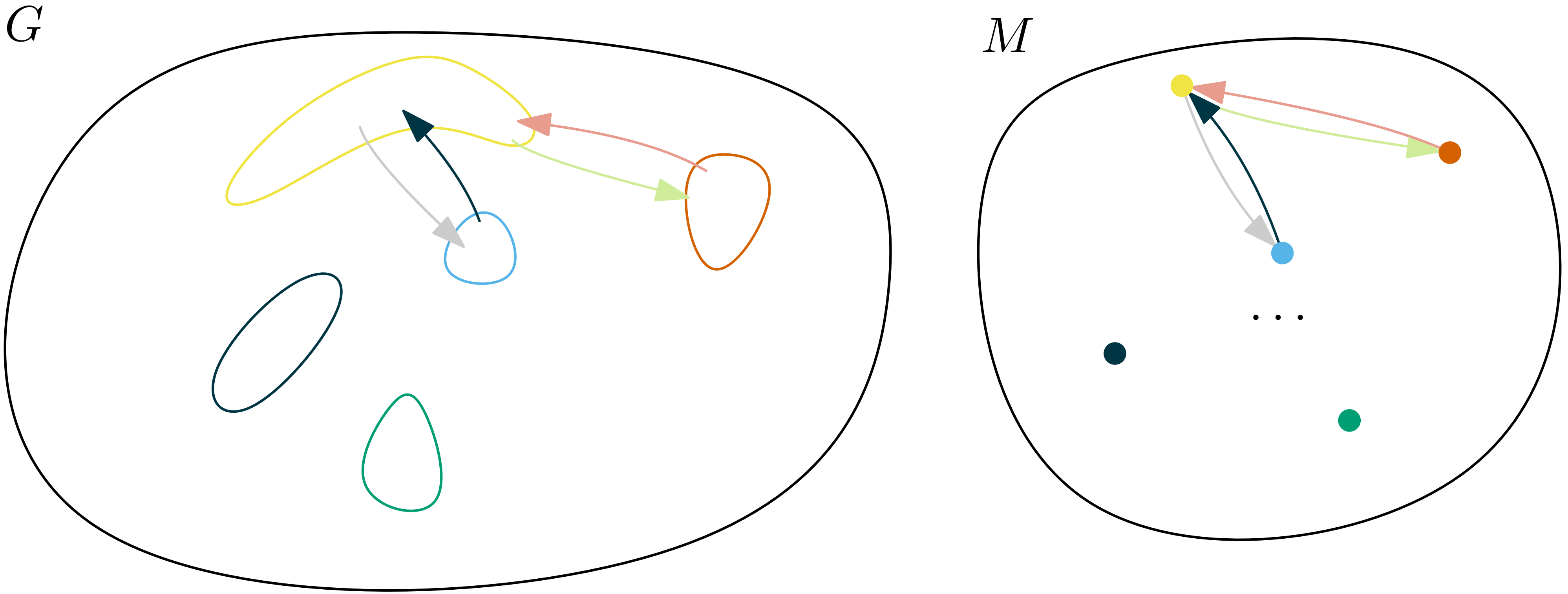
Generalized Tree Doubling Algorithm



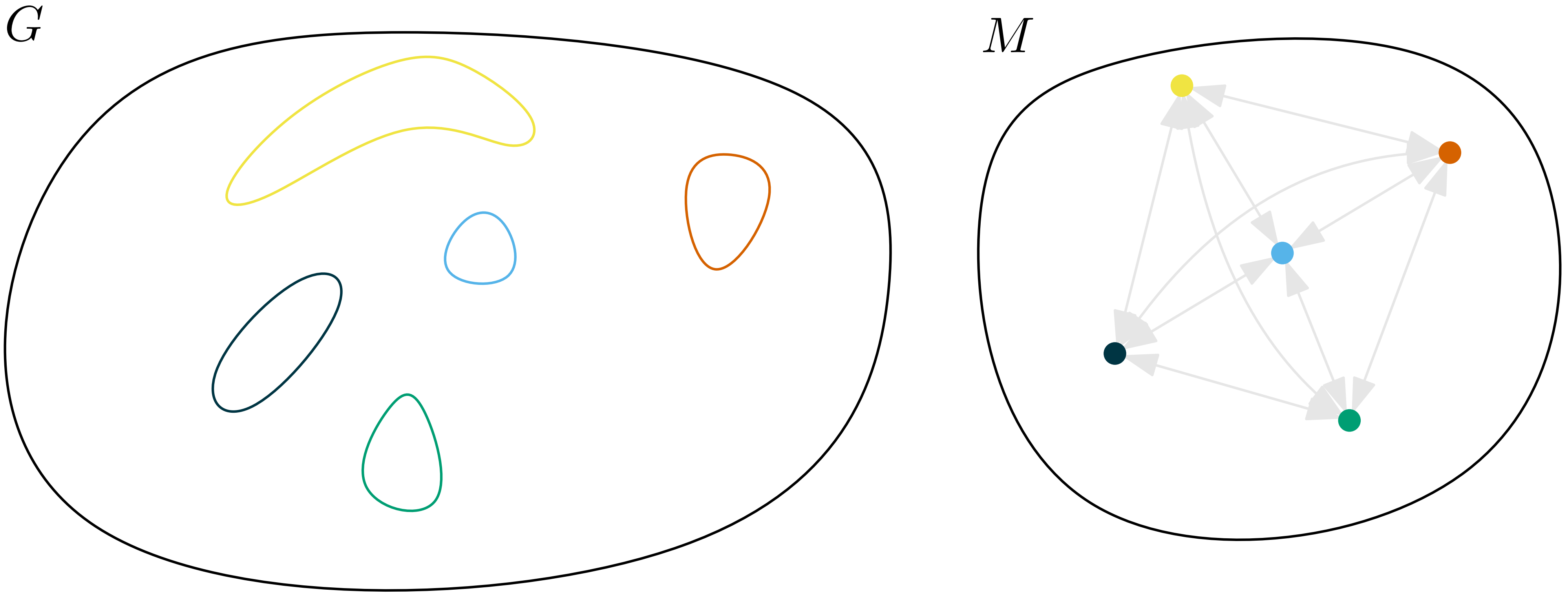
Generalized Tree Doubling Algorithm



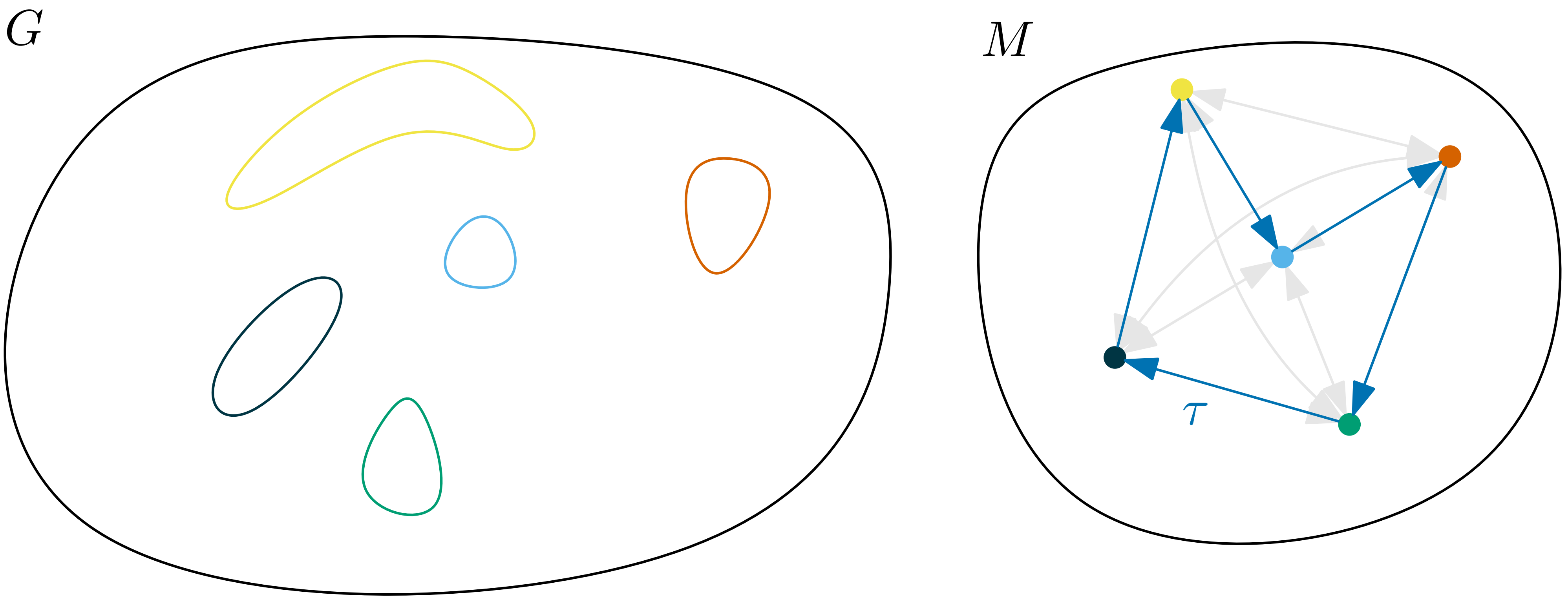
Generalized Tree Doubling Algorithm



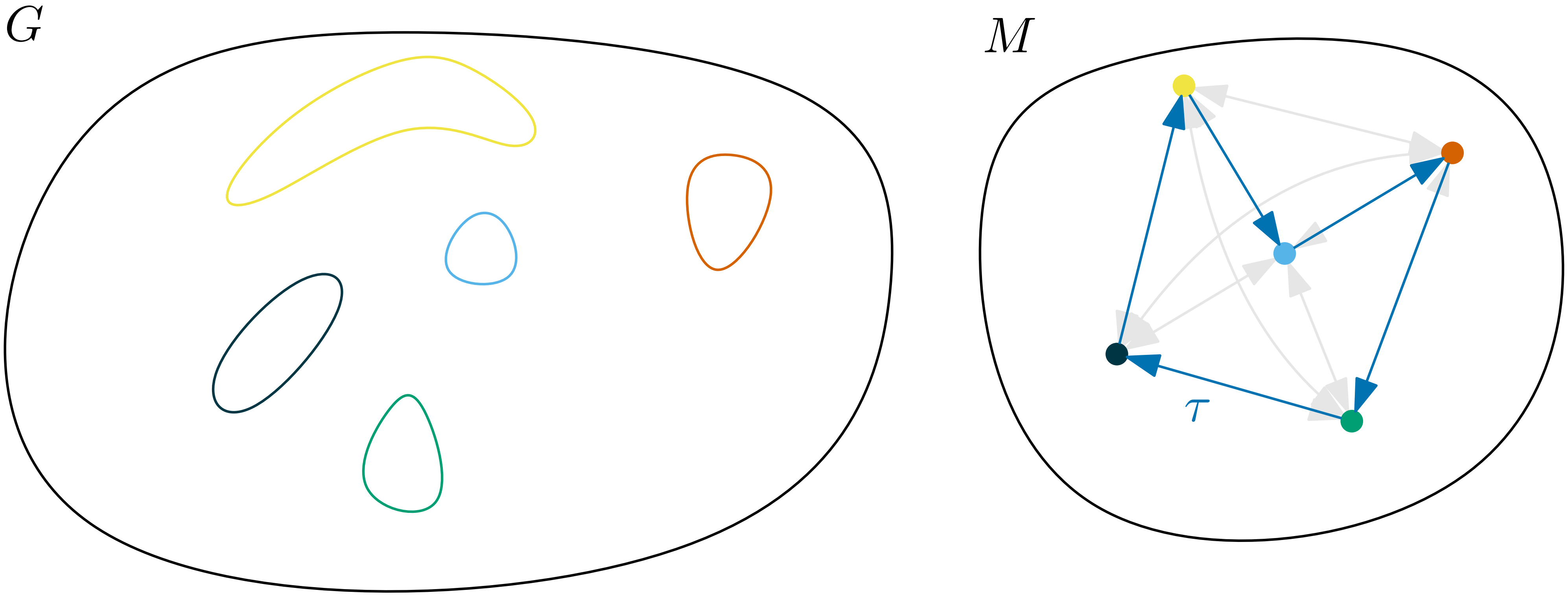
Generalized Tree Doubling Algorithm



Generalized Tree Doubling Algorithm

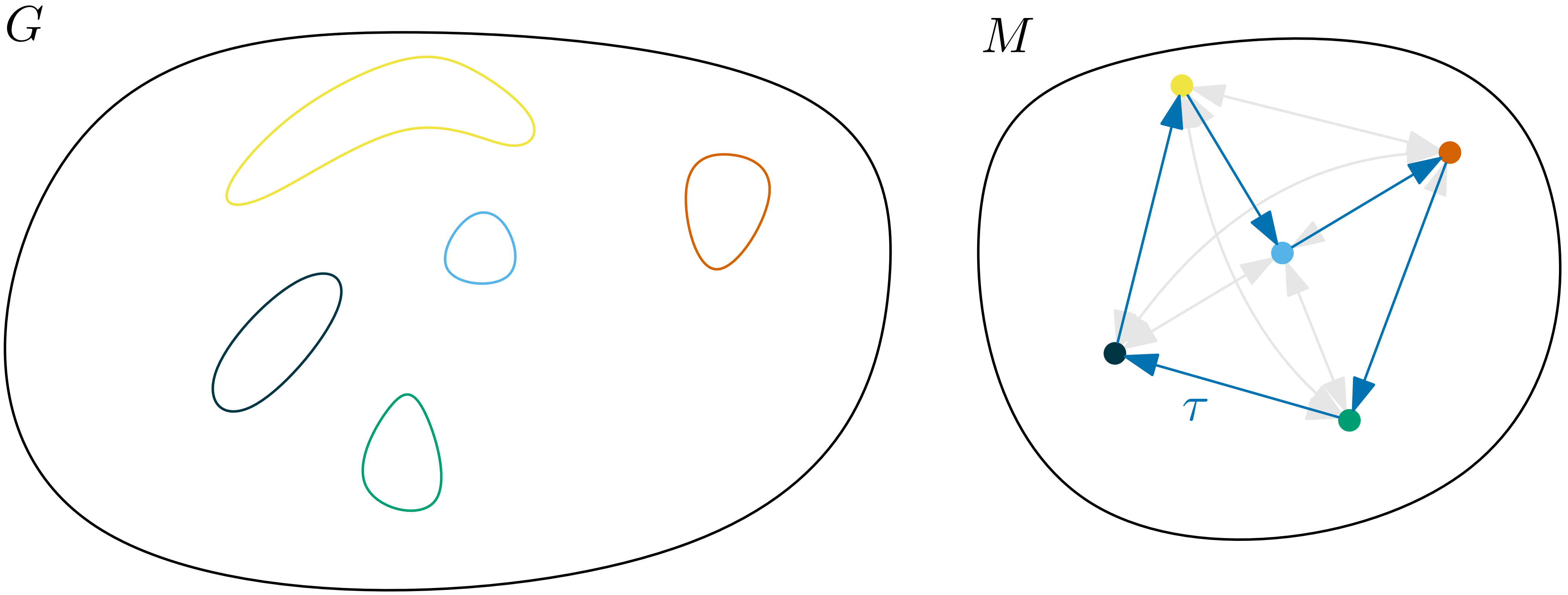


Generalized Tree Doubling Algorithm



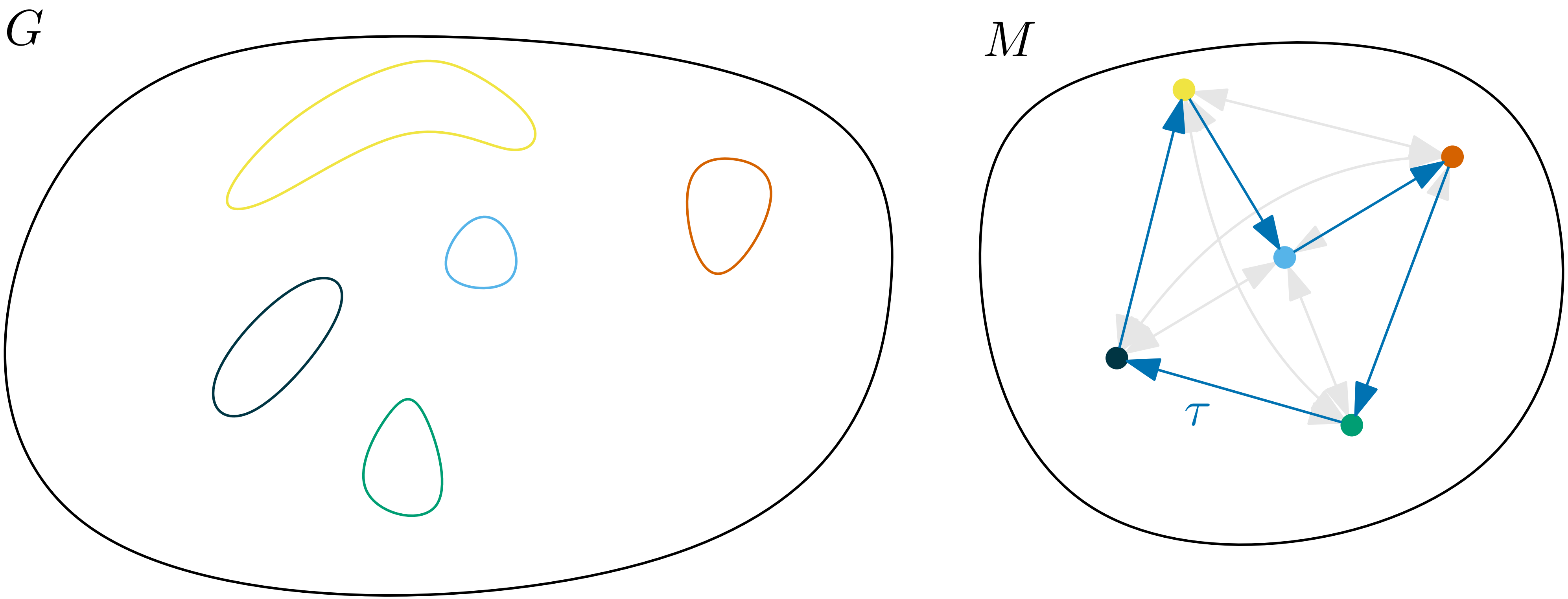
Question: Can we upper bound the cost of τ?

Generalized Tree Doubling Algorithm



Question: Can we upper bound the cost of τ, given that M is minor of G?

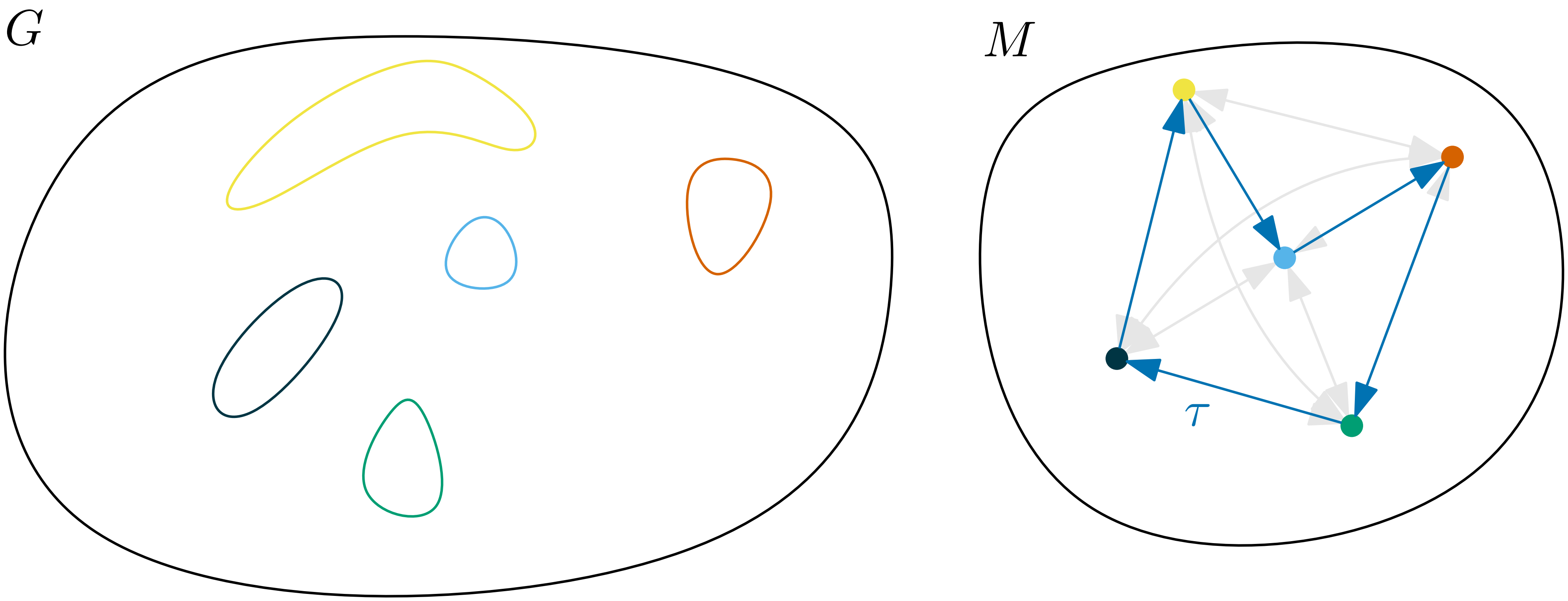
Generalized Tree Doubling Algorithm



Question: Can we upper bound the cost of τ, given that M is minor of G?

Lemma: Let H be a minor of metric graph G, then $TSP(H) \leq TSP(G)$.

Generalized Tree Doubling Algorithm

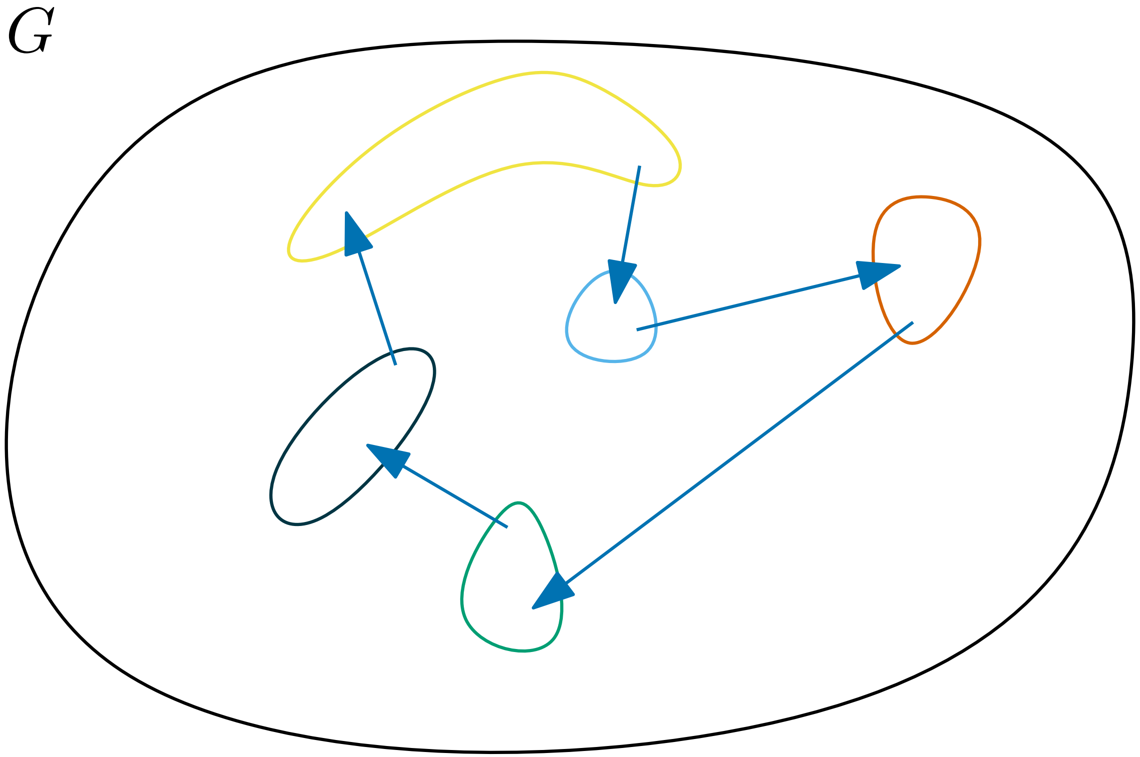


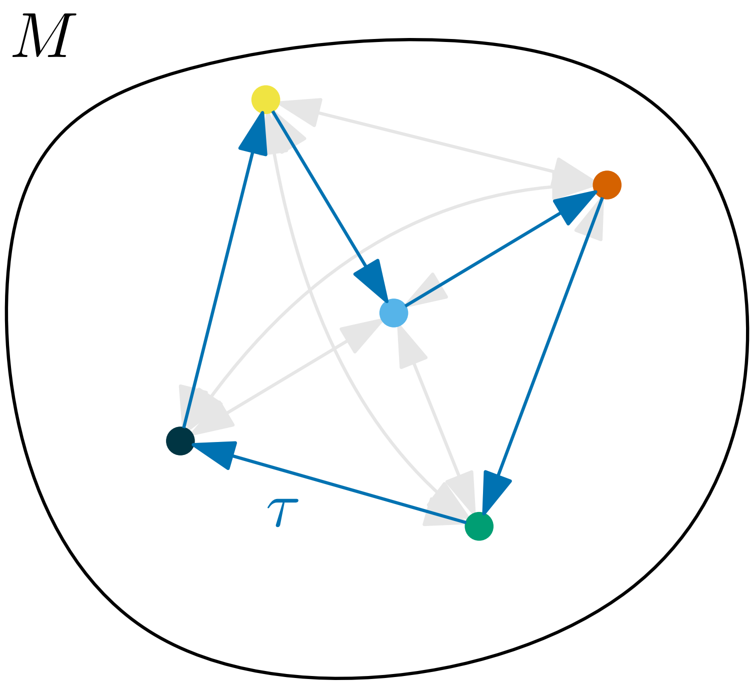
Question: Can we upper bound the cost of τ, given that M is minor of G?

