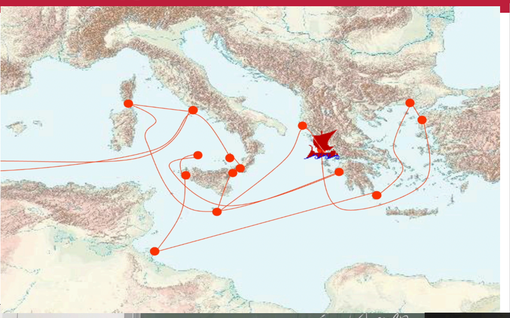
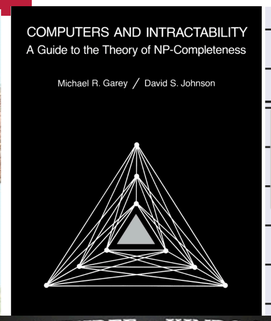
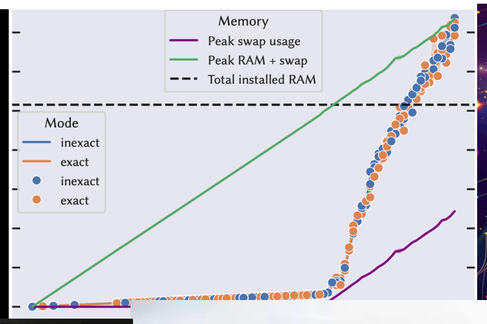
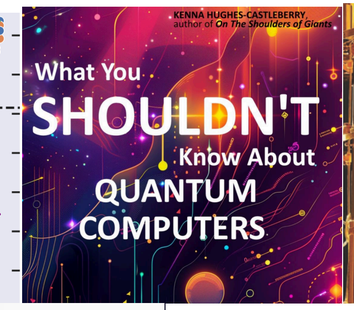
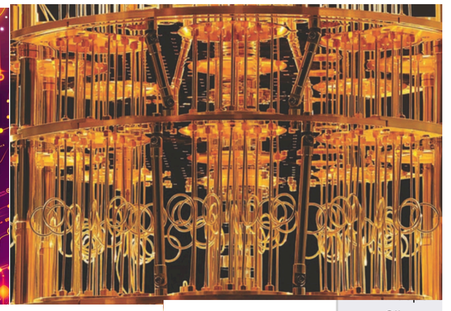
Technische Universität Braunschweig

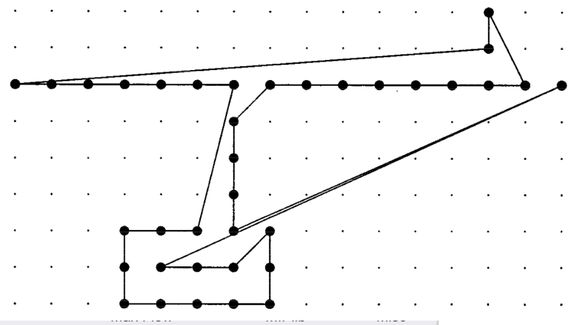


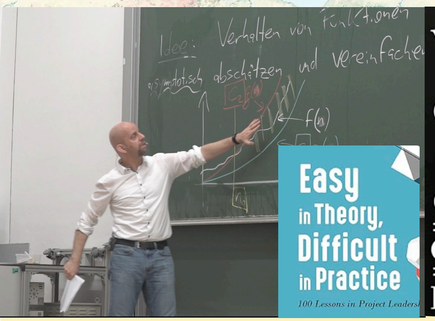




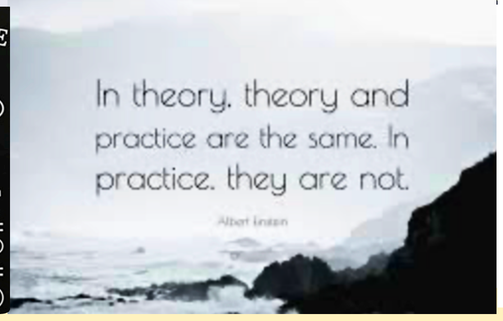


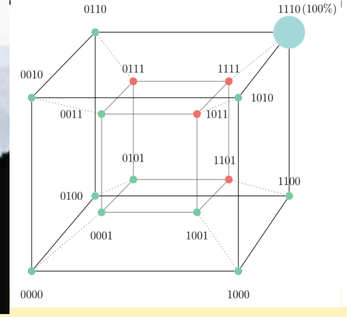


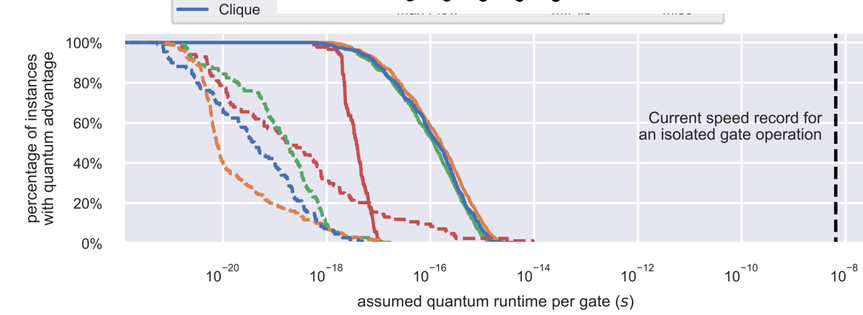












Hard in Theory, Easy in Practice?
Solving Hard Optimization Problems in Geometry and Elsewhere

Sándor P. Fekete
Department of Computer Science
TU Braunschweig

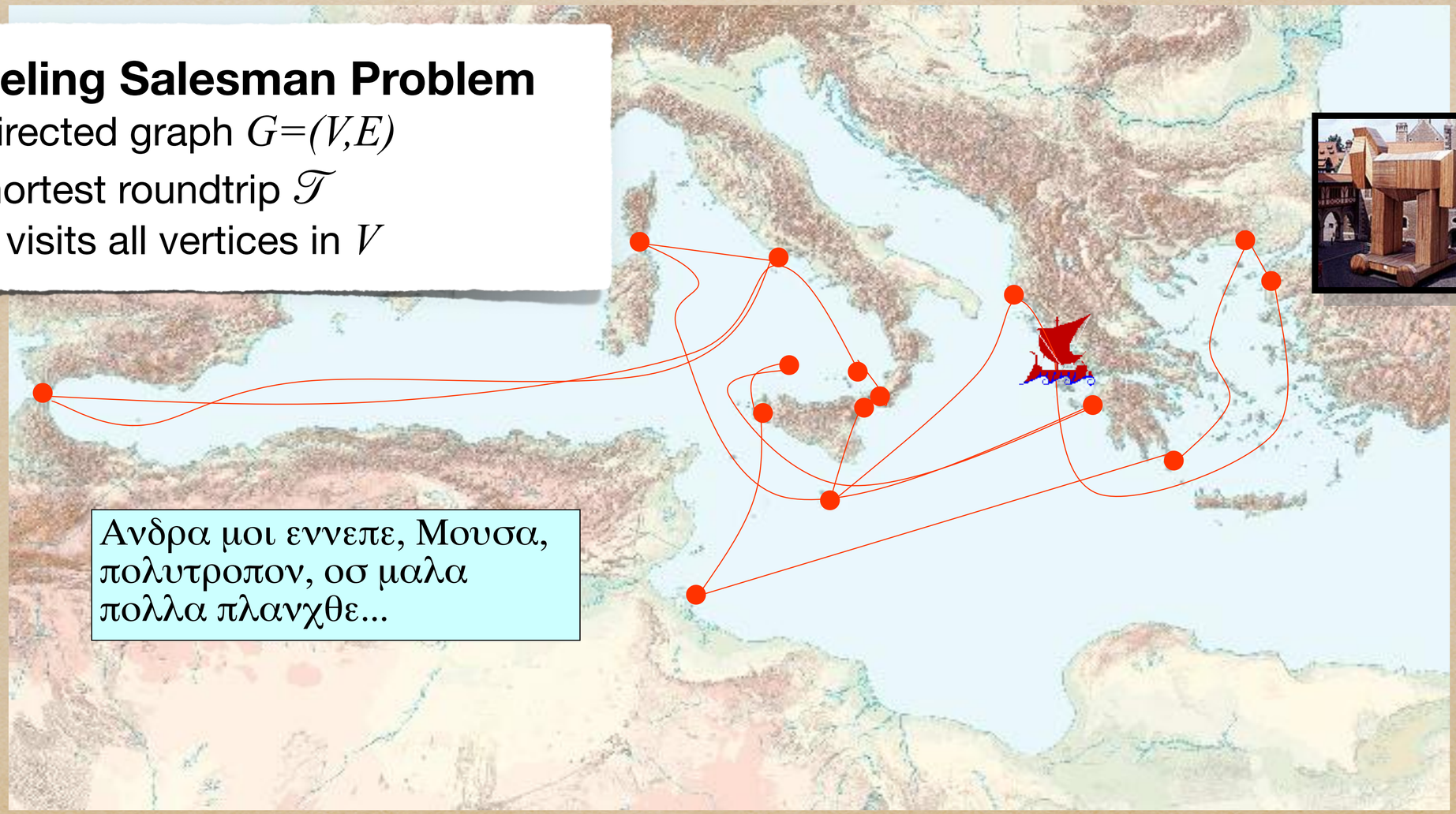
1 Some Ancient and Recent History

3000 Years Ago

The Traveling Salesman Problem

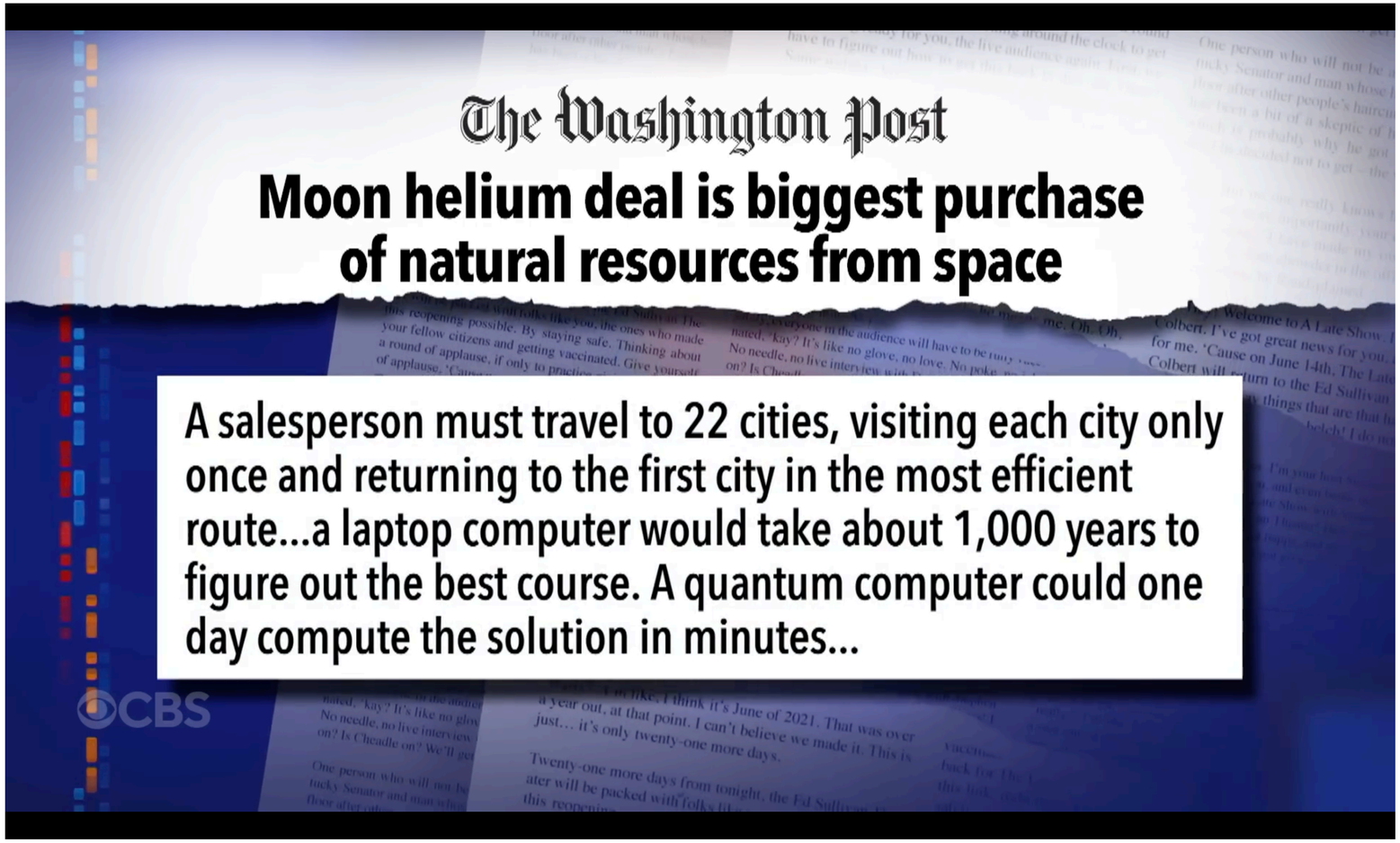
Input: A directed graph $G=(V,E)$

Output: A shortest roundtrip $\mathcal{T}$ that visits all vertices in V




MOOOOON NEWWWWS

Master Plan



The Washington Post

Mostly sunny 37/27 • Tomorrow: Cloudy 47/33 | Democracy Dies in Darkness | MONDAY, FEBRUARY 5, 2018

"It would take a laptop computer 1,000 years to compute the most efficient route between 22 cities, for example. A quantum computer could do this within minutes, possibly seconds."

February 5, 2018

(1) Collect $300M
(2) Fly to the moon
(3) Harvest Helium-3
(4) Bring back to earth
(5) Build quantum computer
(6) Compute 22-city tour

Master Plan



Bill Cook @wjcook · Feb 12

To clarify, computing an optimal TSP route on 22 points takes 0.005 seconds on my iMac, not the 1000 years reported in @washingtonpost An error factor of 6 trillion. Like reporting the US National Debt is $4.

washingtonpost.com/news/innovatio…

The Washington Post

Mostly sunny 37/27 • Tomorrow: Cloudy 47/33 | Democracy Dies in Darkness | MONDAY, FEBRUARY 5, 2018 · $2

"It would take a laptop computer 1,000 years to compute the most efficient route between 22 cities, for example. A quantum computer could do this within minutes, possibly seconds."

February 5, 2018

An official website of the United States government Here's how you know

NIST

Search NIST | Menu

Information Technology

CYBERSECURITY AND PRIVACY

Frameworks

National Cybersecurity Center of Excellence (NCCoE)

NICE

What Is Post-Quantum Cryptography?

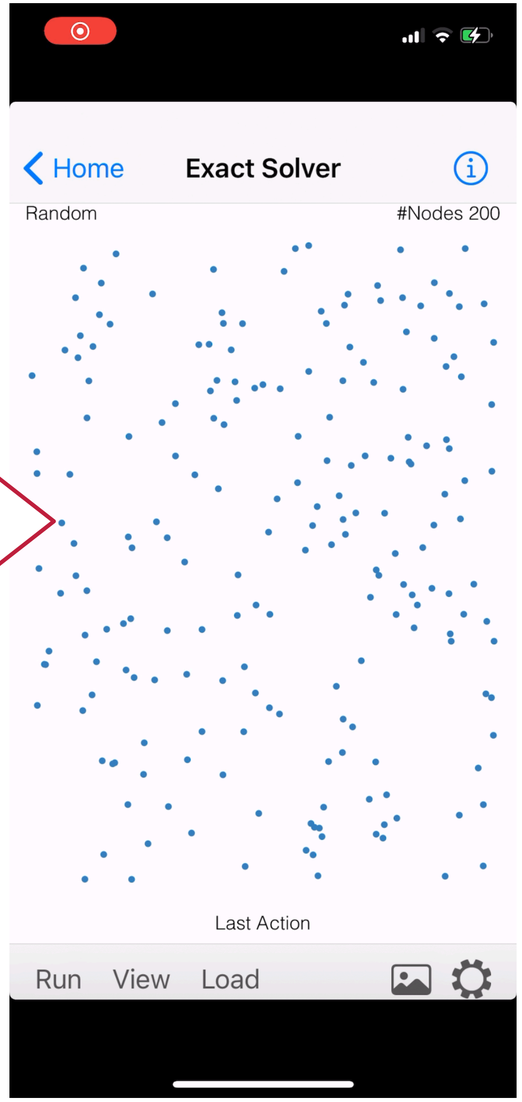
Why are quantum computers being developed if they can potentially cause so much harm?

There are many helpful things quantum computers will likely be able to do. Quantum computers have the potential to accomplish tasks that involve the interplay of complex variables. These tasks include drug design, simulations of complex molecules, and solutions to the classic "traveling salesman" problem — finding the most efficient route through a number of destinations.

„Hard“ and „Hard“ May Differ

$200! \approx 10^{370}$

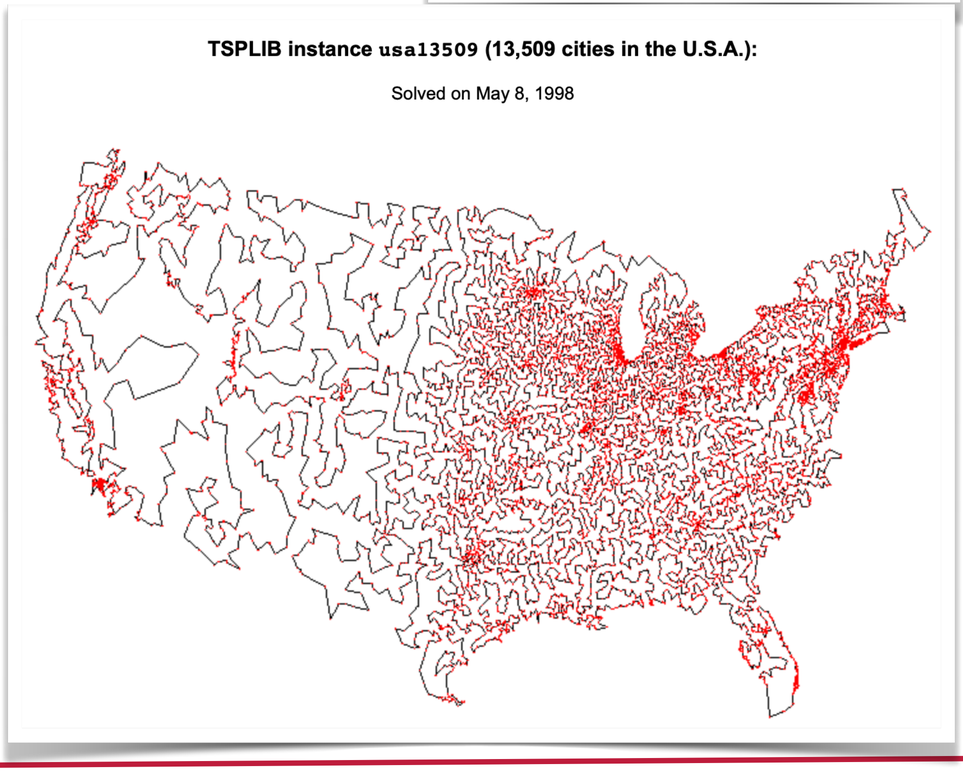
Traveling Salesman:
Shortest perimeter
for 200 points



TSPLIB

TSPLIB is a library of sample instances for the TSP (and related problems) from various sources and of various types.

ulysses16	16	GEO
ulysses22	22	GEO
usa13509	13509	EUC_2D

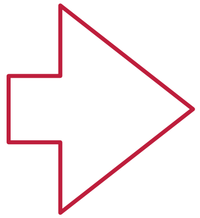


$13509! \approx 10^{50.000}$

„Hard“ and „Hard“ May Differ

$30! \approx 10^{32}$

Area minimization:
Smallest area
for 30 points??



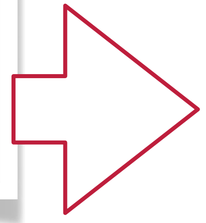
Area-Optimal Simple Polygonalizations: The CG Challenge 2019

Authors: Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, Joseph S. B. Mitchell Authors Info & Claims

ACM Journal of Experimental Algorithmics, Volume 27 • December 2022 • Article No.: 2.4, pp 1–12 • https://doi.org/10.1145/3504000

$200! \approx 10^{370}$

Traveling Salesman:
Shortest perimeter
for 200 points



**Hard in Theory, Easy in Practice:
Solving Large Minimum-Weight Triangulation Instances to Provable Optimality**

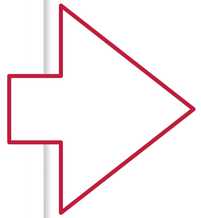
SÁNDOR P. FEKETE, ANDREAS HAAS, and PHILLIP KELDENICH, TU Braunschweig, Department of Computer Science, Germany

We consider practical solution of a well-known geometric optimization that combines the challenges of theoretical complexity with algebraic difficulties: For a given set of points in the two-dimensional plane, the NP-hard *Minimum-Weight Triangulation* problem asks for a maximal set of non-crossing edges of minimum total Euclidean distance.

Based on a comprehensive set of optimization methods and implementation techniques, we demonstrate that Minimum-Weight Triangulation is a practically tractable problem, making it possible to compute provably optimal solutions for benchmark instances of sizes up to 100,000,000 points (including for *all* benchmark sets from other publicly available sources that we are aware of) in reasonable time on commodity hardware. In the process, we also show how to deal with a variety of other important aspects that can impede practical solvability of large geometric optimization problems, such as exactness of Euclidean distances, numerical issues, and memory management. We also address the existence of practically difficult instances, as well as the quality of minimum-weight triangulations as spanners.

$10^{42.000.000}$

Minimum Triangulation:
Cheapest Subdivision
into triangles
for 100.000.000 points!

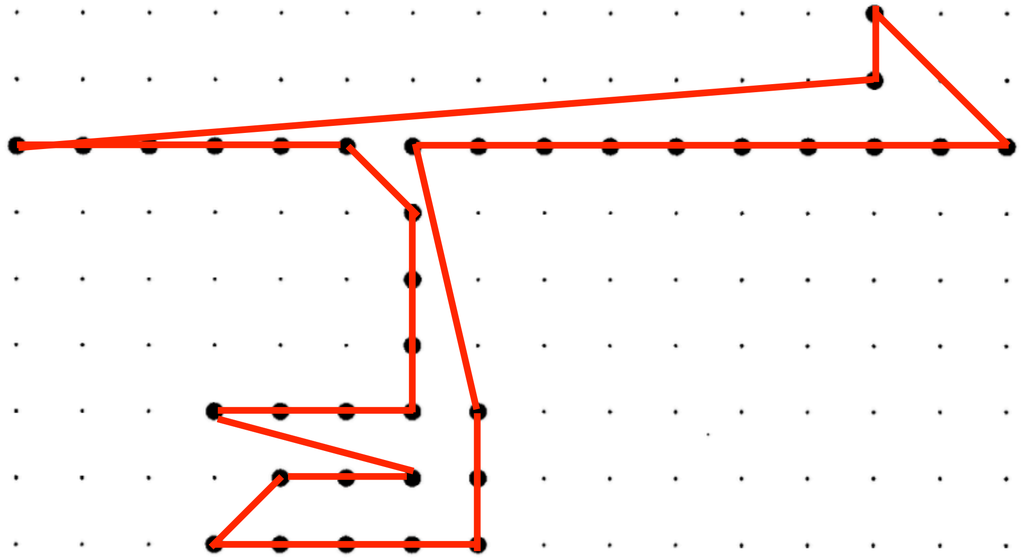


2 Minimum-Area Polygons

Optimal Polygonizations

Given: A planar point set P

Wanted: A simple polygon with vertex set P

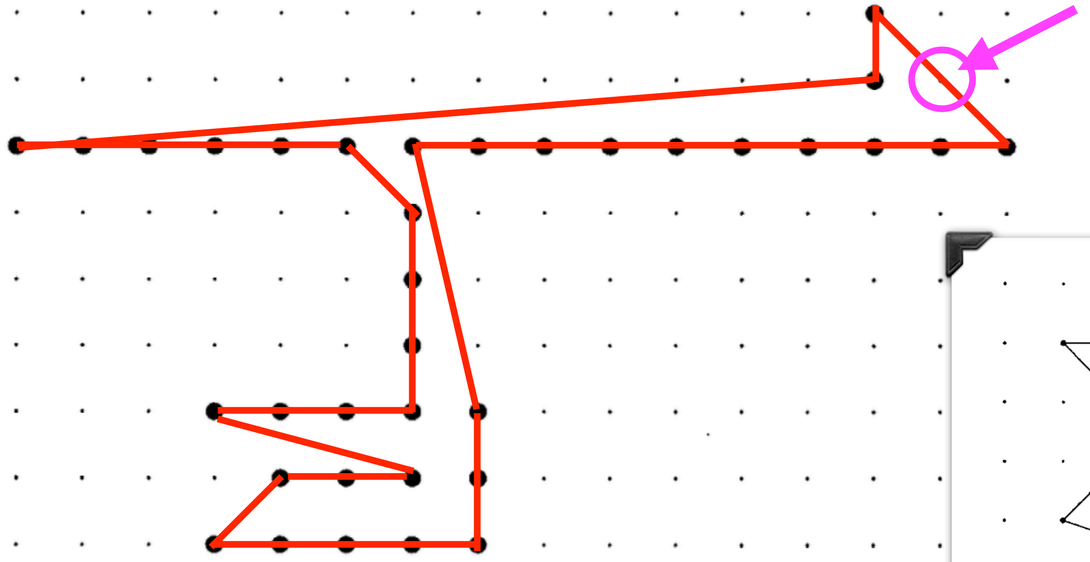


Objective: Minimize Perimeter → Traveling Salesman Problem

Optimal Polygonizations

Given: A planar point set *P*

Wanted: A simple polygon with vertex set *P*



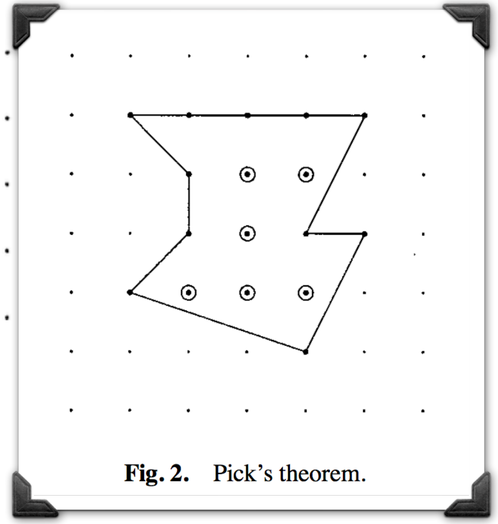


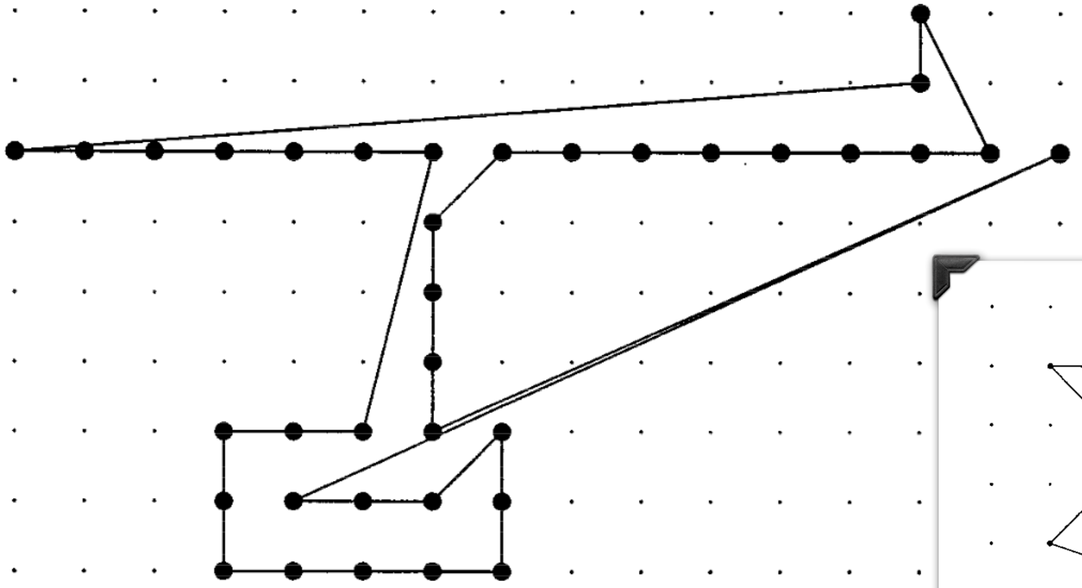
Fig. 2. Pick's theorem.

