Fixed-parameter algorithms for minimum cost edge-connectivity augmentation

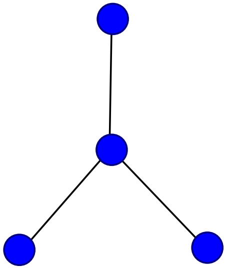
Dániel Marx[1] László A. Végh[2]

[1]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI) Budapest, Hungary

[2]Department of Management, London School of Economics, London, UK

40th International Colloquium on Automata, Languages and Programming (ICALP 2013)
Riga, Latvia
July 8, 2013

The edge-connectivity augmentation problem

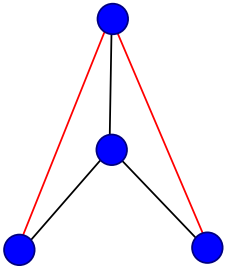


Problem

Input:
Graph $G = (V, E)$, connectivity target k, a cost function for each new edge that can be added to the graph.

Output:
Minimum cost set F of new edges so that $G + F$ is k-edge-connected.

The edge-connectivity augmentation problem



Problem

Input:
Graph $G = (V, E)$, connectivity target k, a cost function for each new edge that can be added to the graph.

Output:
Minimum cost set F of new edges so that $G + F$ is k-edge-connected.

The edge-connectivity augmentation problem

Variants

- Special case $k = 1$:
  Minimum cost spanning tree, polynomial-time solvable
  [Borůvka 1926].

The edge-connectivity augmentation problem

Variants

- Special case $k = 1$:
  Minimum cost spanning tree, polynomial-time solvable [Borůvka 1926].
- Uniform case:
  Adding an arbitrary new edge has unit cost. Polynomial-time solvable for arbitrary k [Watanabe, Nakamura 1987].

The edge-connectivity augmentation problem

Variants

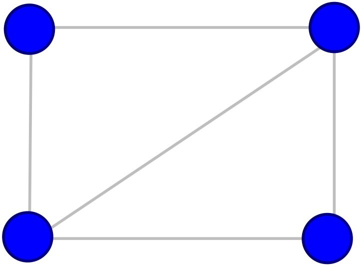
- Special case $k = 1$:
  Minimum cost spanning tree, polynomial-time solvable [Borůvka 1926].
- Uniform case:
  Adding an arbitrary new edge has unit cost. Polynomial-time solvable for arbitrary k [Watanabe, Nakamura 1987].
- Minimum cardinality case:
  Every cost is 1 or ∞. We wish to add a minimum number of new edges from a given set of links E^*.

The edge-connectivity augmentation problem

Variants

- Special case $k = 1$:
  Minimum cost spanning tree, polynomial-time solvable [Borůvka 1926].
- Uniform case:
  Adding an arbitrary new edge has unit cost. Polynomial-time solvable for arbitrary k [Watanabe, Nakamura 1987].
- Minimum cardinality case:
  Every cost is 1 or ∞. We wish to add a minimum number of new edges from a given set of links E^*.
  NP complete for $k \geq 2$.

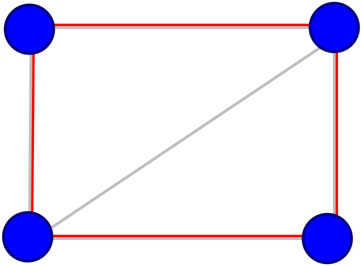
NP-completeness



Proposition

Minimum cardinality edge-connectivity augmentation is NP-complete already for $k = 2$.

NP-completeness



Proposition

Minimum cardinality edge-connectivity augmentation is NP-complete already for $k = 2$.

Proof.

For an arbitrary link graph (V, E^*) and starting graph $(V, \emptyset)$, there exists an augmentation with $|V|$ links in $E^* \Leftrightarrow (V, E^*)$ contains a Hamiltonian cycle. □

The edge-connectivity augmentation problem

Related results

- Polynomial algorithms for variants of the uniform case, e.g. [Watanabe, Nakamura 1987], [Frank 1992],...
- Approximation algorithms for the minimum cost variant e.g. [Agrawal, Klein, Ravi 1995], [Goemans, Williamson 1995], [Jain 2001],...

Important special case

Augmenting connectivity by one: we assume that the input graph is already $(k-1)$-edge-connected.

Fixed-parameter tractability

Main definition

A parameterized problem is **fixed-parameter tractable (FPT)** with some parameter k if there is an $f(k)n^c$ time algorithm for some constant c and function f depending only on k.