Lemma: Let H be a minor of metric graph G, then $TSP(H) \leq TSP(G)$.

$$cost(\tau) = TSP(M) \leq TSP(G)$$

Generalized Tree Doubling Algorithm





Question: Can we upper bound the cost of τ, given that M is minor of G?

Lemma: Let H be a minor of metric graph G, then $TSP(H) \leq TSP(G)$.

$$cost(\tau) = TSP(M) \leq TSP(G)$$

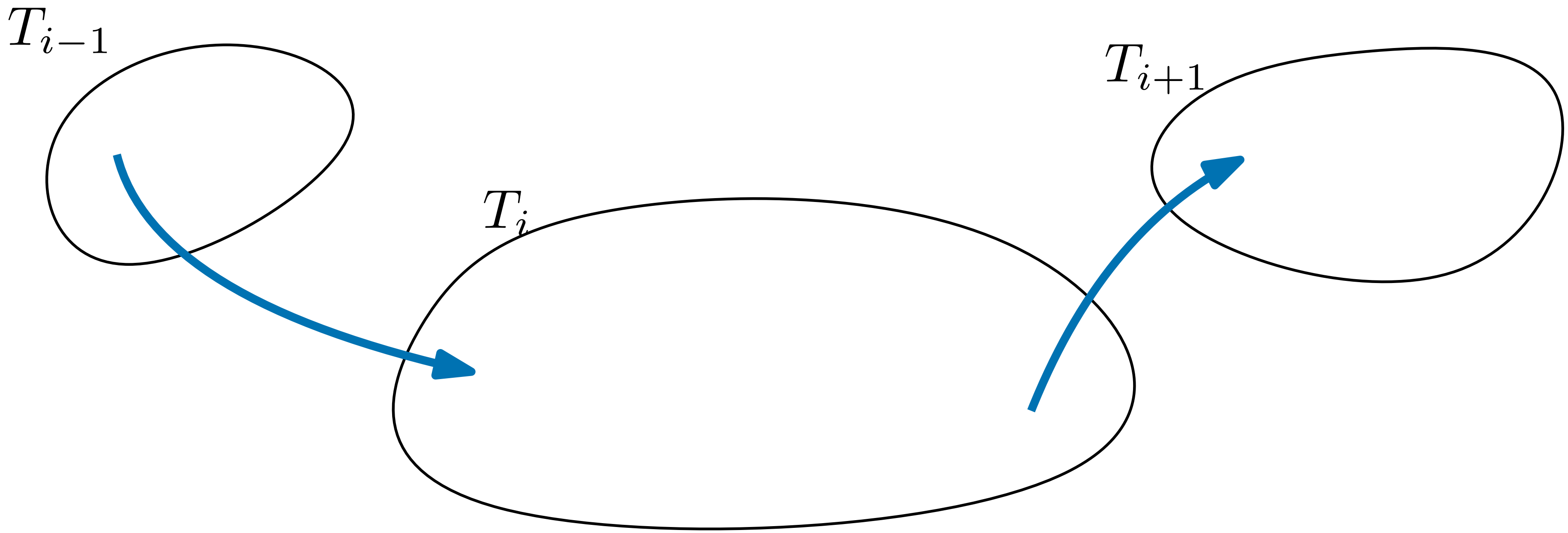
Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

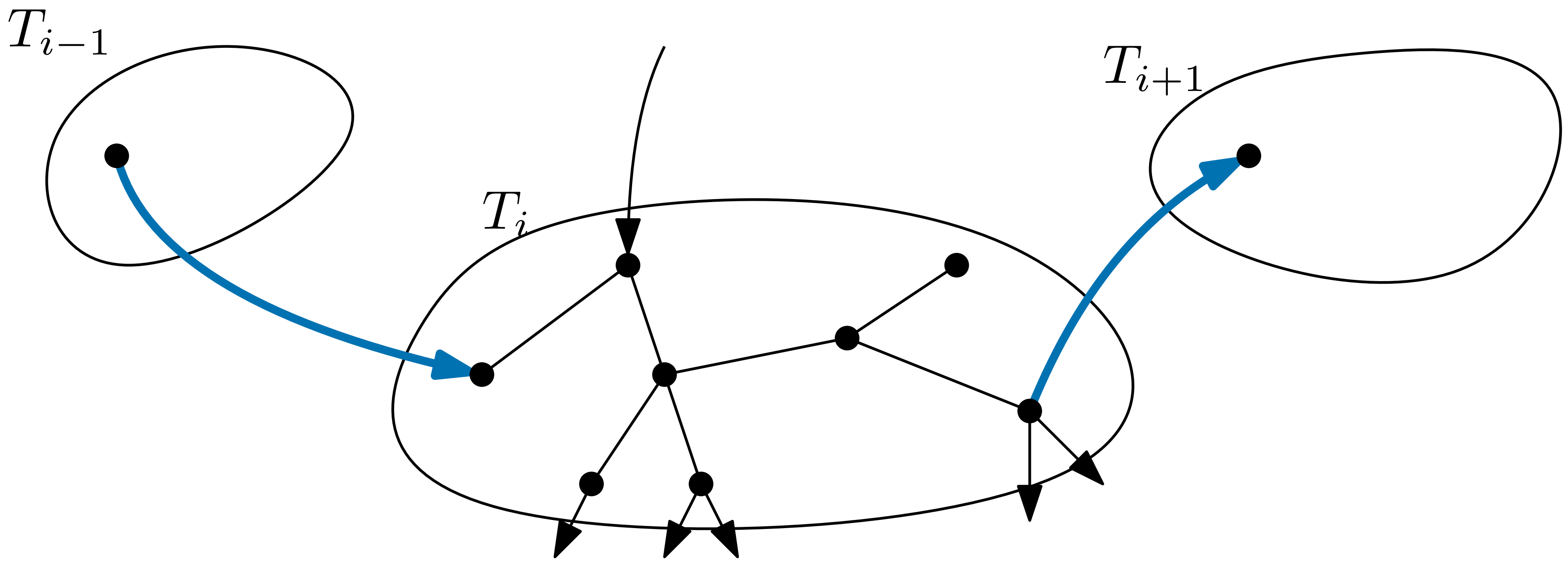
Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:



Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

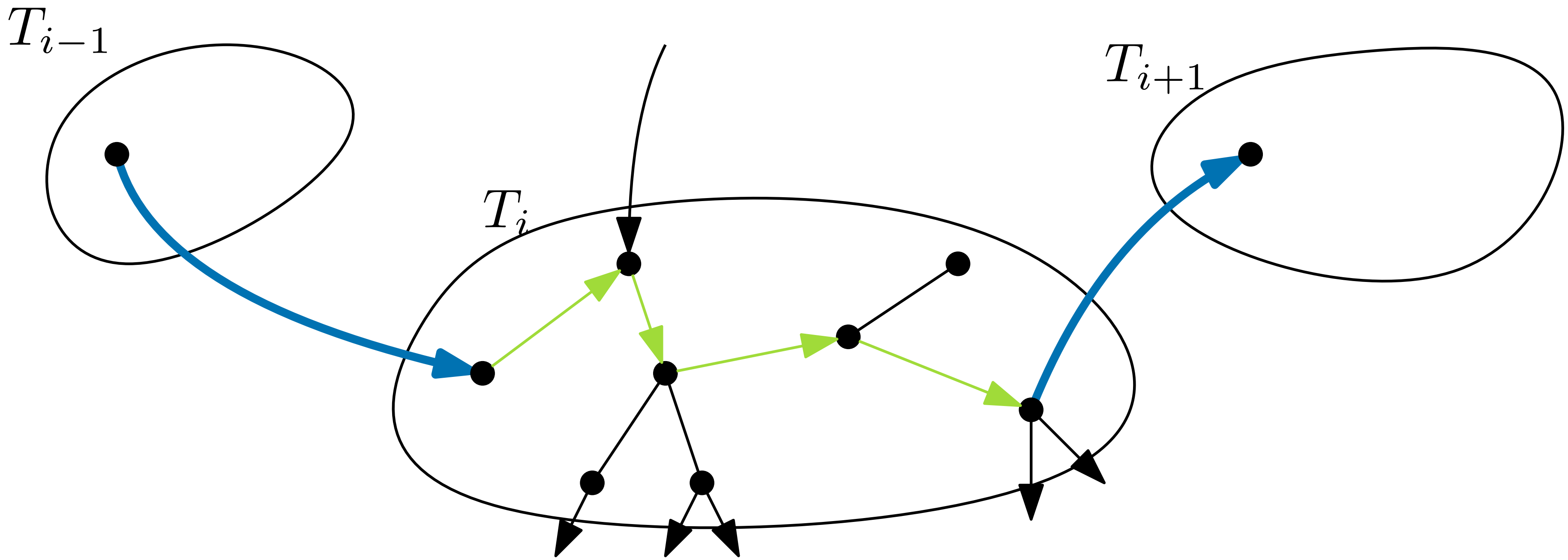
Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:



Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

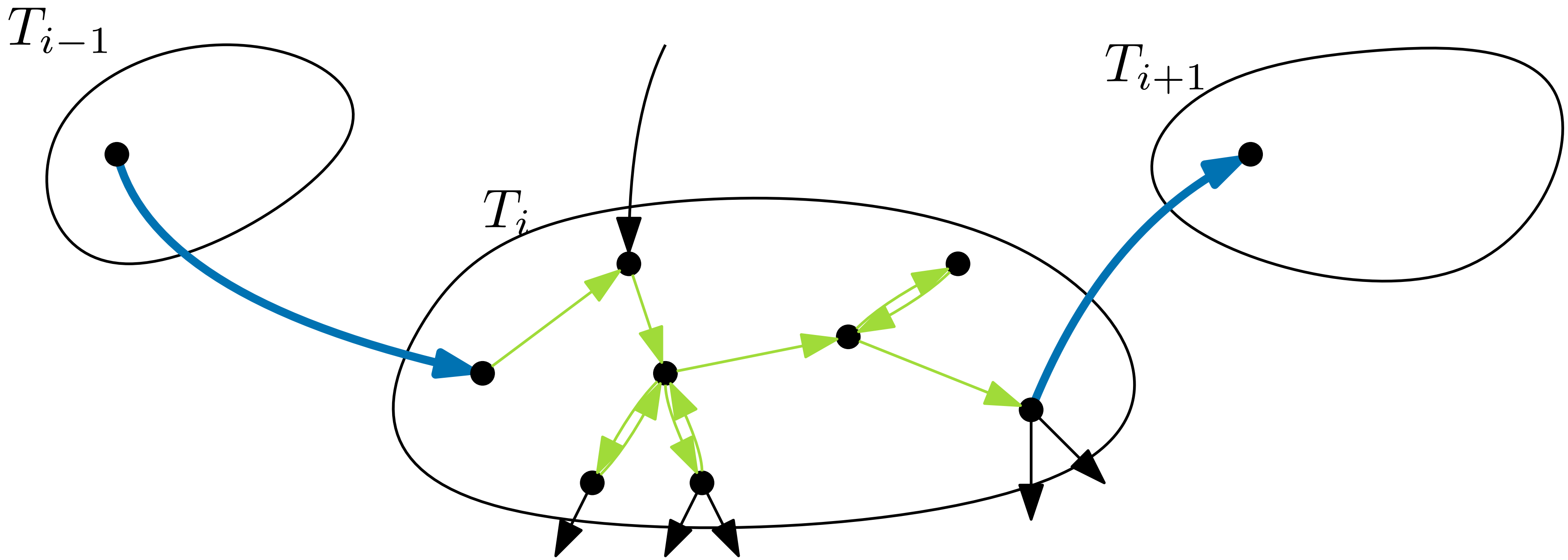
Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:



Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

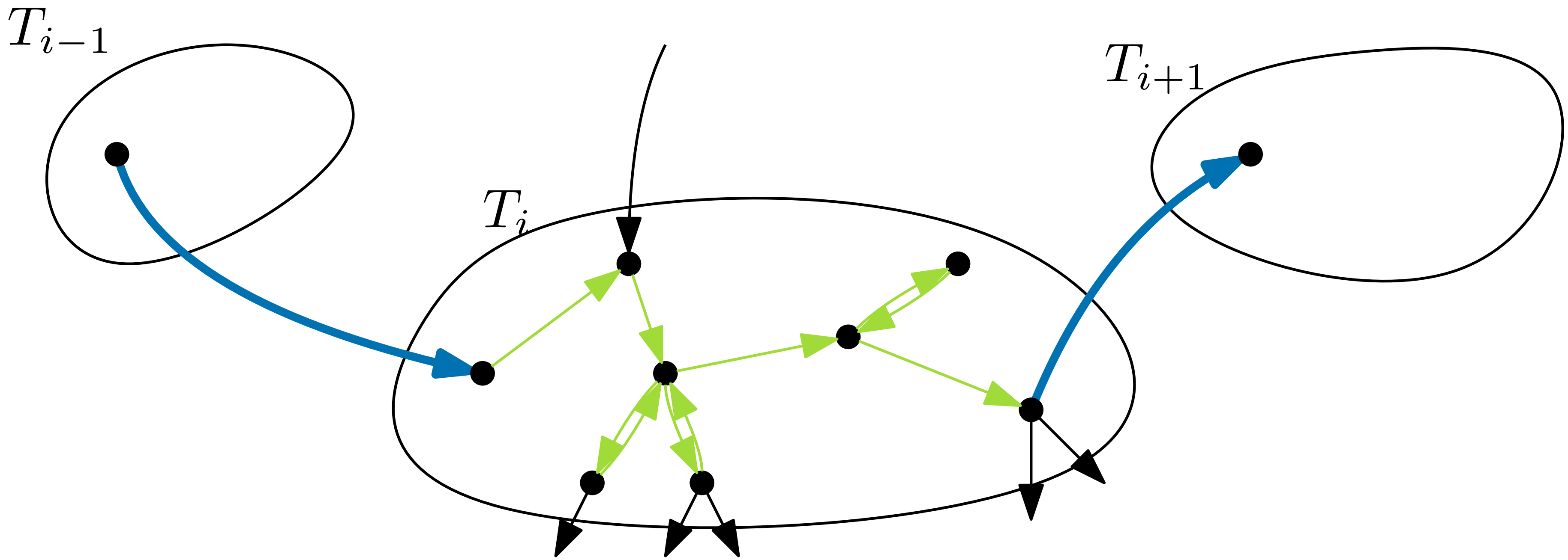
Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:



Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:

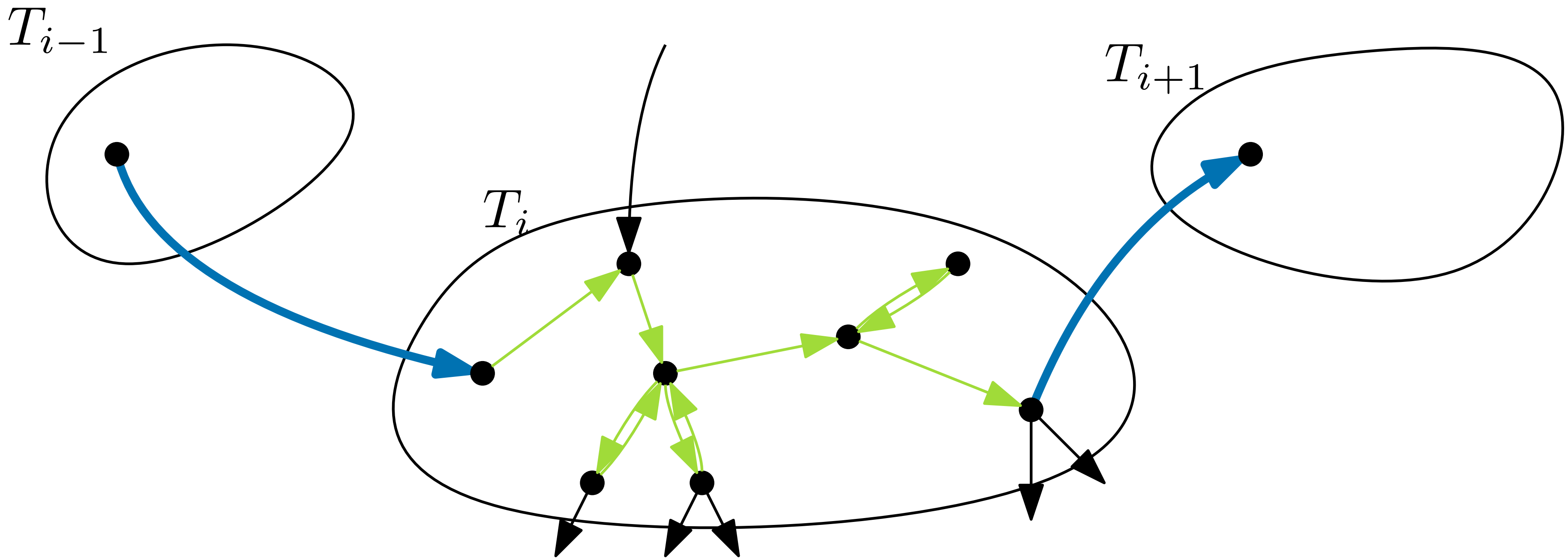


Cost of path p_i through T_i?

Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:

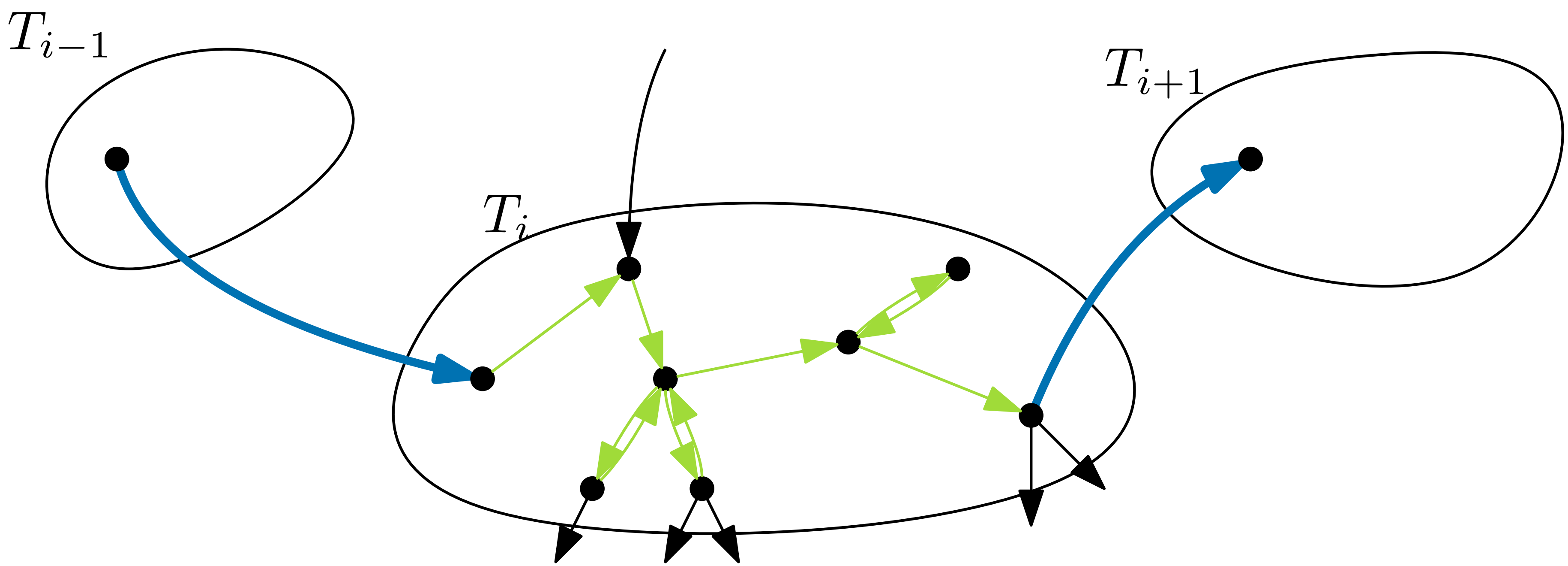


Cost of path p_i through T_i? $c(p_i) \leq 2c(T_i)$

Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:



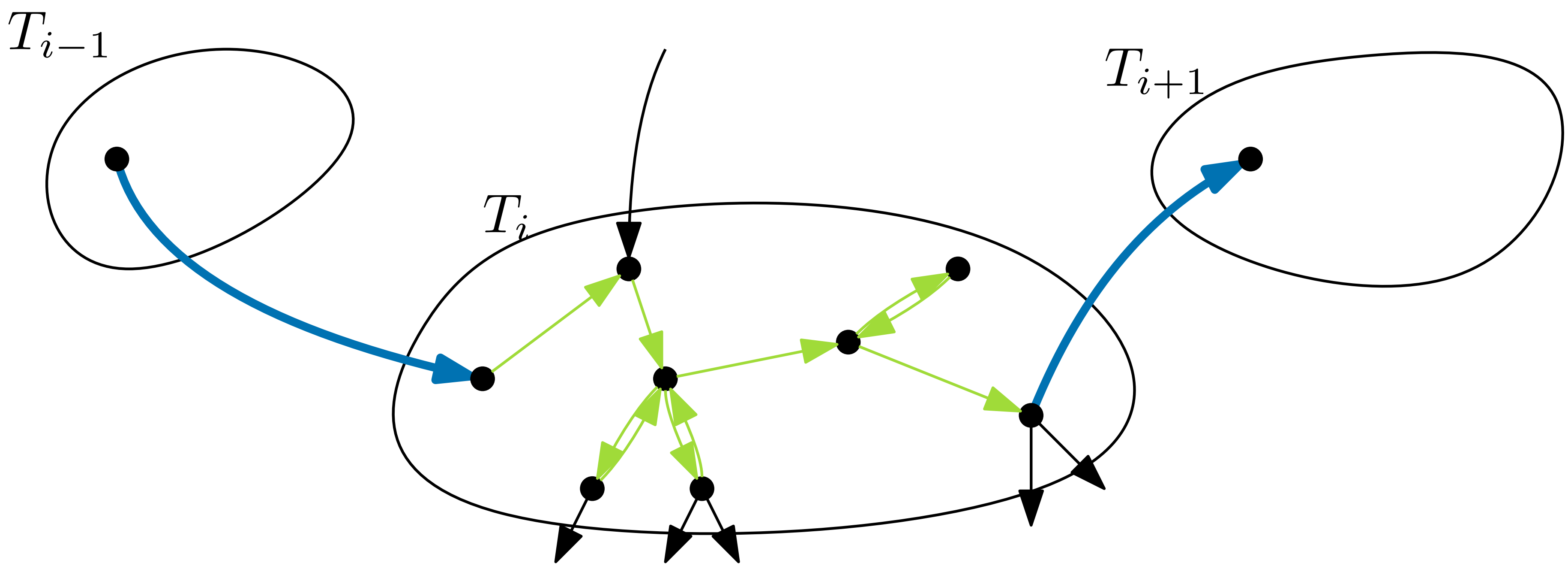
Cost of path p_i through T_i? $c(p_i) \leq 2c(T_i)$

thus $\sum c(p_i) \leq \sum 2c(T_i) \leq 2\,TSP(G)$

Generalized Tree Doubling Algorithm

Now: Connect τ into tour of G

Consider components $T_{i-1}, T_i, T_{i+1} \in \tau$:

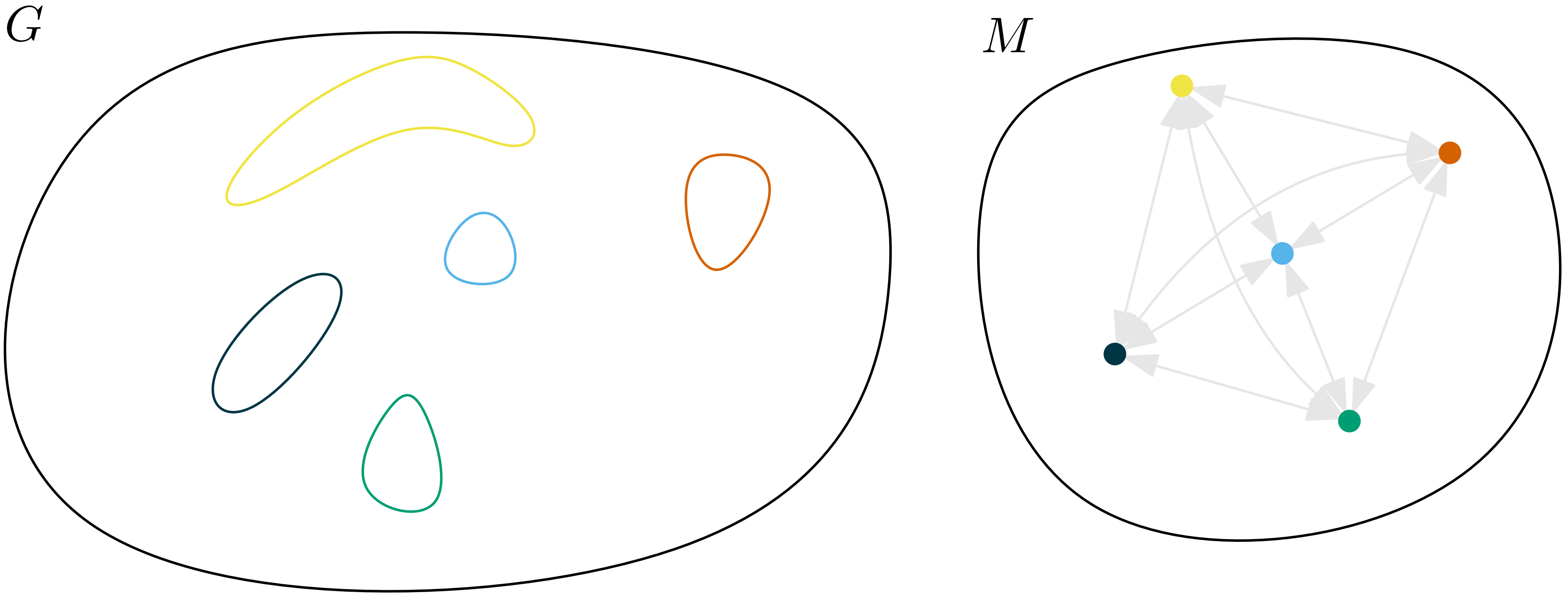


Cost of path p_i through T_i? $c(p_i) \leq 2c(T_i)$

thus $\sum c(p_i) \leq \sum 2c(T_i) \leq 2\,TSP(G)$

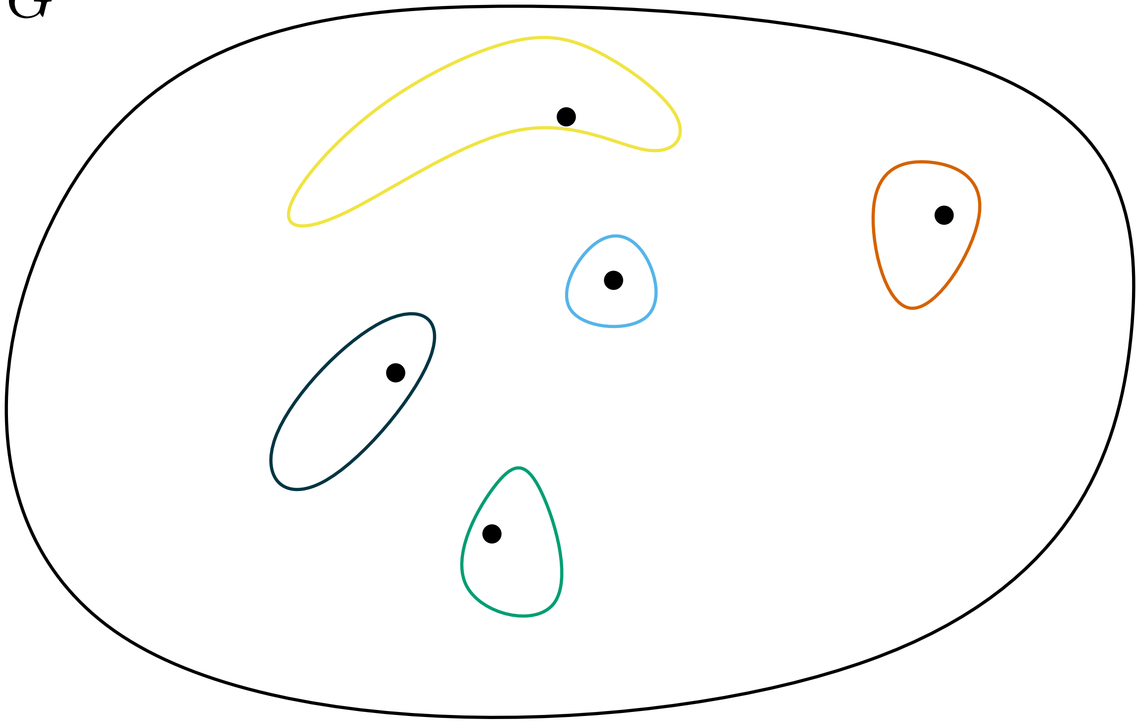
Theorem: The algorithm computes a 3-approx. for ATSP in $\mathcal{O}^*(2^k)$ where k is the number of one-way edges in a given min. spanning arborescence.

Algorithm revisited

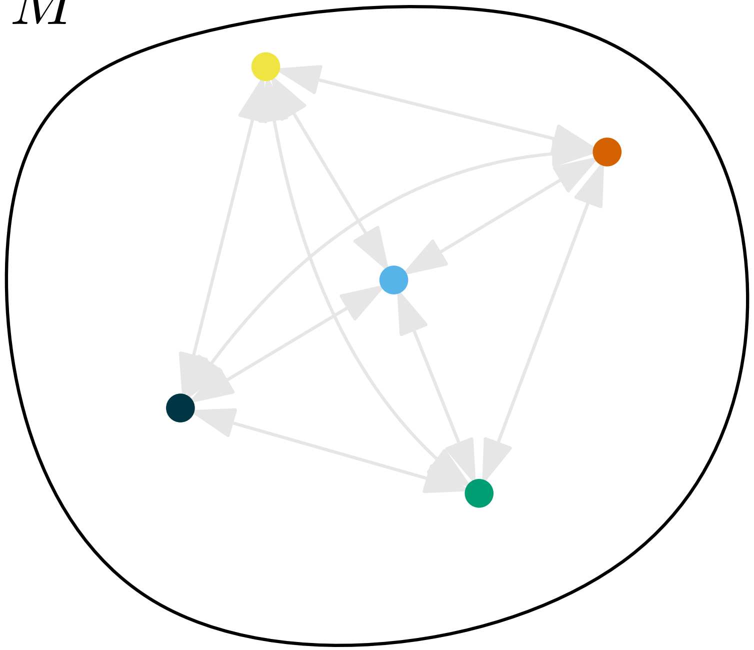


Algorithm revisited

G

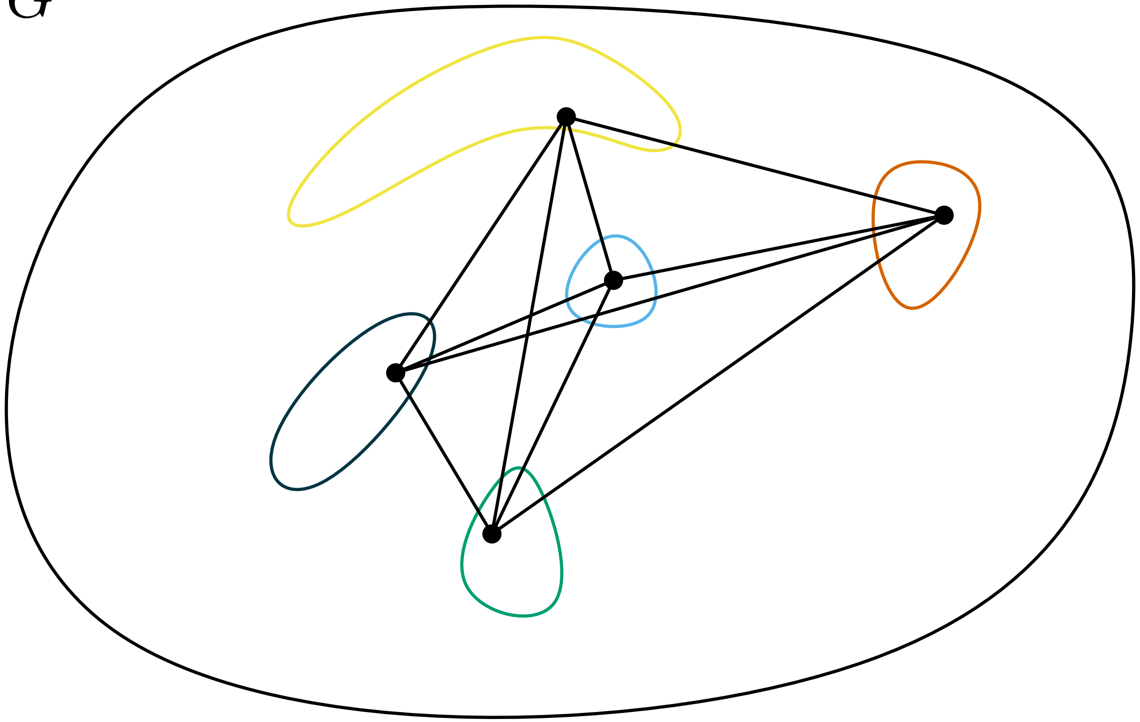


M

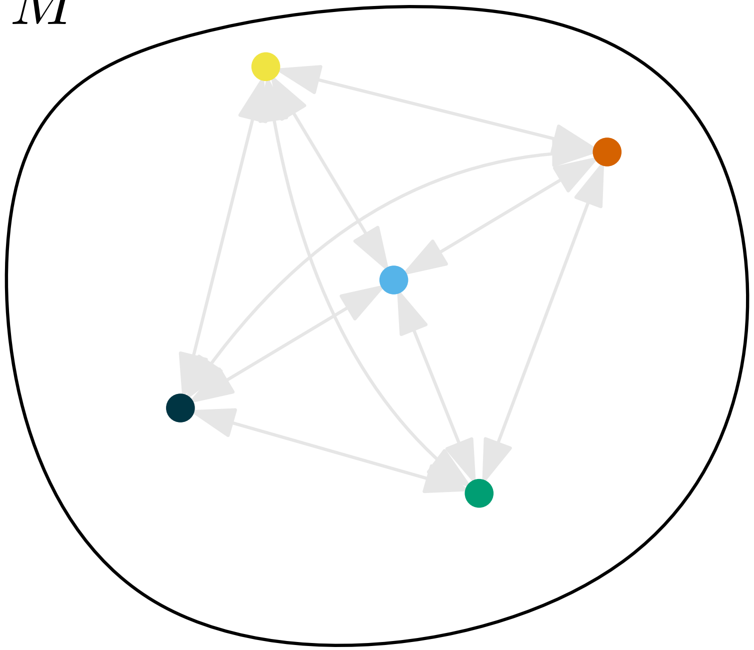


Algorithm revisited

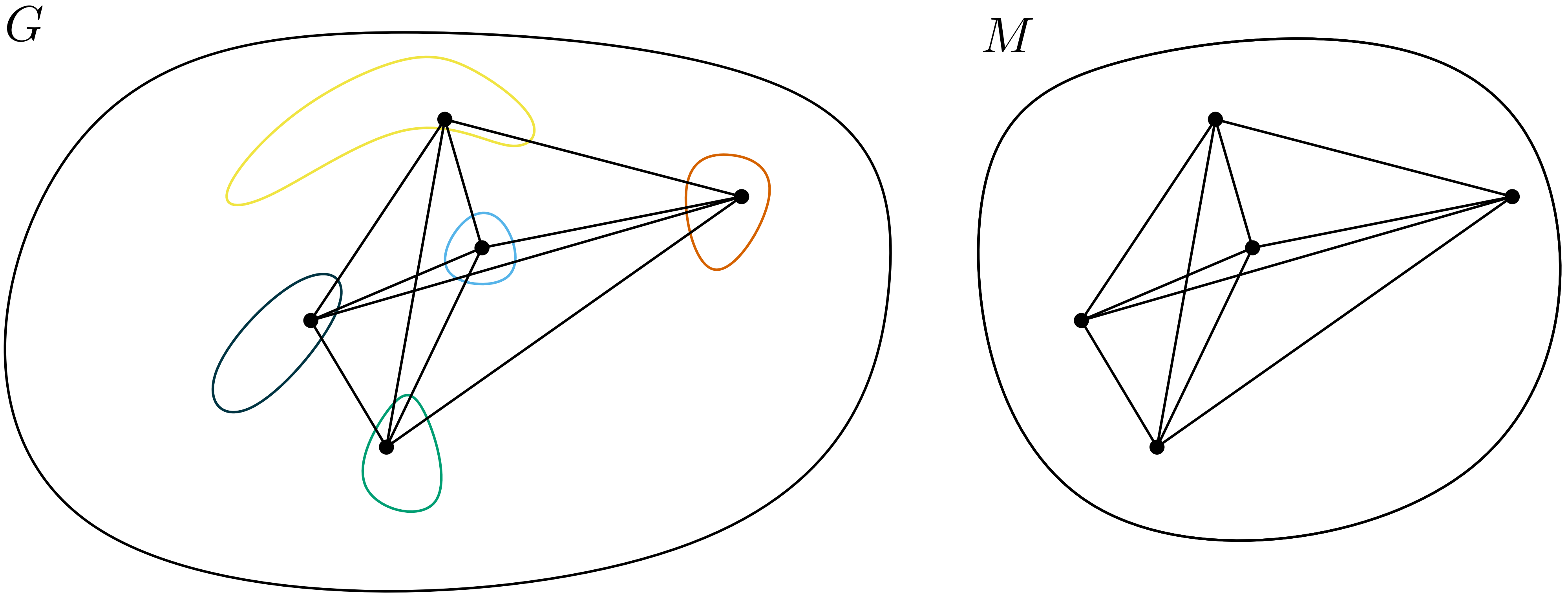
G



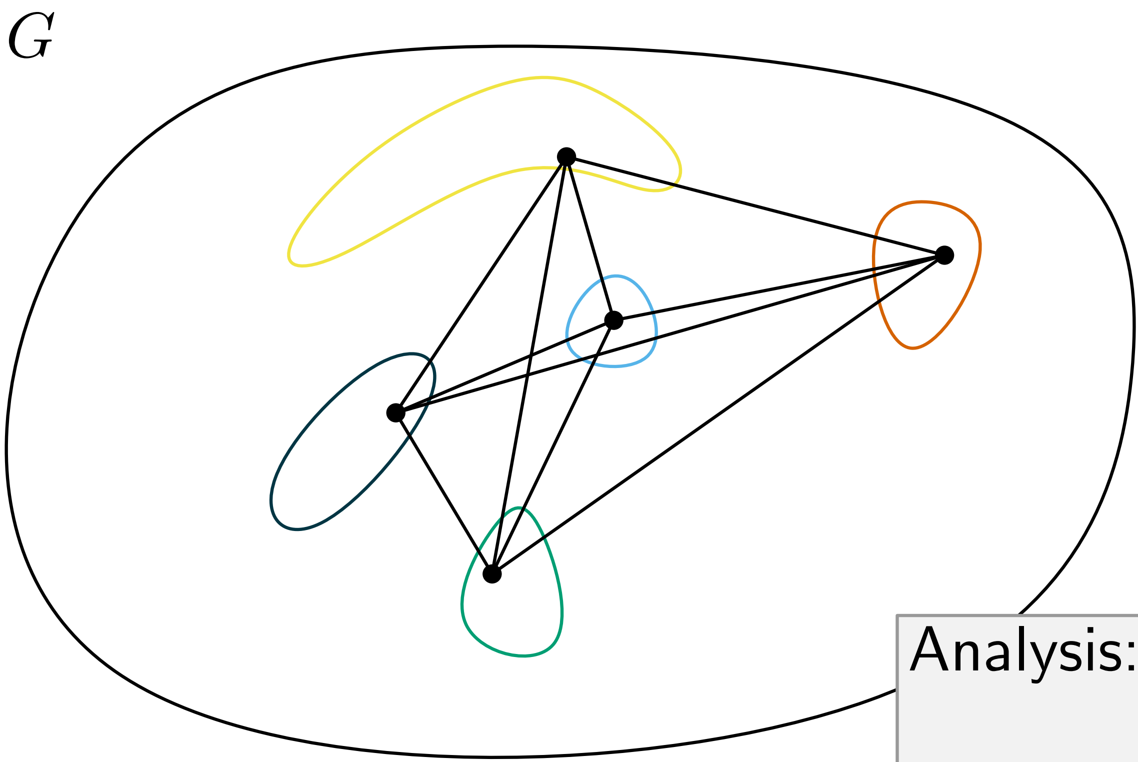
M

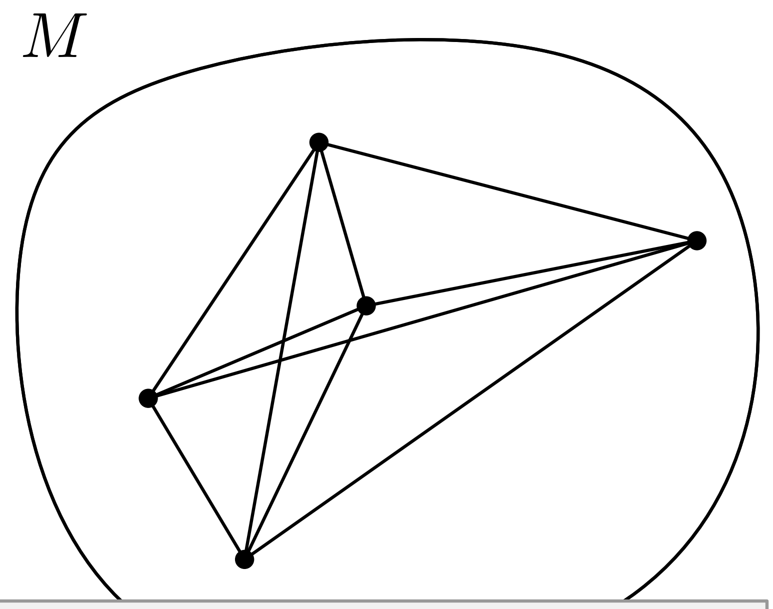


Algorithm revisited



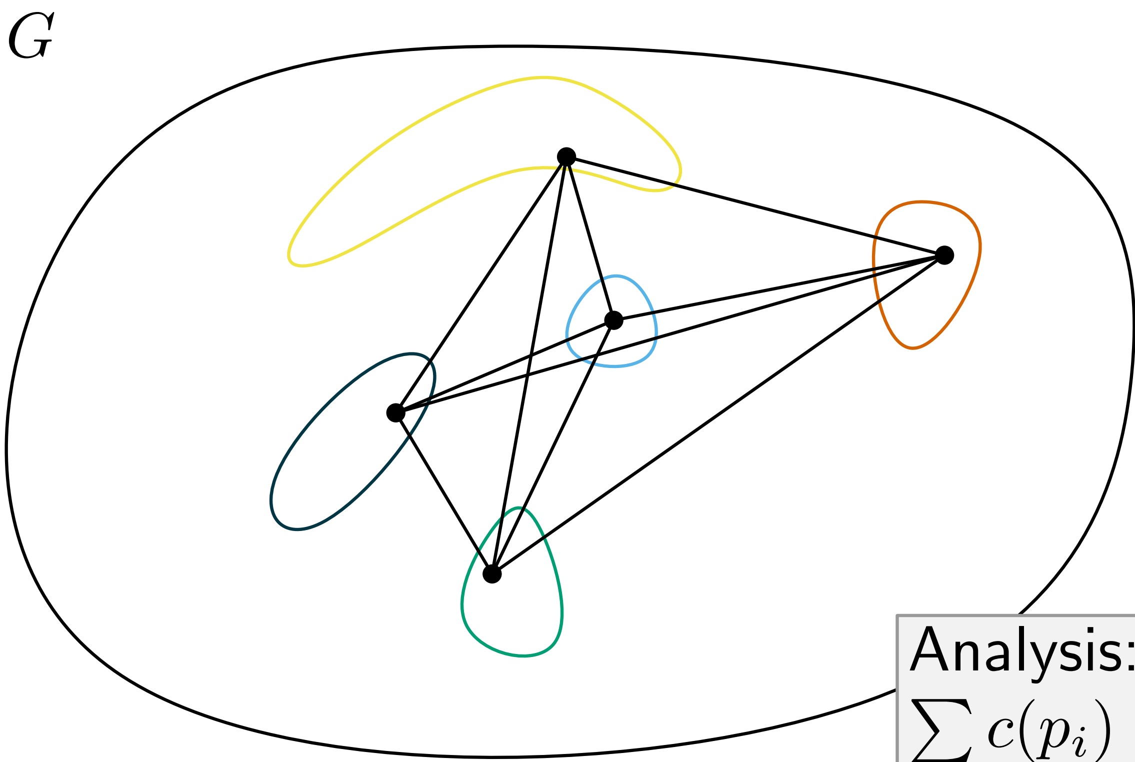
Algorithm revisited

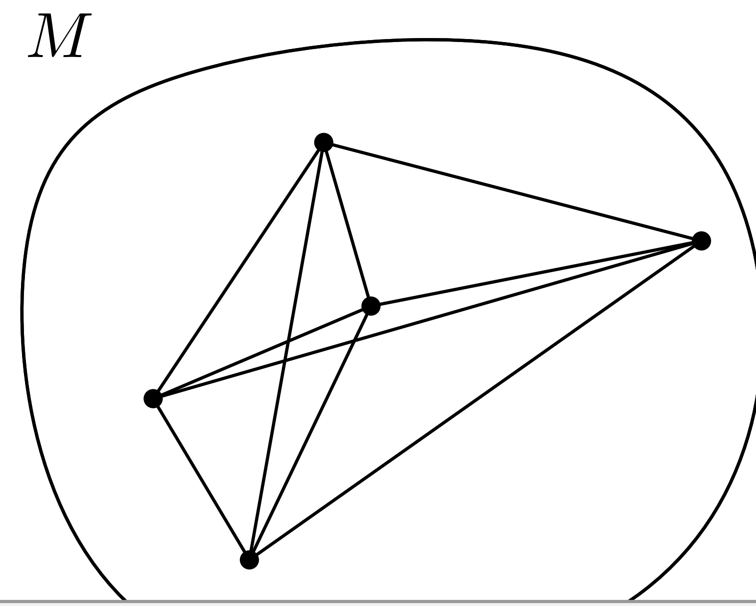




Analysis:

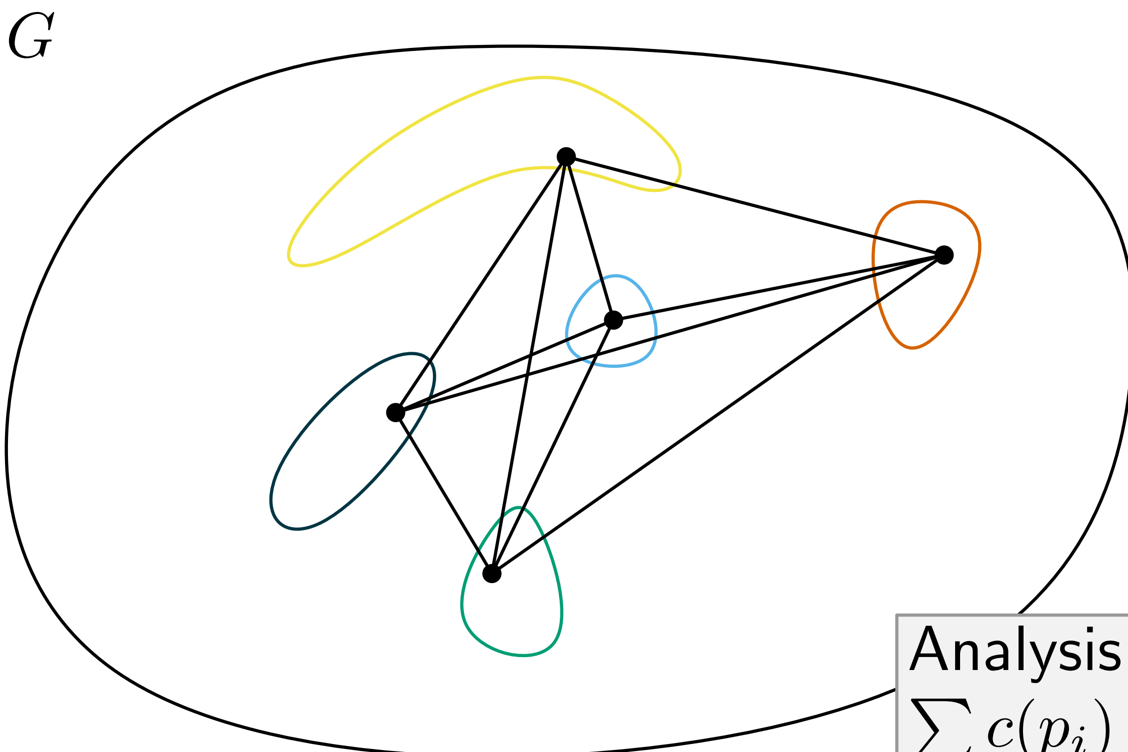
Algorithm revisited

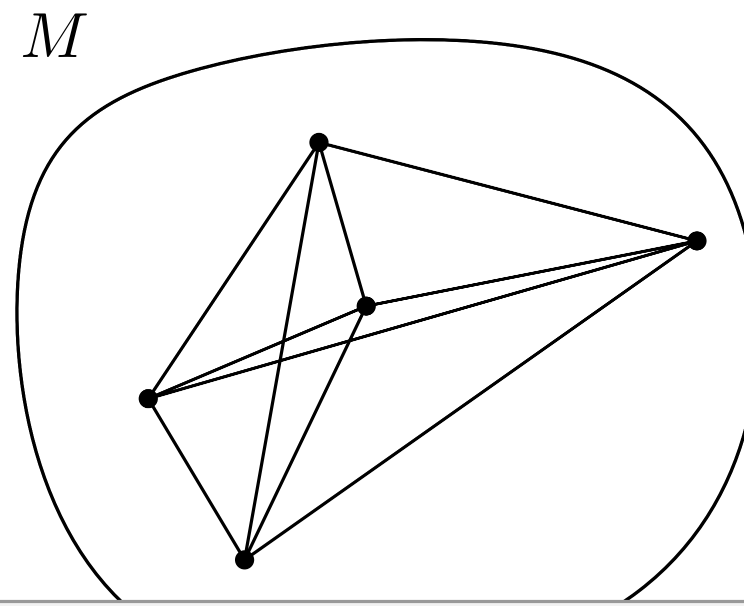




Analysis:
$\sum c(p_i) \leq \sum 2c(T_i) \leq 2TSP(G)$ still holds

Algorithm revisited





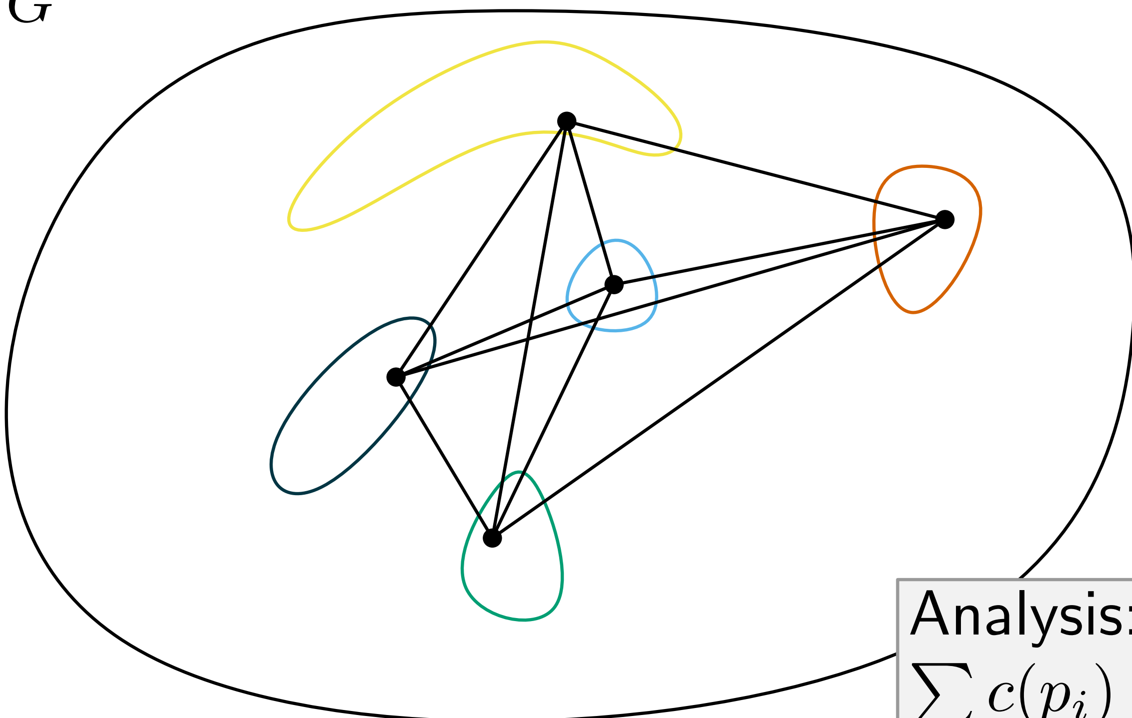
Analysis:

$\sum c(p_i) \leq \sum 2c(T_i) \leq 2TSP(G)$ still holds

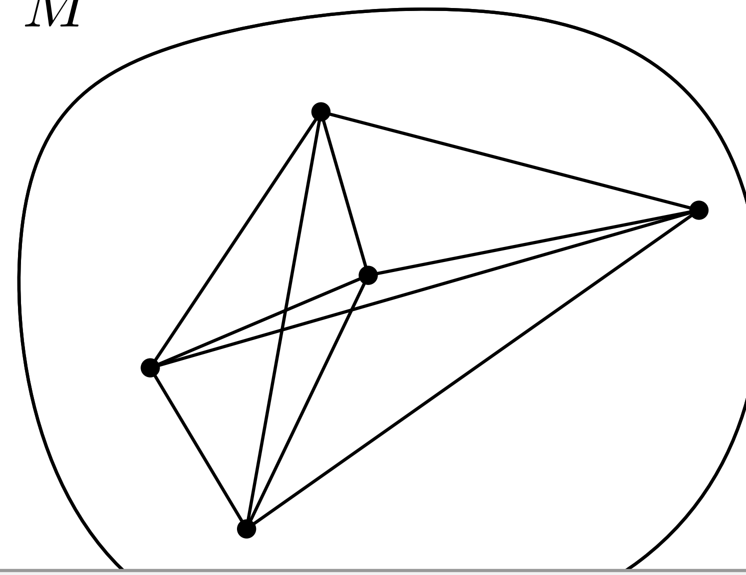
$\tau \leq TSP(G)$ still holds

Algorithm revisited

G



M



Analysis:
$\sum c(p_i) \leq \sum 2c(T_i) \leq 2TSP(G)$ still holds
$\tau \leq TSP(G)$ still holds

$\rightarrow$ 3-approximation

Approximation guarantee vs runtime

Problem: the instances are too asymmetric

Approximation guarantee vs runtime

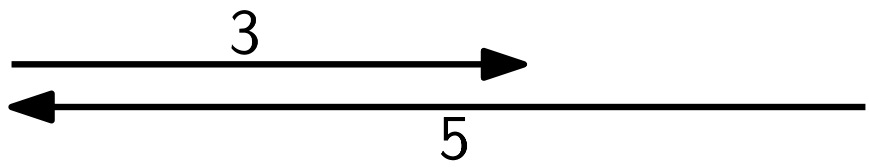
Problem: the instances are too asymmetric

Idea: ignore asymmetries up to a factor of β

Approximation guarantee vs runtime

Problem: the instances are too asymmetric

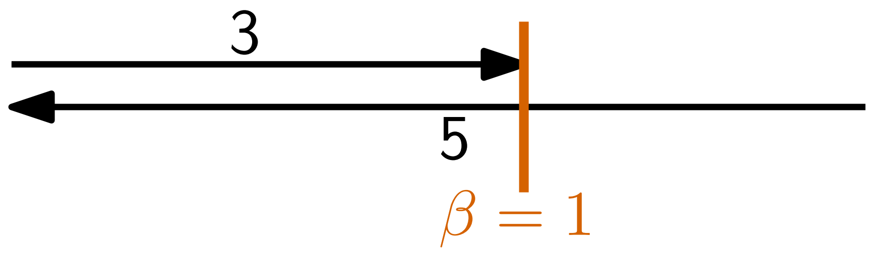
Idea: ignore asymmetries up to a factor of β



Approximation guarantee vs runtime

Problem: the instances are too asymmetric

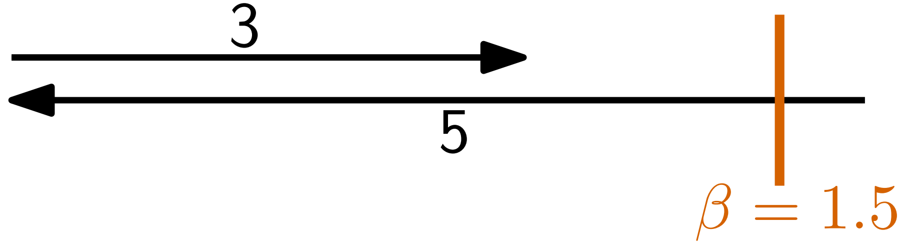
Idea: ignore asymmetries up to a factor of β



Approximation guarantee vs runtime

Problem: the instances are too asymmetric

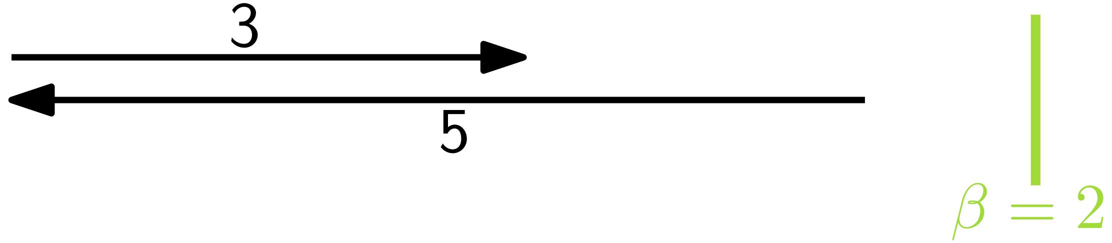
Idea: ignore asymmetries up to a factor of β



Approximation guarantee vs runtime

Problem: the instances are too asymmetric

Idea: ignore asymmetries up to a factor of β



Approximation guarantee vs runtime

Problem: the instances are too asymmetric

Idea: ignore asymmetries up to a factor of β

Generalized Christofides algorithm

Theorem: Metric ATSP can be $(\frac{7}{4} + \frac{3}{4}\beta)$-approximated in $\mathcal{O}^*(2^{k_\beta})$, where k_β is the size of a vertex cover for the graph induced by all β-asymmetric links, for any $\beta \geq 1$.

Approximation guarantee vs runtime

Problem: the instances are too asymmetric

Idea: ignore asymmetries up to a factor of β

Generalized Christofides algorithm

Theorem: Metric ATSP can be $(\frac{7}{4} + \frac{3}{4}\beta)$-approximated in $\mathcal{O}^*(2^{k_\beta})$, where k_β is the size of a vertex cover for the graph induced by all β-asymmetric links, for any $\beta \geq 1$.

Generalized tree doubling algorithm

Theorem: Metric ATSP can be $(2 + \beta)$-approximated in $\mathcal{O}^*(2^{k_\beta})$, where k_β is the number of β-one-way arcs in a given minimum spanning arborescence, for any $\beta \geq 1$.

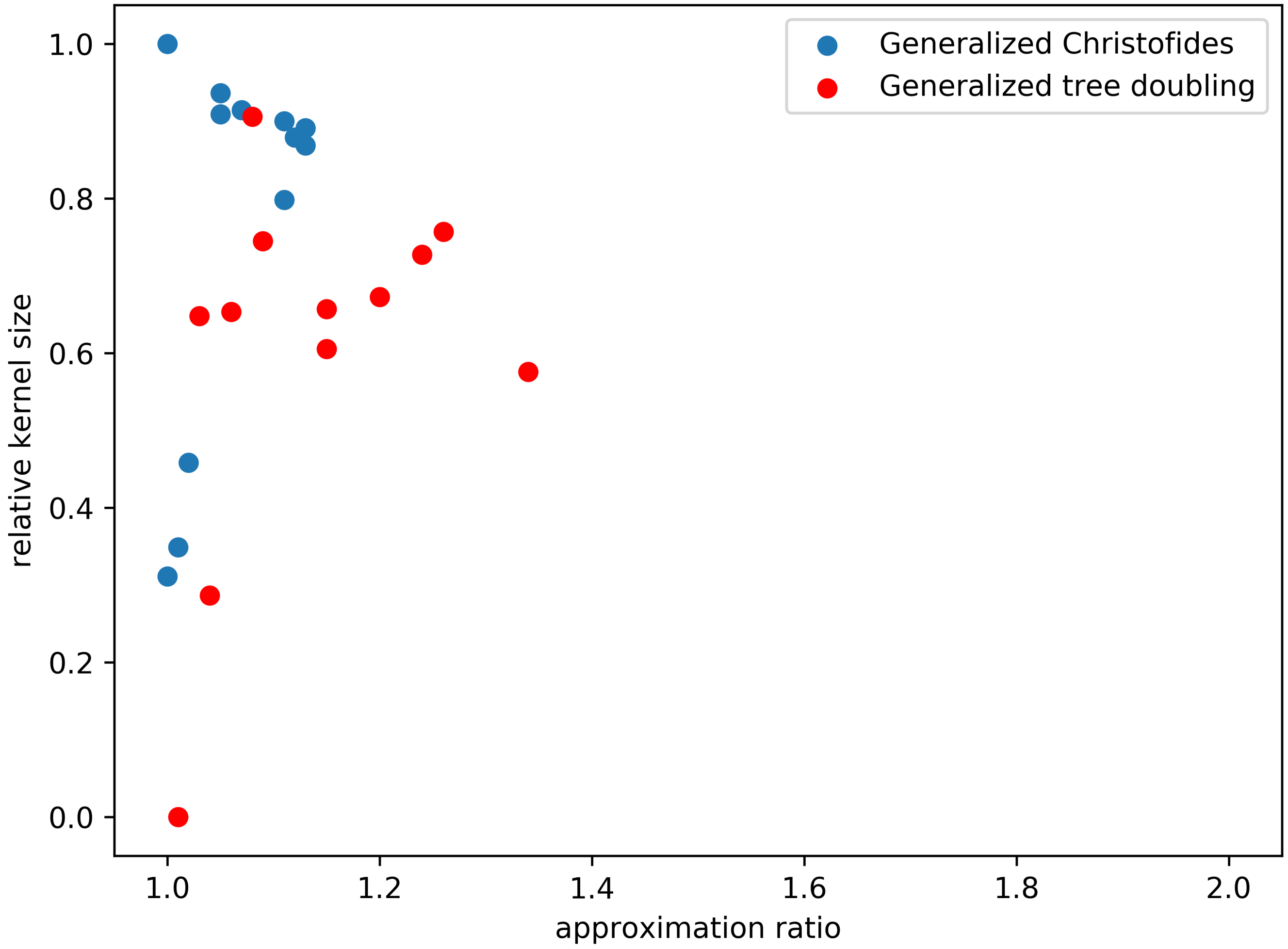
Evaluation

Evaluation

- TSPLIB: most common TSP benchmark; 19 asymmetric instances

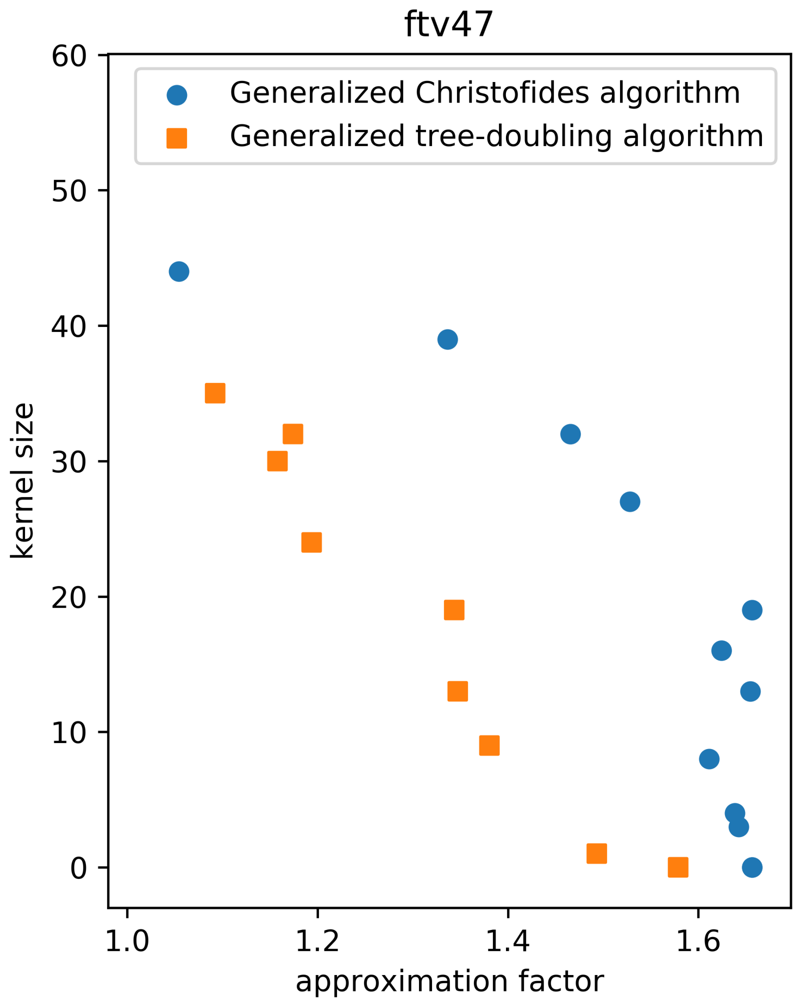
Evaluation

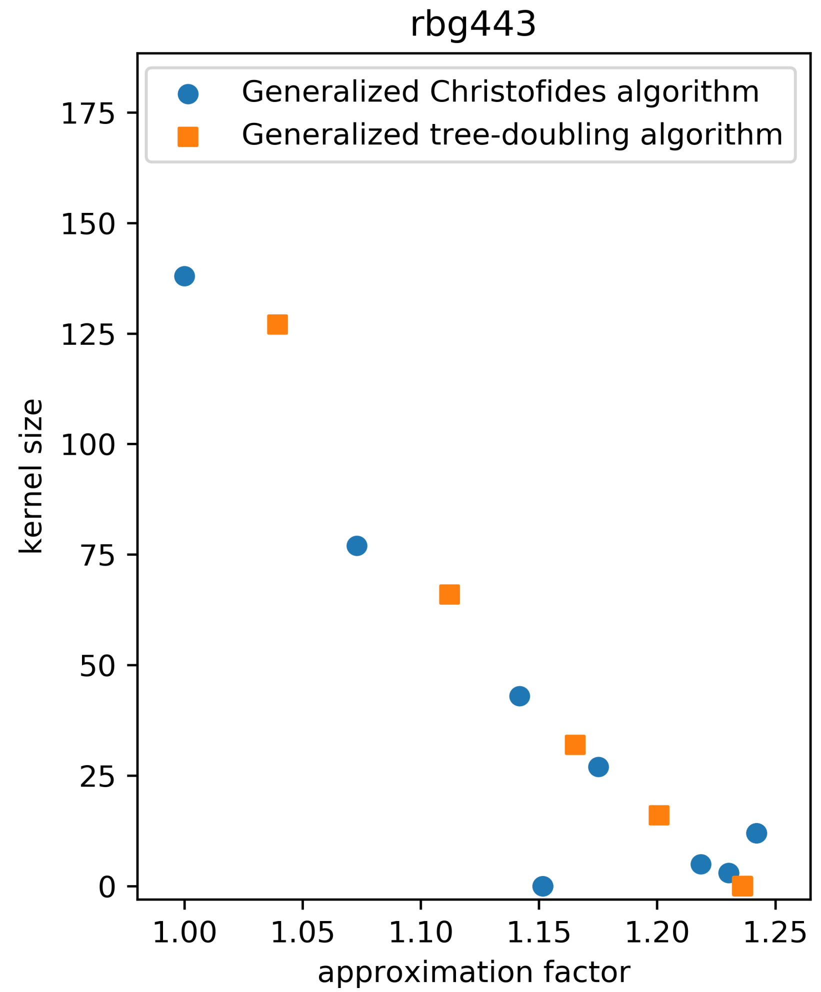
- TSPLIB: most common TSP benchmark; 19 asymmetric instances