Objective: Minimize Area!

Optimal Polygonizations

Given: A planar point set *P*

Wanted: A simple polygon with vertex set *P*



Objective: Minimize Area!

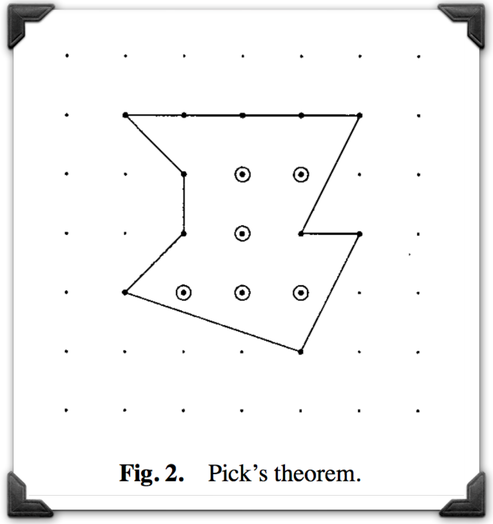
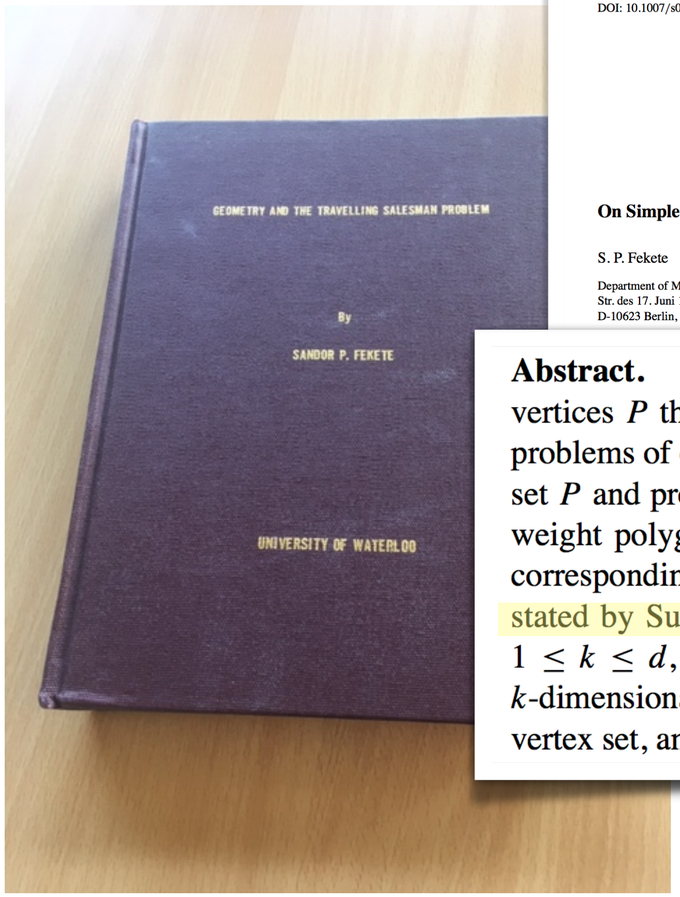


Fig. 2. Pick's theorem.

Some History



Discrete Comput Geom 23:73–110 (2000)
DOI: 10.1007/s004549910005

Discrete & Computational Geometry
© 2000 Springer-Verlag New York Inc.

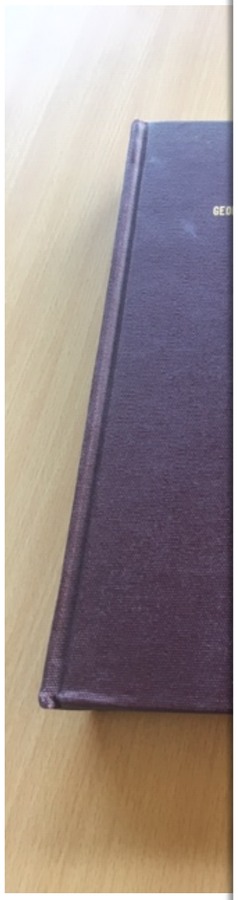
On Simple Polygonalizations with Optimal Area*

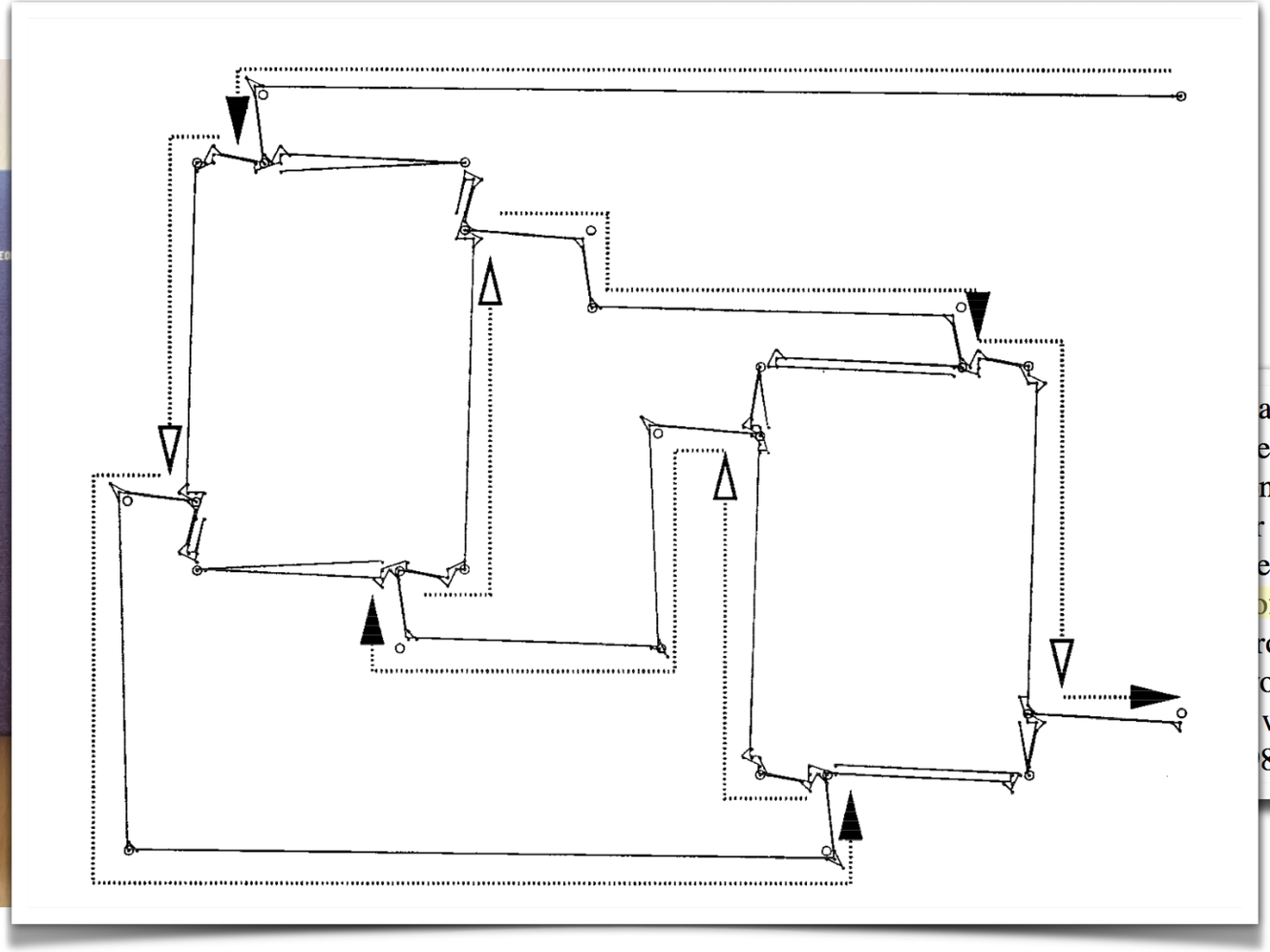
S. P. Fekete

Department of Mathematics, TU Berlin,
Str. des 17. Juni 136,
D-10623 Berlin, Germany

Abstract. We discuss the problem of finding a simple polygonalization for a given set of vertices P that has optimal area. We show that these problems are very closely related to problems of optimizing the number of points from a set Q in a simple polygon with vertex set P and prove that it is NP-complete to find a minimum weight polygon or a maximum weight polygon for a given vertex set, resulting in a proof of NP-completeness for the corresponding area optimization problems. This answers a generalization of a question stated by Suri in 1989. Finally, we turn to higher dimensions, where we prove that, for $1 \leq k \leq d, 2 \leq d$, it is NP-hard to determine the smallest possible total volume of the k-dimensional faces of a d-dimensional simple nondegenerate polyhedron with a given vertex set, answering a generalization of a question stated by O'Rourke in 1980.

Some History





a given set of
ely related to
n with vertex
r a maximum
eness for the
of a question
rove that, for
olume of the
with a given
80.

Some History

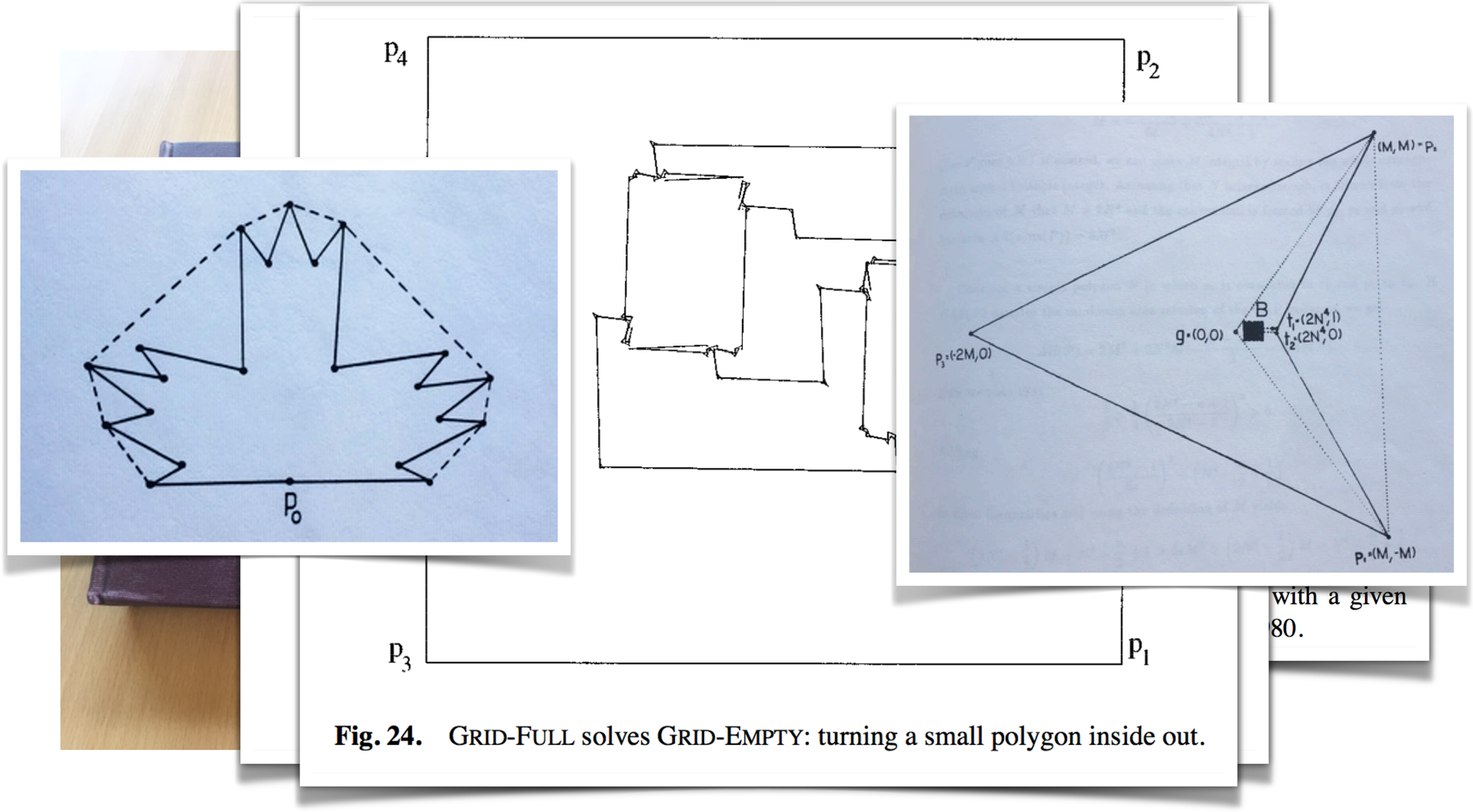


Fig. 24. GRID-FULL solves GRID-EMPTY: turning a small polygon inside out.

The 2019 CG Challenge

Area-Optimal Simple Polygonalizations: The CG Challenge 2019

ERIK D. DEMAINE, CSAIL, MIT, USA
SÁNDOR P. FEKETE, PHILLIP KELDENICH, and DOMINIK KRUPKE, Department of Computer Science, TU Braunschweig, Germany
JOSEPH S. B. MITCHELL, Department of Applied Mathematics and Statistics, Stony Brook University, USA

We give an overview of theoretical and practical aspects of finding a simple polygon of minimum (Min-Area) or maximum (Max-Area) possible area for a given set of n points in the plane. Both problems are known to be $\mathcal{NP}$-hard and were the subject of the 2019 Computational Geometry Challenge, which presented the quest of finding good solutions to more than 200 instances, ranging from $n = 10$ all the way to $n = 1{,}000{,}000$.

CCS Concepts: • **Theory of computation → Design and analysis of algorithms**; *Computational Geometry;*

Additional Key Words and Phrases: Computational geometry, geometric optimization, algorithm engineering, exact algorithms, polygonization, area optimization

ACM Reference format:
Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. 2022. Area-Optimal Simple Polygonalizations: The CG Challenge 2019. *J. Exp. Algorithmics* 27, 2, Article 2.4 (February 2022), 12 pages.
https://doi.org/10.1145/3504000

1 INTRODUCTION

1.1 The Computational Geometry Challenge

The "**CG:SHOP Challenge**" (**Computational Geometry: Solving Hard Optimization Problems**) originated as a workshop at the 2019 **Computational Geometry Week** (**CG Week**) in Portland, Oregon in June, 2019. The goal was to conduct a computational challenge competition that focused attention on a specific hard geometric optimization problem, encouraging researchers to devise and implement solution methods that could be compared scientifically based on how well they performed on a database of instances. While much of computational geometry research has targeted theoretical research, often seeking provable approximation algorithms for $\mathcal{NP}$-hard

This work was funded by DFG project "Computational Geometry: Solving Hard Optimization Problems" (CG:SHOP), FE 407/21-1. J.M. has been partially supported by the National Science Foundation (CCF-2007275, CCF-1526406), the US-Israel Binational Science Foundation (BSF project 2016116), Sandia National Labs, and DARPA (Lagrange).
Authors' addresses: E. D. Demaine, CSAIL, MIT, 32 Vassar Street, Cambridge, MA 02139; email: edemaine@mit.edu; S. P. Fekete, P. Keldenich, and D. Krupke, Department of Computer Science, TU Braunschweig, Mühlenpfordtstr. 32, 38106 Braunschweig, Germany; emails: {s.fekete, p.keldenich, d.krupke}@tu-bs.de; J. S. B. Mitchell, Department of Applied Mathematics and Statistics, 100 Nicolls Rd., Stony Brook University, Stony Brook, NY 11794; email: joseph.mitchell@stonybrook.edu.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2022 Copyright held by the owner/author(s).
1084-6654/2022/02-ART2.4
https://doi.org/10.1145/3504000

The 2019 CG Challenge

The contest consists of a total of **247 instances**, as follows. For n= {10,15,20,25,30,35,40,45,50,60,70,80,90,100,200,300,400,500,600,700,800,900,1000, 2000,3000,4000,5000,6000,7000,8000,9000,10000,20000,30000,40000,50000,60000, 70000,80000,90000,100000}, there are six instances each. In addition, there will be one instance of size n=1000000.
These instances are of three different types:

- uniform: uniformly at random from a square
- edge: randomly generated according to the distribution of the rate of change (the "edges") of an image
- illumination: randomly generated according to the distribution of brightness of an image (such as an illumination map)

Each instance consists of n points in the plane with even integer coordinates. (This ensures that the area of any simple polygon will be an integer.)

For each instance, the **score** is the ratio between the achieved area, divided by the area of the convex hull, i.e., a number between 0 and 1. For instances without a feasible solution, the default score is 1 (for minimization) or 0 (for maximization). The total score is the sum of all 247 individual scores.

The contest will be run in several different categories. These categories include:

(Score_min) The best total score for minimum area polygons
(Score_max) The best total score for maximum area polygons
(Opt_min) The largest number of instances solved to optimality for a minimum area polygon
(Opt_max) The largest number of instances solved to optimality for a maximum area polygon
(Bound_min) The best bounds for minimization
(Bound_max) The best bounds for maximization

Contest opens 24:00 (midnight, CET), February 28, 2019.
Contest closes 24:00 (midnight, CET), May 31, 2019.

Some Recent Progress

Technische Universität Braunschweig

Bachelor's Thesis

New algorithm approaches f area-optimal pol

Michael Perk

June 29, 2018

Institute of Operating Systems and Co
Prof. Dr. Sándor Fekete

Supervisors:
Arne Schmidt, M. Sc.
Andreas Haas, M. Sc.
Phillip Keldenich, M. Sc

Computing Area-Optimal Simple Polygonizations

SÁNDOR P. FEKETE, ANDREAS HAAS, PHILLIP KELDENICH, MICHAEL PERK, and ARNE SCHMIDT, Department of Computer Science, TU Braunschweig

We consider methods for finding a simple polygon of minimum (Min-Area) or maximum (Max-Area) possible area for a given set of points in the plane. Both problems are known to be NP-hard; at the center of the recent CG Challenge, practical methods have received considerable attention. However, previous methods focused on heuristic methods, with no proof of optimality. We develop exact methods, based on a combination of geometry and integer programming. As a result, we are able to solve instances of up to $n = 25$ points to provable optimality. While this extends the range of solvable instances by a considerable amount, it also illustrates the practical difficulty of both problem variants.

CCS Concepts: • **Theory of computation** → **Design and analysis of algorithms**; *Computational Geometry;*

Additional Key Words and Phrases: Computational Geometry, geometric optimization, algorithm engineering, exact algorithms, polygonization, area optimization

ACM Reference format:
Sándor P. Fekete, Andreas Haas, Phillip Keldenich, Michael Perk, and Arne Schmidt. 2022. Computing Area-Optimal Simple Polygonizations. *J. Exp. Algorithmics* 27, 2, Article 2.6 (May 2022), 23 pages.
https://doi.org/10.1145/3503607

1 INTRODUCTION

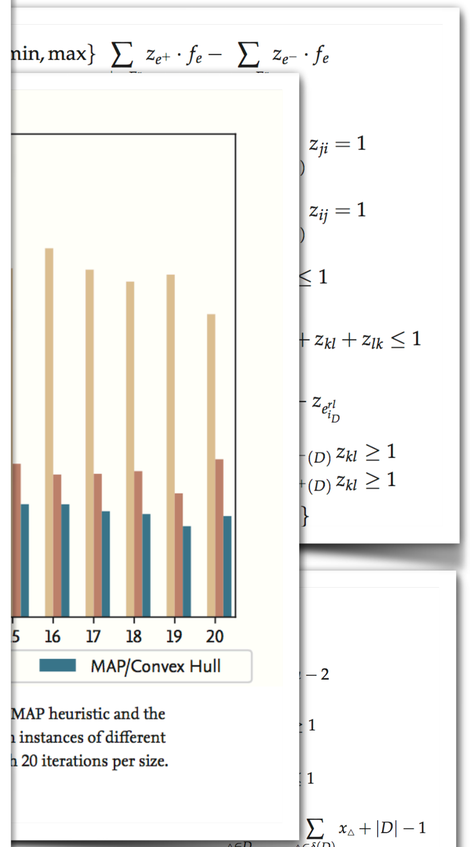
While the classic geometric **Traveling Salesman Problem** (**TSP**) is to find a (simple) polygon with a given set of vertices that have the shortest perimeter, it is natural to look for a simple polygon with a given set of vertices that minimizes another basic geometric measure: the enclosed area. The problem Min-Area asks for a simple polygon with minimum enclosed area, while Max-Area demands one of the maximum area; see Figure 1 for an illustration.

Both problem variants were shown to be $\mathcal{NP}$-complete by Fekete [2, 3, 6], who also showed that no **polynomial-time approximation scheme** (**PTAS**) exists for Min-Area problem and gave a $\frac{1}{2}$-approximation algorithm for Max-Area.

1.1 Related Work

Most previous practical work has focused on finding heuristics for both problems. Taranilla et al. [14] proposed three different heuristics. Peethambaran [11, 12] later proposed randomized

This work was supported by DFG grant FE407/21-1, "Computational Geometry: Solving Hard Optimization Problems (CG:SHOP)".
Authors' address: S. P. Fekete, A. Haas, P. Keldenich, M. Perk, and A. Schmidt, Department of Computer Science, TU Braunschweig, Braunschweig 38106, Germany; emails: {s.fekete, a.haas, p.keldenich, m.perk, arne.schmidt}@tu-bs.de.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1084-6654/2022/05-ART2.6 $15.00
https://doi.org/10.1145/3503607



$\min, \max\} \sum z_{e^+} \cdot f_e - \sum z_{e^-} \cdot f_e$

$z_{ji} = 1$

$z_{ij} = 1$

≤ 1

$+ z_{kl} + z_{lk} \leq 1$

$- z_{e_{i_D}^l}$

$(D)\ z_{kl} \geq 1$

$(D)\ z_{kl} \geq 1$

$- 2$

$: 1$

$: 1$

$\sum_{\triangle \in \delta(D)} x_\triangle + |D| - 1$

$\forall \triangle \in T: \quad x_\triangle \in \{0,1\}$

MAP heuristic and the
instances of different
20 iterations per size.

