New Approximation Algorithms for $(1,2)$-TSP

Anna Adamaszek and Matthias Mnich and

Katarzyna Paluch

University of Wrocław

$(1, 2)$-TSP

Input: A complete undirected graph G with weights one and two on the edges.

Task: Compute a traveling salesman tour of G of minimum weight.

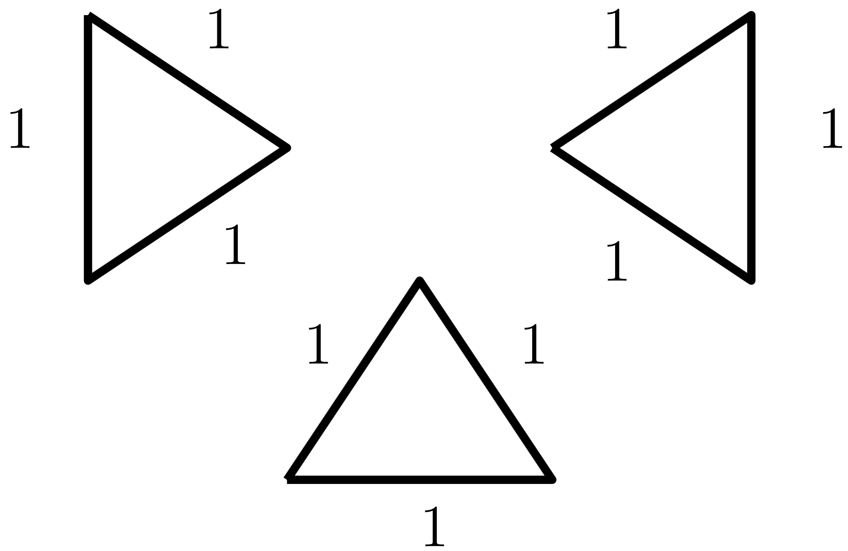
Observation: Every instance of $(1, 2)$-TSP satisfies the triangle inequality.

Hardness of $(1, 2)$-TSP

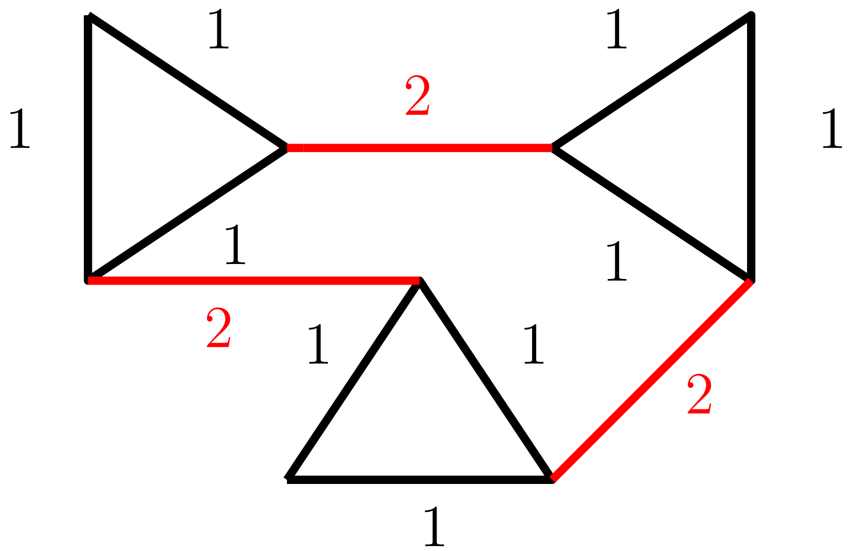
- It is one of Karp's 21 NP-complete problems.
- Proved to be APX-hard [Papadimitriou and Yannakakis 1993].
- Best known inapproximability bound for $(1, 2)$-TSP is $\frac{535}{534}$ [Karpinski and Schmied 2012]