Evaluation

- TSPLIB: most common TSP benchmark; 19 asymmetric instances



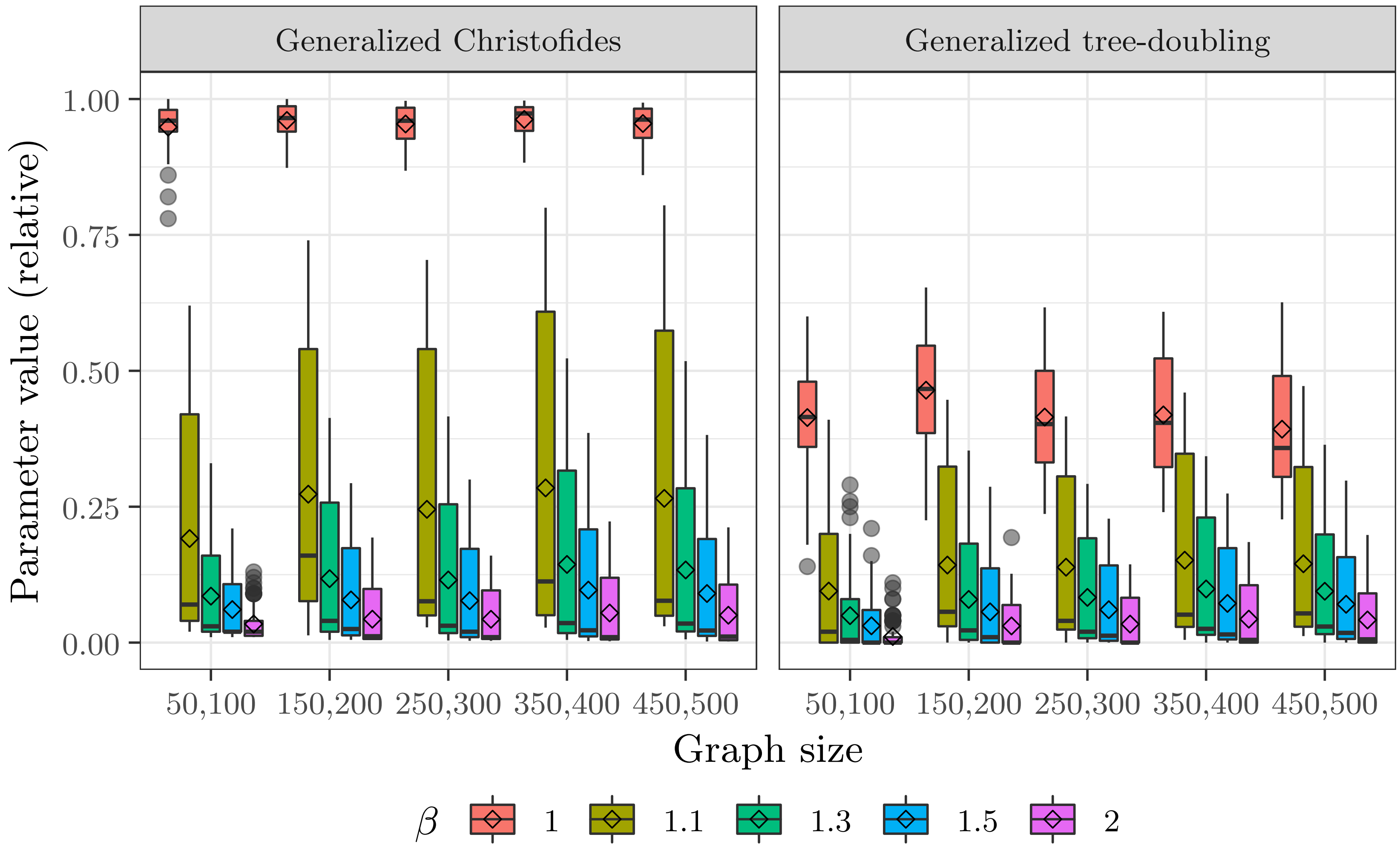


Evaluation

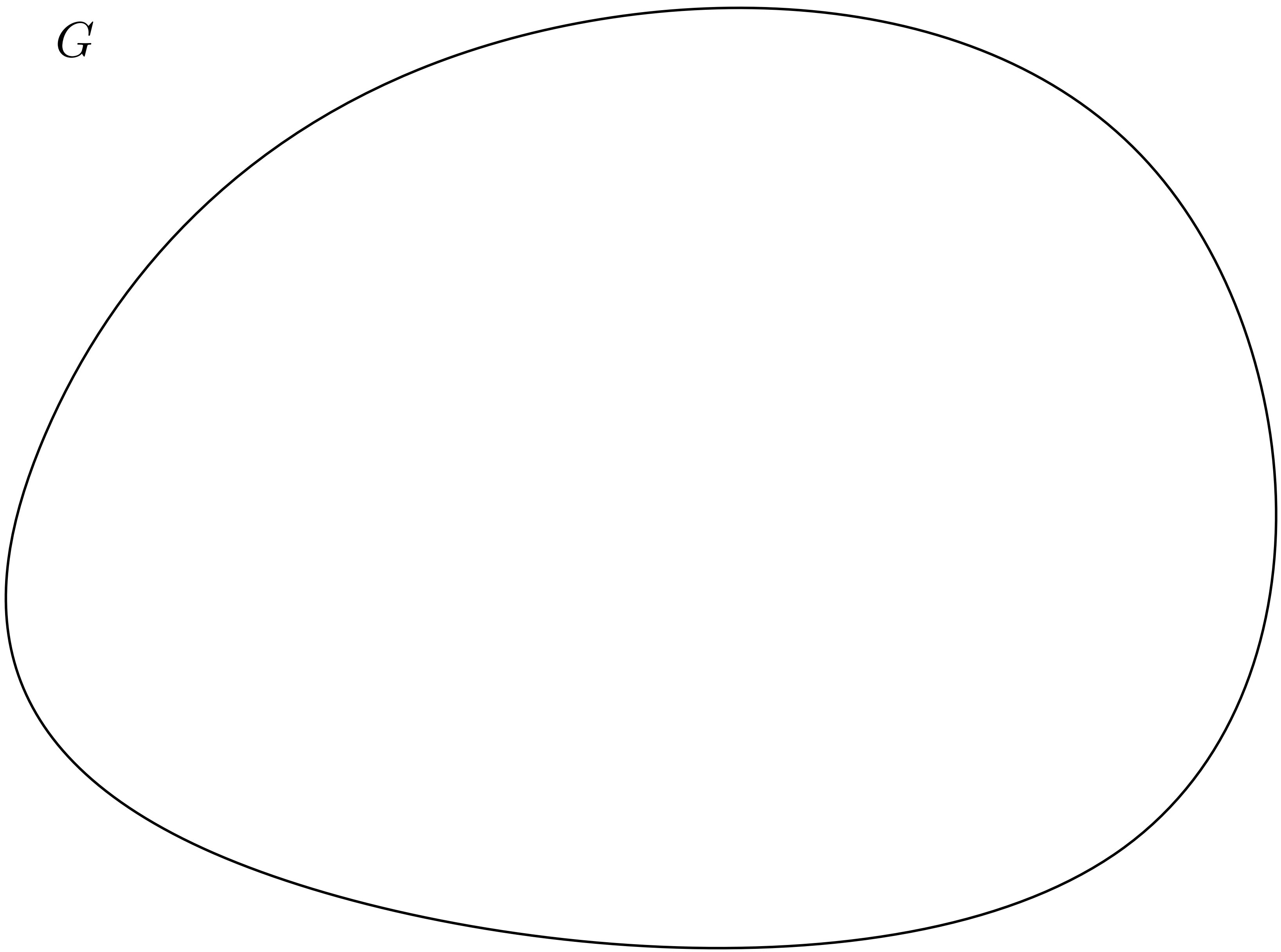
- road networks by Rodríguez and Ruiz

Evaluation

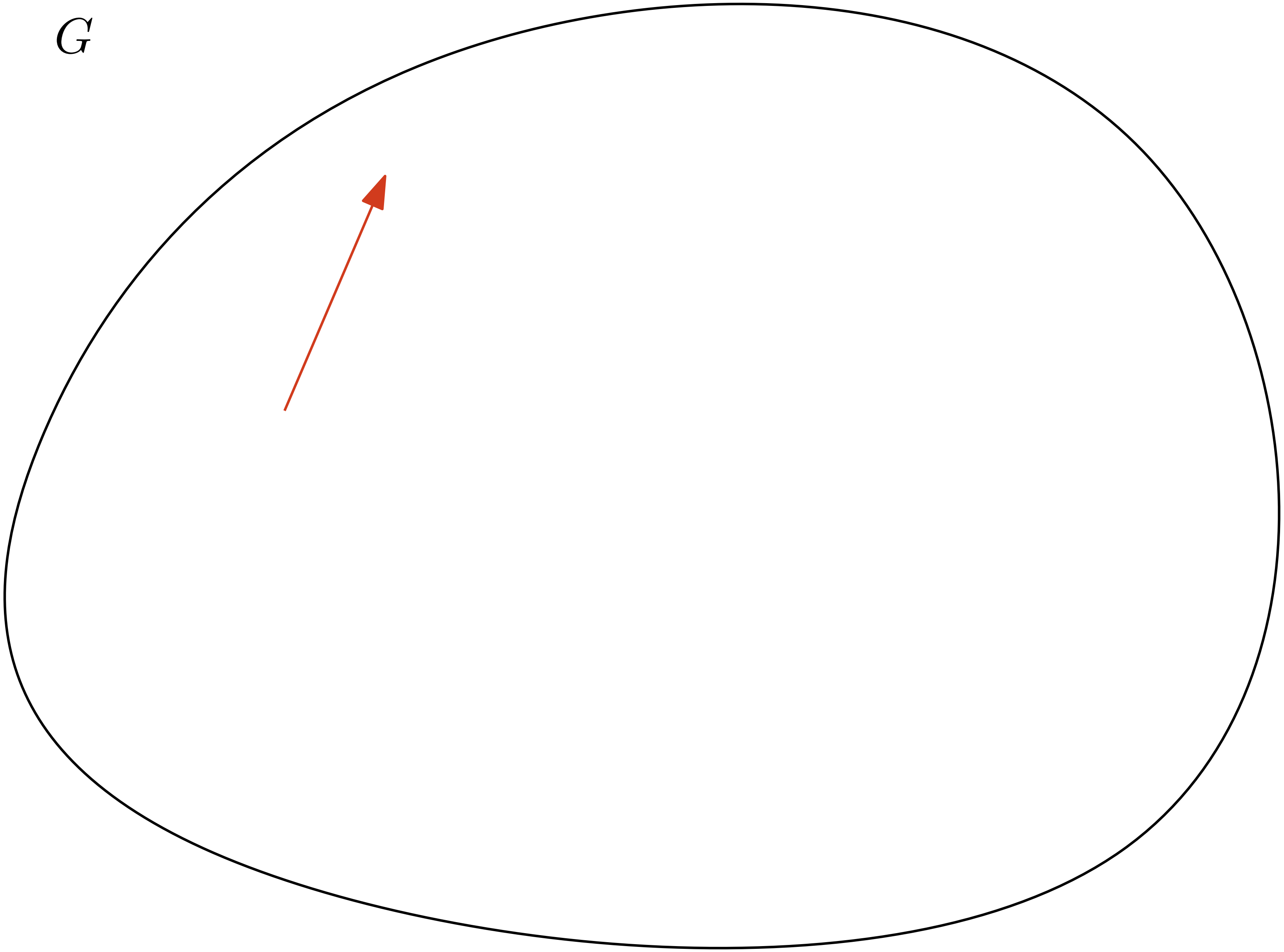
- road networks by Rodríguez and Ruiz



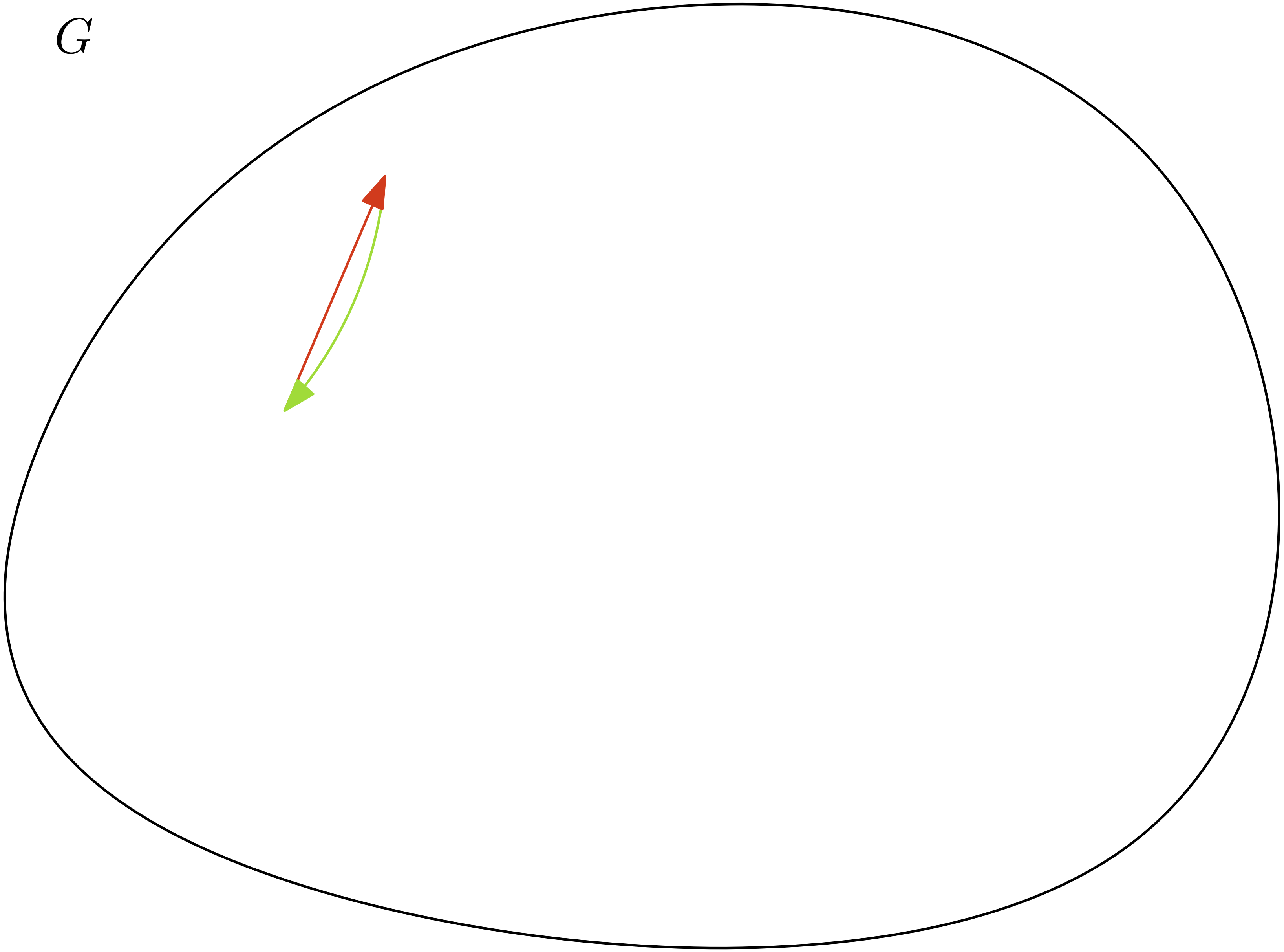
The generalized Christofides algorithm



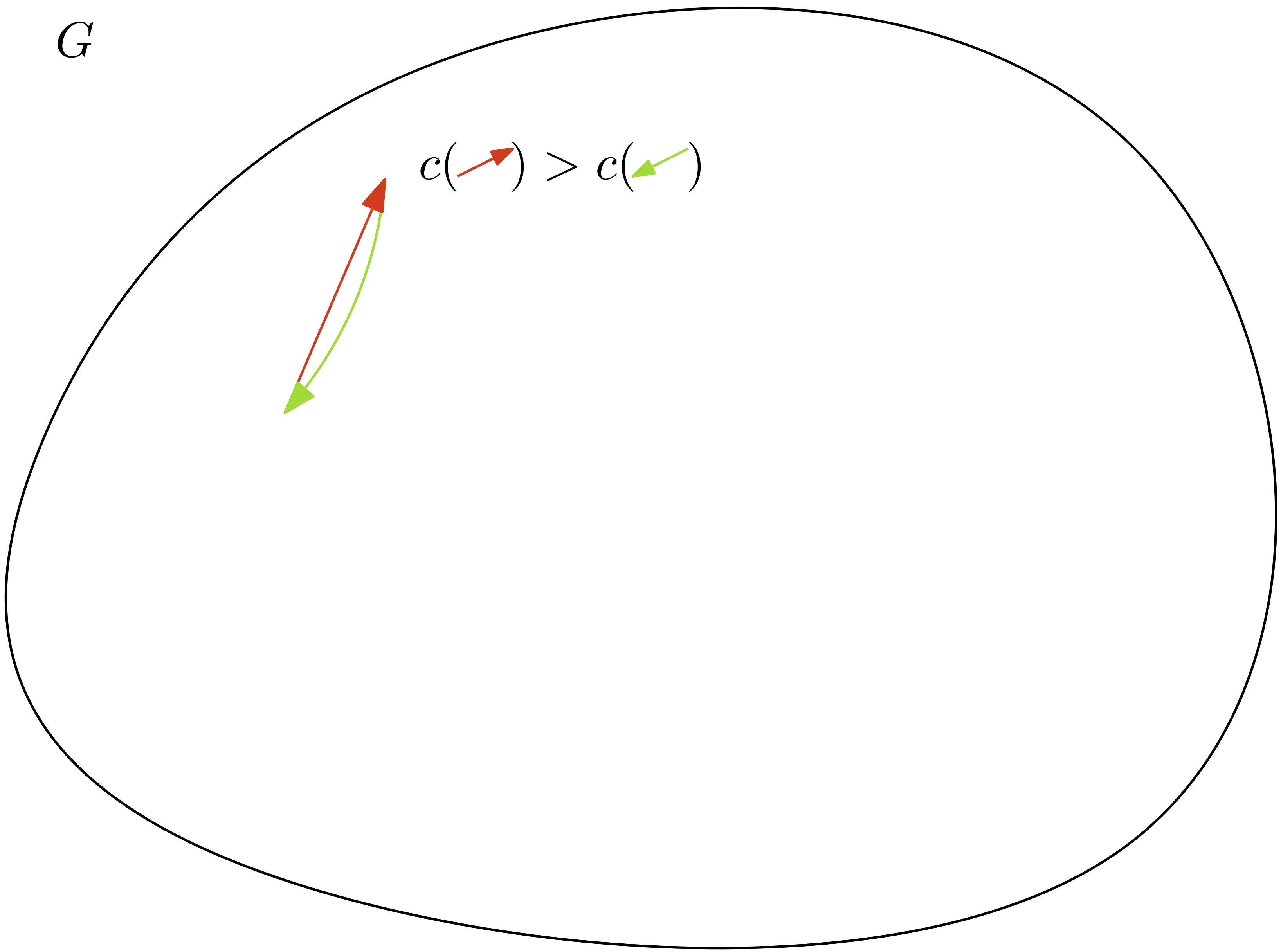
The generalized Christofides algorithm



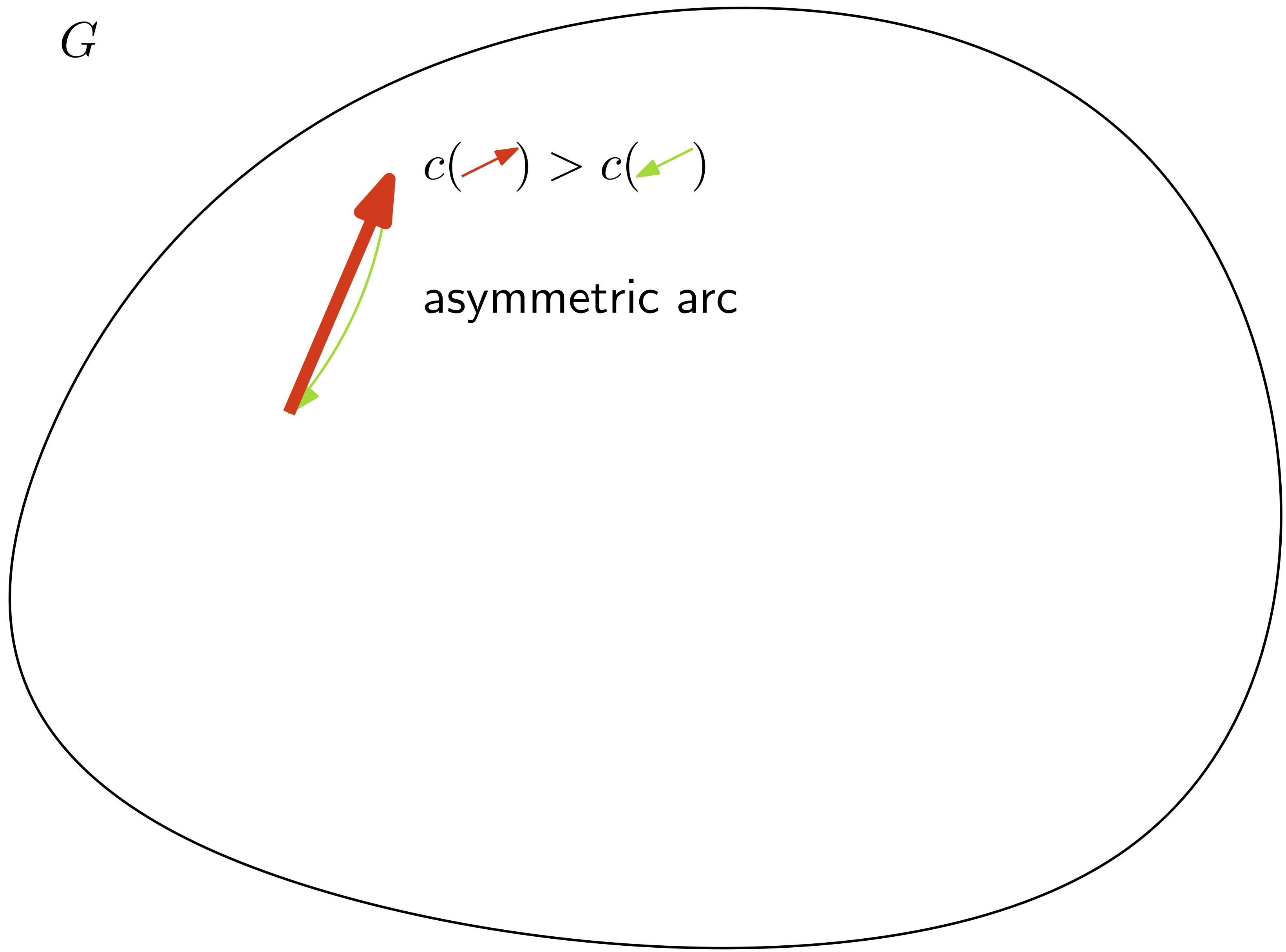
The generalized Christofides algorithm



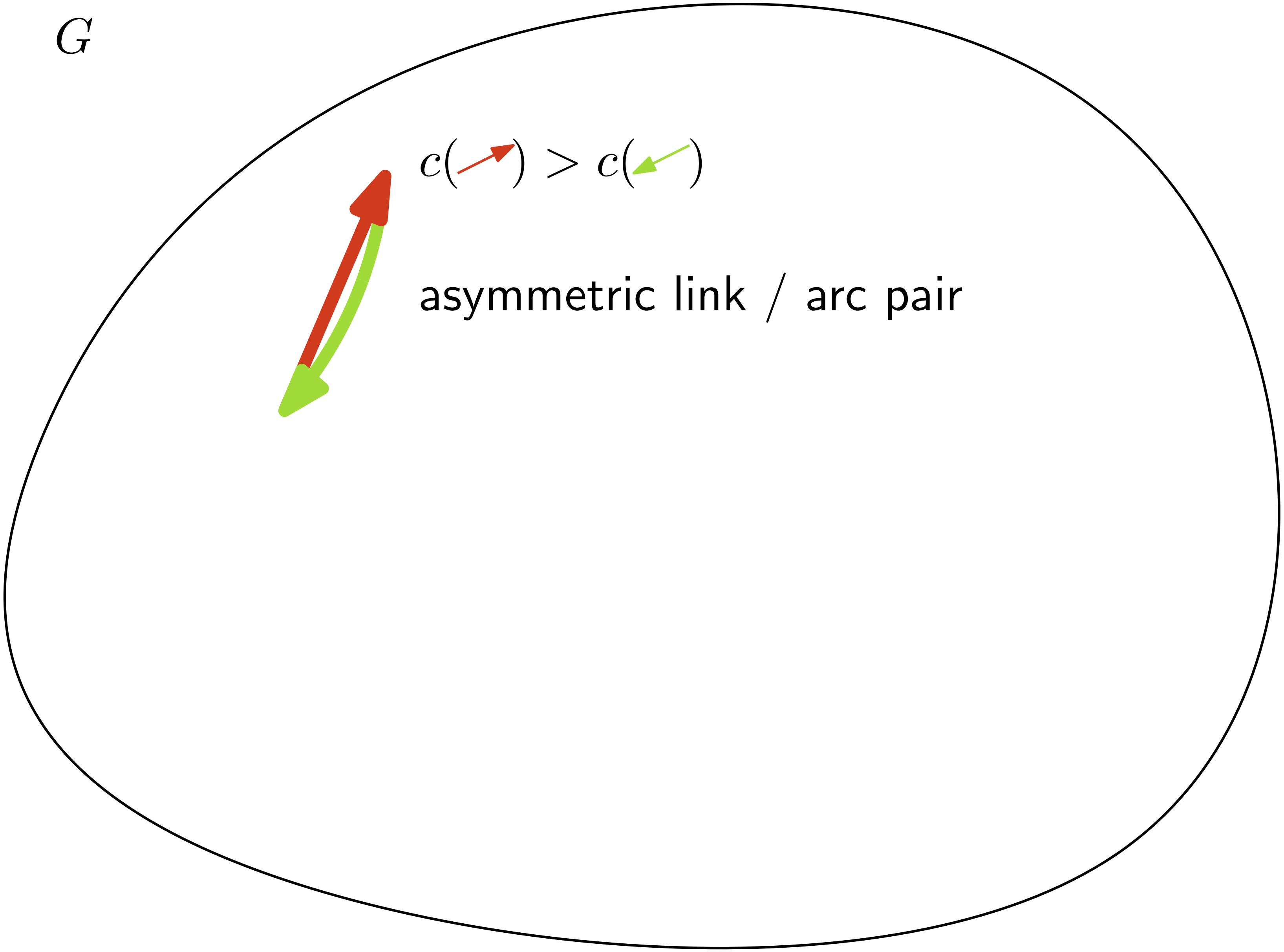
The generalized Christofides algorithm



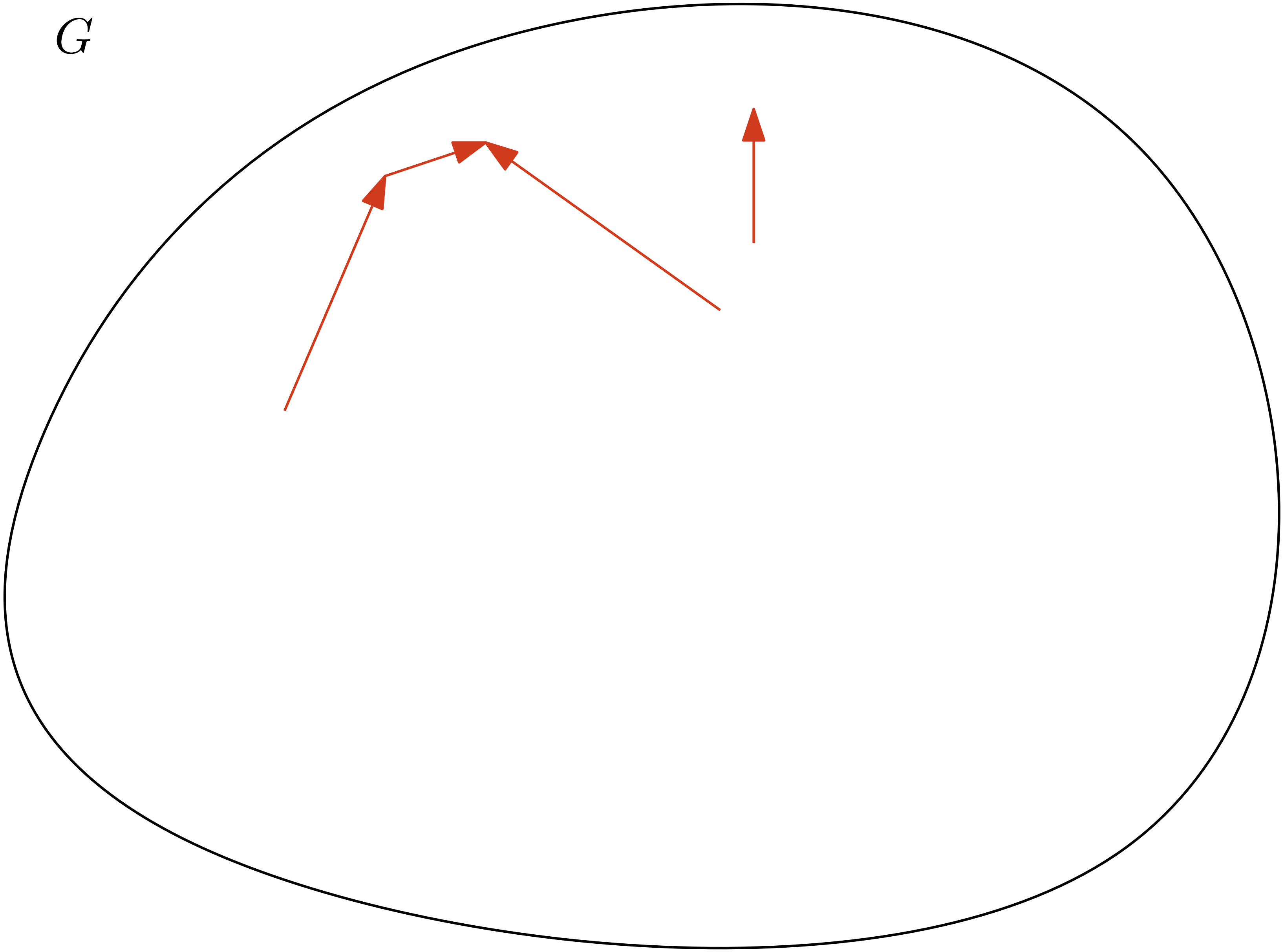
The generalized Christofides algorithm



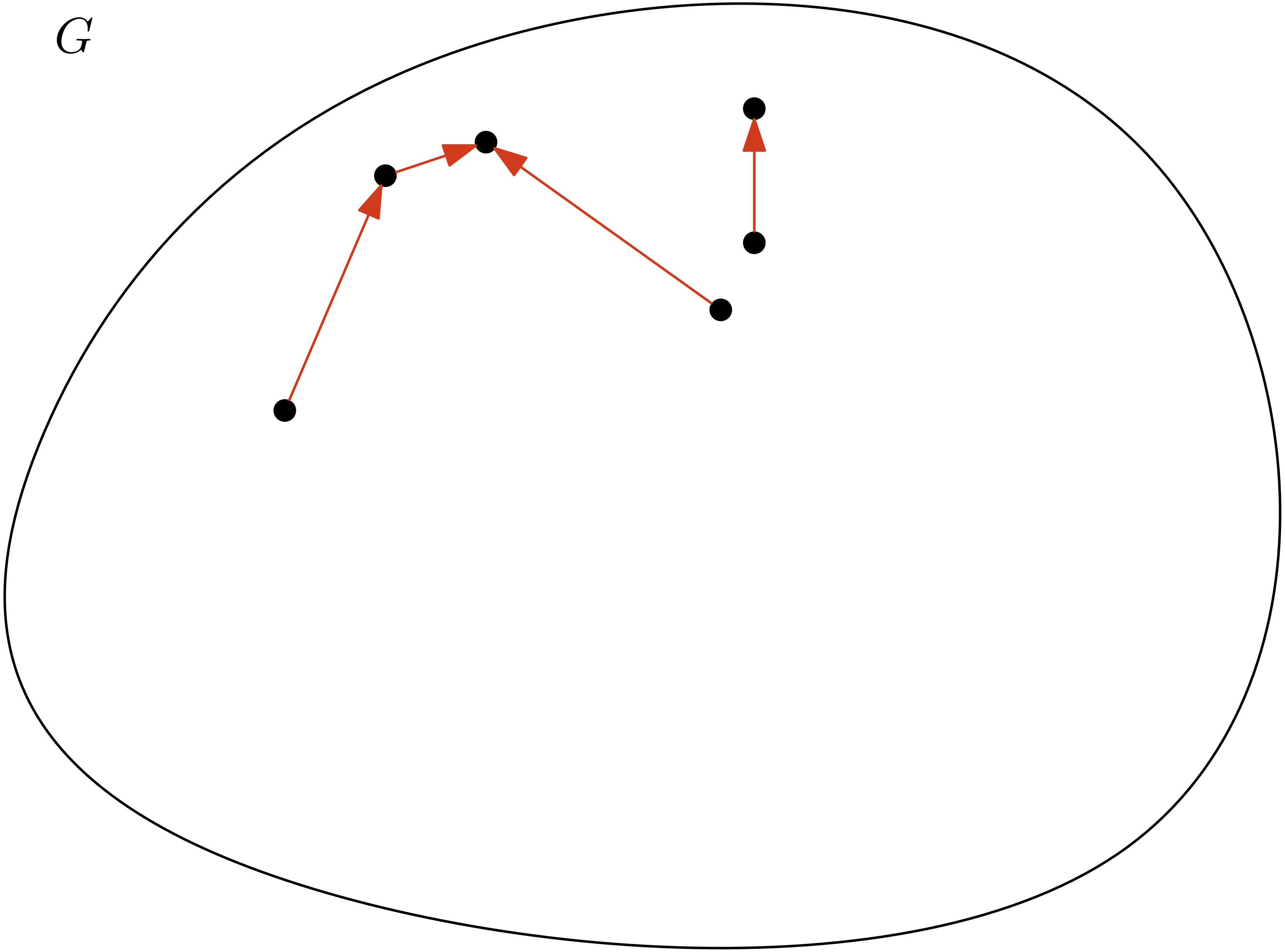
The generalized Christofides algorithm