Project and Instances

CG:SHOP Help Instances Publications Competitions Login Register

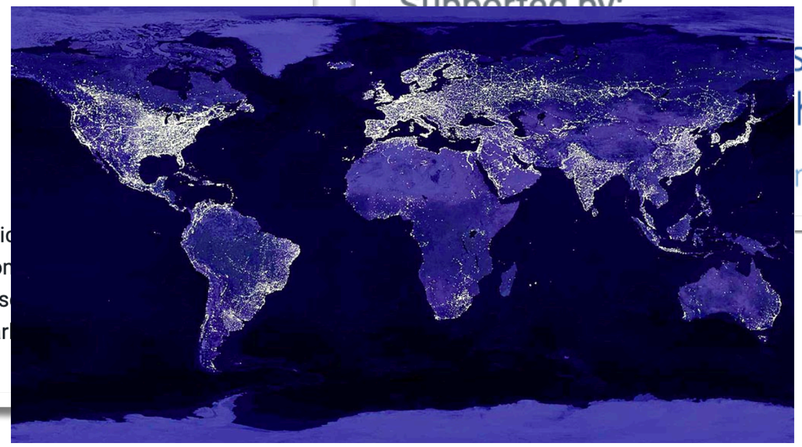
Computational Geometry: Solving Hard Optimization Problems

Geometric Optimization Challenges

The CG:SHOP Chall
tackling specific, dif
(CG Week) in 2019,
computational perfo
from traditional con

Supported by:

sche
hungsgemeinschaft
n Research Foundation



Hard instances project

Home Search

Instance Search

Instance Type: Points
Generation Type: artificial, imag
Size: 4 — 5000000

Deselect All Select All

a280
Type: Points
Size: 279
Generator: tsplib
Generation Type: artificial
Add to package

ali535
Type: Points
Size: 506
Generator: tsplib
Generation Type: artificial
Add to package

att48
Type: Points
Size: 48
Generator: tsplib
Generation Type: artificial
Add to package

att532
Type: Points
Size: 532
Generator: tsplib
Generation Type: artificial
Add to package

berlin52

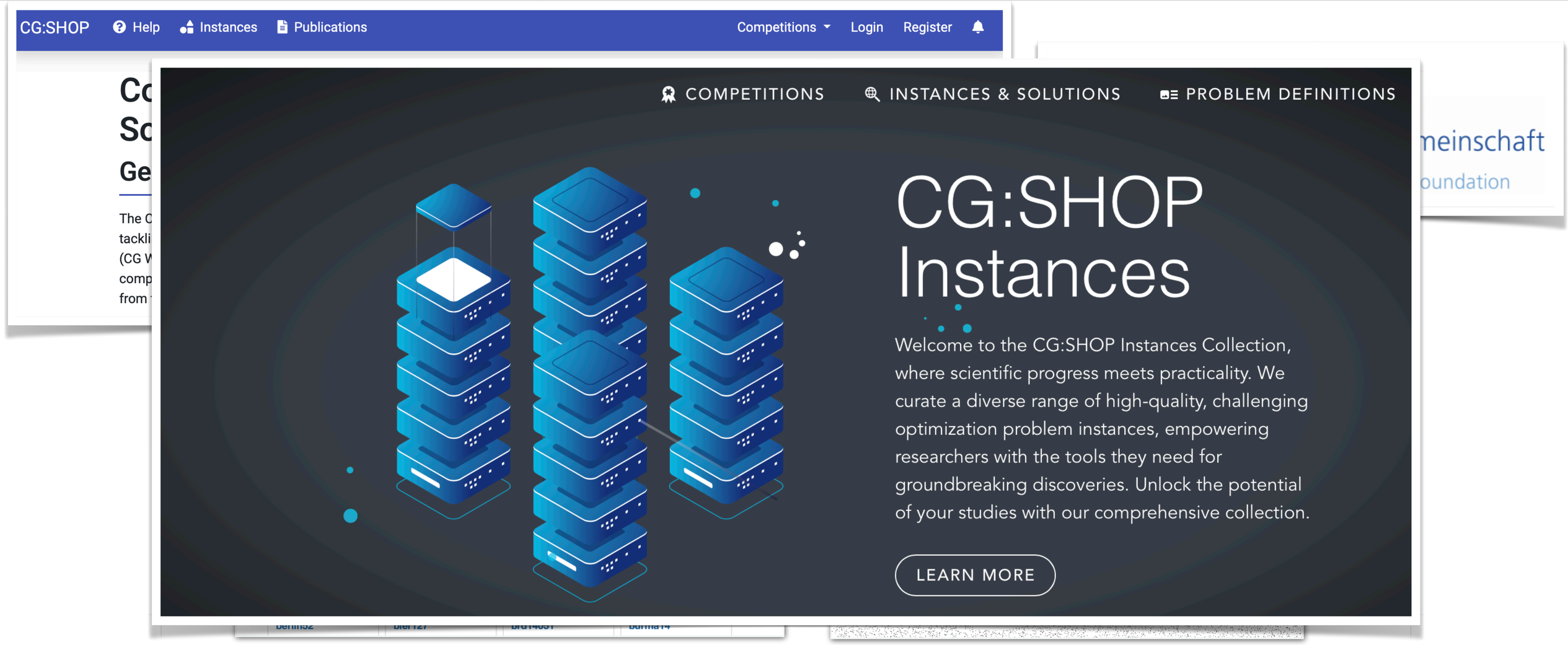
bier127

brd14051

burma14



Project and Instances



Project and Instances

CG:SHOP Help

Our Problem Collection



Maximum Polygon Packing

Optimal packing a set of objects into a given container.

LEARN MORE



Minimum Convex Partition

Partition a point set into a smallest set of convex faces.

LEARN MORE



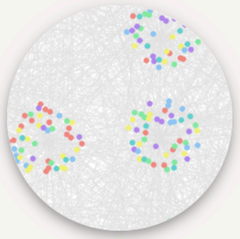
Multi Agent Path Finding

Compute a set of feasible trajectories a given set of robots and targets.

LEARN MORE



Minimum Coverage by Convex Polygons



Minimum Partition into Plane Subgraphs

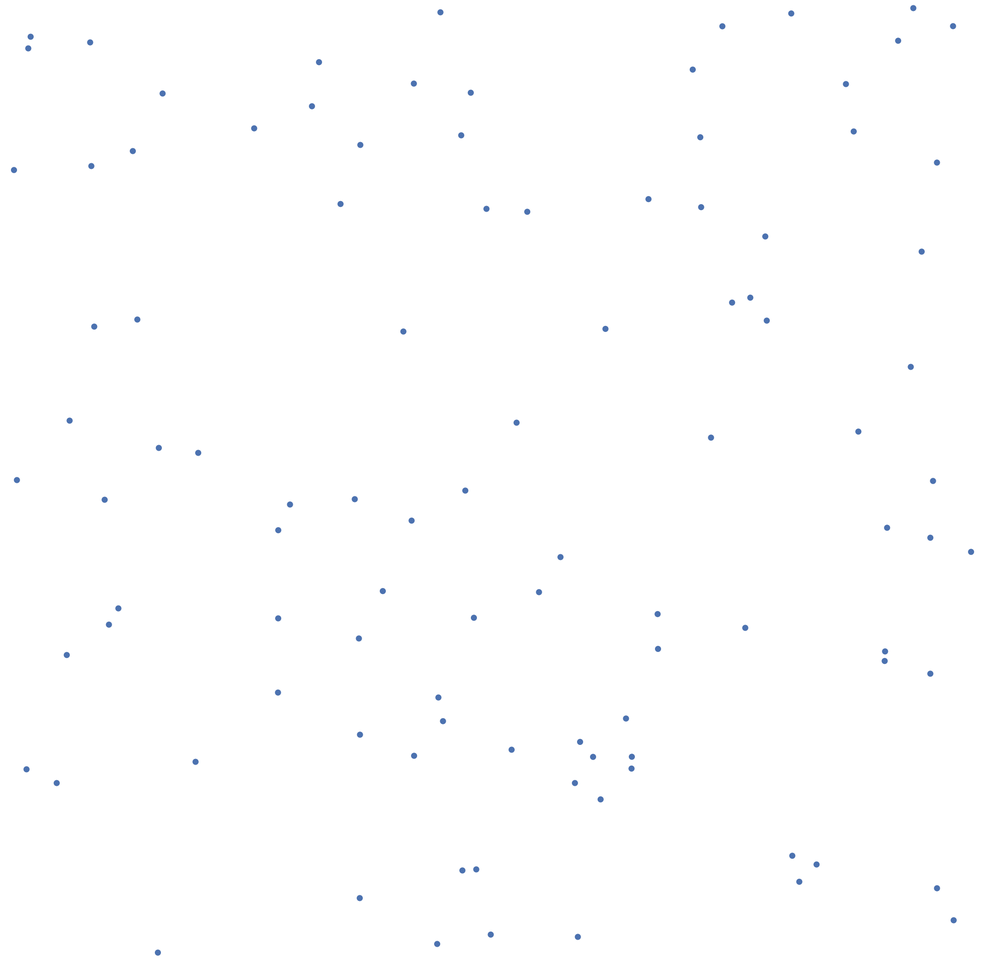


Minimum Non-Obtuse Triangulation

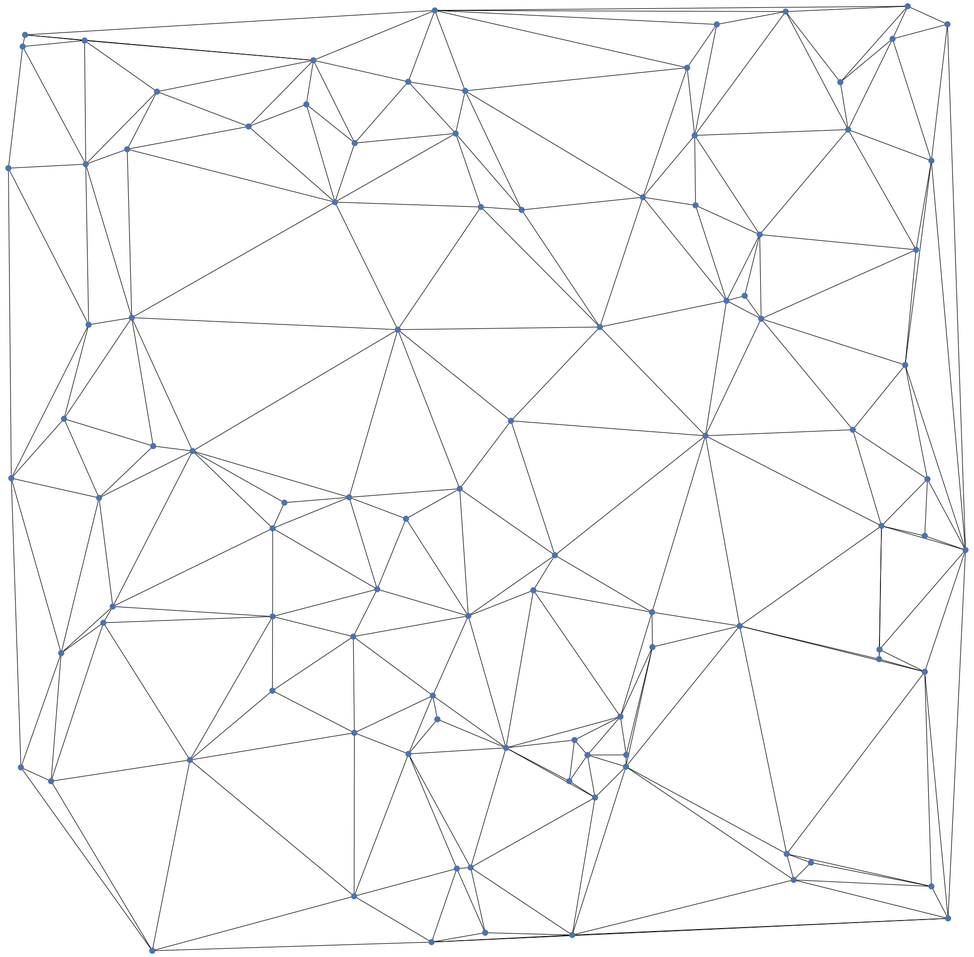
meinschaft
oundation

3 Minimum-Weight Triangulation

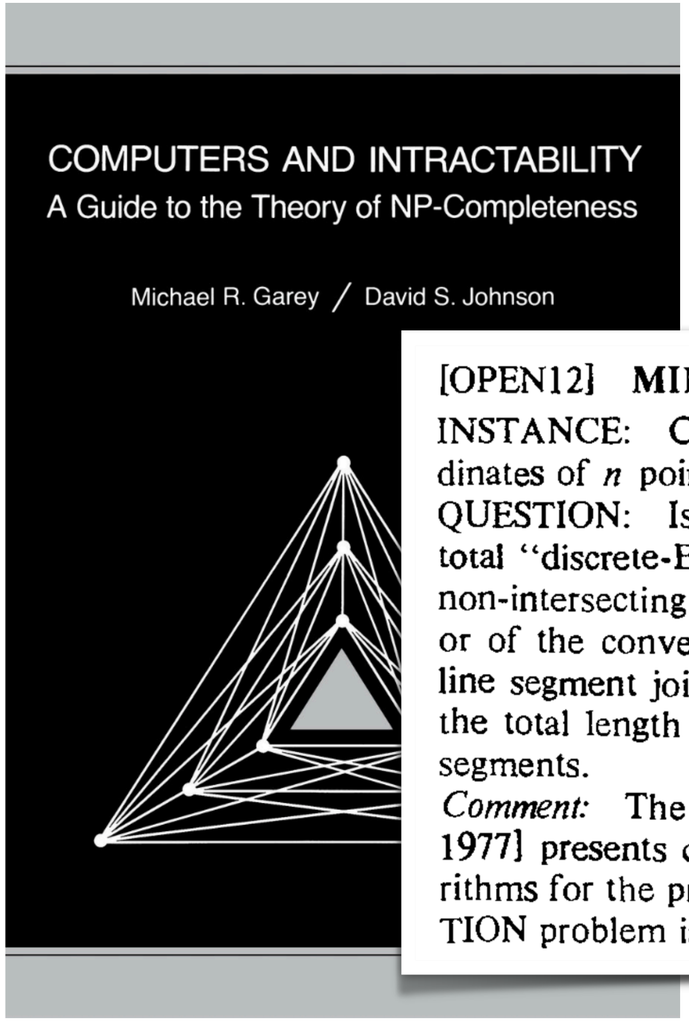
Minimum-Weight Triangulation



Minimum-Weight Triangulation



Minimum-Weight Triangulation



[OPEN12] MINIMUM LENGTH TRIANGULATION

INSTANCE: Collection $C = \{(a_i, b_i): 1 \leqslant i \leqslant n\}$ of pairs of integers, giving the coordinates of n points in the plane, and a positive integer B.

QUESTION: Is there a triangulation of the set of points represented by C that has total "discrete-Euclidean" length B or less? Here a triangulation is a collection of non-intersecting line segments, each joining two points in C, that divides the interior of the convex hull into triangular regions. The discrete-Euclidean length of a line segment joining (a_i, b_i) and (a_j, b_j) is given by $\lceil ((a_i - a_j)^2 + (b_i - b_j)^2)^{1/2} \rceil$, and the total length of a triangulation is the sum of the lengths of its constituent line segments.

Comment: The analogous problem for the rectilinear metric is also open. [Lloyd, 1977] presents counterexamples to a number of conjectured polynomial time algorithms for the problem and proves that the related CONSTRAINED TRIANGULATION problem is NP-complete.

MWT is NP-hard [Mulzer and Rote, 2008]

Minimum-Weight Triangulation is NP-hard

WOLFGANG MULZER
Princeton University
and
GÜNTER ROTE
Freie Universität Berlin

A triangulation of a planar point set S is a maximal plane straight-line graph with vertex set S. In the *minimum-weight triangulation* (MWT) problem, we are looking for a triangulation of a given point set that minimizes the sum of the edge lengths. We prove that the decision version of this problem is NP-hard, using a reduction from PLANAR 1-IN-3-SAT. The correct working of the gadgets is established with computer assistance, using dynamic programming on polygonal faces, as well as the β-skeleton heuristic to certify that certain edges belong to the minimum-weight triangulation.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Geometrical problems and computations; G.2.2 [**Graph Theory**]: Graph algorithms

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Optimal triangulations, PLANAR 1-IN-3-SAT

1. INTRODUCTION

Given a set S of points in the Euclidean plane, a *triangulation* T of S is a maximal plane straight-line graph with vertex set S. The *weight* of T is defined as the total Euclidean length of all edges in T. A triangulation that achieves the minimum weight is called a *minimum-weight triangulation* (MWT) of S.

The problem of computing a triangulation for a given planar point set arises naturally in many applications such as stock cutting, finite element analysis, terrain modeling, and numerical approximation. The *minimum-weight* triangulation has attracted the attention of many researchers, mainly due to its natural definition of optimality, and not so much because it would be important for the mentioned applications. We show that computing a minimum-weight triangulation is NP-hard. Note that it is not known whether the MWT problem is in NP, because it is an open problem whether sums of radicals (namely, Euclidean distances) can be compared in polynomial time [Blömer 1991]. This problem is common to many geometric optimization problems, like, perhaps most famously, the Euclidean Traveling Salesperson Problem. To get a variant of the problem that is in NP, one can take the weight of an edge e as the rounded value $\lceil \|e\|_2 \rceil$. Our proof also shows that this variant is NP-complete.

Our proof uses a polynomial time reduction from POSITIVE PLANAR 1-IN-3-SAT, a variant of the well-known PLANAR 3-SAT problem, which is a standard tool for showing NP-hardness of geometric problems.

1.1 Optimal Triangulations

Usually, a planar point set has (exponentially) many different triangulations, and many applications call for triangulations with certain good properties (see Figure 1). Optimal triangulations under various optimality criteria were extensively surveyed

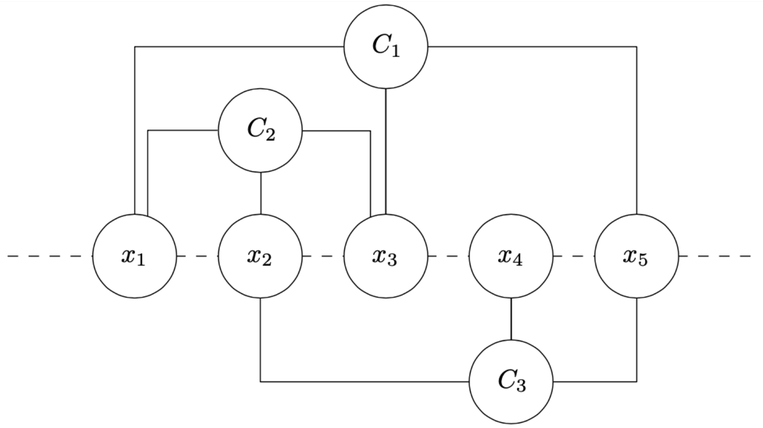
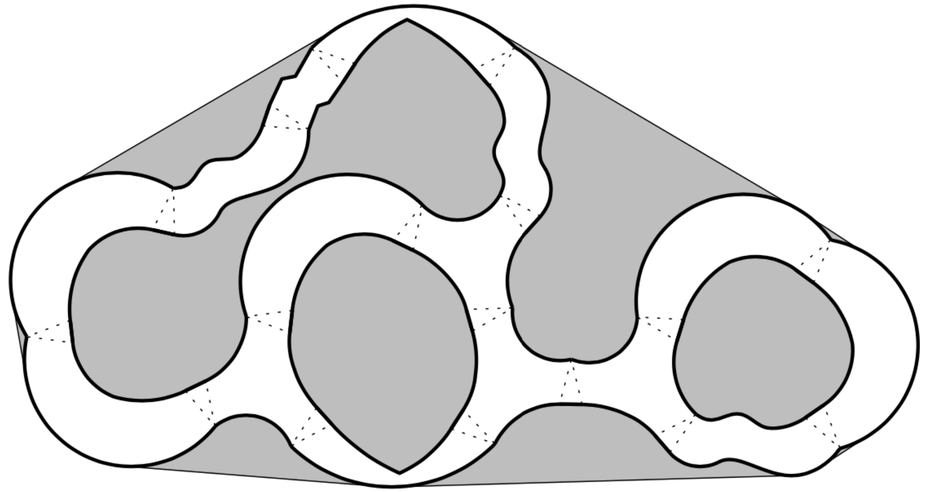


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.



MWT is NP-hard [Mulzer and Rote, 2008]

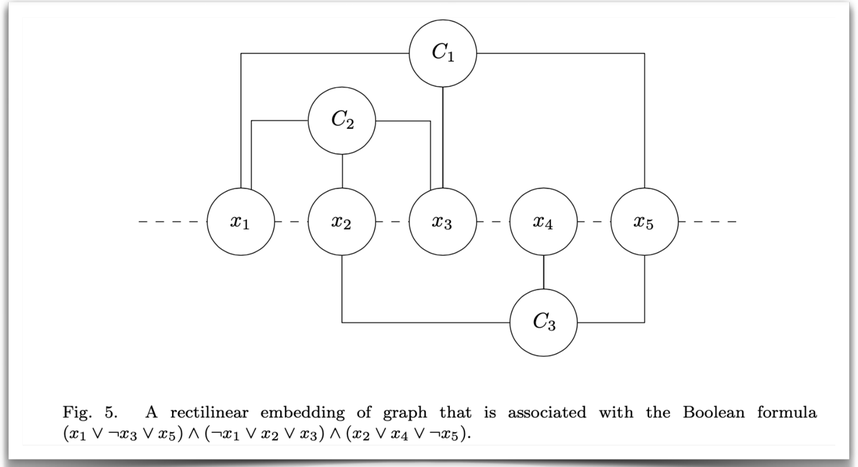
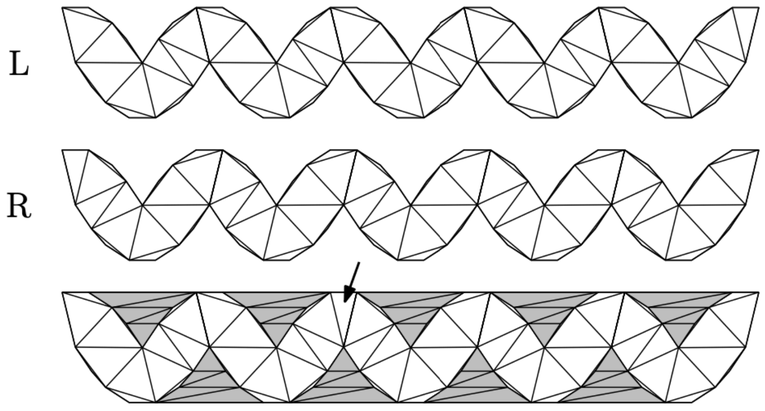


Fig. 5. A rectilinear embedding of graph that is associated with the Boolean formula $(x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_4 \vee \neg x_5)$.



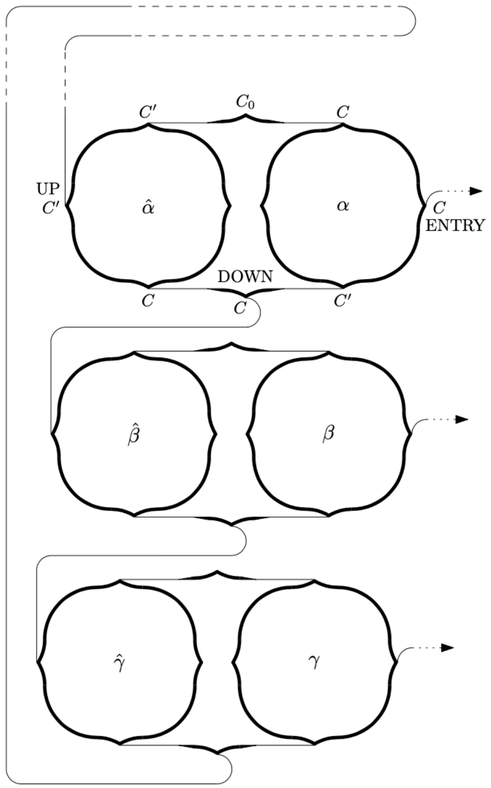
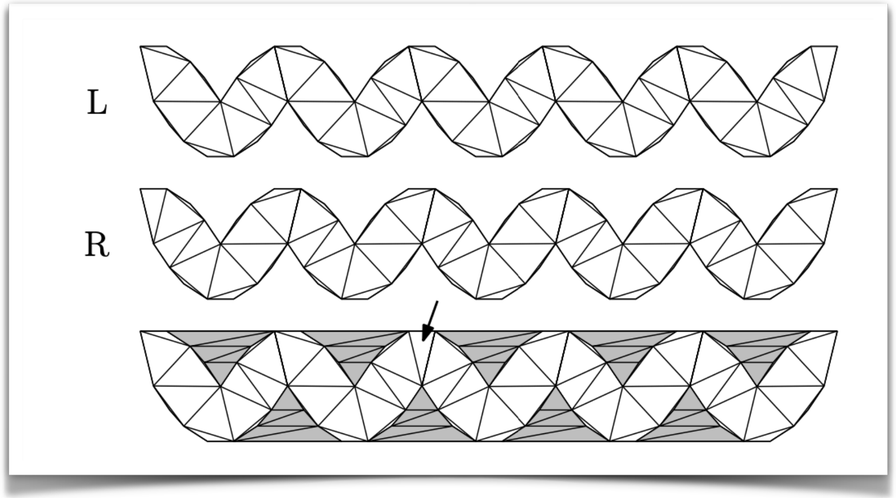


Fig. 19. Schematic view of the clause gadget. The dashed parts of the wires are long enough for sufficiently many copies of the wire-piece and the extended wire-piece to ensure that the wire can reach from the lowest C connection to the left C' connection of $\hat{\alpha}$ (Lemma 6.1).

MWT is NP-hard [Mulzer and Rote, 2008]



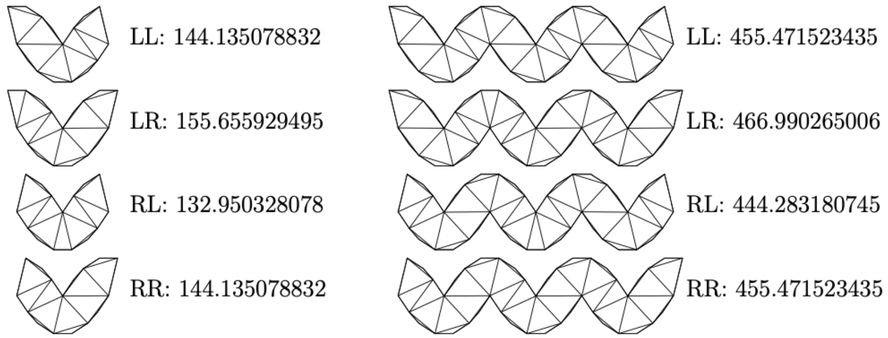


Fig. 17. Optimal solutions for all cases for the wire-piece and the extended wire-piece

pattern	multiplicity	internal cost c	reduced internal cost $\bar{c}$	relative reduced cost $\tilde{c}$
LL	1	455.471 523 435	455.471 523 435	0.000 000 000
LR	2	466.990 265 006	455.679 921 006	0.208 397 570
RL	1	444.283 180 745	455.593 524 745	0.122 001 310
RR	1	455.471 523 435	455.471 523 435	0.000 000 000

Table III. Analysis of the extended wire-piece

LEMMA 6.1. *Consider two small vertical terminal triangles at the same height with a horizontal distance $d > 230\,000$ that is a multiple of 0.01. Then the two triangles can be connected by a sequence of wire-pieces and extended wire-pieces.*
An analogous statement holds for vertical connections.