Starting point - a cycle cover

- Compute a cycle cover C_{min} of G of minimum weight, where
  a cycle cover of G – a collection of cycles such that each vertex of G belongs to exactly one cycle in the collection.
- $w(C_{min}) \leq OPT$
- Remove the heaviest edge from each cycle c of C_{min}.
- Patch the obtained paths in an arbitrary way so that they form a traveling salesman tour.
- From a cycle c of length k we obtain a path of weight at most $\frac{k+1}{k}w(c)$. (In the worst case a 1-edge is replaced with a 2-edge.)
- Therefore we have a $4/3$-approximation.



$$w(C_{min}) = 3 \cdot 3 = 9$$



$$w(C_{min}) = 3 \cdot 3 = 9$$

$$w(Sol) = 3 \cdot 4 = 12$$

Hartvigsen's algorithm

Computing a minimum weight cycle cover C_{min} of a graph is easy - by reducing to matchings.

[Hartvigsen] There is an $O(n^3)$ algorithm that, given a complete graph G with edge weights 1 and 2, computes a triangle-free cycle cover of G with minimum weight.

Approximations algorithms

- $\frac{9}{7}$ not using Hartvigsen's algorithm [Papadimitriou, Yannakakis 1993] $O(n^3)$
- $\frac{7}{6}$ using Hartvigsen's algorithm [Papadimitriou, Yannakakis 1993] $O(n^3)$
- $\frac{65}{56}$ using Hartvigsen's algorithm [Bläser, Ram 2005] $O(n^3)$
- $\frac{8}{7}$ local search, not using Hartvigsen's algorithm[Berman, Karpinski 2006] $O(n^9)$

Our results:

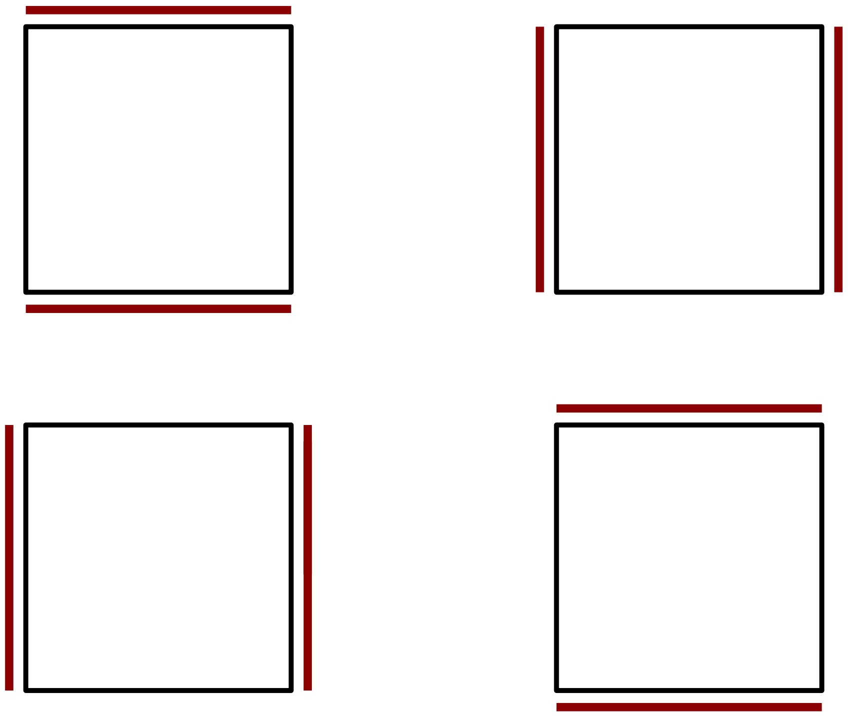
- $\frac{7}{6}$ not using Hartvigsen's algorithm $O(n^{2.5})$
- $\frac{8}{7}$ using Hartvigsen's algorithm $O(n^3)$

Goal

The goal is to maximize the average length of a path consisting of 1-edges.