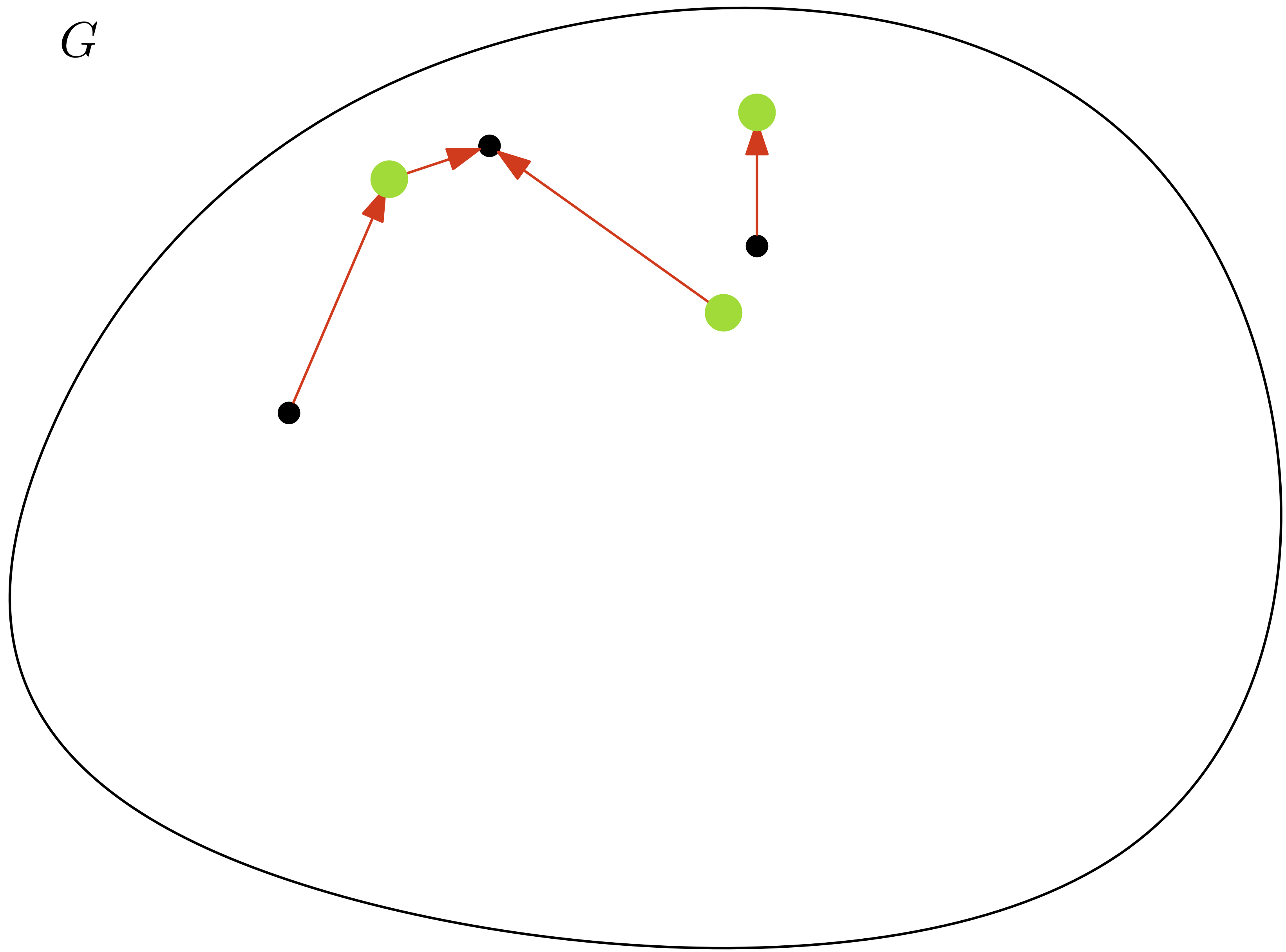
The generalized Christofides algorithm



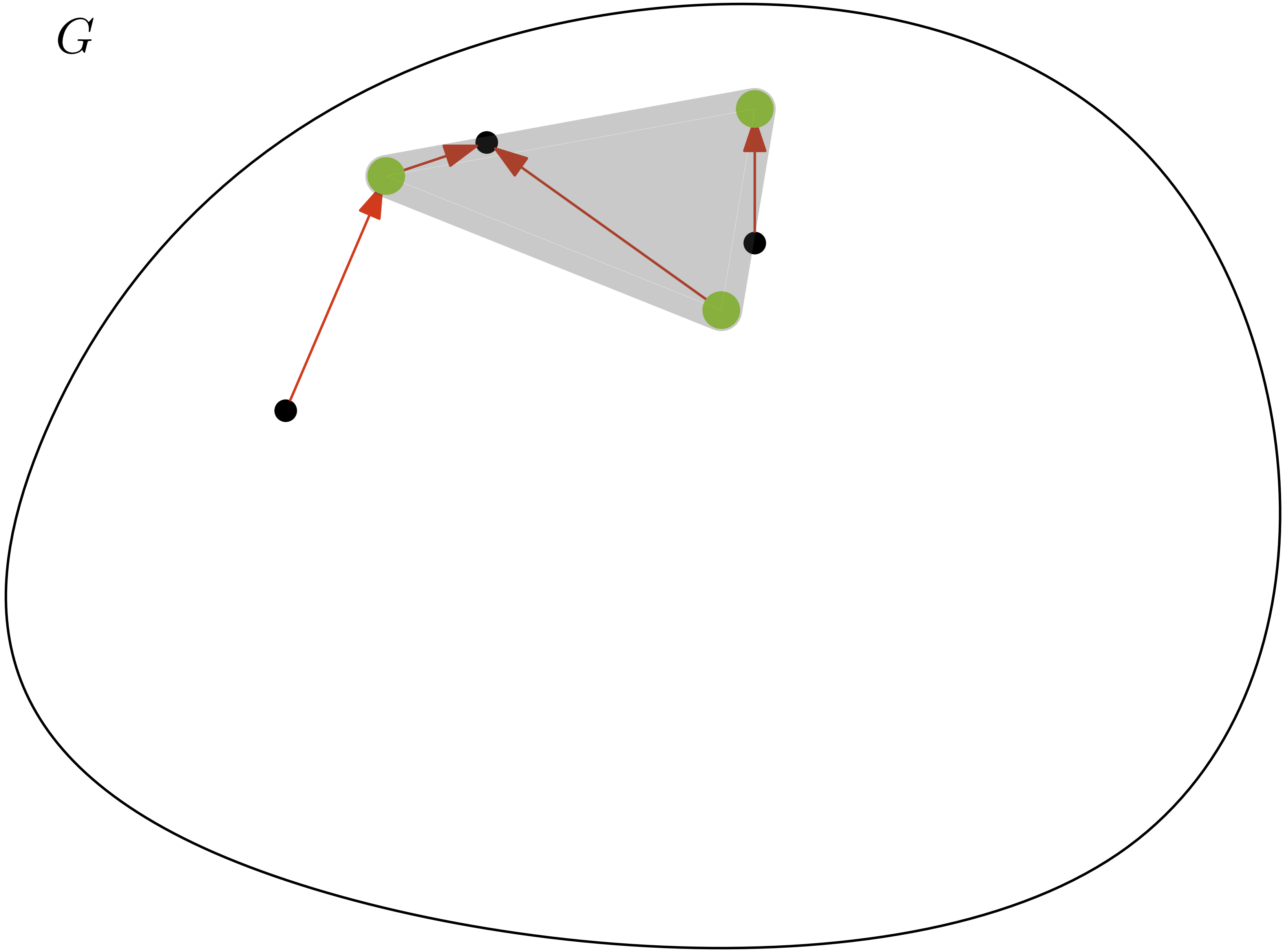
The generalized Christofides algorithm



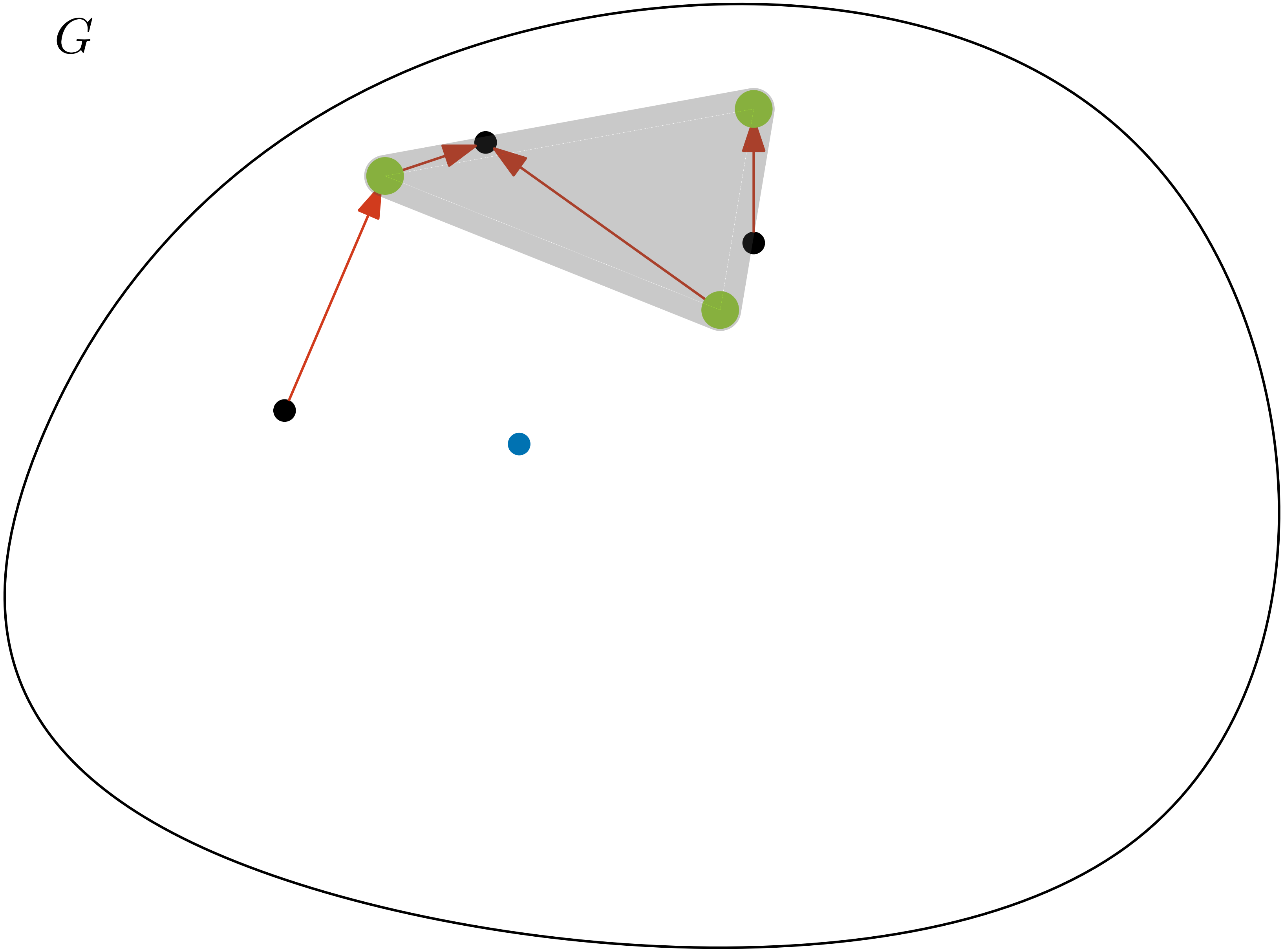
The generalized Christofides algorithm



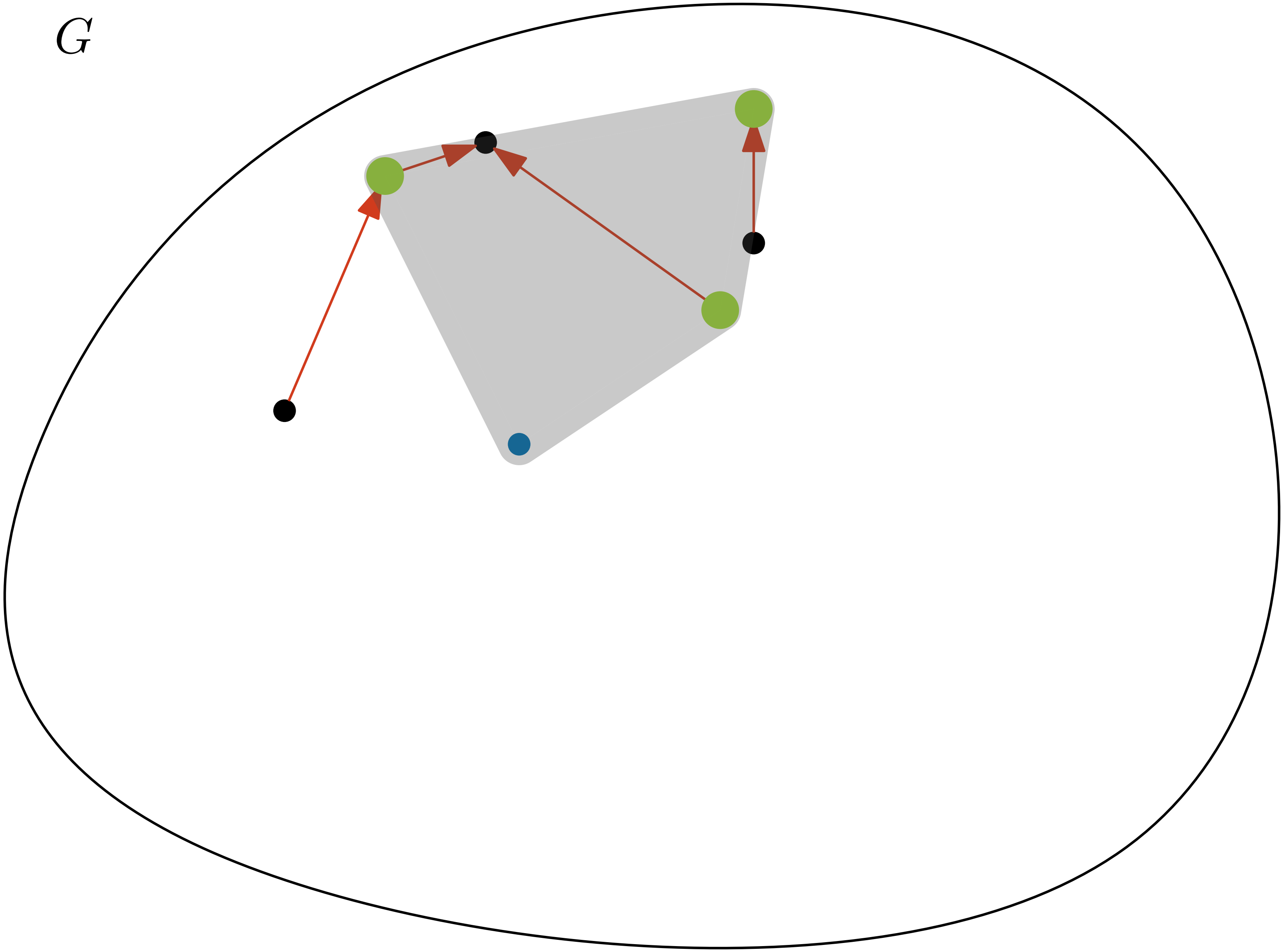
The generalized Christofides algorithm



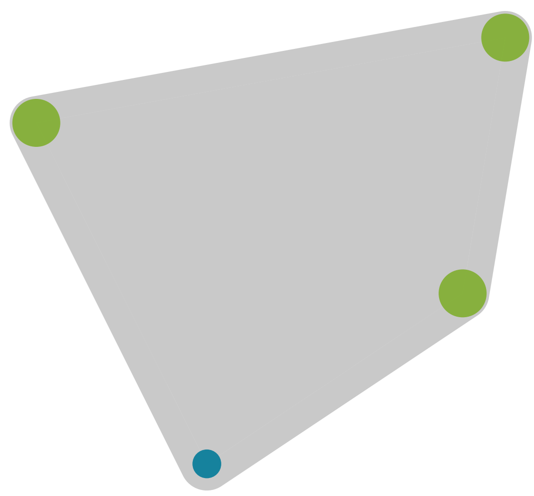
The generalized Christofides algorithm



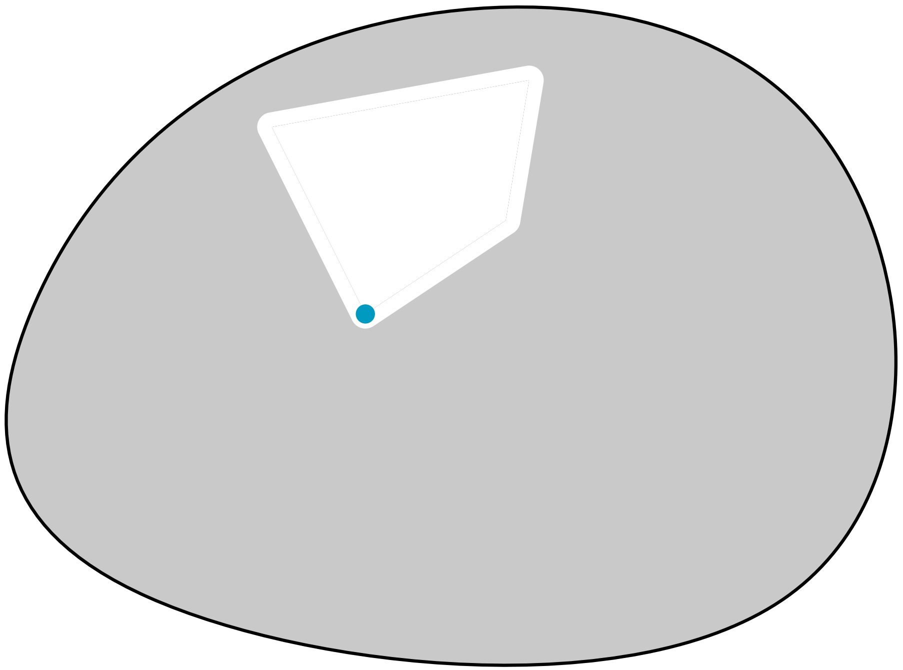
The generalized Christofides algorithm



The generalized Christofides algorithm

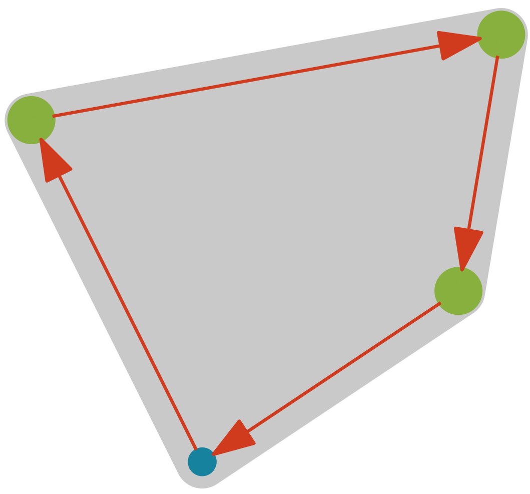


asymmetric subgraph

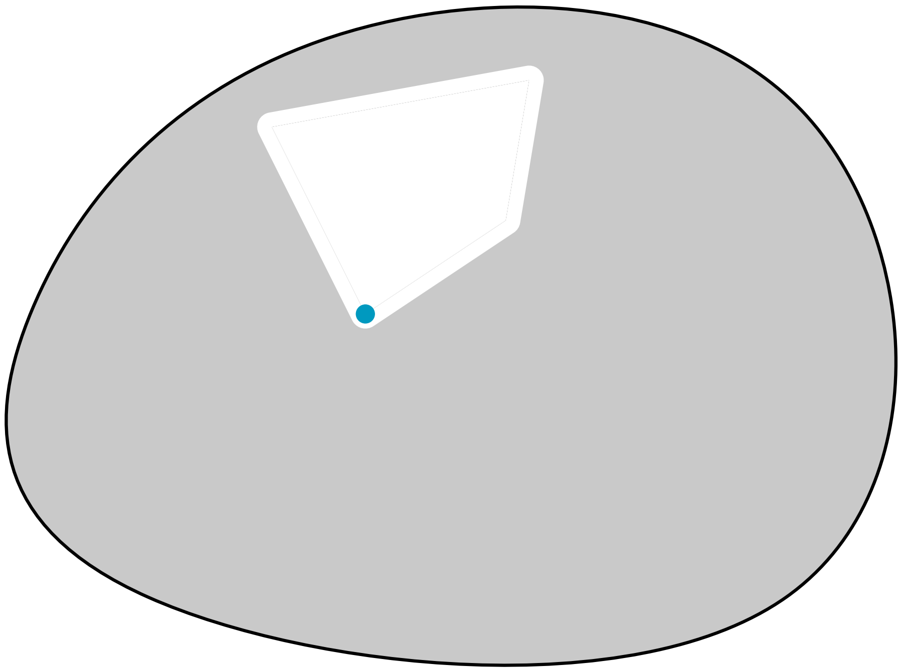


symmetric subgraph

The generalized Christofides algorithm

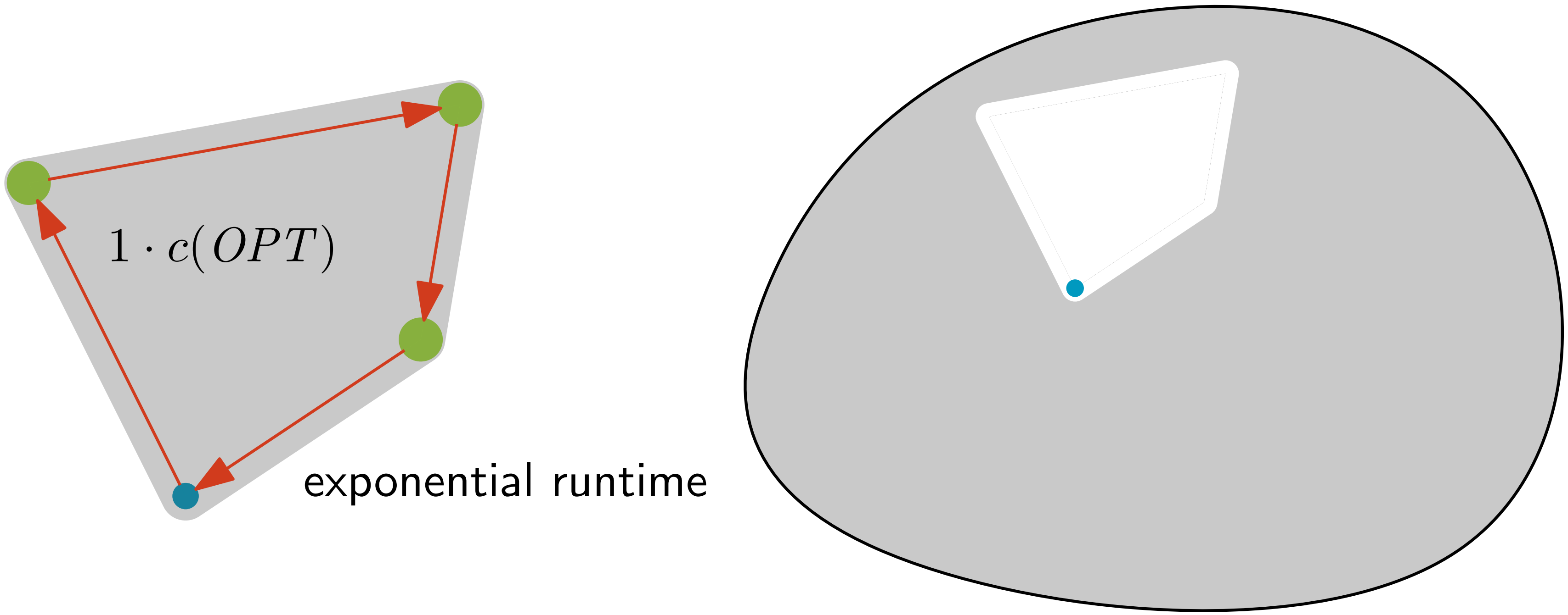


asymmetric subgraph



symmetric subgraph

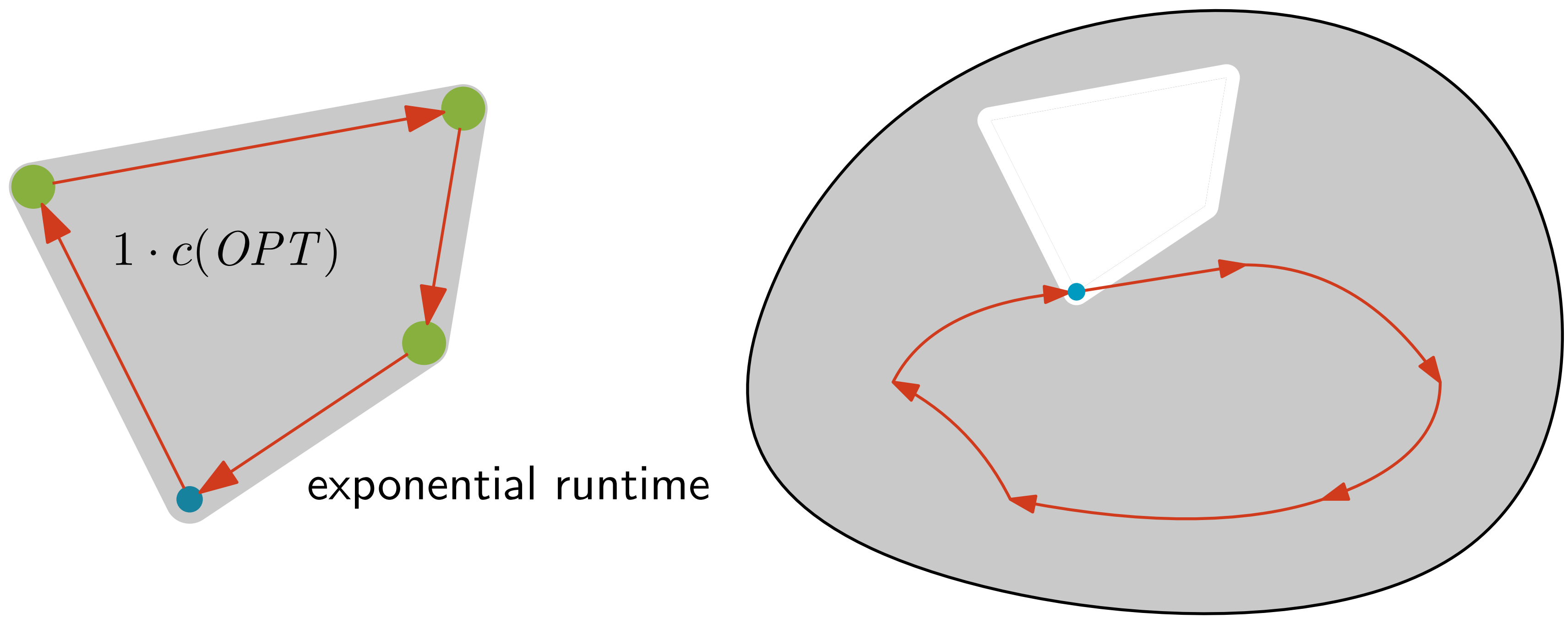