PROOF. If the distance between two terminal triangles is $d = 0.01 \cdot z$ for some integer $z \geq 3 \cdot 2740 \cdot 2739 = 22\,514\,580$, they can be connected by concatenating $y := z \bmod 2740$ extended wire-pieces with $\lfloor z/2740 \rfloor - 3y$ wire-pieces. □

MWT is NP-hard [Mulzer and Rote, 2008]

	wire-piece	extended wire-piece	thickening adapter	thinning adapter
LL	0.000 000 000	0.000 000 000	$0.000\,051\,402\ldots = \varepsilon_1$	0.000 000 000
LR	0.210 506 663	0.208 397 570	0.018 887 246	0.018 887 246
RL	0.125 593 246	0.122 001 310	0.014 627 250	0.014 627 250
RR	0.000 000 000	0.000 000 000	0.000 000 000	$0.000\,051\,402\ldots = \varepsilon_1$

	C_0 connection	left bend	right bend	thick left bend
LL	0.000 000 000	0.000 000 000	0.000 000 000	0.000 000 000
LR	0.044 001 701	0.086 460 895	0.210 506 663	0.891 261 046
RL	0.020 571 757	0.125 593 246	0.125 593 246	0.020 571 757
RR	0.000 000 000	0.000 000 000	0.000 000 000	0.000 000 000

Table IV. The relative reduced costs $\tilde{c}$ for the pieces with two terminals

C		relative reduced cost	C'	
LLl		$0.003\,861\,076\ldots = \delta_1$		RRr
LLr		$0.000\,001\,575\ldots = \varepsilon_2$		RRl
LRl		0.029 994 716		LRr
LRr		0.026 135 215		LRl
RLl		0.020 571 757		RLr
RLr		0.020 573 332		RLl
RRl		$0.000\,000\,000 = 0$		LLr
RRr		$0.004\,630\,878\ldots = \delta_2$		LLl

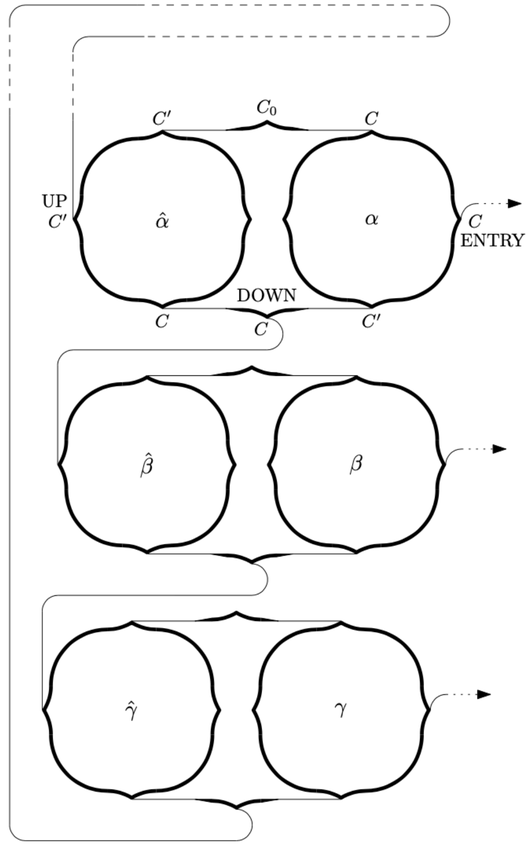


Fig. 19. Schematic view of the clause gadget. The dashed parts of the wires are long enough for sufficiently many copies of the wire-piece and the extended wire-piece to ensure that the wire can reach from the lowest C connection to the left C' connection of $\hat{\alpha}$ (Lemma 6.1).

This Talk

Hard in Theory, Easy in Practice: Solving Large Minimum-Weight Triangulation Instances to Provable Optimality

SÁNDOR P. FEKETE, ANDREAS HAAS, and PHILLIP KELDENICH, TU Braunschweig, Department of Computer Science, Germany

We consider practical solution of a well-known geometric optimization that combines the challenges of theoretical complexity with algebraic difficulties: For a given set of points in the two-dimensional plane, the NP-hard *Minimum-Weight Triangulation* problem asks for a maximal set of non-crossing edges of minimum total Euclidean distance.

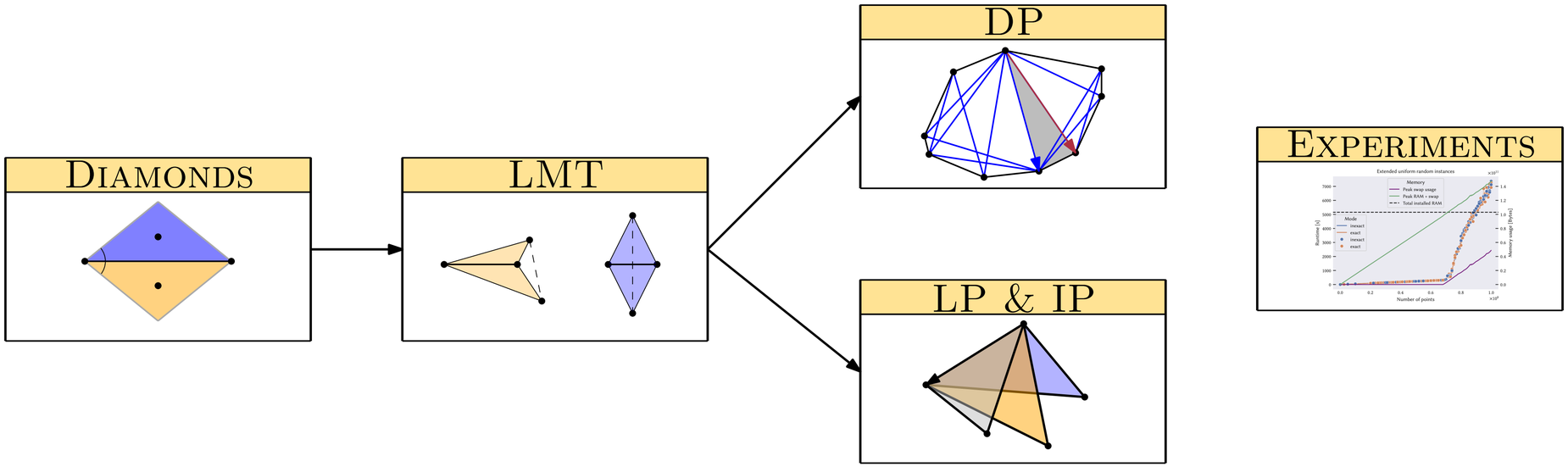
Based on a comprehensive set of optimization methods and implementation techniques, we demonstrate that Minimum-Weight Triangulation is a practically tractable problem, making it possible to compute provably optimal solutions for benchmark instances of sizes up to 100,000,000 points (including for *all* benchmark sets from other publicly available sources that we are aware of) in reasonable time on commodity hardware. In the process, we also show how to deal with a variety of other important aspects that can impede practical solvability of large geometric optimization problems, such as exactness of Euclidean distances, numerical issues, and memory management. We also address the existence of practically difficult instances, as well as the quality of minimum-weight triangulations as spanners.

Two Notes

Goal instance size: 10^8 points for “non-pathological“ instances.
Kills any idea requiring $\Omega(n^2)$ runtime or memory (in practice).

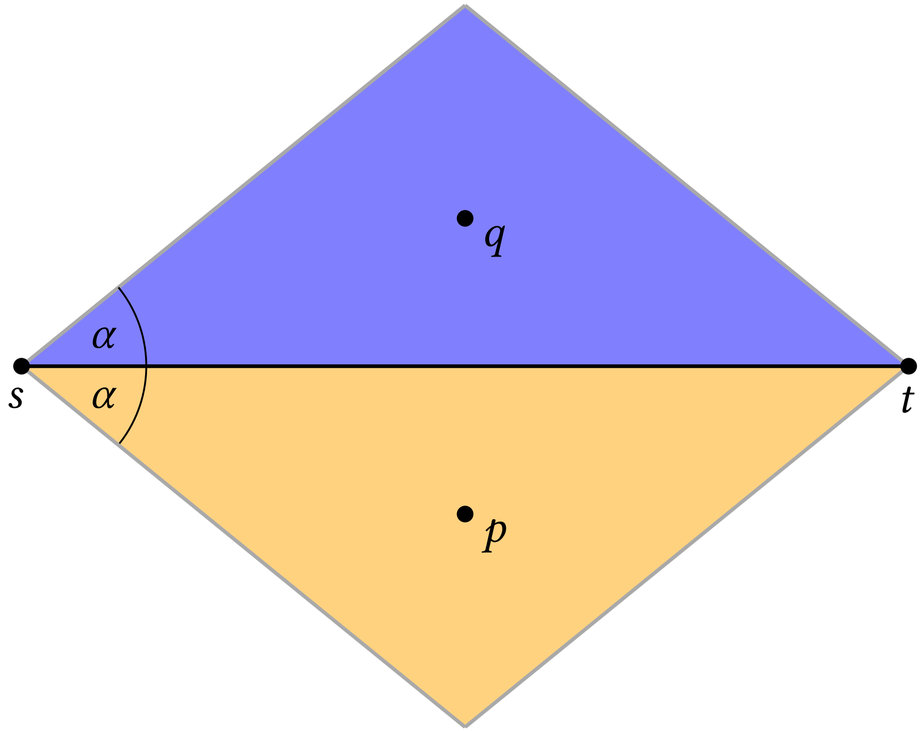
Numerical Exactness & Square Root Sums:
We aim for numerical exactness in our solution.
We cannot escape the square root sum problem everywhere.

Outline


DIAMONDS
LMT
DP
LP & IP
EXPERIMENTS
Extended uniform random instances
Memory
Peak swap usage
Peak RAM + swap
Total installed RAM
Mode
inexact
exact
inexact
exact
Runtime (s)
Memory usage [Bytes]
Number of points

Edge Enumeration: Diamond Filter

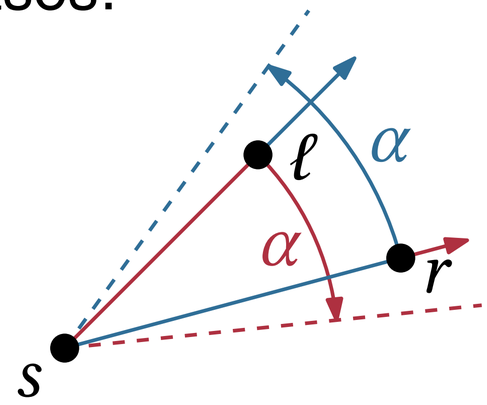
st has the **diamond property** iff one of the triangles is empty;
only edges with the diamond property can be in the MWT.
Largest known $\alpha = \pi/4.6$ (Drysdale et al. (2001)).



Dead Sectors

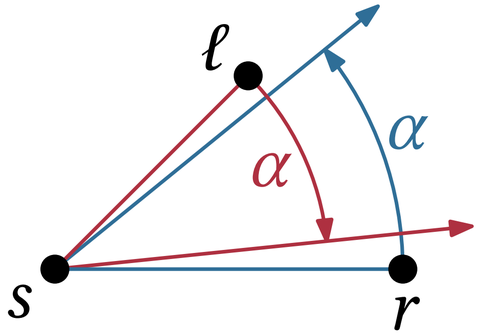
Dead sector representation: Polar angle interval + activation distance

Three cases:



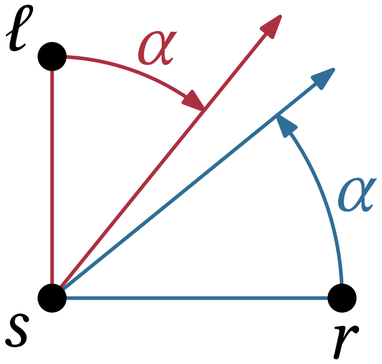
$\angle \ell sr \leq \alpha$

bounded by points



$\alpha < \angle \ell sr < 2\alpha$

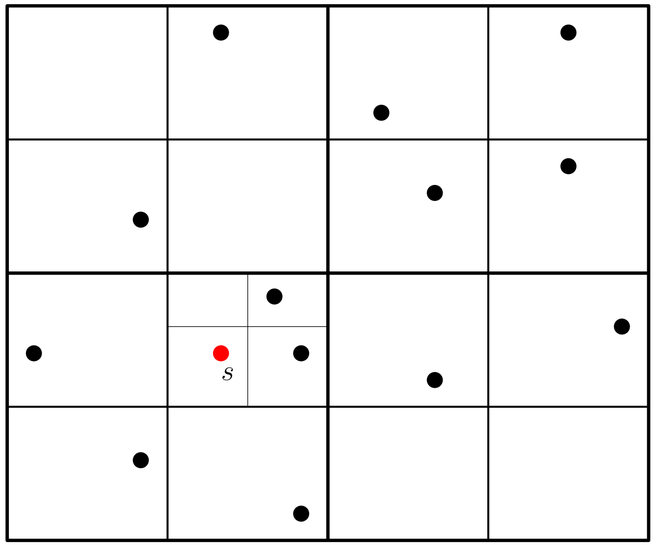
only angle interval



$2\alpha \leq \angle \ell sr$

empty/no sector

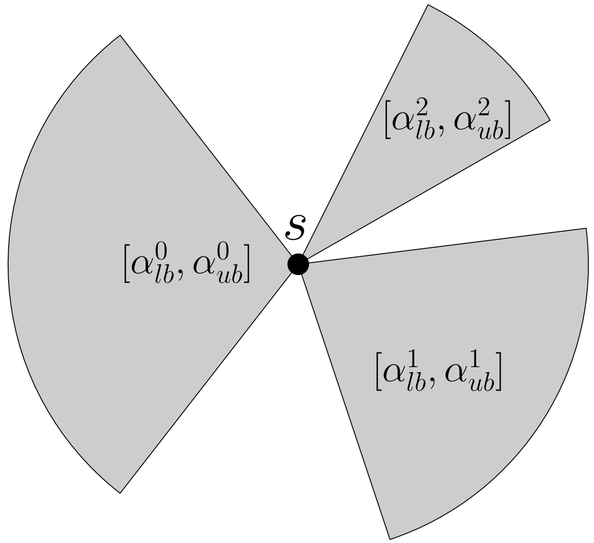
Candidate Edges Enumeration: Data Structures



point quadtree

$[\varphi^1_{lb}, \varphi^1_{ub}], \rho^1_A$
$[\varphi^2_{lb}, \varphi^2_{ub}], \rho^2_A$
$[\varphi^3_{lb}, \varphi^3_{ub}], \rho^3_A$
$[\varphi^4_{lb}, \varphi^4_{ub}], \rho^4_A$
$[\varphi^5_{lb}, \varphi^5_{ub}], \rho^5_A$
$\ldots$
$[\varphi^k_{lb}, \varphi^k_{ub}], \rho^k_A$

inactive sectors
FIFO

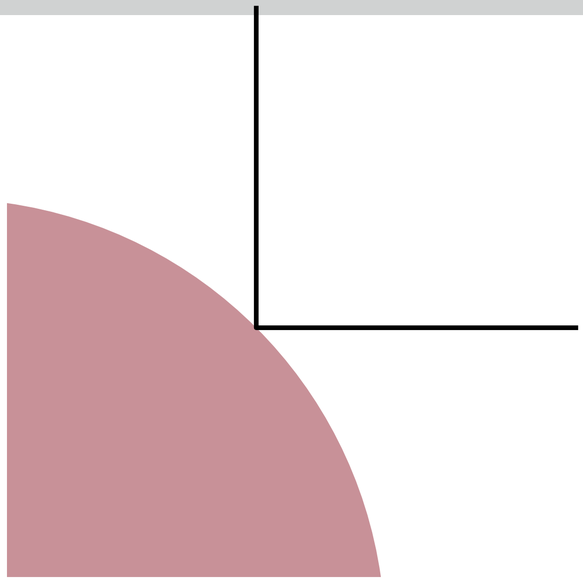


active sectors
sorted list

+ priority queue of nodes and points

+ angularly sorted set of discovered points

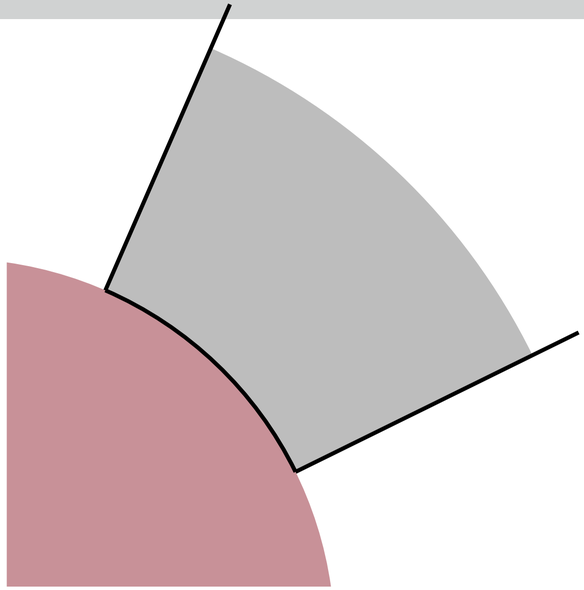
Event Types



- node covered by sectors?

➡ ignore

- else expand node



- activate dead sector
- insert angle interval
- potentially merge

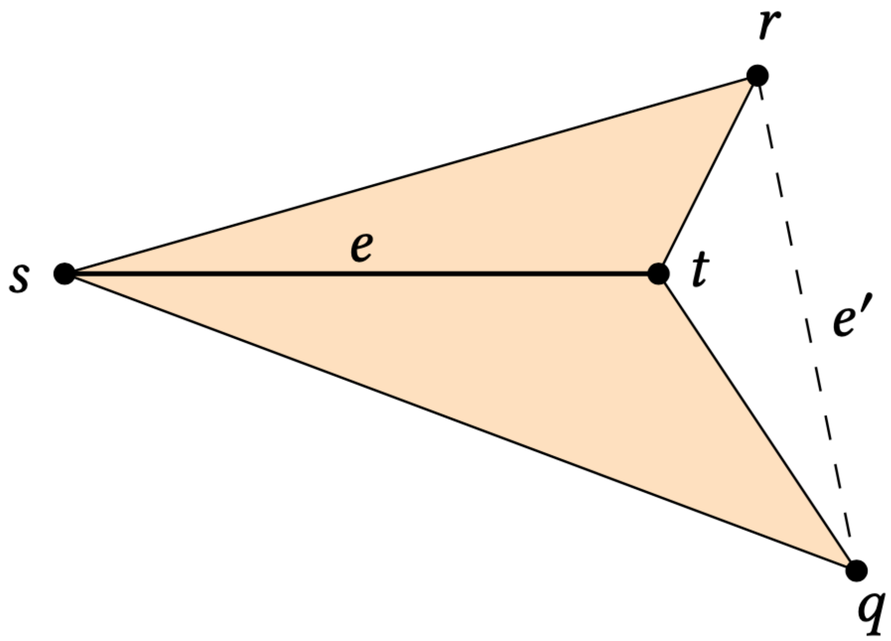
Parallelization & Exactness

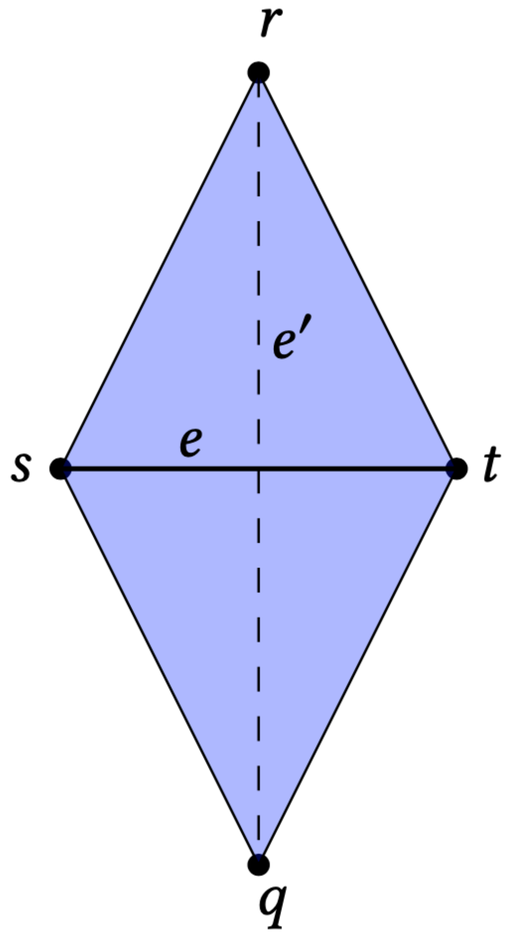
Parallelization: trivial, consider multiple sources individually
Exactness challenges:

- Ensure exactly correct order of events
- Except for squared distance comparisons, only interval arithmetic
- **Angles are challenging → pseudoangles**

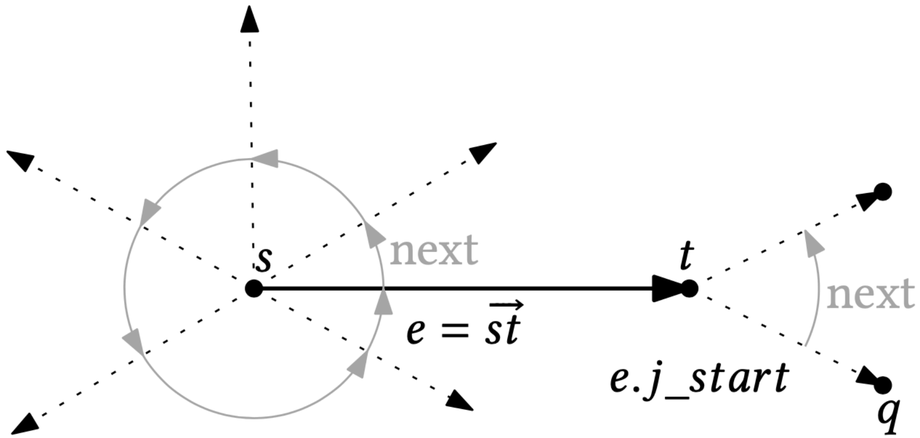
LMT Skeleton

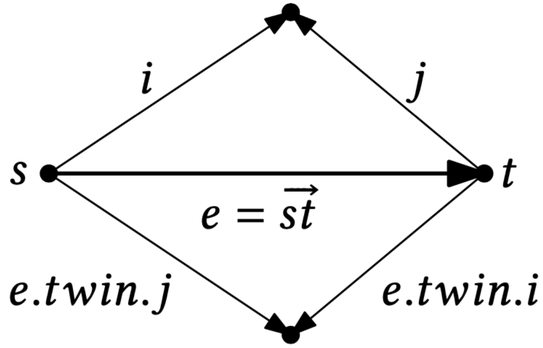
Edges must be locally minimal:



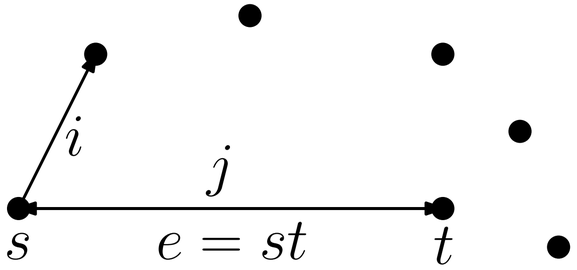


LMT Data Structure



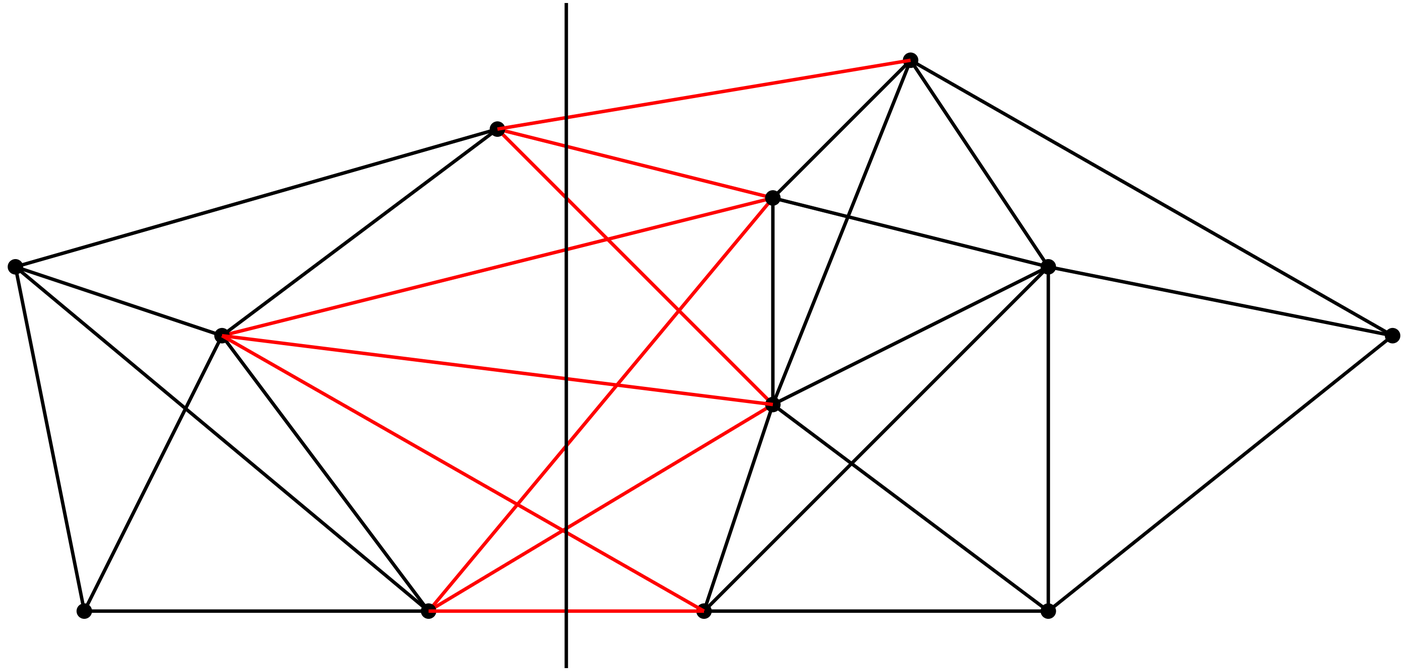


Find next triangle left of e:

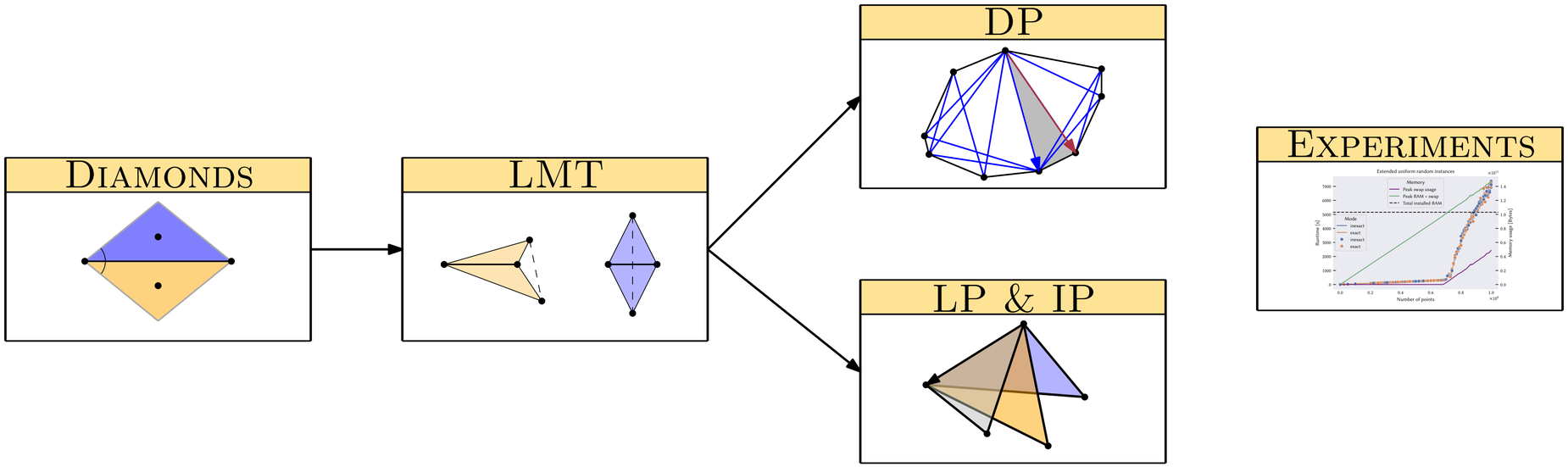


Parallelization & Exactness

Exactness: Only exact predicates needed
Parallelization: Data partitioning, no locking

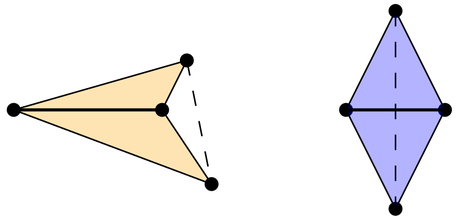


Outline

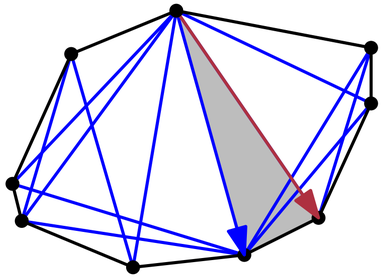

DIAMONDS
LMT
DP
LP & IP
EXPERIMENTS
Extended uniform random instances
Memory
Peak swap usage
Peak RAM + swap
Total installed RAM
Mode
inexact
exact
inexact
exact
Runtime (s)
Memory usage [Bytes]
Number of points

Outline

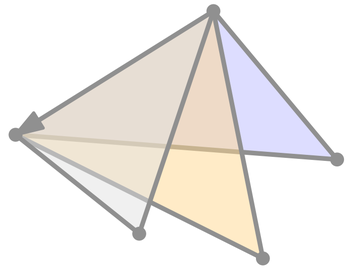
LMT



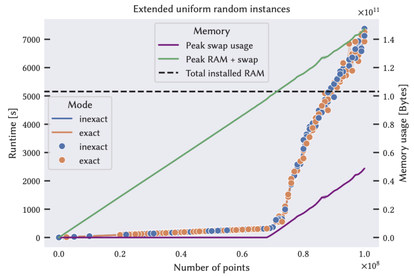
DP



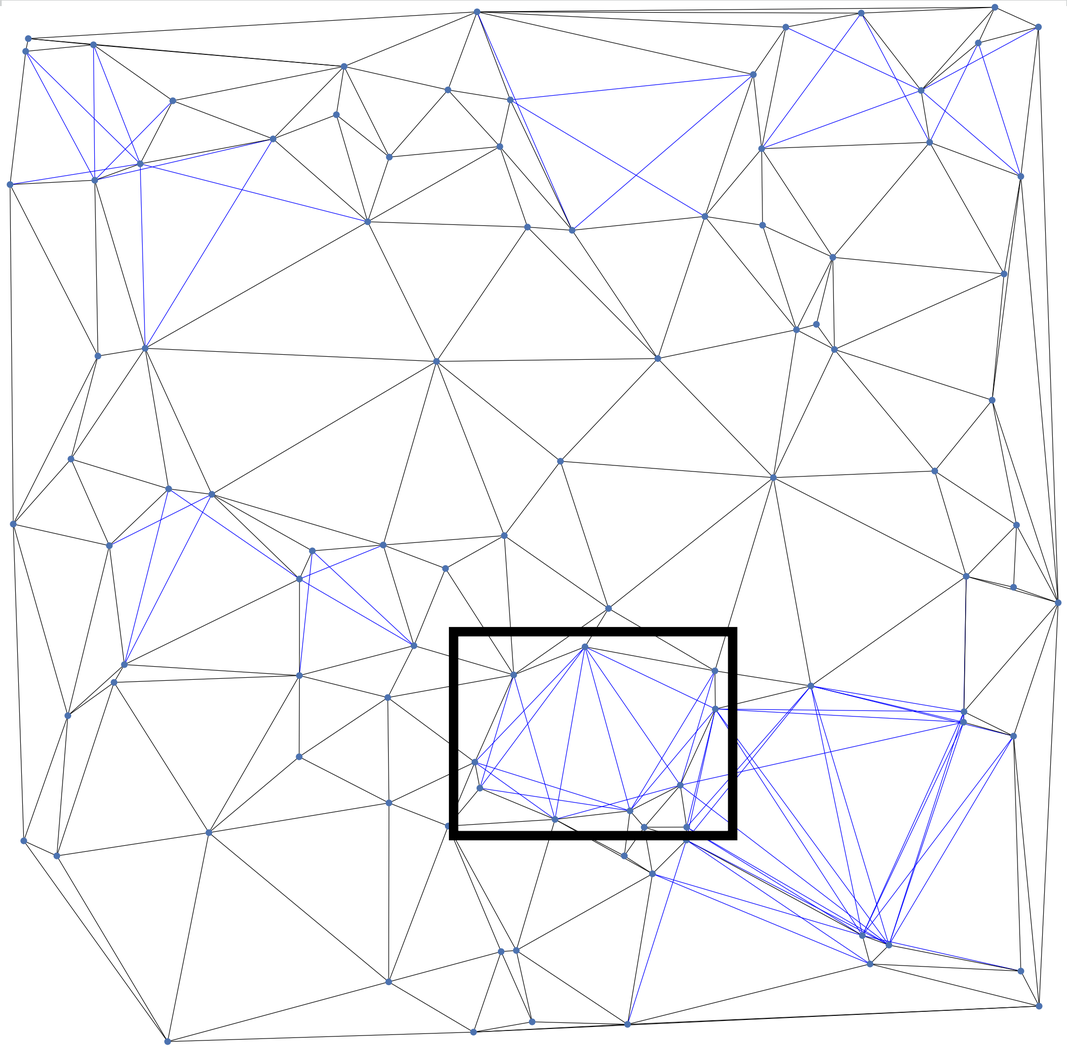
LP & IP



EXPERIMENTS

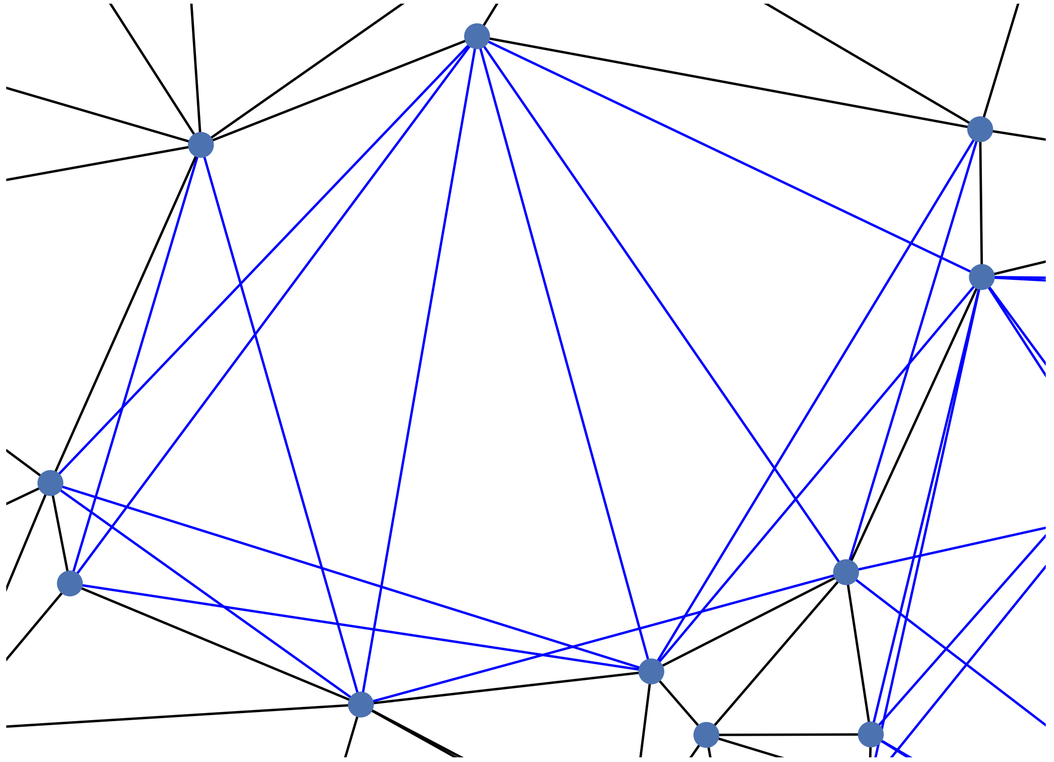


Dynamic Program



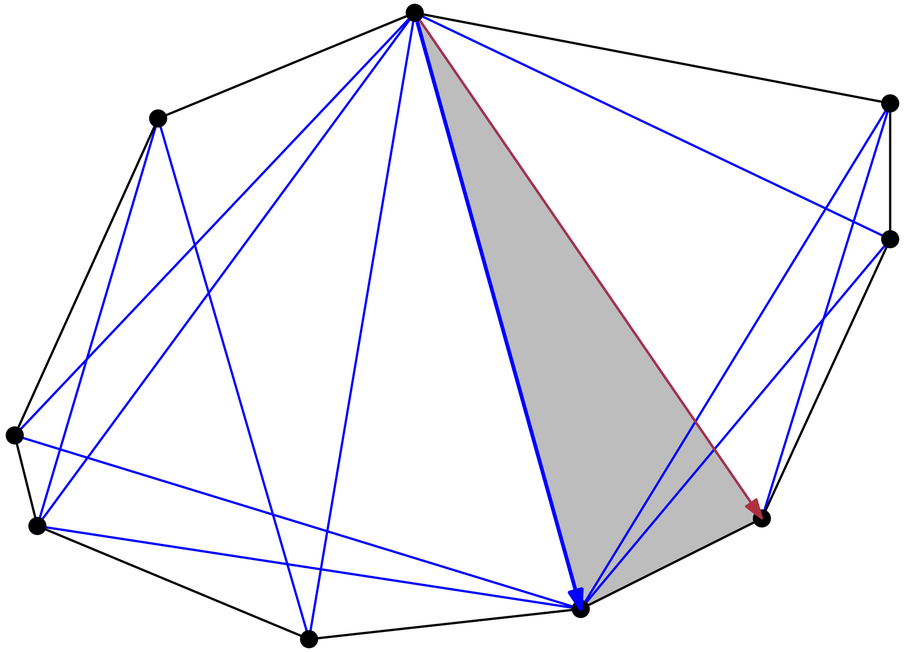
Dynamic Program

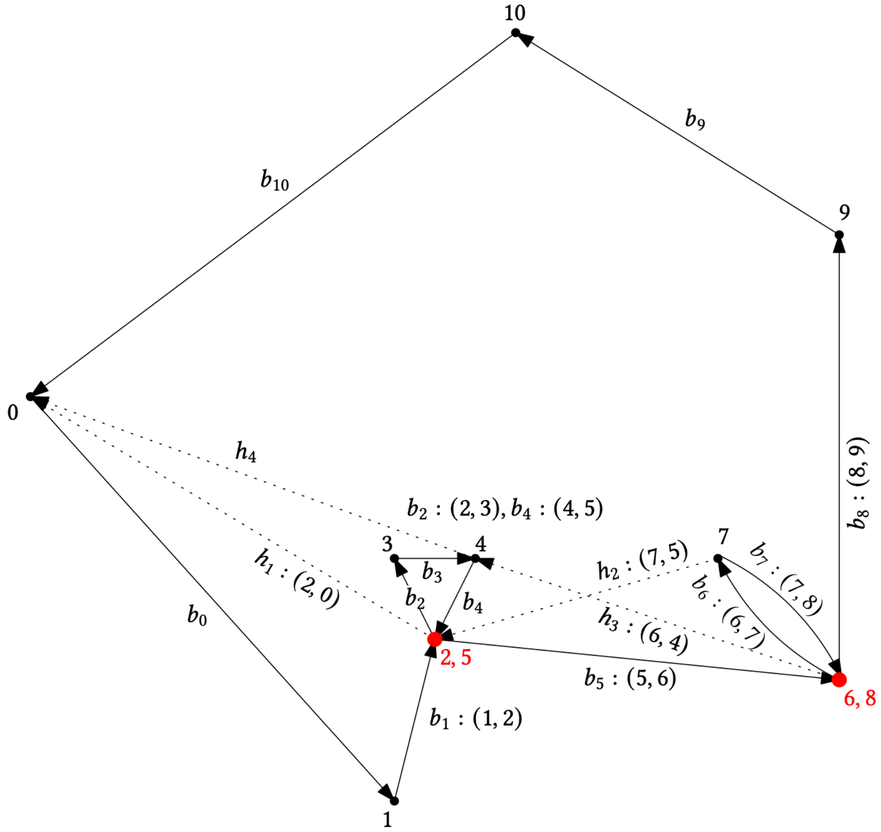
“simple hole”



Challenges

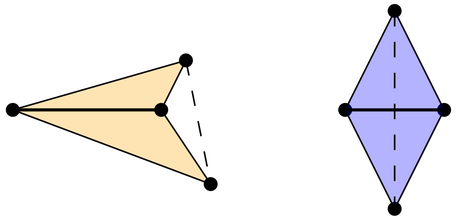
Well-known, classical DP, some caveats.



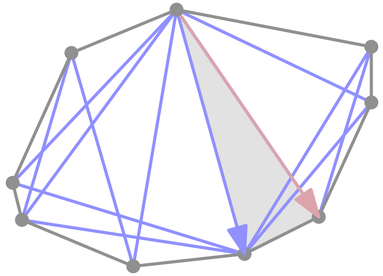


Outline

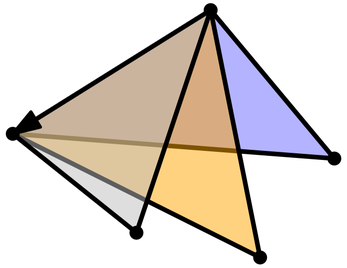
LMT



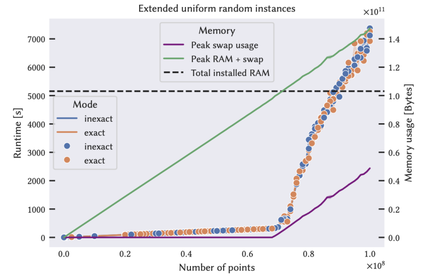
DP



LP & IP

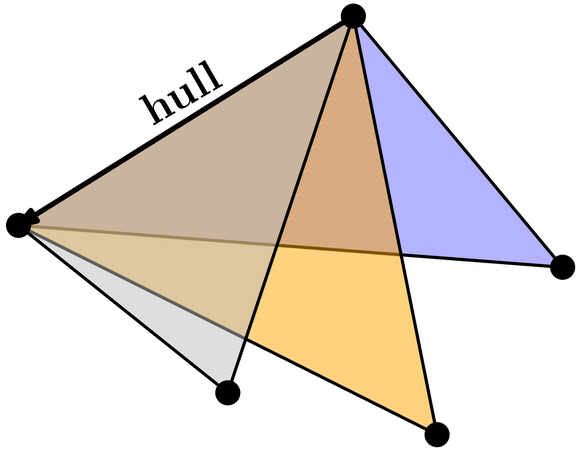


EXPERIMENTS

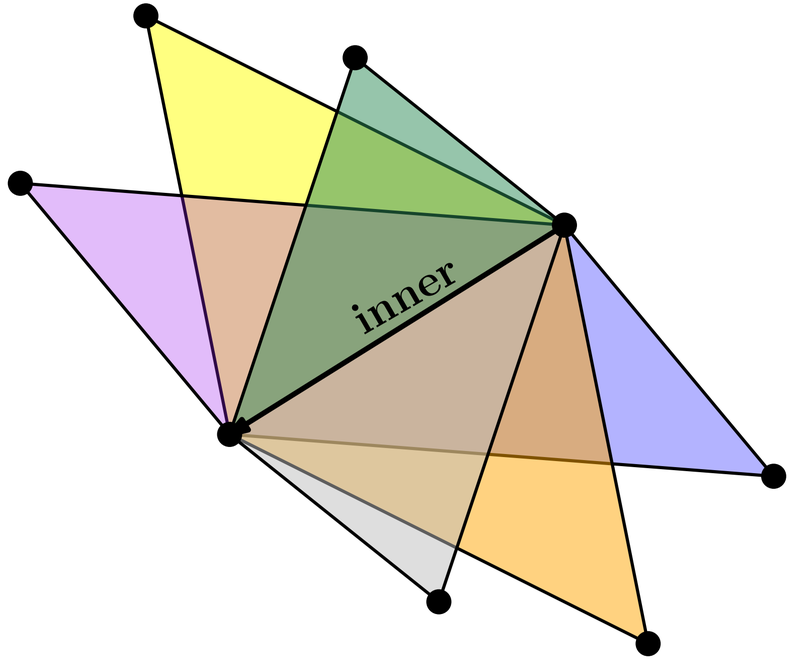


Linear Program (LP)

Well-behaved instances: Diamonds + LMT + DP normally sufficient
Non-simple holes: LP (and possibly IP)
Formulation: One variable $x_\Delta \geq 0$ per empty triangle Δ



$$x_{\triangle} + x_{\triangle} + x_{\triangle} = 1$$



$$x_{\triangle} + x_{\triangle} + x_{\triangle} - x_{\triangle} - x_{\triangle} - x_{\triangle} = 0$$

Linear Program (LP): Option B

B. Solve the LP inexactly, give good error bounds

primal

$$\min \sum_{\Delta} c_\Delta x_\Delta \text{ s.t.}$$

$$\forall e \in H: \sum_{\Delta \in L(e)} x_\Delta = 1$$

$$\forall e \notin H: \sum_{\Delta \in L(e)} x_\Delta - \sum_{\Delta \in R(e)} x_\Delta = 0$$

$$x_\Delta \geq 0$$

$$x_\Delta \leq 1$$

dual

$$\max \sum_{e \in H} z_e + \sum_{\Delta} y_\Delta \quad \text{s.t.}$$

$$\forall \Delta: \sum_{e:\Delta \in L(e)} z_e - \sum_{e:\Delta \in R(e)} z_e + y_\Delta \leq c_\Delta$$

$$z_e \text{ free}$$

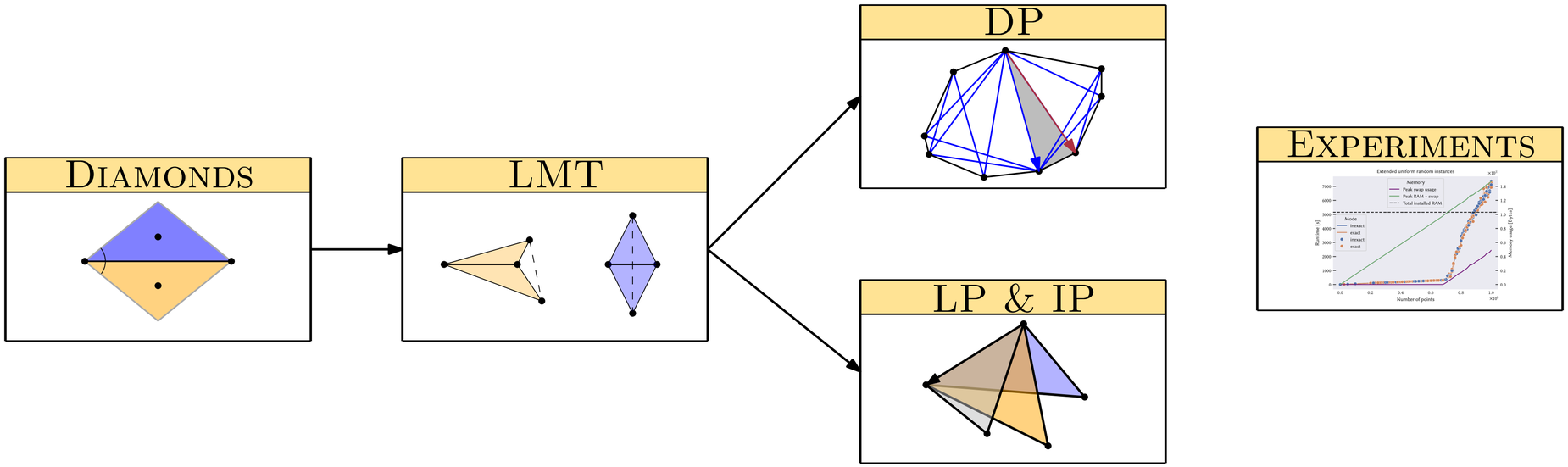
$$y_\Delta \leq 0$$

Easy fix for dual infeasibility!

When all else fails: IP

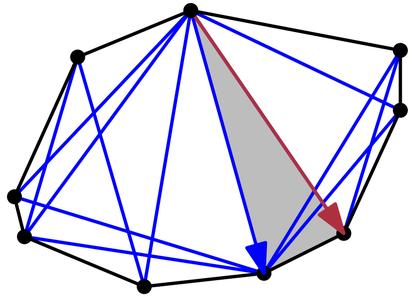
Non-integral LP relaxation: use IP instead (inexact).