Main goal of parameterized complexity: to find FPT problems.

Fixed-parameter tractability

Main definition

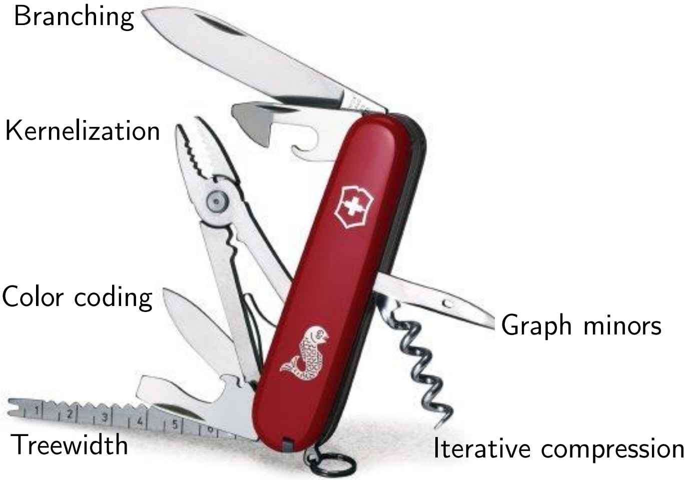
A parameterized problem is **fixed-parameter tractable (FPT)** with some parameter k if there is an $f(k)n^c$ time algorithm for some constant c and function f depending only on k.

Main goal of parameterized complexity: to find FPT problems.

Examples of NP-hard problems that are FPT:

- Finding a vertex cover of size k.
- Finding a path of length k.
- Finding k disjoint triangles.
- Drawing the graph in the plane with k edge crossings.
- Finding disjoint paths that connect k pairs of points.
- ...

Fixed-parameter tractability



Kernelization

A particularly nice way of proving fixed-parameter tractability:

Definition

A polynomial kernel is a polynomial-time reduction creating an equivalent instance whose size is polynomial in the parameter k.

Intuitively, a polynomial kernel means that the problem can be solved by preprocessing + brute force:

- Compute the equivalent instance whose size is polynomial in k.
- Use whatever method available to solve the kernel in time exponential in its size.

Fixed-parameter tractability of connectivity augmentation

What is the right parameter?

- k: connectivity target

  The problem is NP complete for any fixed $k \geq 2$.

- p: the maximum number of augmenting edges allowed.

  Trivial $n^{O(p)}$ algorithms, but fixed-parameter tractability is a challenging question!

Fixed-parameter tractability of connectivity augmentation

What is the right parameter?

- k: connectivity target

  The problem is NP complete for any fixed $k \geq 2$.
- p: the maximum number of augmenting edges allowed.

  Trivial $n^{O(p)}$ algorithms, but fixed-parameter tractability is a challenging question!

Previous results

- [Nagamochi 2003]: Minimum cardinality edge-connectivity augmentation from 1 to 2 is FPT with parameter p.
- [Guo, Uhlman 2010]: Minimum cardinality edge-connectivity augmentation from 1 to 2 has a kernel on $O(p^2)$ nodes, $O(p^2)$ edges; also for node-connectivity.

Main result

MINIMUM COST EDGE-CONNECTIVITY AUGMENTATION BY ONE

- **Input:**
  - $k \in \mathbb{Z}_+$: connectivity target
  - $(V, E \cup E^*)$, E: edges, E^*: links.
  - $G = (V, E)$ is $(k-1)$-edge connected.
  - $c : E^* \to \mathbb{R}_+$: cost
  - $p \in \mathbb{Z}_+$: maximum number of allowed links
- **Output:**
  Minimum cost $F \subseteq E^*$ s.t. $(V, E \cup F)$ is k-edge-connected and $|F| \leq p$.

Theorem

MINIMUM COST EDGE-CONNECTIVITY AUGMENTATION BY ONE admits a kernel of $O(p^4)$ nodes, $O(p^4)$ edges and $O(p^4)$ links, with all costs integers of $O(p^8 \log p)$ bits.

Overview

Key steps

- Formulate a slightly more general weighted problem.
- Observation: the problem can be formulated as covering every minimum cut with an edge.
- $k = 2, 3$:
    - Reduce to trees/cactus graphs via contractions.
    - Reduce to metric instances.
    - Kernelization for metric instances.
- $k \geq 4$: reduce to $k = 2$ or $k = 3$ via cactus representation of minimum cuts.
- Reduce cost sizes by simultaneous Diophantine approximation.