The generalized Christofides algorithm



asymmetric subgraph

symmetric subgraph

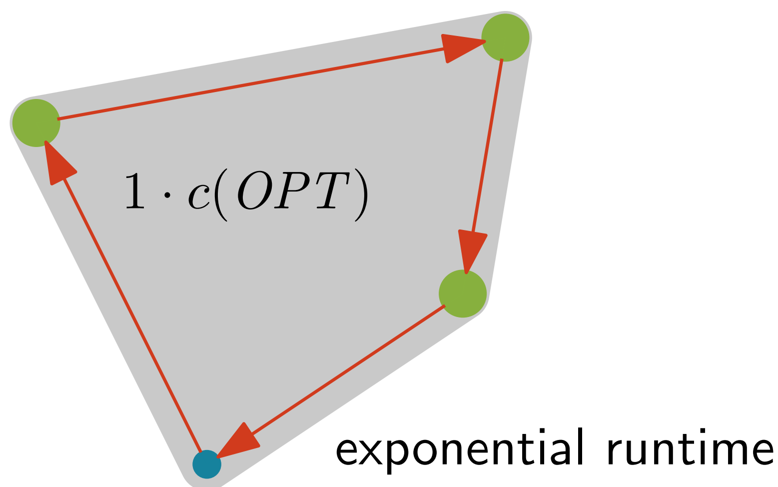
The generalized Christofides algorithm

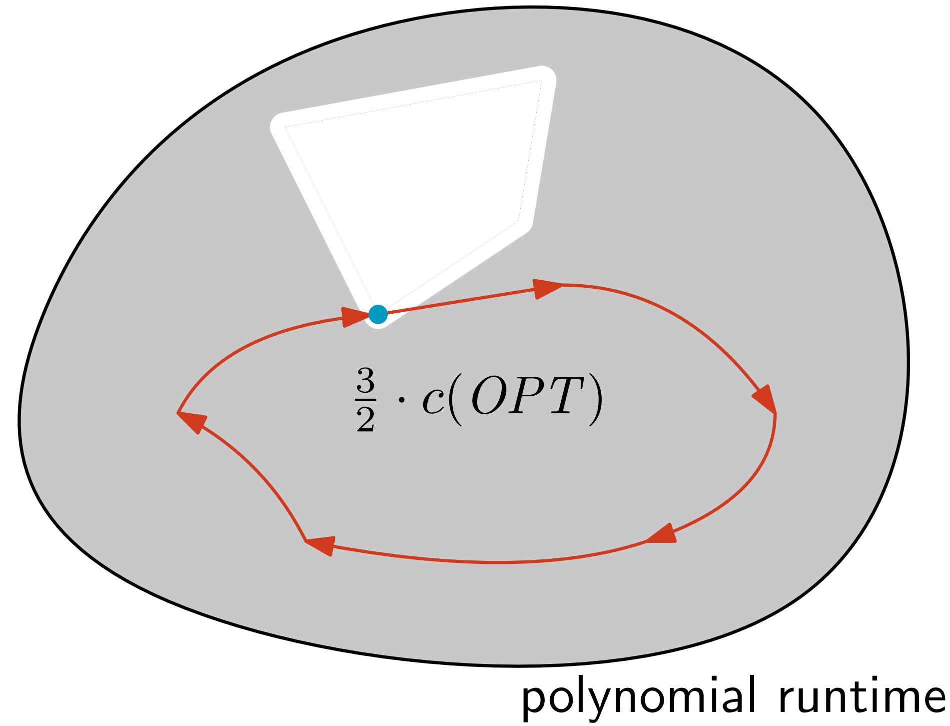


asymmetric subgraph

symmetric subgraph

The generalized Christofides algorithm

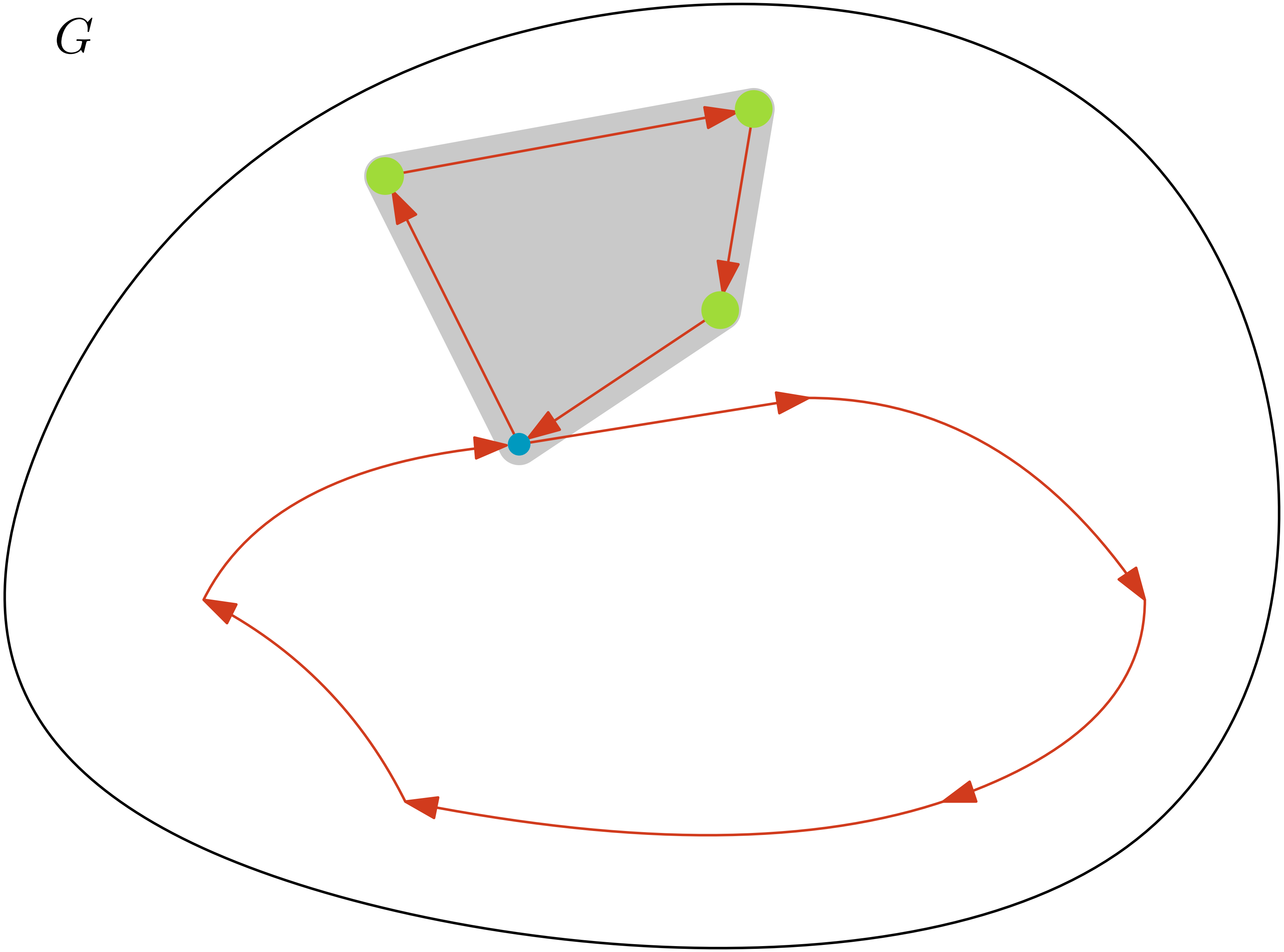




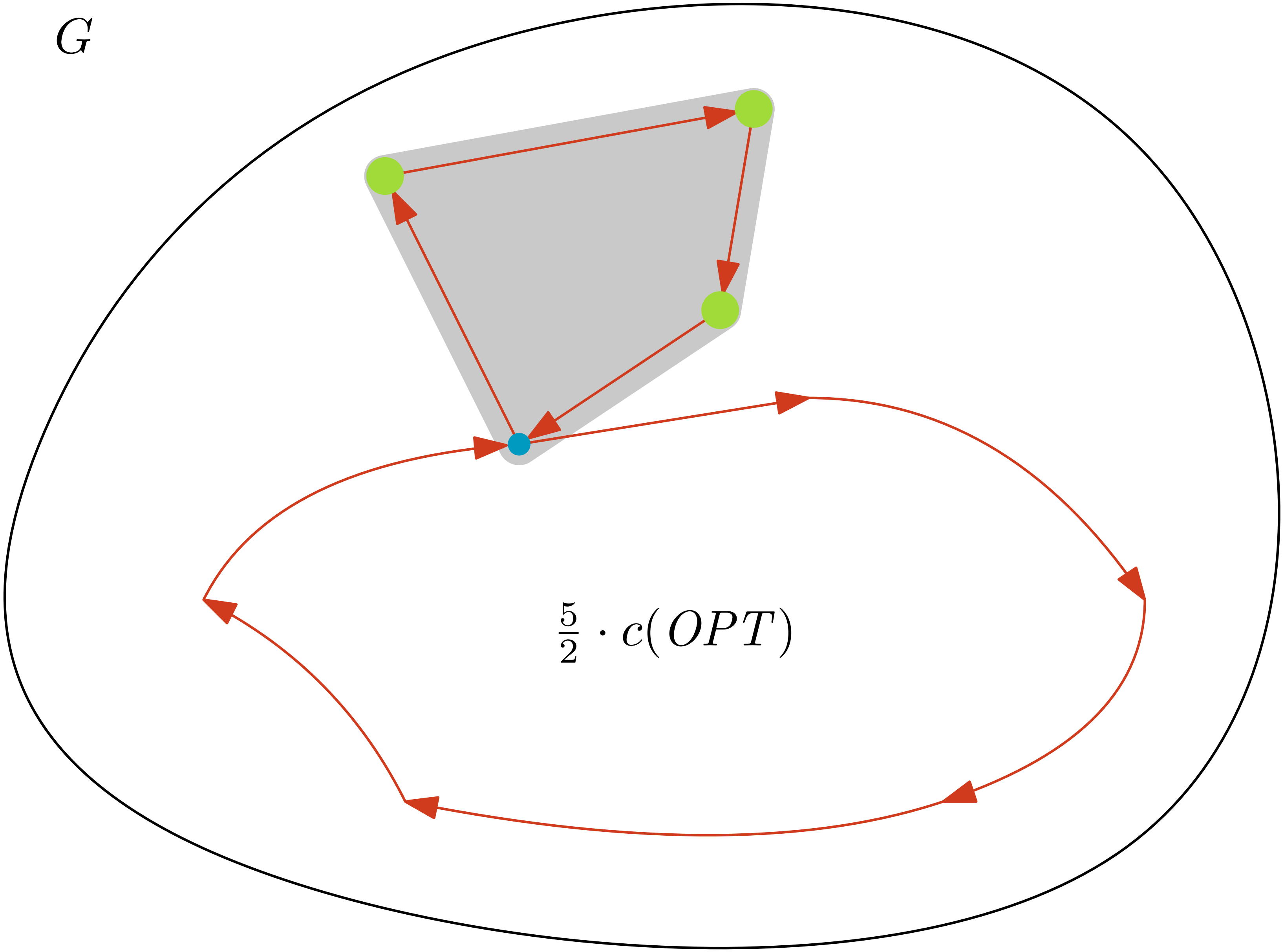
asymmetric subgraph

symmetric subgraph

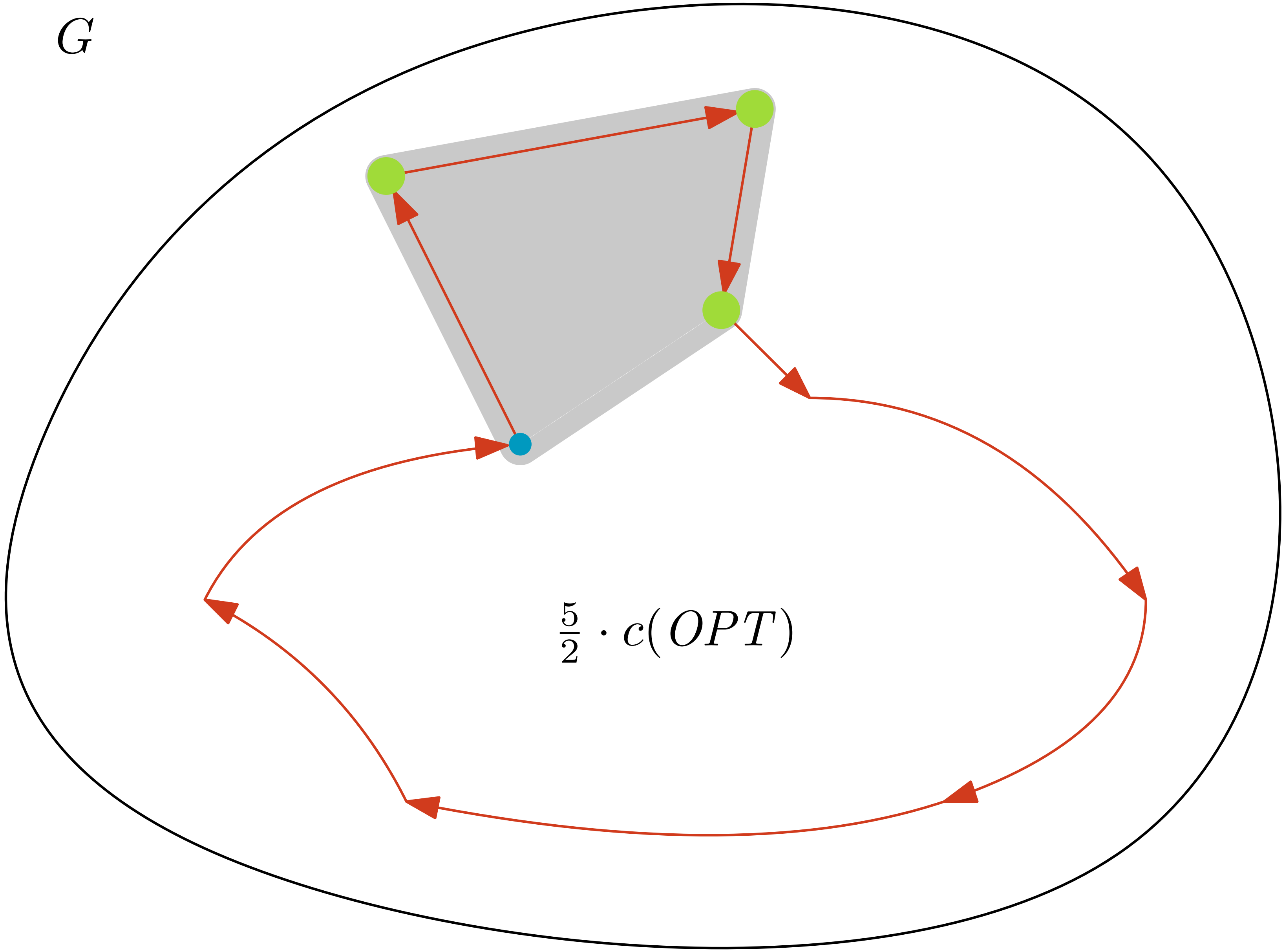
The generalized Christofides algorithm



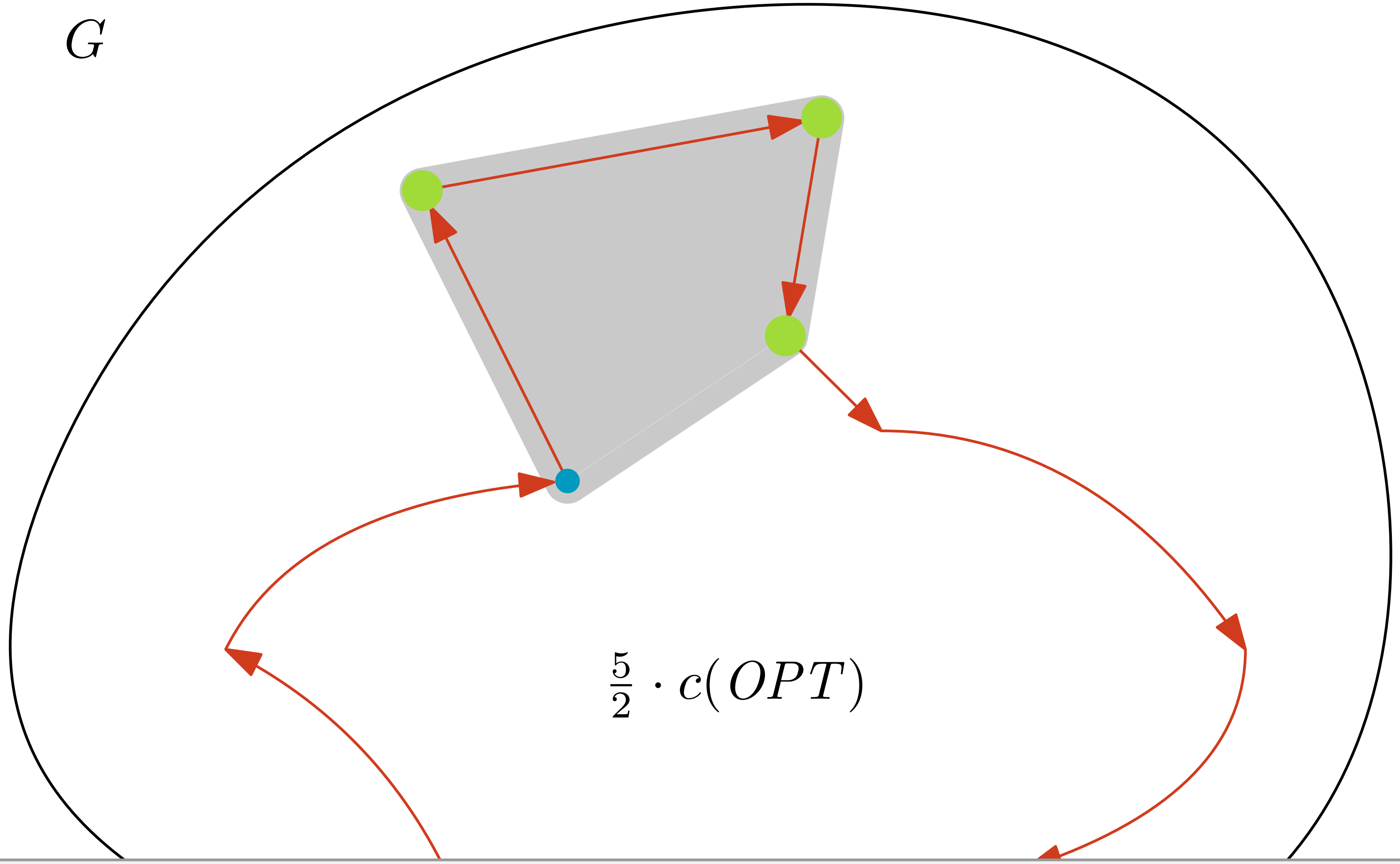
The generalized Christofides algorithm



The generalized Christofides algorithm



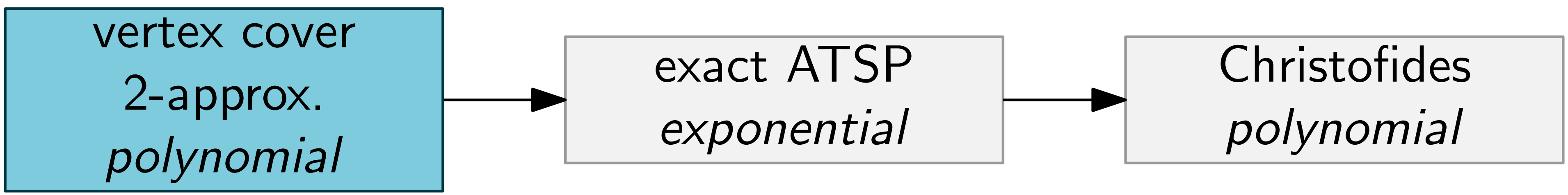
The generalized Christofides algorithm



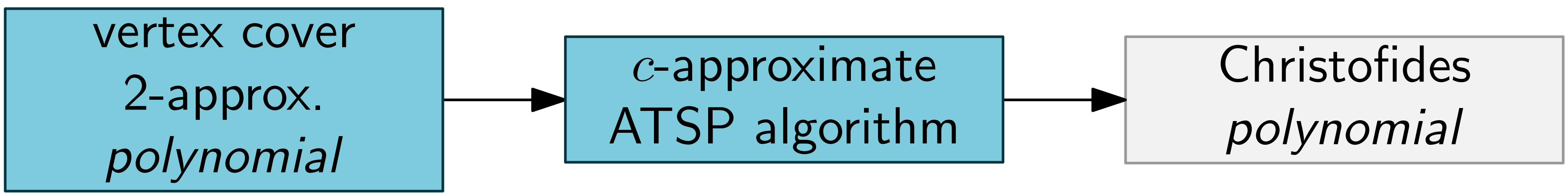