Outline

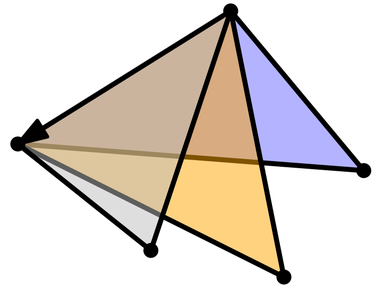

DIAMONDS
LMT
DP
LP & IP
EXPERIMENTS
Extended uniform random instances
Memory
Peak swap usage
Peak RAM + swap
Total installed RAM
Mode
inexact
exact
inexact
exact
Runtime (s)
Memory usage [Bytes]
Number of points

Outline

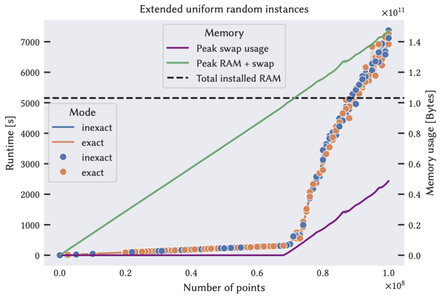
DP



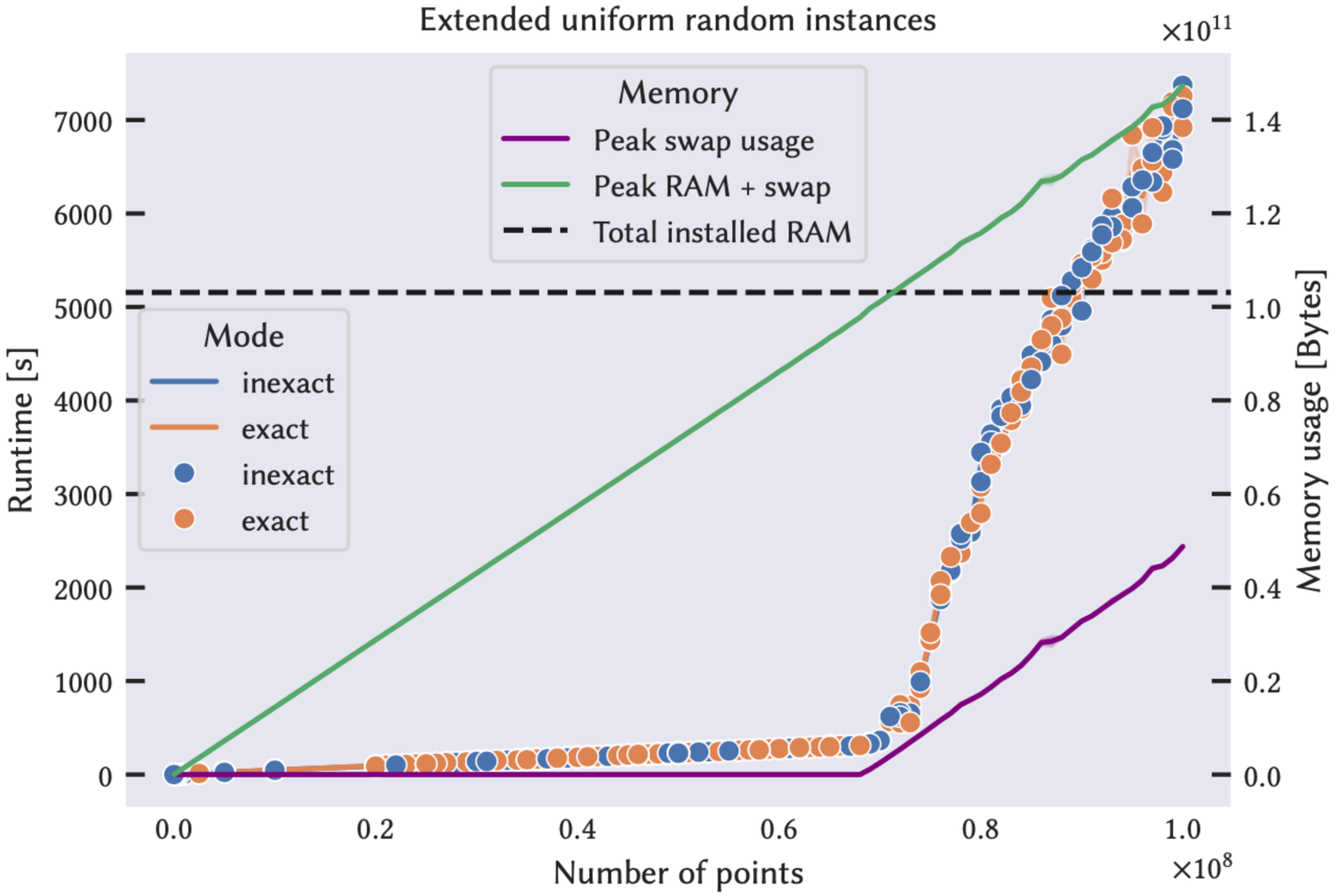
LP & IP



Experiments

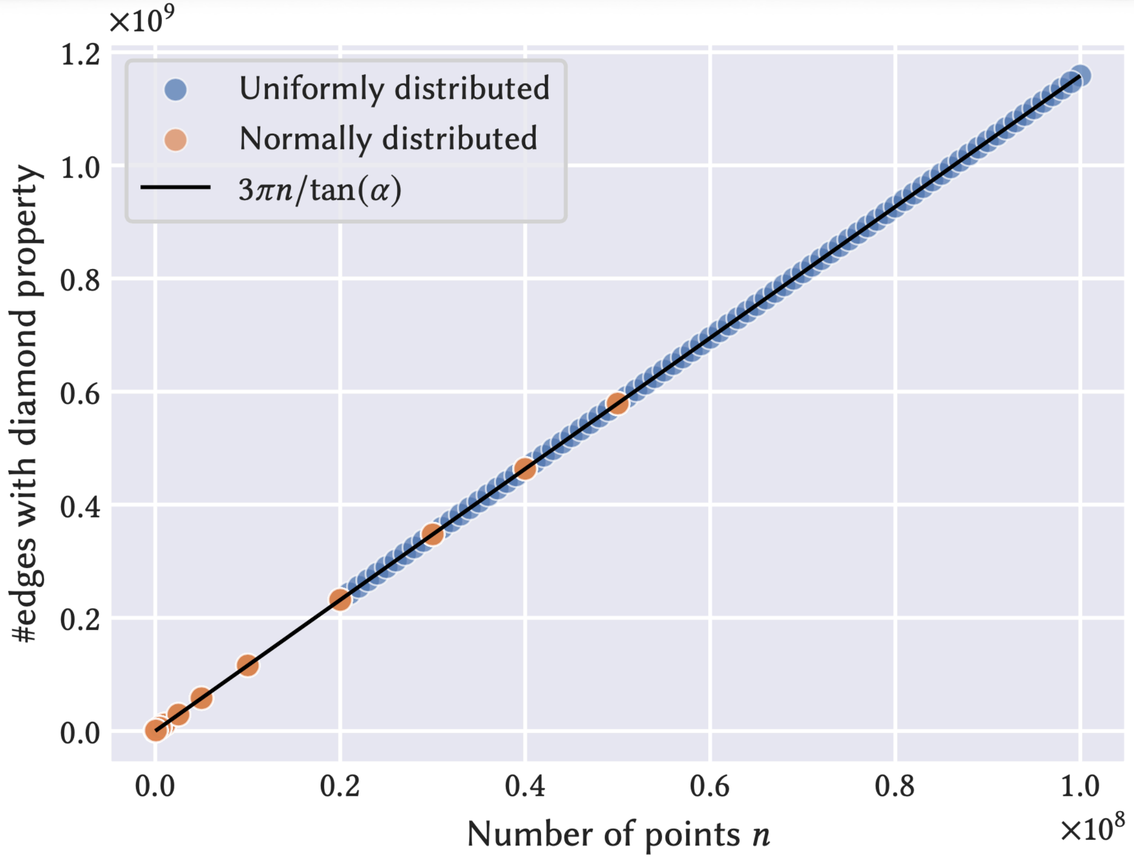


Larger Random Instances

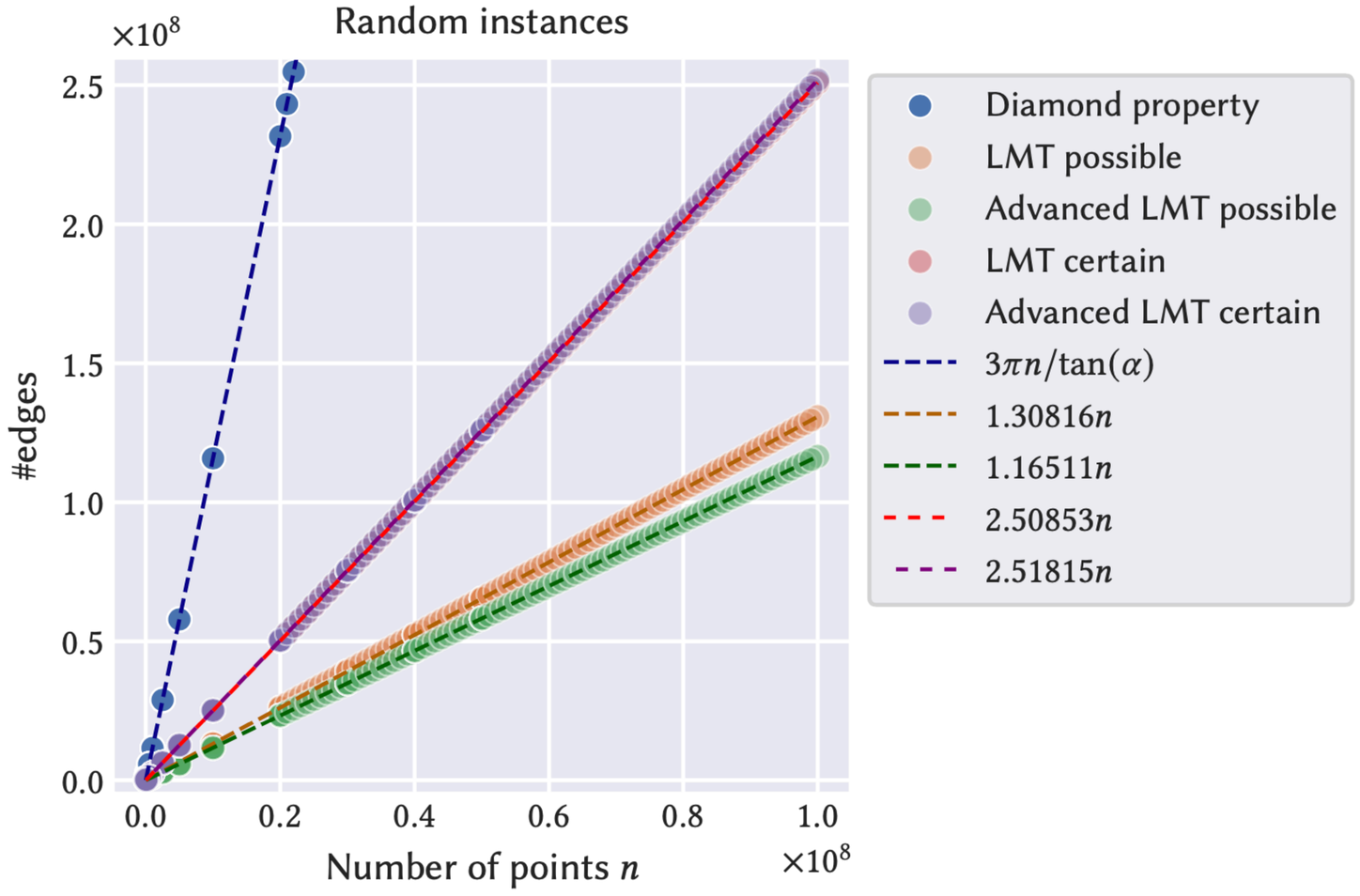


Number of Diamond Edges

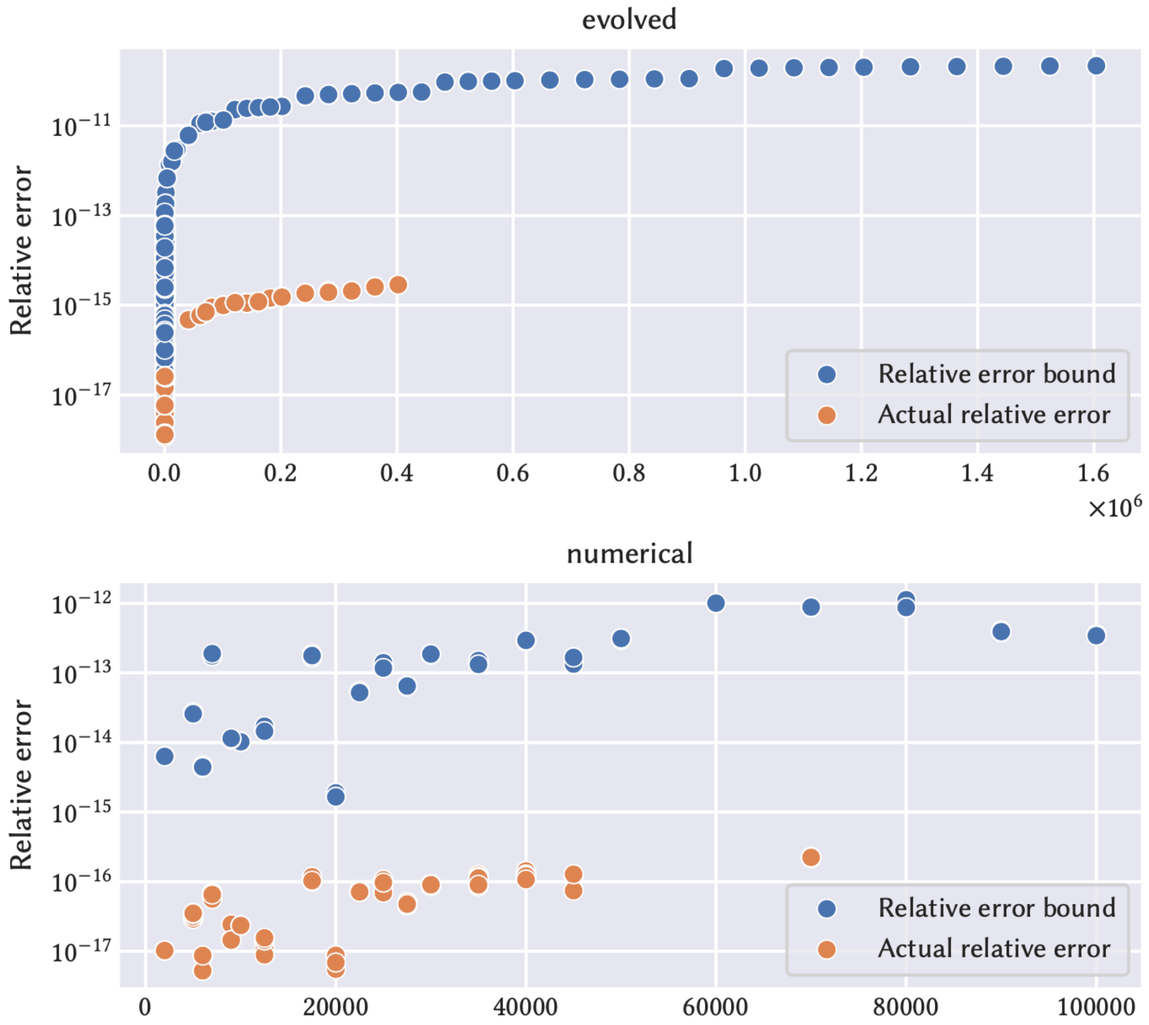
THEOREM 2.1. *Let P be a uniformly distributed point set in the plane with $|P| = n$ and let $\alpha \leq \pi/3$ be the base angle for the diamond property. Then the expected number of edges that pass the diamond test is less than $3\pi n/\tan(\alpha)$.*



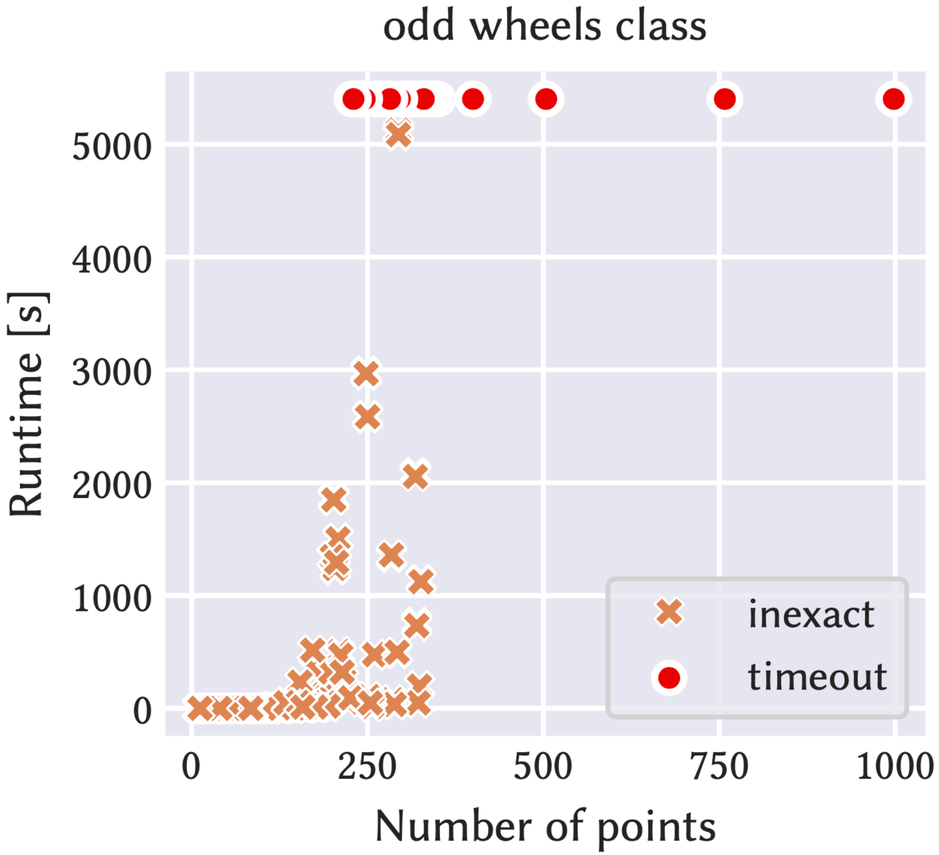
Number of Surviving Edges

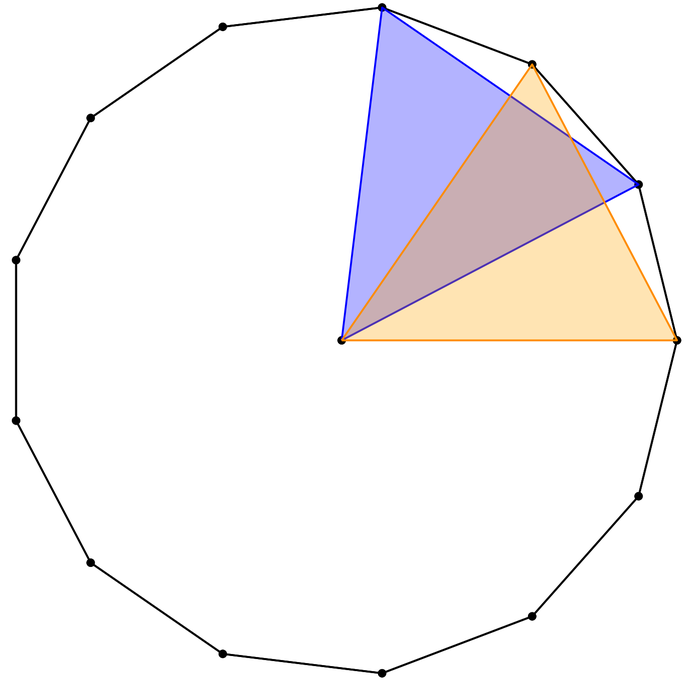


Errors and Error Bounds



IP

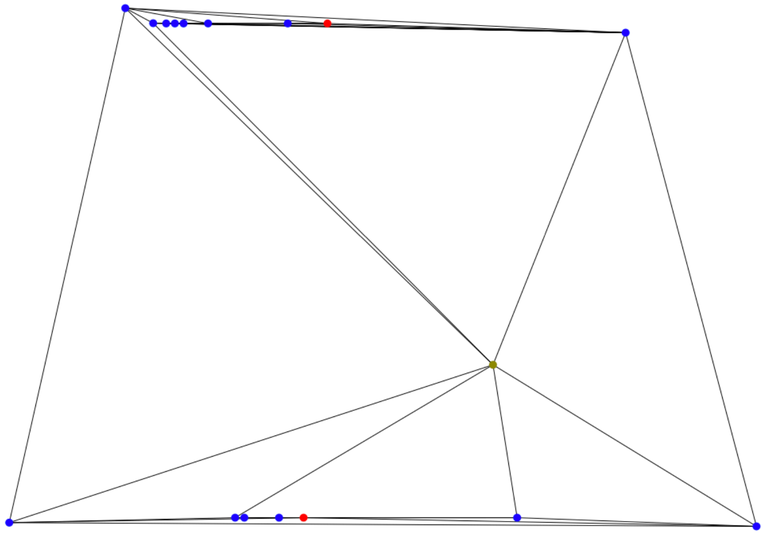


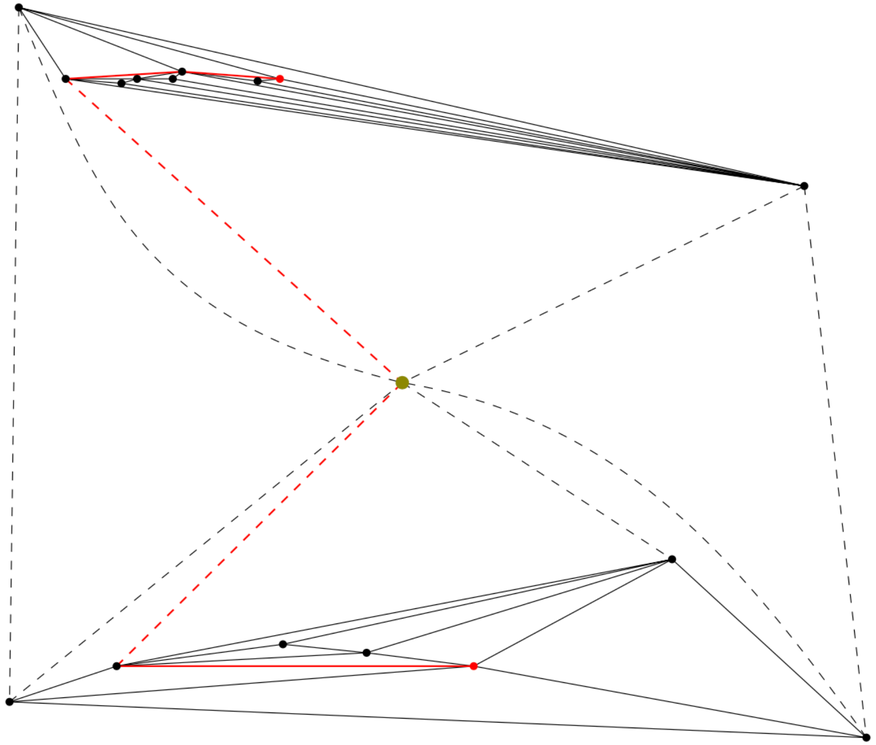


regular 13-gon + center

Dilation of the MWT

THEOREM 7.1. *The worst-case stretch factor of the MWT is at least 2.070148.*



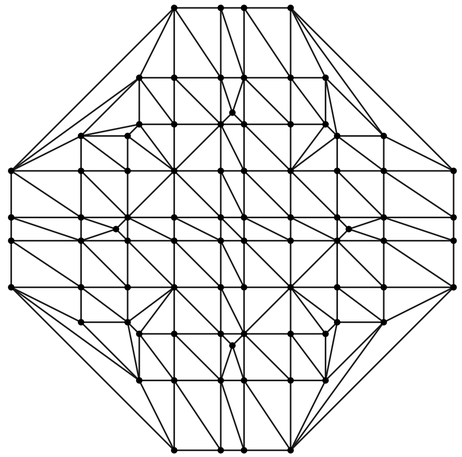


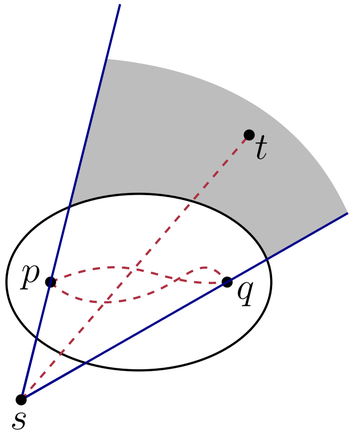
4 Minimum-Dilation Triangulations

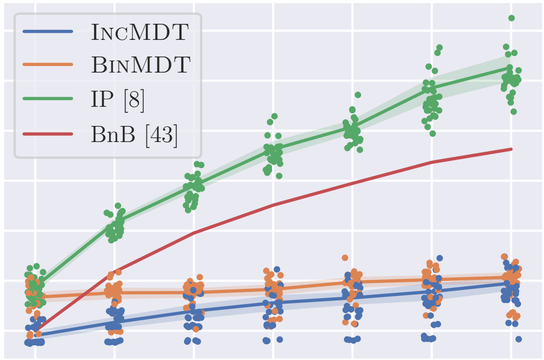
Technische Universität Braunschweig

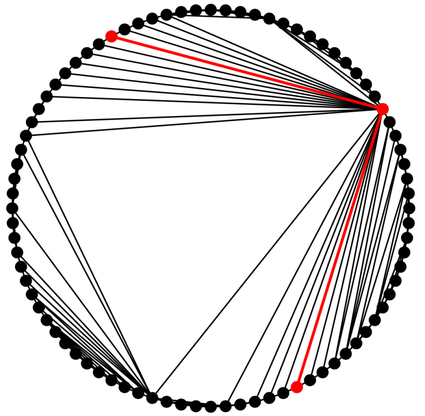
Institute of Operating Systems and Computer Networks
Algorithms