A more general problem

Weighted minimum cost edge-connectivity augmentation by one

- **Input:**
  - $k \in \mathbb{Z}_+$: connectivity target.
  - $(V, E \cup E^*)$, E: edges, E^*: links.
  - $G = (V, E)$ is $(k-1)$-edge connected.
  - $c : E^* \to \mathbb{R}_+$: cost, $w : E^* \to \mathbb{Z}_+$: weight.
  - $p \in \mathbb{Z}_+$: maximum total weight of allowed links.
- **Output:**
  Minimum cost $F \subseteq E^*$ s.t. $(V, E \cup F)$ is k-edge-connected and $w(F) \leq p$.

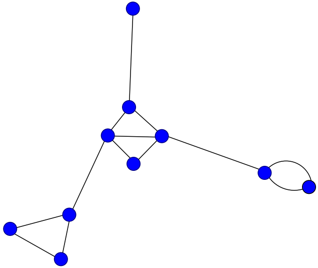
Theorem

Weighted Minimum Cost Edge-Connectivity Augmentation by One admits a kernel of $O(p)$ nodes, $O(p)$ edges and $O(p^3)$ links, with all costs integers of $O(p^6 \log p)$ bits.

$k = 2$ — Reduction step 1: Contraction

Proposition

We can contract all 2-connected blocks to obtain an equivalent instance.

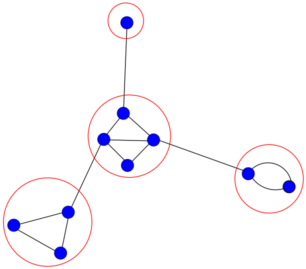


$\Rightarrow$

$k = 2$ — Reduction step 1: Contraction

Proposition

We can contract all 2-connected blocks to obtain an equivalent instance.

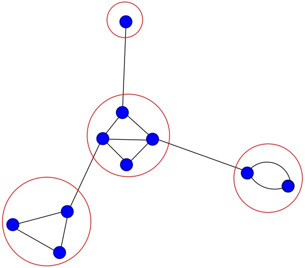


$\Rightarrow$

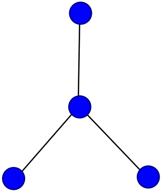
$k = 2$ — Reduction step 1: Contraction

Proposition

We can contract all 2-connected blocks to obtain an equivalent instance.



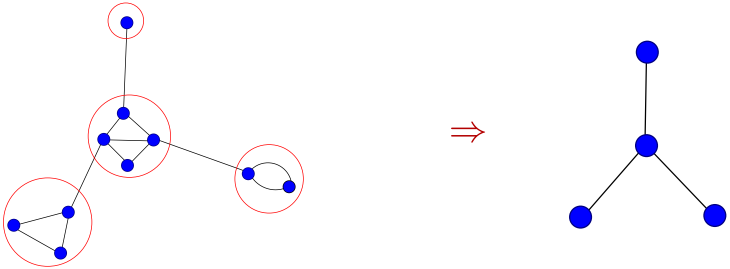
$\Rightarrow$



$k = 2$ — Reduction step 1: Contraction

Proposition

We can contract all 2-connected blocks to obtain an equivalent instance.

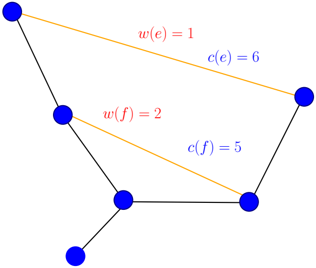


Proposition

We may assume that the input $G = (V, E)$ is a tree.

$k = 2$ — Metric instances

w: weight, c: cost.

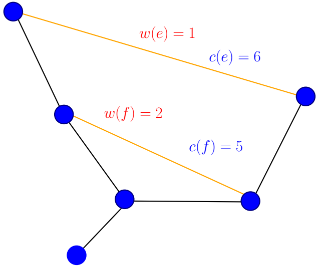


Definition

The link f is a shadow of e if the path in E between the endpoints of f is a subset of that for e, and $w(e) \leq w(f)$.

$k = 2$ — Metric instances

w: weight, c: cost.



Definition

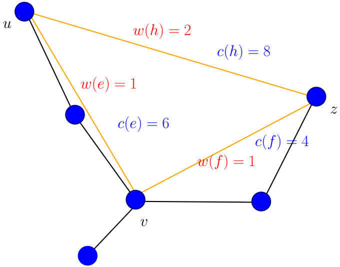
The link f is a shadow of e if the path in E between the endpoints of f is a subset of that for e, and $w(e) \leq w(f)$.