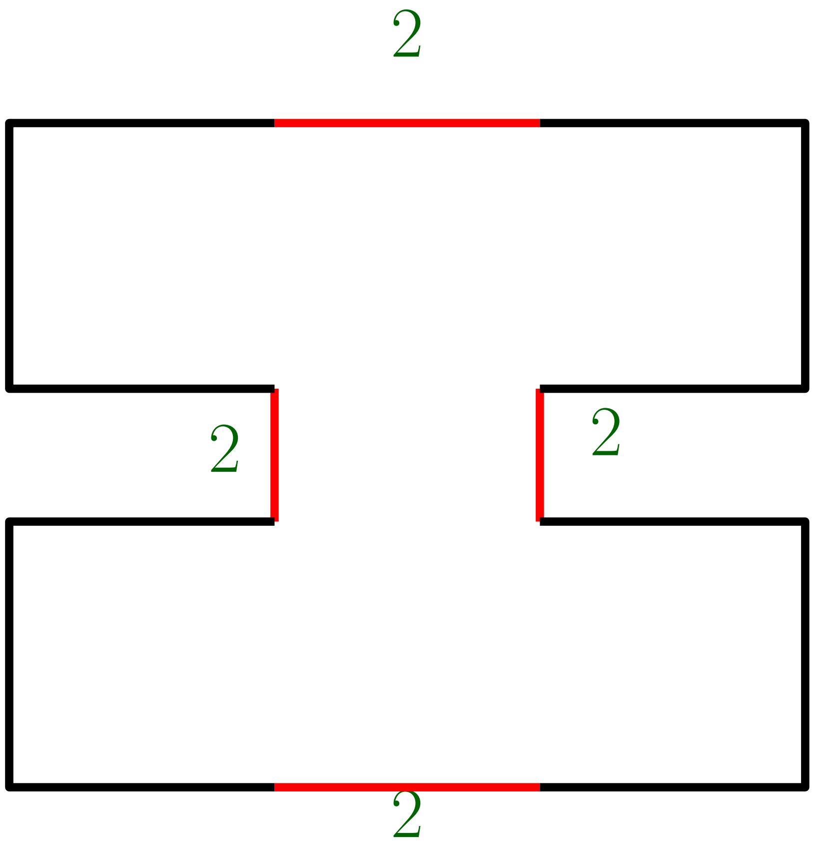
M_{min}- a perfect matching of minimum weight

- A minimum weight perfect matching M_{min} satisfies $w(M_{min}) \leq OPT/2$ (assuming the graph has an even number of vertices).
- We can use M_{min} to connect cycles of C_{min} and form longer paths consisting of 1-edges.
- It works only if each short cycle c of C_{min} has an incident 1-edge of M_{min} connecting it with a different cycle of C_{min}.



$$w(C_{min}) = 4 \cdot 4 = 16$$

$$w(M_{min}) = 8$$



$$w(C_{min}) = 4 \cdot 4 = 16$$

$$w(M_{min}) = 8$$

$$w(Sol)/w(C_{min}) = \frac{20}{16} = \frac{5}{4}$$

A good matching

We say that a matching M is *good* if it connects each square (and hexagon) c of C_{min} to somewhere outside of c.

The weight of a minimum weight perfect good matching is a lower bound on OPT.