Theorem: The algorithm computes a 2.5-approx. for ATSP in $\mathcal{O}(2^k k^2 + n^3)$, where k is the size of a minimum vertex cover on the graph induced by the asymmetric links.

Going polynomial

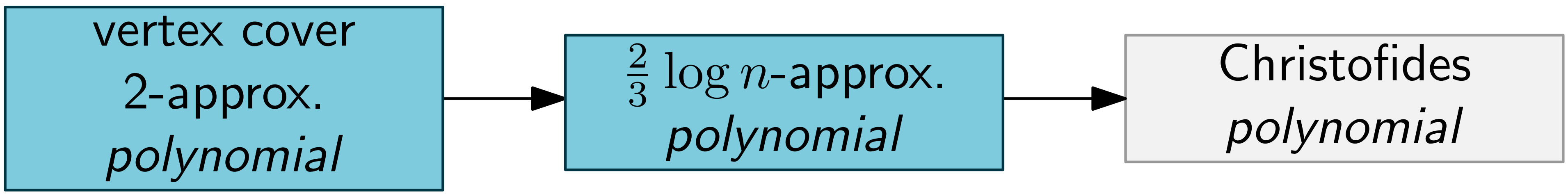
Going polynomial



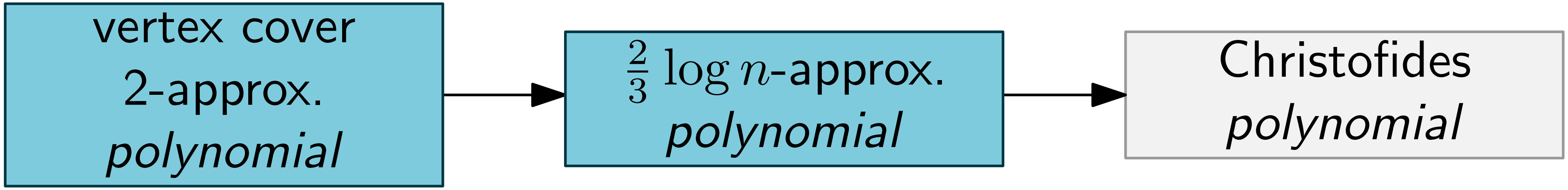
Going polynomial



Going polynomial

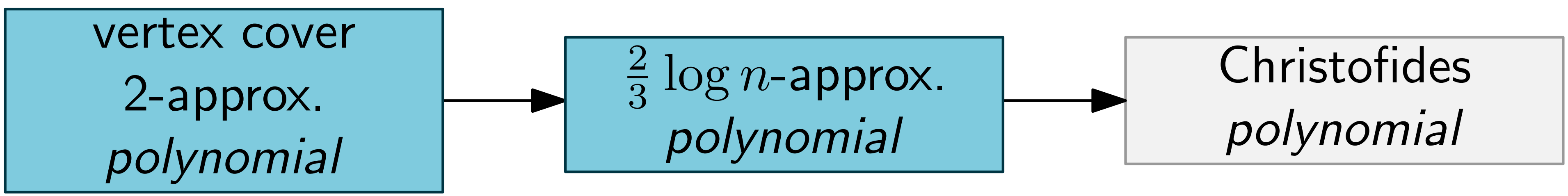


Going polynomial



Approximation factor $\frac{2}{3}\log n + 1.5$?

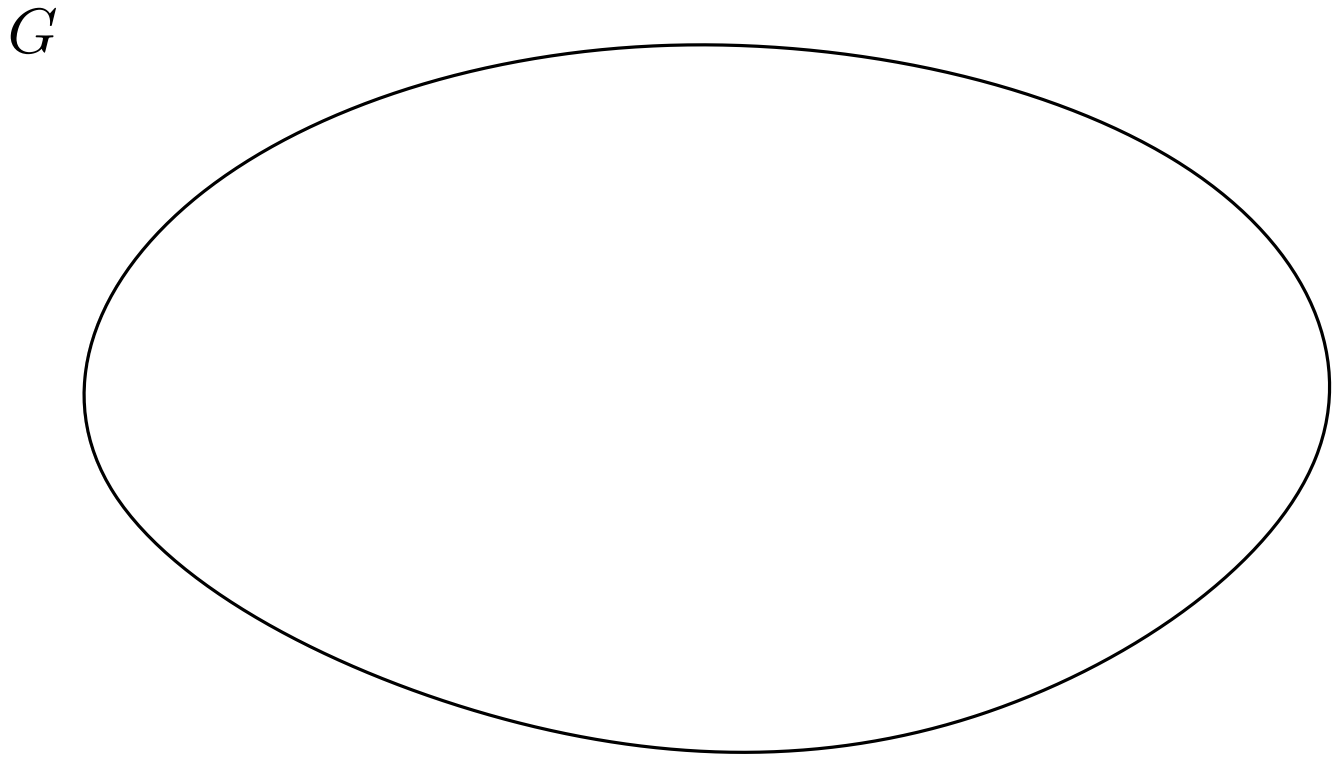
Going polynomial



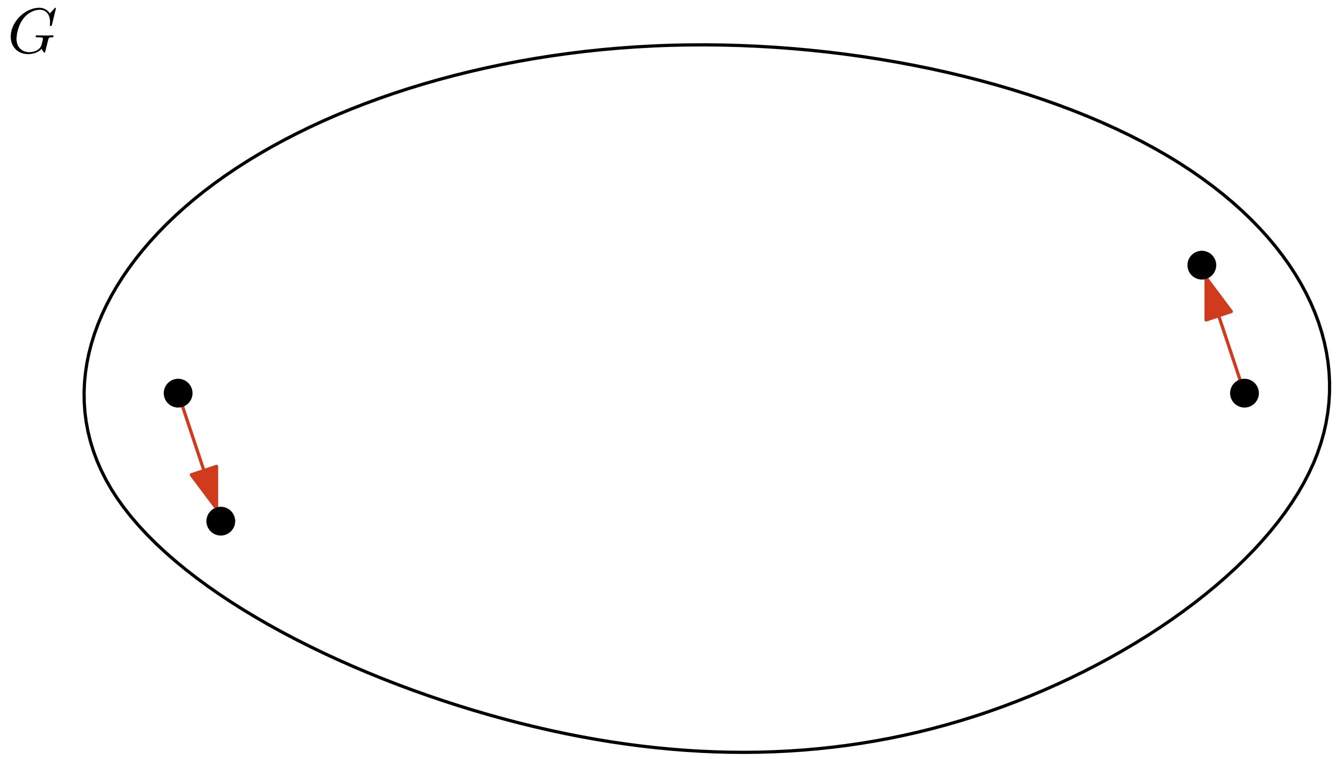
Approximation factor $\cancel{\frac{2}{3}\log n + 1.5}$?

$\frac{2}{3}\log k + 1.5$!

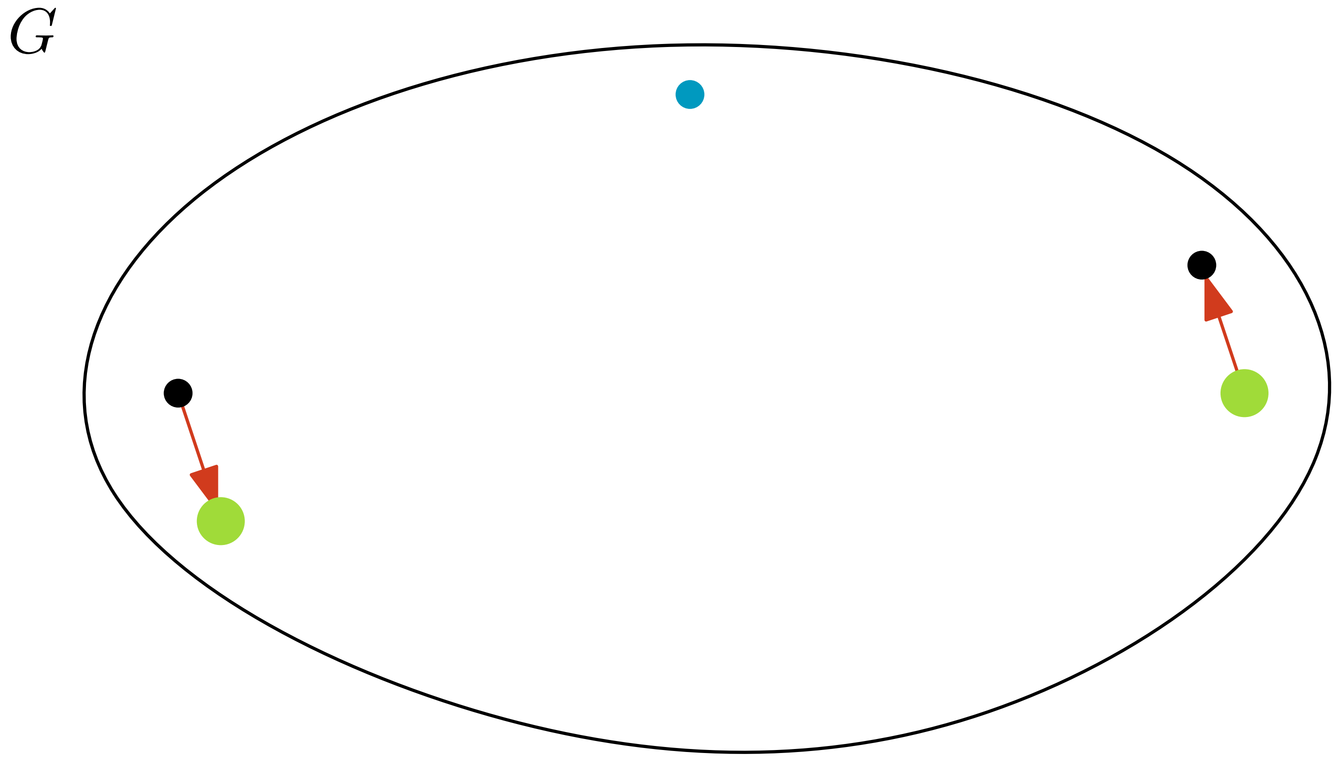
Unfavourable edge-case?



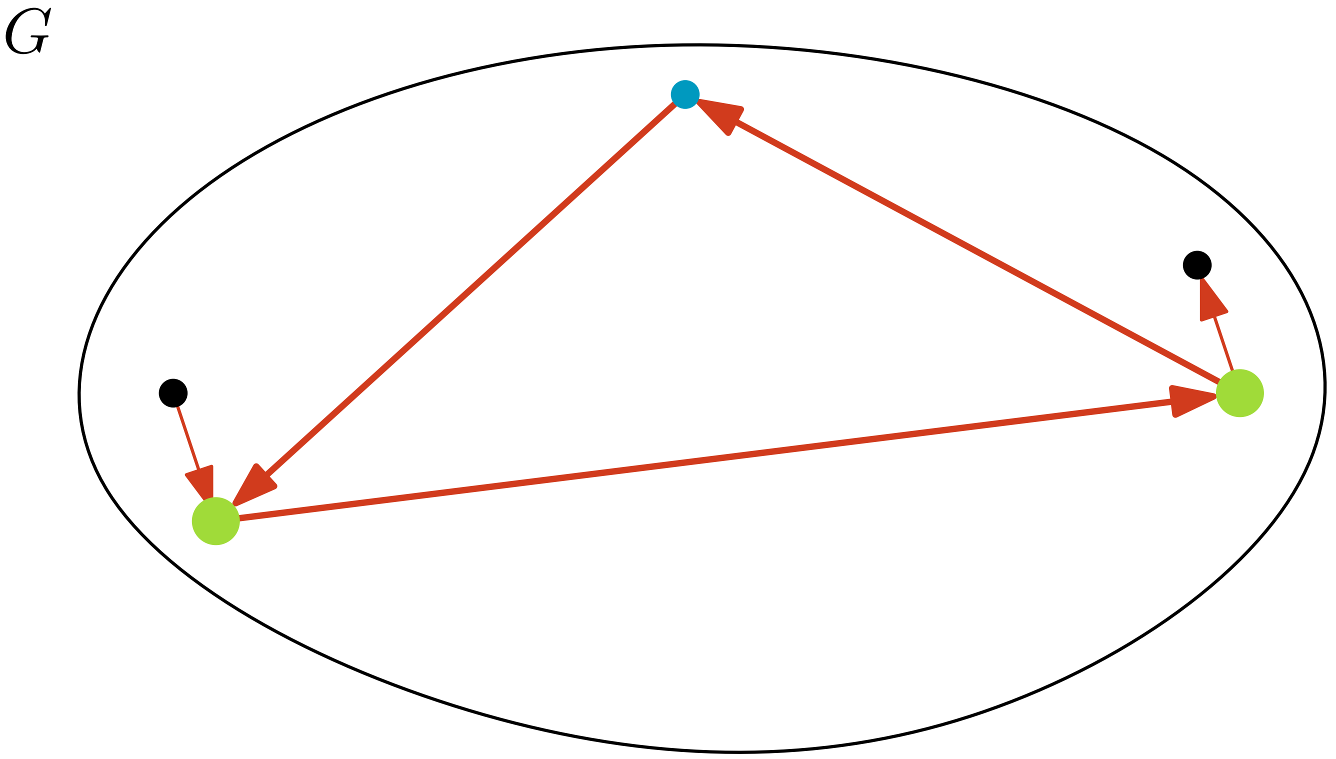
Unfavourable edge-case?



Unfavourable edge-case?



Unfavourable edge-case?



Unfavourable edge-case?