Intuition

Link e is better than f: it provides more connectivity (however, its cost might be larger)

$k = 2$ — Metric instances

w: weight, c: cost.



Definition

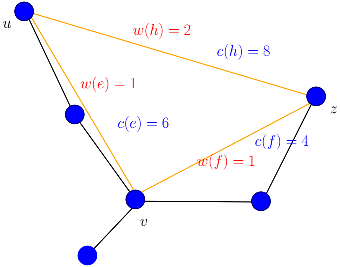
The instance is *metric*, if

(i) $c(f) \leq c(e)$ holds whenever the link f is a shadow of link e.

(ii) For $e = (u, v)$, $f = (v, z)$ and $h = (u, z)$ with $w(h) \geq w(e) + w(f)$, we must have $c(h) \leq c(e) + c(f)$.

$k = 2$ — Metric instances

w: weight, c: cost.



Definition

The instance is *metric*, if

(i) $c(f) \leq c(e)$ holds whenever the link f is a shadow of link e.

(ii) For $e = (u, v)$, $f = (v, z)$ and $h = (u, z)$ with $w(h) \geq w(e) + w(f)$, we must have $c(h) \leq c(e) + c(f)$.

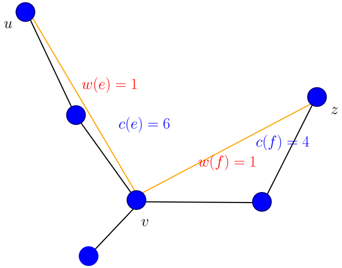
Intuition

(i) If $c(e) < c(f)$, then replacing f by e can only make the solution better.

(ii) If $c(e) + c(f) < c(h)$, then substituting h by e and f can only make the solution better.

$k = 2$ — Metric completion

w: weight, c: cost.



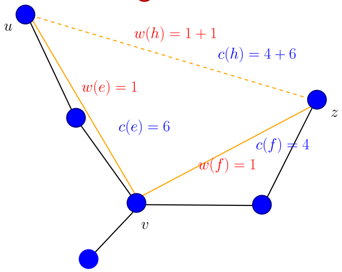
Lemma

Every instance can be replaced by an equivalent metric instance via a simple metric completion algorithm.

Remark: The metric completion is the reason for considering the weighted version of the problem.

$k = 2$ — Metric completion

w: weight, c: cost.



Lemma

Every instance can be replaced by an equivalent metric instance via a simple metric completion algorithm.

Remark: The metric completion is the reason for considering the weighted version of the problem.

$k = 2$ — Kernelization of metric instances

- Every leaf in G must have an incident link added.
- If the ($\#$ leaves $> 2p$), then the problem is infeasible.
- Otherwise, it follows that

$$(\#\text{leaves}) + (\#\text{ branching nodes}) \leq 4p - 2.$$

$k = 2$ — Kernelization of metric instances

- Every leaf in G must have an incident link added.
- If the (# leaves $> 2p$), then the problem is infeasible.
- Otherwise, it follows that

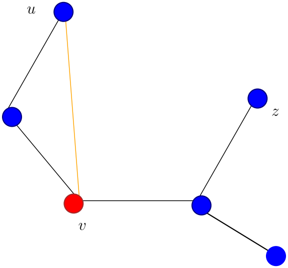
$$(\#\text{leaves}) + (\#\text{ branching nodes}) \leq 4p - 2.$$

- Key lemma: For every metric instance, there exists an optimal solution with every link incident to leaves and branching nodes only.
- We obtain a kernel on $\leq 4p - 2$ nodes by replacing every path of degree 2 nodes by a single edge.

$k = 2$ — Kernelization of metric instances

Key lemma

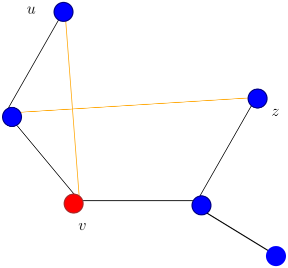
For every metric instance, there exists an optimal solution with every link incident to leaves and branching nodes only.



$k = 2$ — Kernelization of metric instances

Key lemma

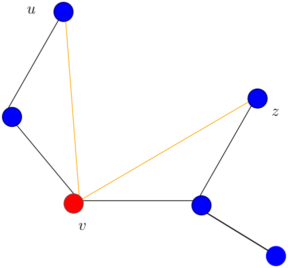
For every metric instance, there exists an optimal solution with every link incident to leaves and branching nodes only.