Computational hardness of useful matchings

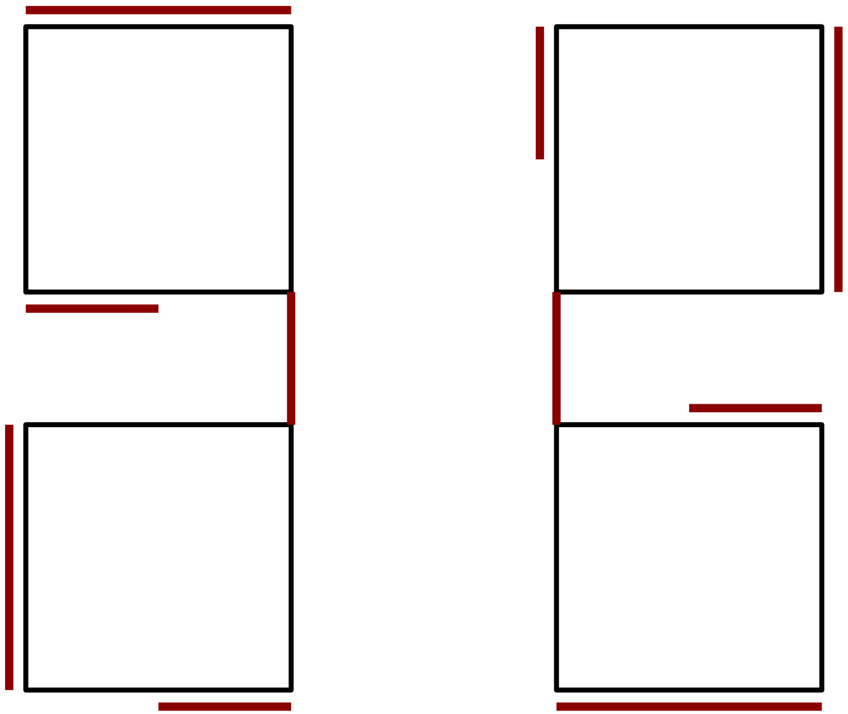
Computing a minimum weight perfect useful matching is NP-hard.

A matching that allows half-edges

.

A *half-edge* of the edge e is, informally speaking, a half of the edge e that contains exactly one of the endpoints of e.

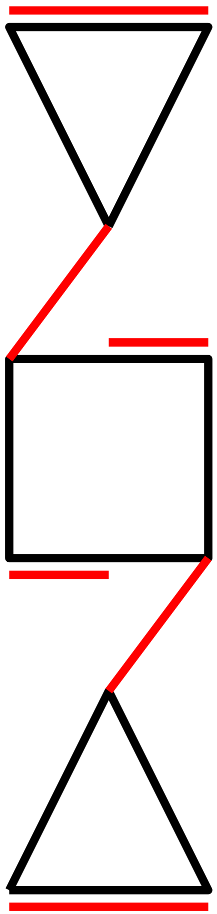
Theorem 1 *A minimum weight perfect matching with half-edges $M^{\frac{1}{2}}$ that connects each square (and hexagon) c of C_{min} to some vertex not on c can be computed in polynomial time.*



$$w(C_{min}) = 4 \cdot 4 = 16$$

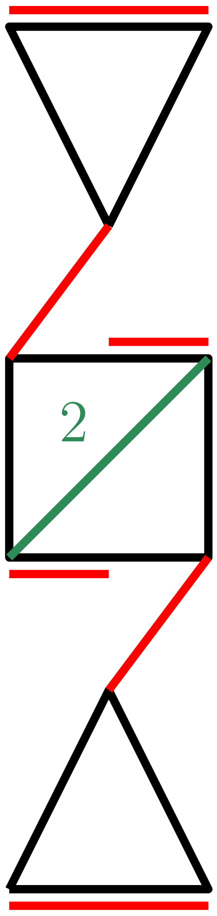
$$w(M^{\frac{1}{2}}) = 6 \cdot 1 + 4 \cdot \frac{1}{2} = 8$$