Exact Algorithms for Minimum Dilation Triangulation







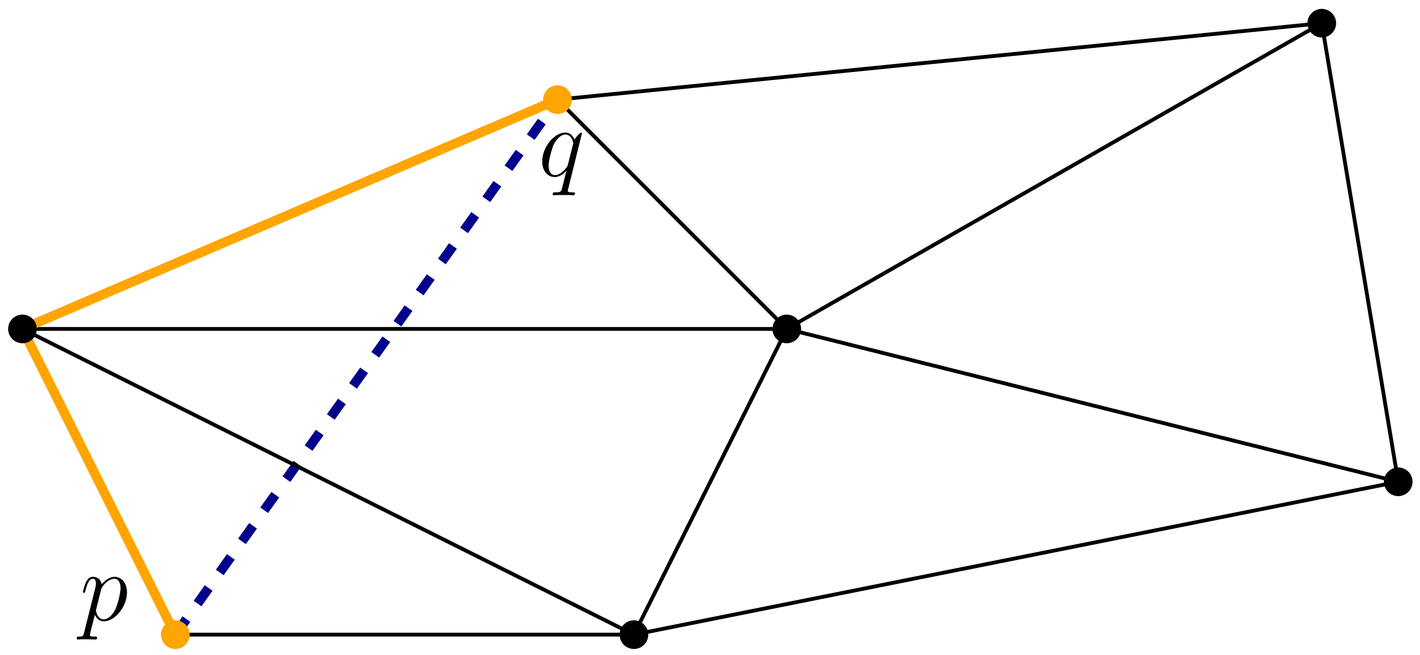








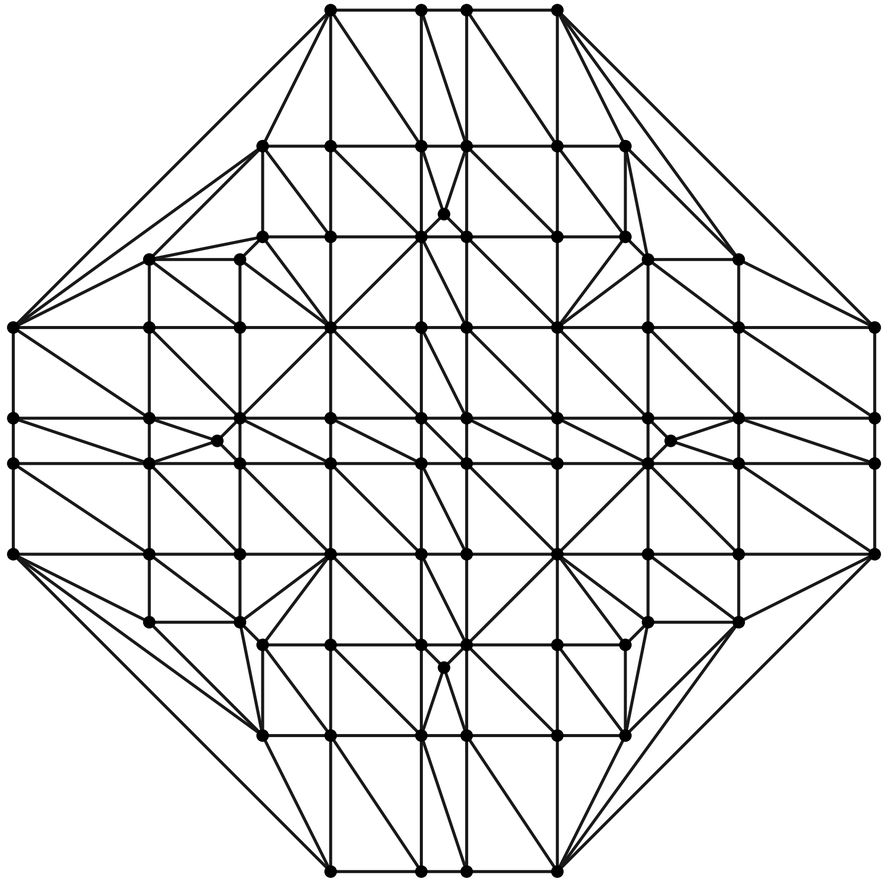
Dilation



$$\mathrm{dil}_G(p,q) = \frac{\mathrm{dist}_G(p,q)}{\|pq\|_2}$$

$$\mathrm{dil}(G) = \max_{p \neq q \in V} \left\{ \mathrm{dil}_G(p,q) \right\}$$

Minimum Dilation Triangulation (MDT)



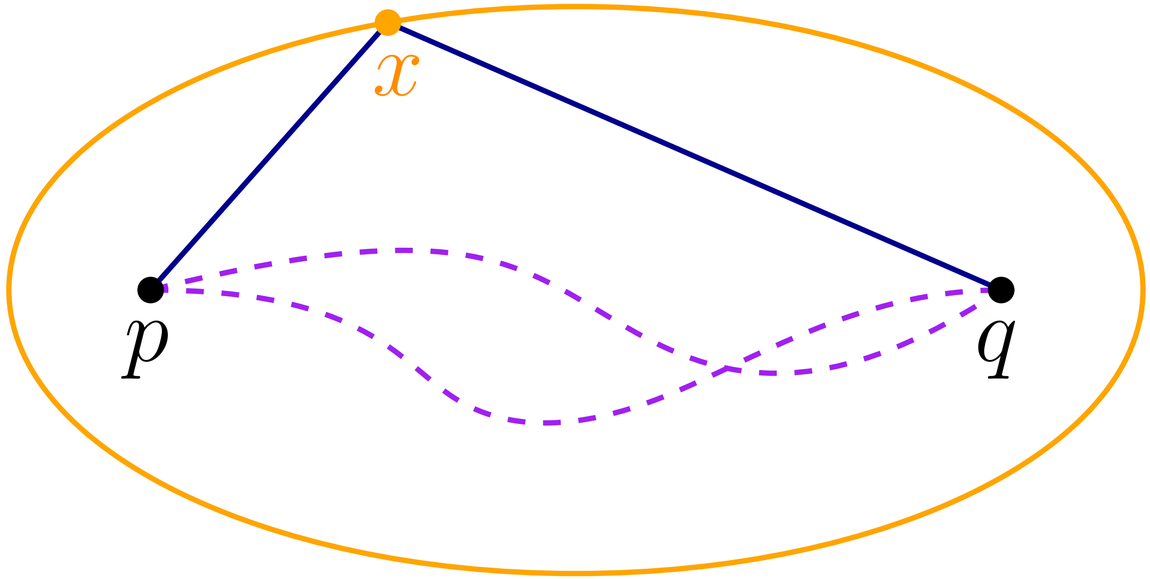
complexity: no PTIME-algorithm known, no NP-hardness proof

4.1: Edge candidates

Ellipse Property

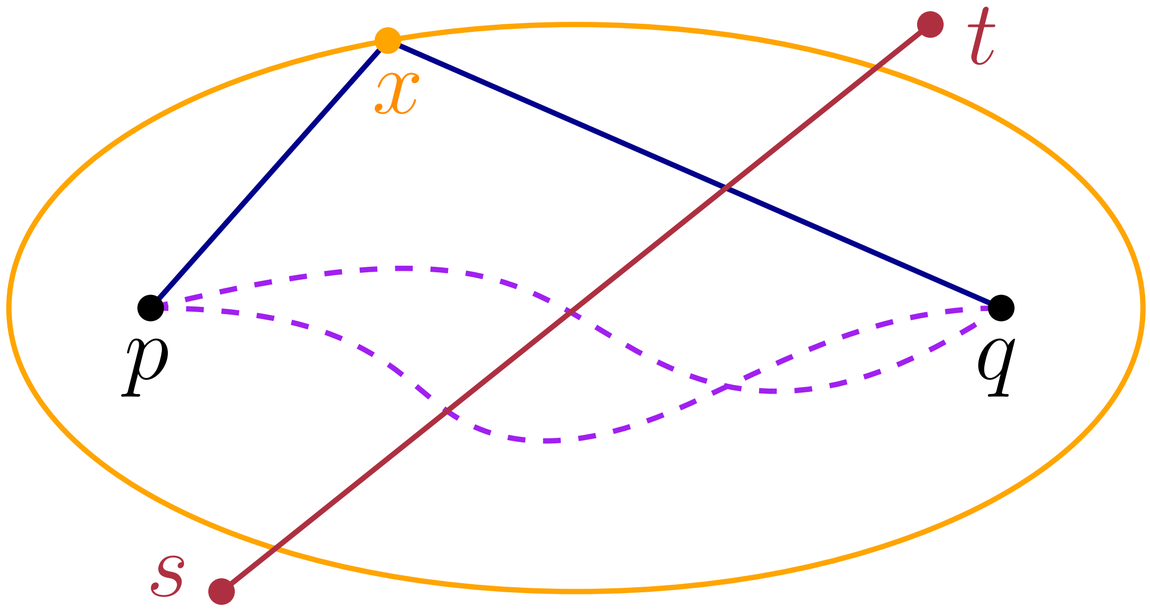
existing solution (e.g., Delaunay): dilation ρ

$$\|xp\|_2 + \|xq\|_2 = \rho \cdot \|pq\|_2$$



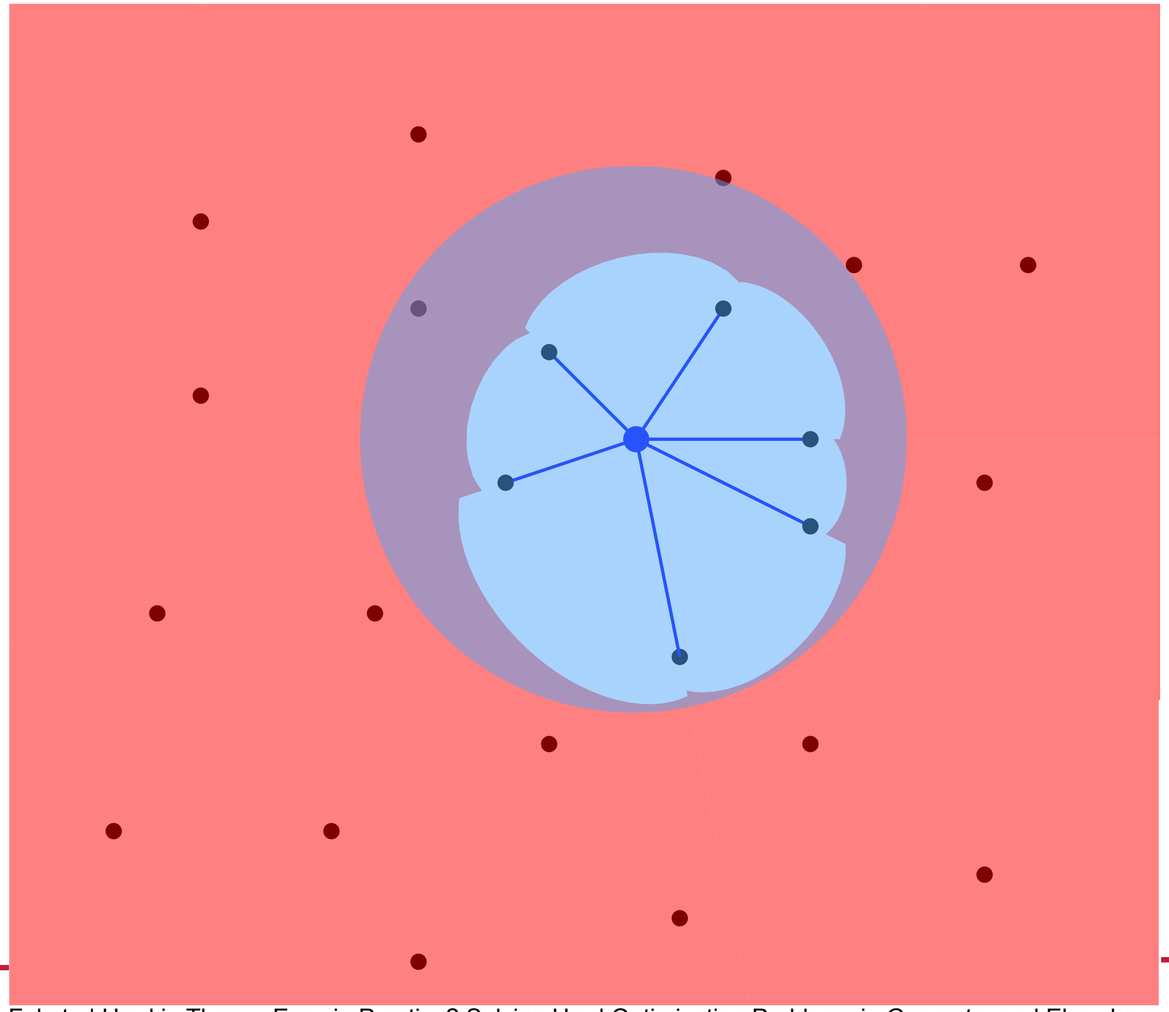
Ellipse Property

existing solution (e.g., Delaunay): dilation ρ



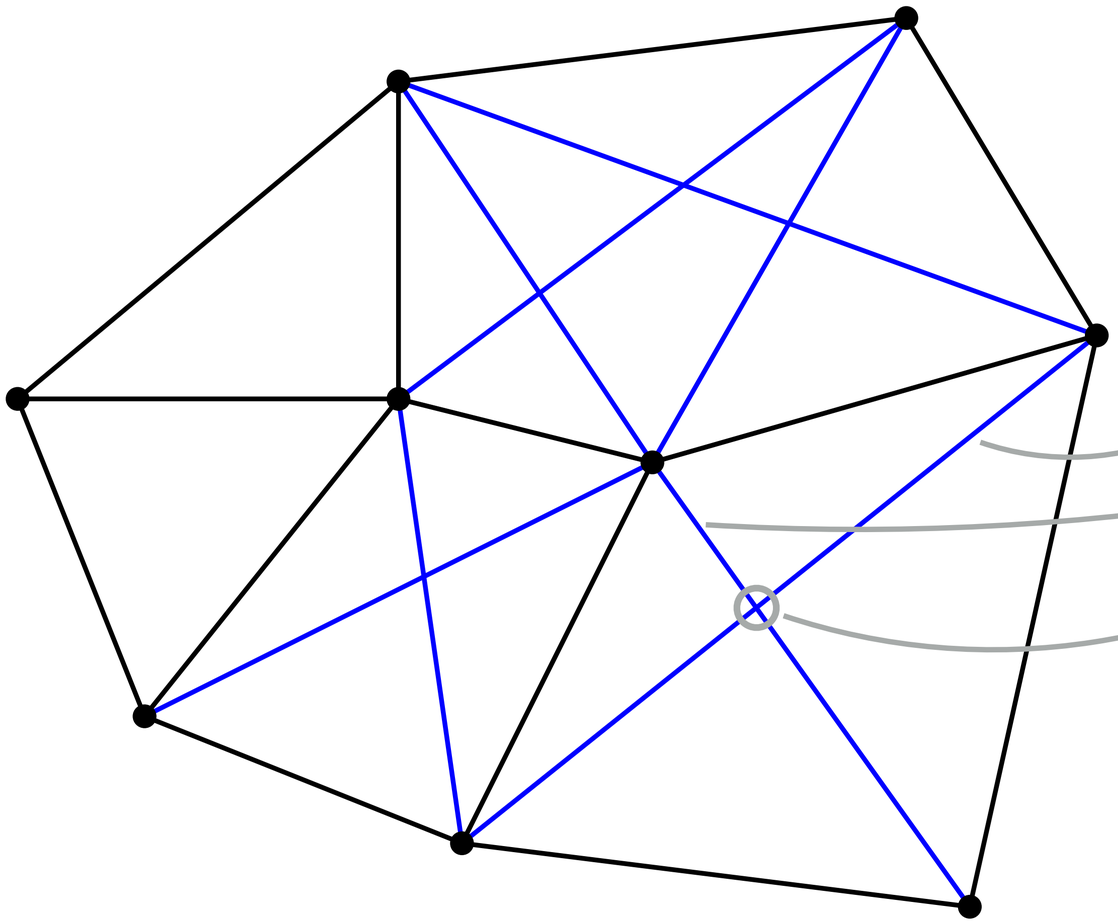
Candidate Edge Enumeration

Idea: enumerate superset of neighbors from each source point s



4.2: SAT model & optimization scheme

SAT Model



Edge variable x_e

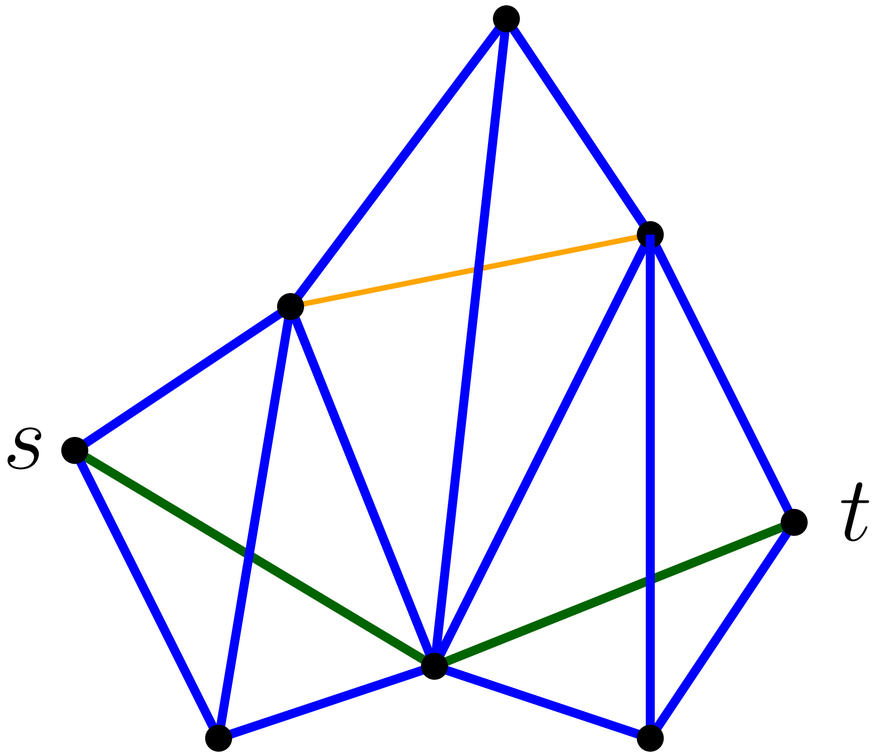
Clause: e is present or crossed

$x_e \vee x_f$

No crossing edges:

$\overline{x_e} \vee \overline{x_f}$

Separation Problem



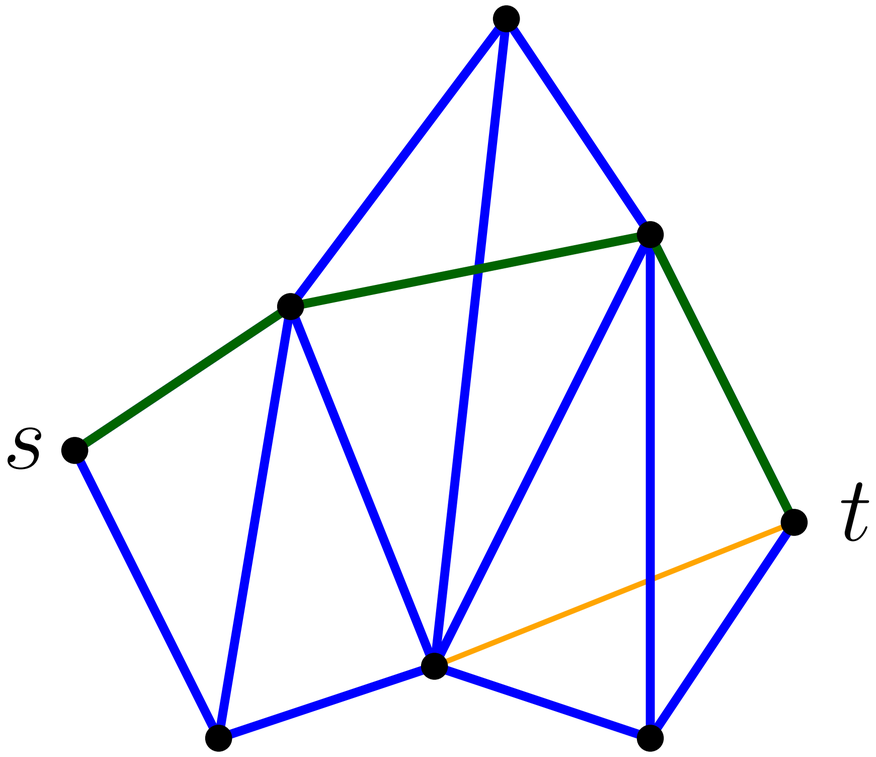
Given: Bound $\rho > 1$ and triangulation T,
$\mathrm{dil}(T) \geq \rho$
(say s, t with $\mathrm{dil}_T(s,t) \geq \rho$)

Find clause C:

- Violated by T
- Satisfied by any triangulation U with $\mathrm{dil}(U) < \rho$

C: unused edges cutting all short $s-t$-paths

Separation Problem



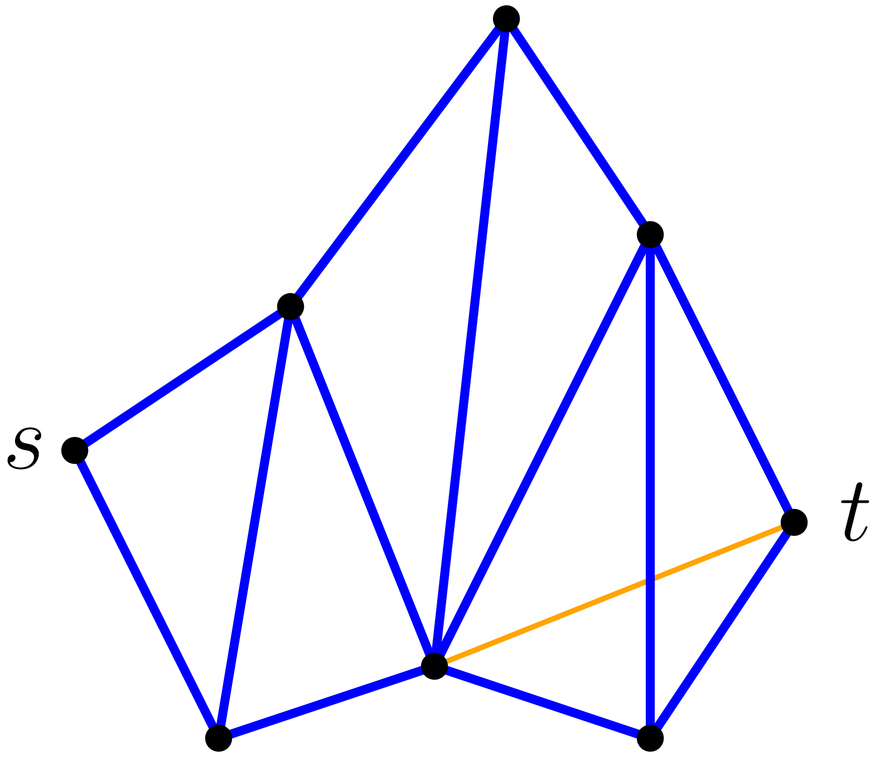
Given: Bound $\rho > 1$ and triangulation T, $\mathrm{dil}(T) \geq \rho$
(say s, t with $\mathrm{dil}_T(s,t) \geq \rho$)

Find clause C:

- Violated by T
- Satisfied by any triangulation U with $\mathrm{dil}(U) < \rho$

C: unused edges cutting all short $s-t$-paths

Separation Problem



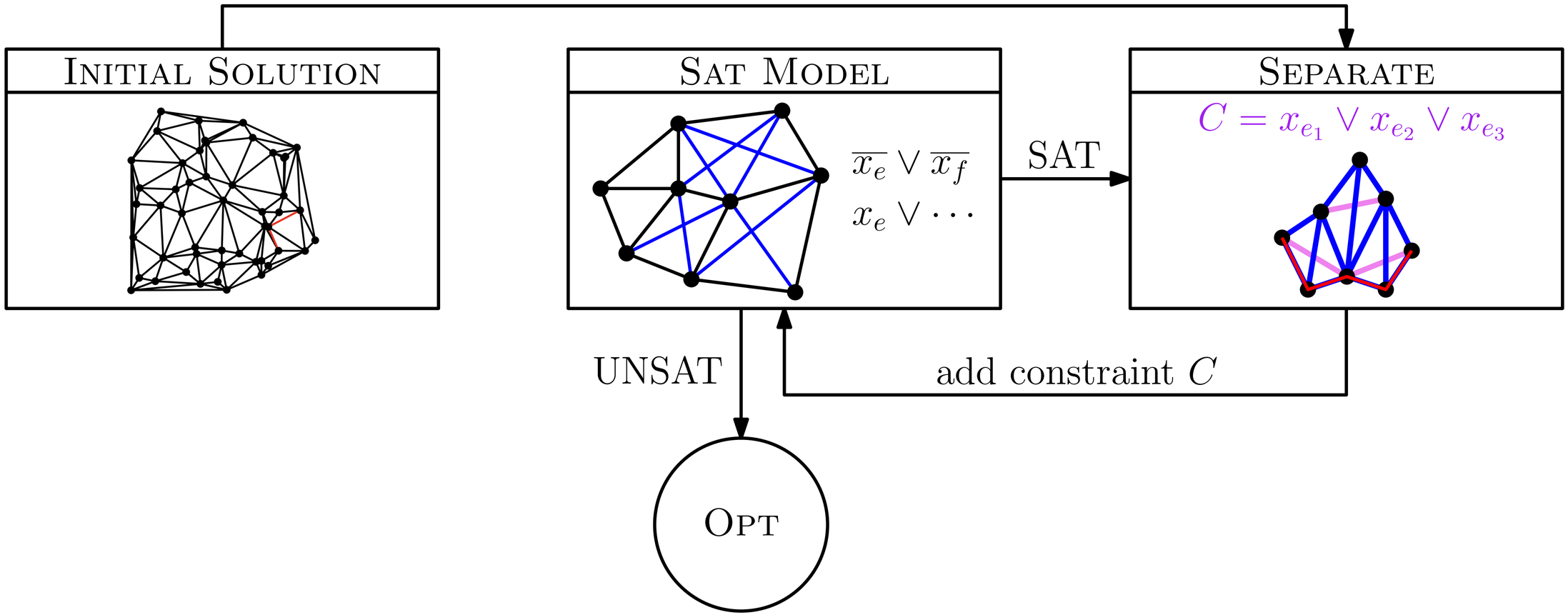
Given: Bound $\rho > 1$ and triangulation T, $\mathrm{dil}(T) \geq \rho$
(say s, t with $\mathrm{dil}_T(s,t) \geq \rho$)