$k = 2$ — Kernelization of metric instances

Key lemma

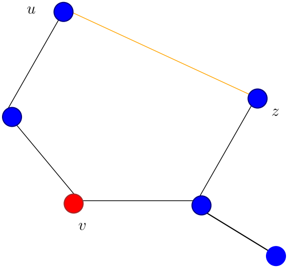
For every metric instance, there exists an optimal solution with every link incident to leaves and branching nodes only.



$k = 2$ — Kernelization of metric instances

Key lemma

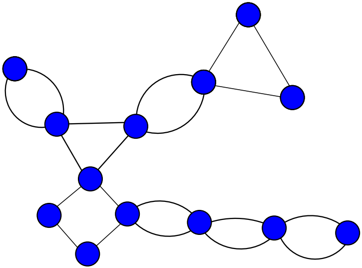
For every metric instance, there exists an optimal solution with every link incident to leaves and branching nodes only.



$k > 3$ — Reduction to $k \in \{2, 3\}$

Cactus graph: every 2-connected block is a cycle.

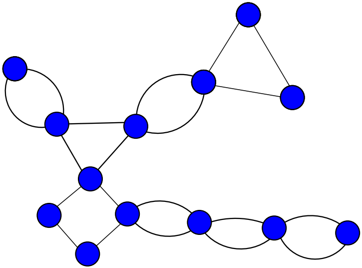
$k = 3$ is similar to $k = 2$, but on a cactus graph instead of a tree.



$k > 3$ — Reduction to $k \in \{2, 3\}$

Cactus graph: every 2-connected block is a cycle.

$k = 3$ is similar to $k = 2$, but on a cactus graph instead of a tree.



$k > 3$ can be reduced to $k = 3$:

Theorem [Dinitz, Karzanov, Lomonosov 1976]

For every graph $G = (V, E)$, there exists a mapping of $\varphi : V \to U$ to the node set of a cactus $H = (U, L)$ s.t. there is a 1-1 correspondence between the minimum cuts.

Reducing the size of the cost

Technical issue about kernels for minimum cost problems:

- The cost c in the input can consist of arbitrary real numbers, thus the kernel consists of a graph with $O(p^4)$ edges and the $O(p^4)$ real numbers for the $O(p^4)$ links.
- The kernel should contain numbers of bounded bitsize only.

Reducing the size of the cost

Technical issue about kernels for minimum cost problems:

- The cost c in the input can consist of arbitrary real numbers, thus the kernel consists of a graph with $O(p^4)$ edges and the $O(p^4)$ real numbers for the $O(p^4)$ links.
- The kernel should contain numbers of bounded bitsize only.
- We can use [Frank, Tardos, 1987] on simultaneous Diophantine approximation to replace the costs by integers of $O(p^6 \log p)$ bits.
  - We want that a solution is optimum with the new costs iff it is optimum with the original cost.
  - What we need is that the cost of any two sets of at most p edges have the same relation in the original and new costs.
- This technique should be essential for other kernelization problems involving costs!

Further results and open questions

- Node-connectivity: We prove that Weighted Minimum Cost Node-Connectivity Augmentation from 1 to 2 admits a kernel.
- Node-connectivity in any other setting: OPEN.

Further results and open questions

- Node-connectivity: We prove that WEIGHTED MINIMUM COST NODE-CONNECTIVITY AUGMENTATION FROM 1 TO 2 admits a kernel.
- Node-connectivity in any other setting: OPEN.
- Augmenting arbitrary input graph to 2-edge-connectivity: We give an FPT algorithm that has a branching step, but existence of a polynomial kernel is OPEN.
- Augmenting arbitrary input graph to k-edge-connectivity: OPEN.

Further results and open questions

- Node-connectivity: We prove that Weighted Minimum Cost Node-Connectivity Augmentation from 1 to 2 admits a kernel.
- Node-connectivity in any other setting: OPEN.
- Augmenting arbitrary input graph to 2-edge-connectivity: We give an FPT algorithm that has a branching step, but existence of a polynomial kernel is OPEN.
- Augmenting arbitrary input graph to k-edge-connectivity: OPEN.
- Directed graph, hypergraphs, nonuniform connectivity requirements: a whole world of connectivity-augmentation problems mostly unexplored from the viewpoint of fixed-parameter tractability!