Bad configurations of half-edges



$w(C_{min}) = 10$

$w(M^{\frac{1}{2}}) = 5$



$$w(C_{min}) = 10$$

$$w(M^{\frac{1}{2}}) = 5$$

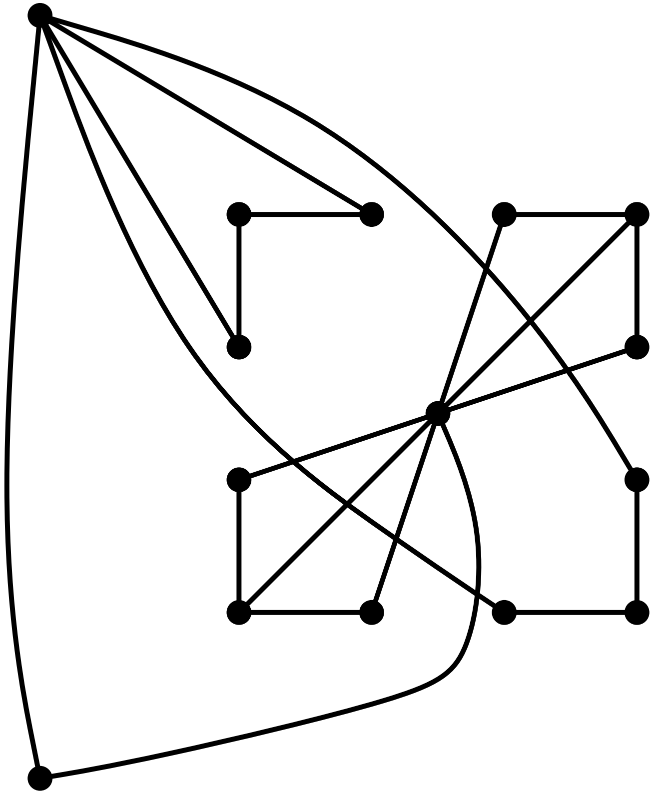
$$w(Sol)/w(C_{min}) = \frac{12}{10} = \frac{6}{5}$$

A good matching with half-edges

To compute a minimum weight good matching with half-edges we use (a, b)-matchings and gadgets.

Given two functions $a, b : V \to N$, an (a, b)-matching is any set $M \subseteq E$ such that $a(v) \leq deg_M(v) \leq b(v)$.

Gadgets



A $7/6$-approximation algorithm

- Compute a minimum weight cycle cover C_{min} of G.
- Find a minimum cost matching with half-edges (and some additional properties) $M^{\frac{1}{2}}$.
- Based on and $M^{\frac{1}{2}}$, construct a multigraph G^1 on vertex set $V(G)$ with at least $\frac{5}{2}\alpha_{\text{opt}} - \beta_{\text{opt}}$ edges of weight 1 from G.
- Path-3-color the edges of G^1. (Color the edges of G^1 with three colors so that each color class consists of vertex-disjoint paths.)
- Extend the set of edges of G^1 from the largest color class arbitrarily to a tour $\mathcal{T}$ of G.

An $8/7$-approximation algorithm

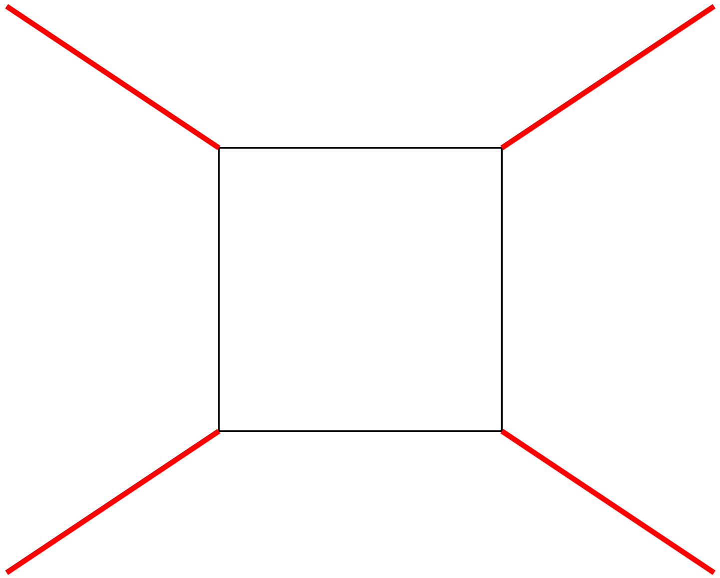
- Using Hartvigsen's algorithm compute a minimum weight triangle-free cycle cover C_{min} of G.
- Find a minimum cost matching with half-edges (and some additional properties) $M^{\frac{1}{2}}$.
- Based on and $M^{\frac{1}{2}}$, construct a multigraph G^1 on vertex set $V(G)$ with at least $\frac{7}{2}\alpha_{\text{opt}} - \beta_{\text{opt}}$ edges of weight 1 from G.
- Path-4-color the edges of G^1. (Color the edges of G^1 with four colors so that each color class consists of vertex-disjoint paths.)
- Extend the set of edges of G^1 from the largest color class arbitrarily to a tour $\mathcal{T}$ of G.

Method of path-3-coloring G^1

We color the multigraph G^1 cycle-wise - by considering each cycle c of C_{min} in turn and coloring all edges incident to c.

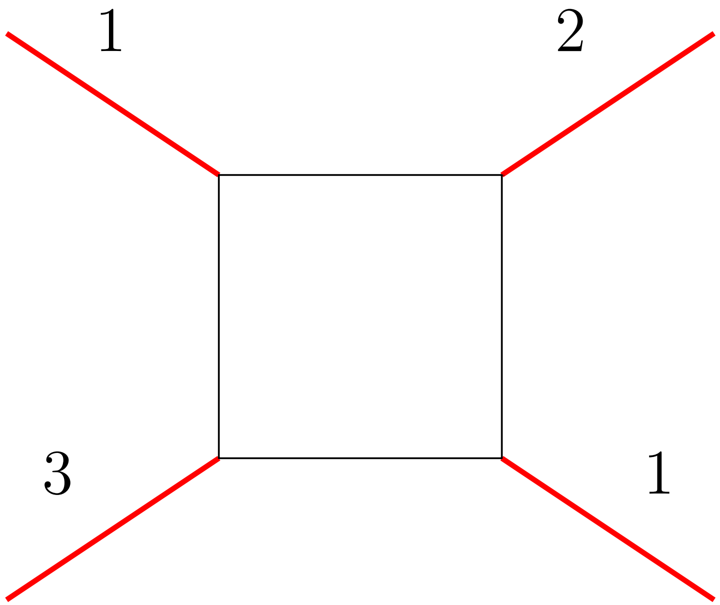
An edge $e = (u, v)$ of G_1 is *safe* if no matter how we color the so far uncolored edges of G_1 edge e is guaranteed not to belong to any monochromatic cycle.

Path-3-coloring



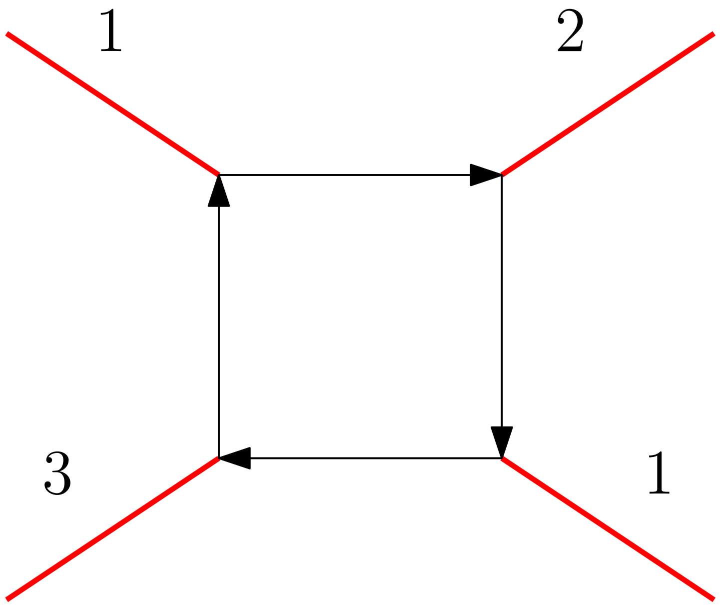
A black square belongs to C_{min}, red edges to $M^{\frac{1}{2}}$.

Path-3-coloring



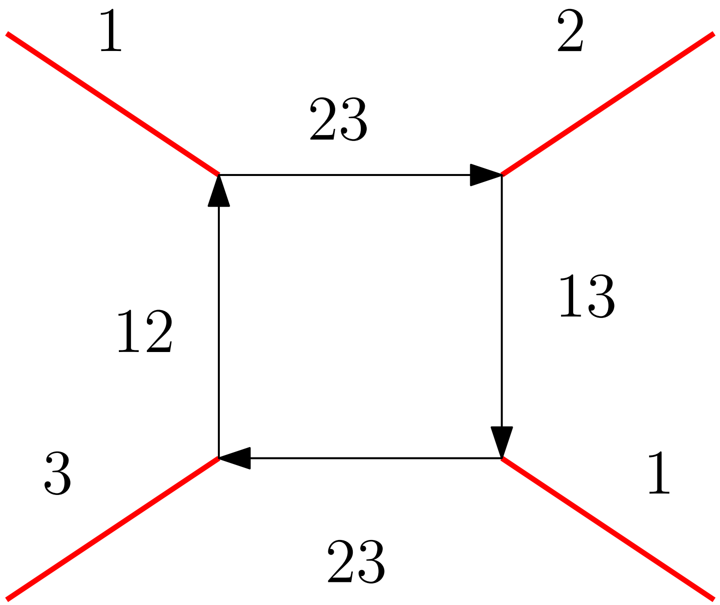
We color the edges of $M^{\frac{1}{2}}$.

Path-3-coloring



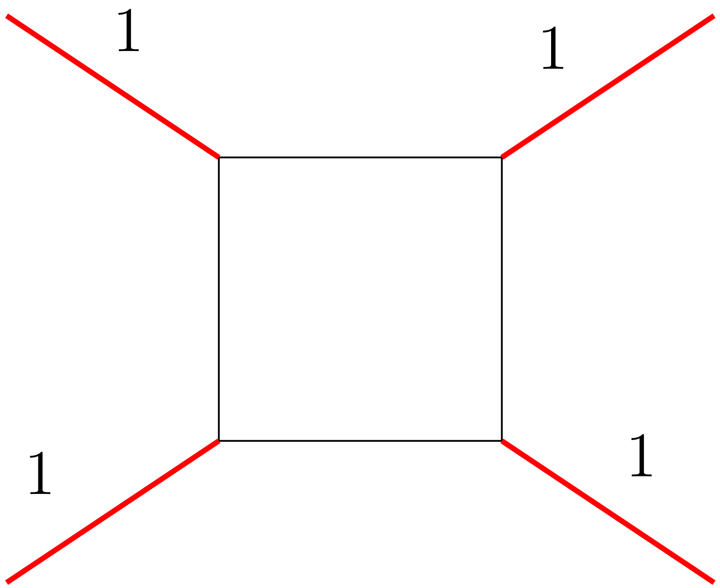
We direct the square - only for the purpose of coloring.

Path-3-coloring



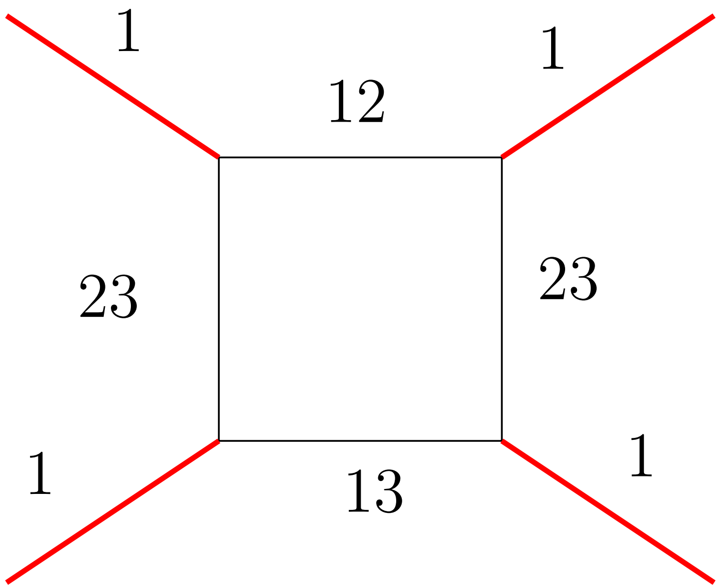
Each colored edge is safe.

Path-3-coloring cd



Each edge already colored is safe.

Path-3-coloring cd



Each colored edge is safe.