Find clause C:

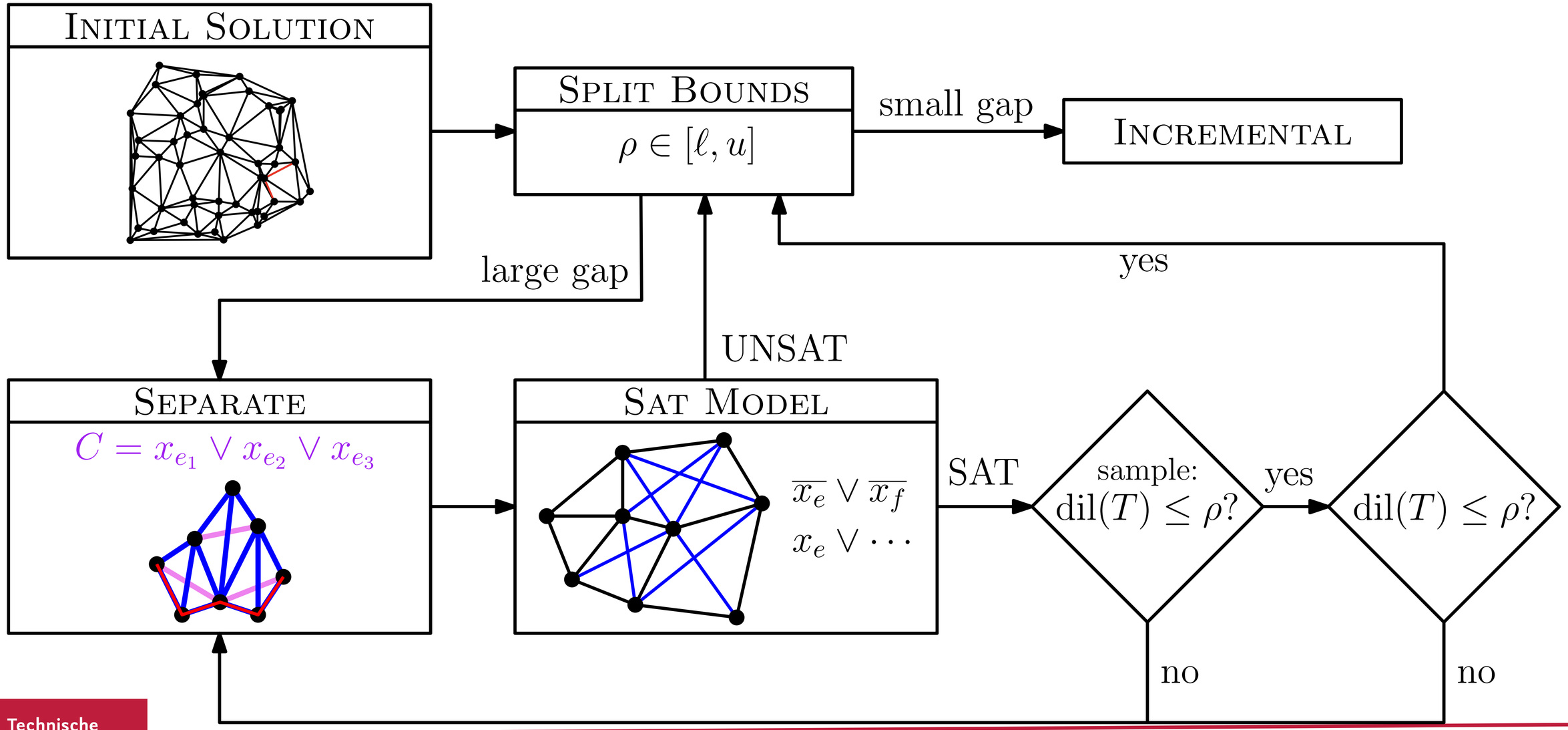
- Violated by T
- Satisfied by any triangulation U with $\mathrm{dil}(U) < \rho$

C: unused edges cutting all short $s-t$-paths

Incremental Approach

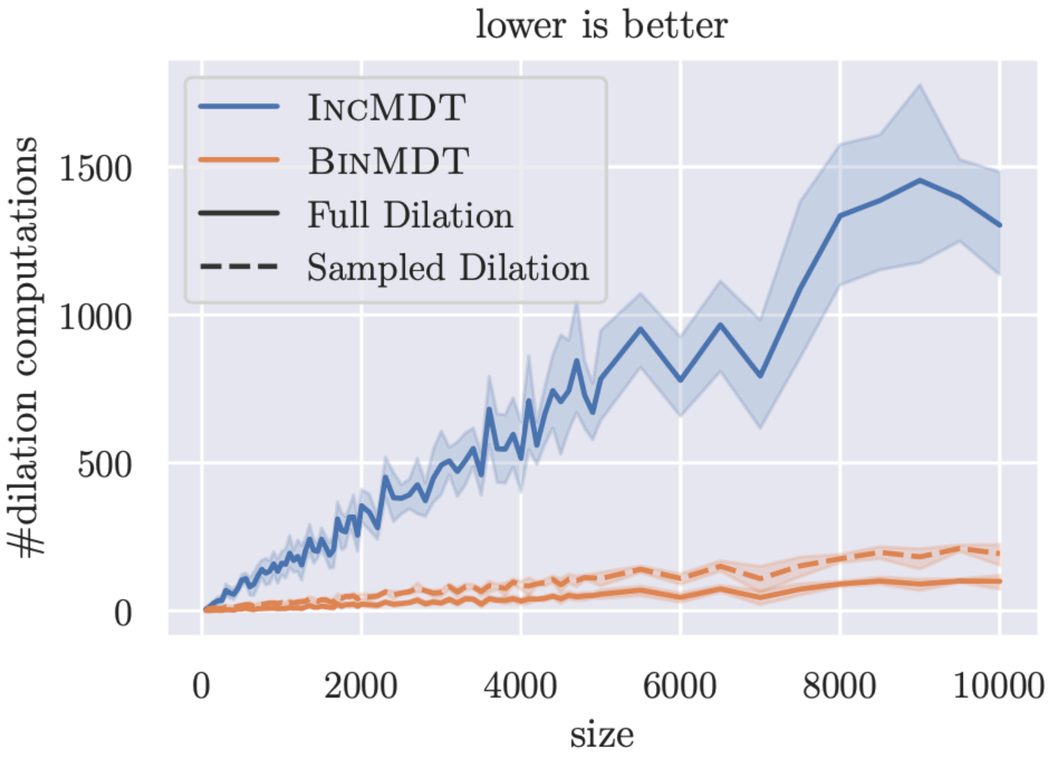


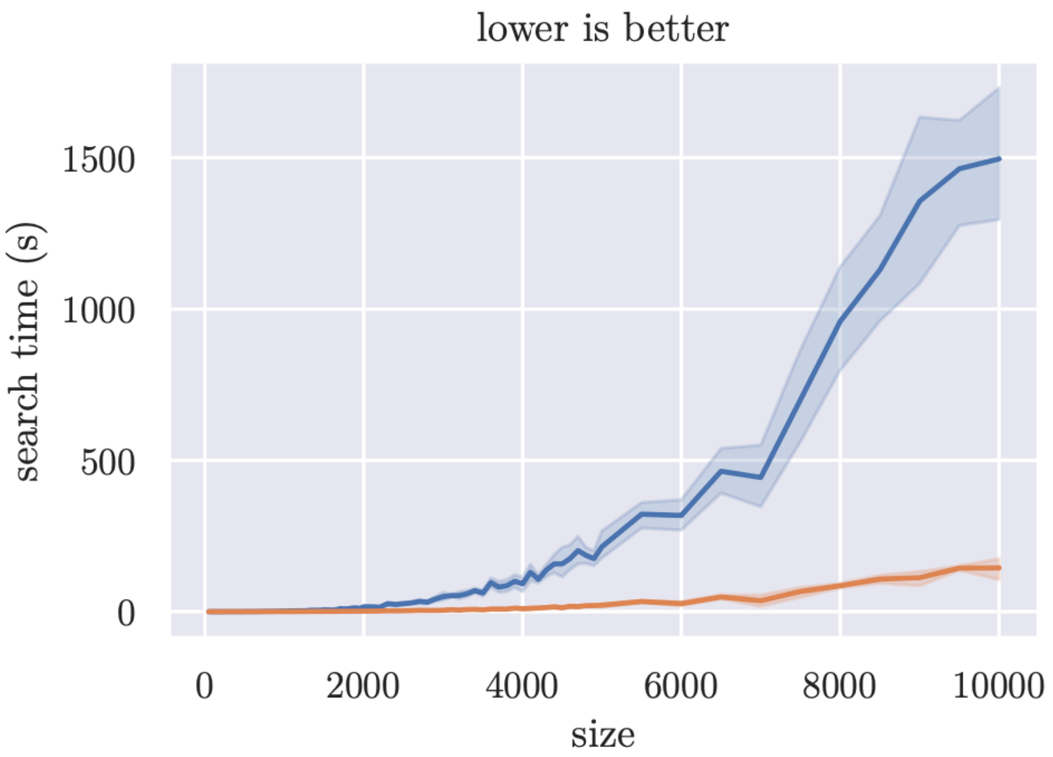
Binary Search Approach



4.3: Experiments

Experiments

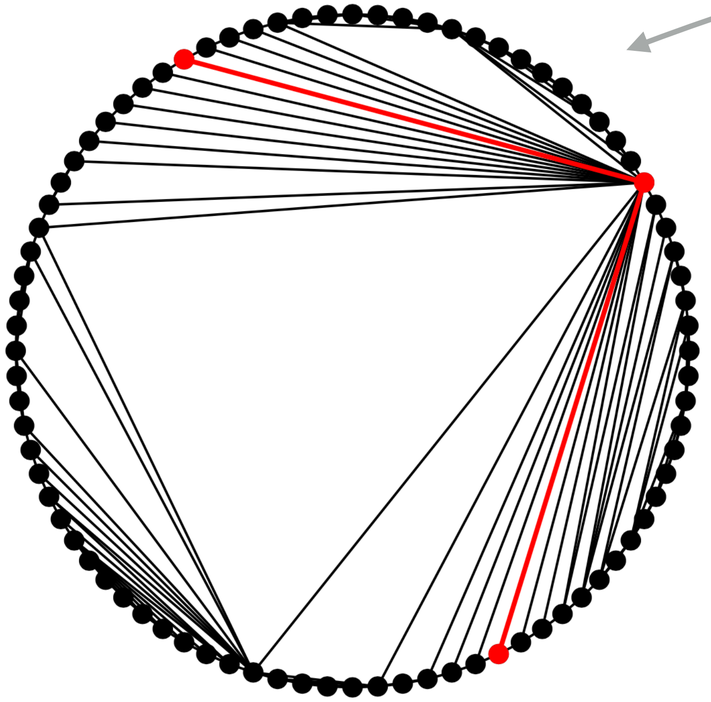


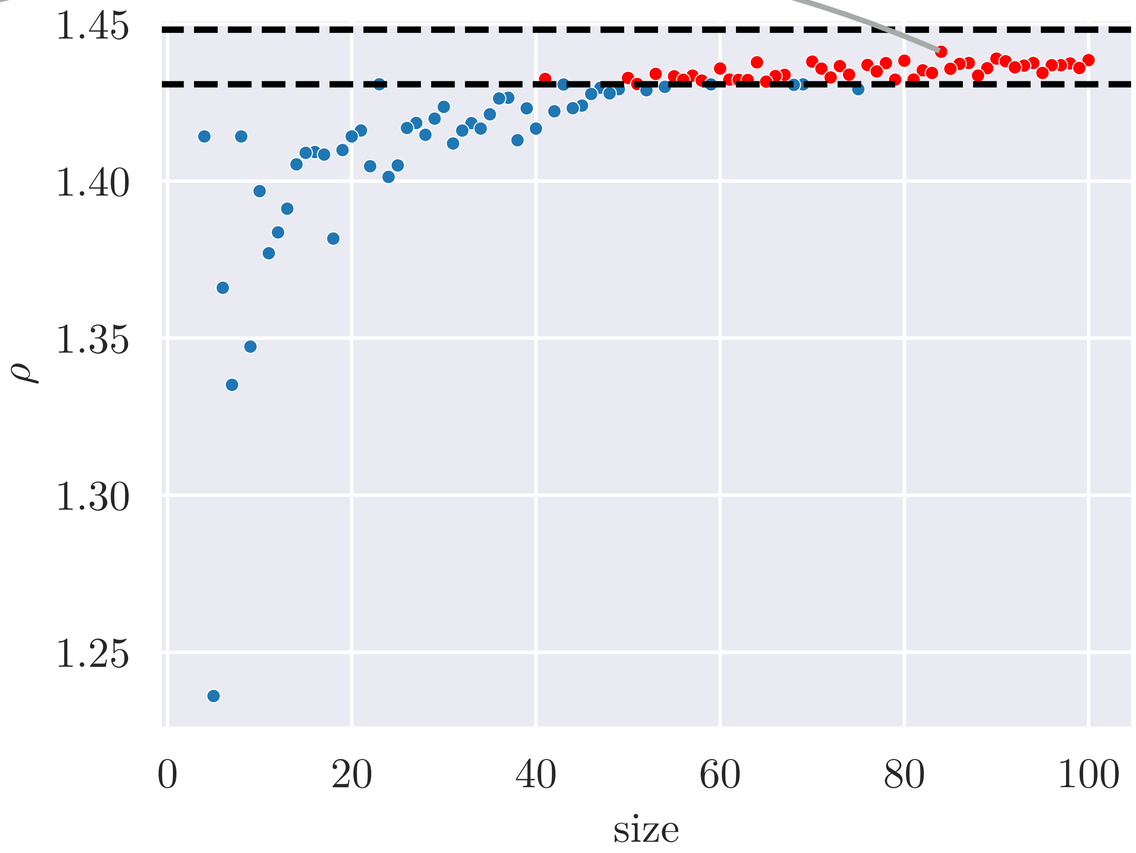


Experiments

▶ **Theorem 6.** *Let P be a set of $n = 84$ points placed at the vertices of a regular n-gon. Then the dilation of the MDT of P is $\rho \geq 1.44116645381$.*

Previous bounds:
[1.4308, 1.4482]





6 Quantum Computing

Quantum Computing

bdr.

Group Solutions Fields of use Innovation hub

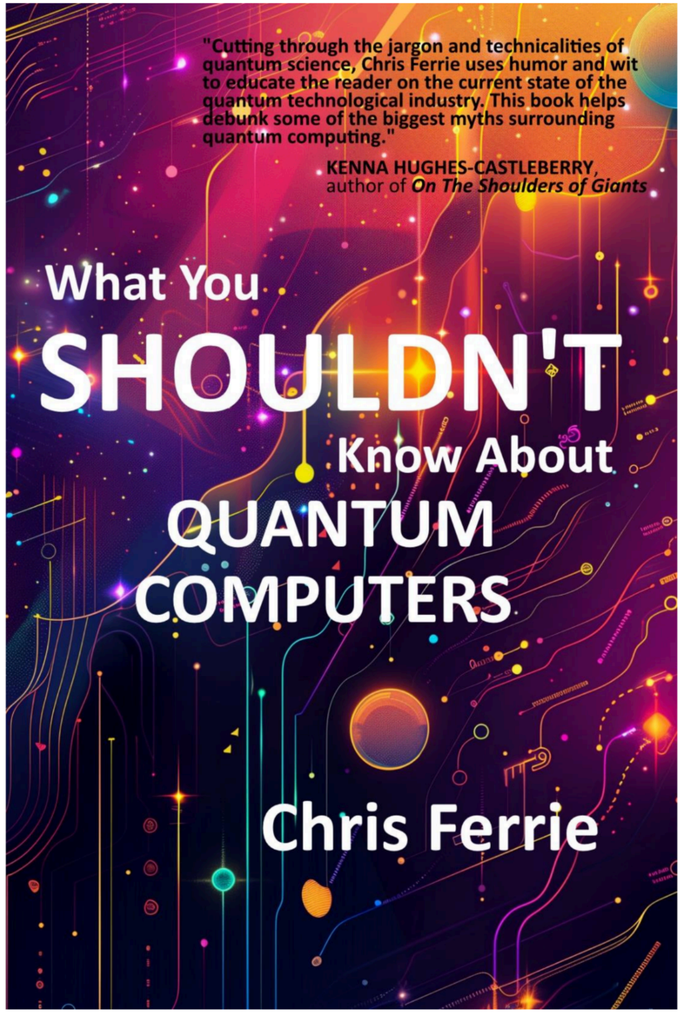


Quantum computing – new rules thanks to supercomputers

published on 15.03.2022

With these computers, everything centres around a quantum effect, i.e., the so-called superposition in combination with smart parallel information processing on the qubits of the quantum computer. Roughly speaking, superposition means that parallel computing is possible. This is precisely what distinguishes quantum computers from conventional computers that process information sequentially, one at a time, like pearls being strung on a necklace. A quantum computer, on the other hand, can calculate all values parallel, which would mean an exponential increase in performance.

Quantum Computing



Myth 2: Qubits Are Zero And One At The Same Time

Myth 3: Quantum Computers Try All Solutions At Once

Myth 5: Quantum Computers Will Replace Digital Computers

Myth 6: Quantum Computers Will Break the Internet

Projects

Bundesministerium für Wirtschaft und Klimaschutz

ProvideQ

Quantum Readiness for Optimization Providers

Enabling the Enablers



Sándor Fekete

TU Braunschweig

Read more



David Gross

Universität zu Köln

Read more



Tobias Osborne

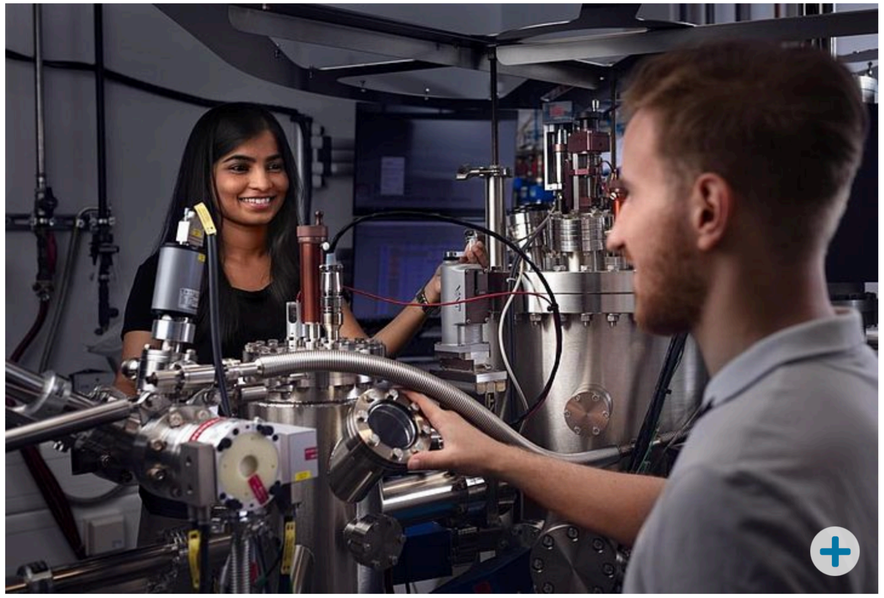
Leibniz Universität Hannover

Read more

BMBF quanten technologien

Projekt

QuBRA



Quantencomputing verspricht Durchbrüche in der Entwicklung innovativer Lösungen für industrielle Herausforderungen.

Quantum Simplex

$$\max_{x \in \mathbb{R}^n} \{c^T x \mid Ax \leq b,\ x \geq 0\}$$

Fast Quantum Subroutines for the Simplex Method

Giacomo Nannicini

Published Online: 18 Oct 2022 | https://doi.org/10.1287/opre.2022.2341

elements per column or row of the basis, basis condition number κ, and optimality tolerance ϵ, pricing can be performed in $\tilde{O}(\frac{1}{\epsilon}\kappa d\sqrt{n}(d_c n + dm))$ time, where the $\tilde{O}$ notation hides polylogarithmic factors; classically, pricing requires $O(d_c^{0.7} m^{1.9} + m^{2+o(1)} + d_c n)$ time in the worst case using the fastest known algorithm for sparse matrix multiplication. For well-conditioned sparse problems the quantum subroutines scale better in

Realistic Runtime Analysis for Quantum Simplex Computation*

Sabrina Ammann† Sándor P. Fekete‡ Paulina L. A. Goedicke§ David Gross‡
Maximilian Hess¶ Andreea Lefterovici‖ Tobias J. Osborne¶ Michael Perk†
Debora Ramacciotti¶ Antonio Rotundo¶ S. E. Skelton¶ Sebastian Stiller*
Timo de Wolff*

16 Nov 2023

Estimating Quantum Simplex

3 Hybrid Benchmarking

In analogy to classical runtime analysis, we are benchmarking the expected computational effort in the standard model for asymptotic quantum algorithm analysis. Thus we are not benchmarking the performance of some assumed quantum device. Instead, we focus on quantifying or estimating (under a very generous set of assumptions) the runtime of a quantum algorithm by counting the minimum expected number of elementary gates required to run it. This allows us to compare

LEMMA 4.1. (ITERATIONS FOR **QSearch**) *Let X be a list of length $|X|$, with t marked items. The expected number $n_Q(|X|,t)$ of iterations that* QSearch *needs to find a marked item is*

$$n_Q(|X|,t) = \sum_{k=1}^{k_{max}} \frac{m_k}{2}\left[\prod_{l=1}^{k-1}\frac{1}{2} + \frac{\sin(4(m_l+1)\theta)}{4(m_l+1)\sin(2\theta)}\right],$$

$$k_{max} = \left\lceil \log_\lambda \frac{|X|}{2\sqrt{|X|-1}} \right\rceil + 4\,,$$

with $\sin^2\theta = t/|X|$, $m_k = \lfloor \min(\lambda^k, \sqrt{|X|}) \rfloor$, $\lambda = 6/5$.

LEMMA 4.2. (**QLSA – Fourier**) *The gate complexity of* QLSA – Fourier *that solves the linear equation $Ax = b$ to precision ε has a lower bound of*

$$\mathcal{C}[\mathrm{QLS}\,(A,b,\varepsilon)] \geq 10tw\left(\frac{\pi}{2\arcsin(\alpha^{-1})}+1\right)\cdot (\|A\|_1 - d^2\gamma)\left(\left\lceil \log\left(\frac{\|A\|_1}{\gamma}-d^2\right)\right\rceil - 1\right),$$

with the smallest integer w satisfying $\frac{e^w}{w^w} \leq \frac{\varepsilon_{seg}^2}{2}$ and $\alpha, \gamma, \Delta_z, K, t, \varepsilon_{seg}$ defined as

$$\alpha = 2\sqrt{\pi}\frac{\kappa}{\kappa+1}\sum_{k=-K}^{K}|k|\Delta_z e^{-\frac{(k\Delta_z)^2}{2}}, \quad t = 2\sqrt{2}\kappa\log\left(1+\frac{8\kappa}{\varepsilon}\right),$$

$$\Delta_z = \frac{2\pi}{\kappa+1}\left[\log\left(1+\frac{8\kappa}{\varepsilon}\right)\right]^{-1/2}, \quad \varepsilon_{seg} = \frac{\varepsilon}{90\gamma t d^2\left\lceil \frac{\|A\|_{\max}}{\gamma}\right\rceil},$$

$$K = \left\lfloor \frac{\kappa+1}{\pi}\log\left(1+\frac{8\kappa}{\varepsilon}\right)\right\rfloor, \quad \gamma = \frac{\varepsilon}{\sqrt{2}d^3t}.$$

Here $||A||_{\max} := \max_{i,j}|A_{ij}|$ is the largest element of A in absolute value.

Estimating Quantum Simplex

$$\langle n_{\mathrm{QMin}_{finite}} \rangle = 3\lceil s_{\max} \rceil \sum_{s=0}^{n-m-1} \frac{n_Q(n-m,s)}{s+1},$$

where $s = \sin^2(\theta)|n-m|$, $n_Q(n-m,s)$ denotes the number of times we need to apply the Grover operator in QSearch and $s_{\max} = \log_3 1/\varepsilon$. Putting everything into Equation (B.11), we finally obtain that

$$\begin{aligned}\mathcal{C}[\mathrm{QStER}] \geq & \left(2^{n_{\varepsilon',\delta'}+1}-1\right) 3\lceil s_{\max}\rceil \\ & \sum_{s=0}^{n-m-1} \frac{n_Q(n-m,s)}{s+1} \\ & \cdot \mathcal{C}[\mathrm{QLS}(A_B, A_k, \varepsilon/c_{\max} 10\sqrt{2})].\end{aligned}$$

Now setting $\delta' = 1/4$ and $\varepsilon' \to \varepsilon/8\sqrt{3}\pi c_{\max}$ we obtain

$$\begin{aligned}\mathcal{C}[\mathrm{QStER}] \geq & \left(\frac{40\sqrt{3}\pi c_{\max}}{\varepsilon} - 1\right) 3\lceil s_{\max}\rceil \cdot \\ & \sum_{s=0}^{n-m-1} \frac{n_Q(n-m,s)}{s+1} \cdot \\ & \mathcal{C}[\mathrm{QLS}(A_B, A_k, \varepsilon/c_{\max} 10\sqrt{2})],\end{aligned}$$

completing the proof. □

bounds for the terms on r.h.s. are proven in [18]

$$\left|\frac{1}{x} - h_3(x)\right| \leq 2\kappa\left(\frac{1}{1-\exp\left(-2\pi^2/\Delta_z^2\right)} - 1\right),$$

$$\begin{aligned}\left|h_3(x) - h_2(x)\right| \leq & \kappa \exp\left(-\frac{(\kappa y_J)^2}{2}\right) \\ & + 2\kappa\left(\frac{1}{1-\exp\left[-\frac{1}{2}\left(2\pi/\Delta_z - y_J\right)^2\right]} - 1\right),\end{aligned}$$

$$\left|h_2(x) - h_1(x)\right| \leq \frac{4\kappa}{\sqrt{2\pi}} \int_{z_K}^{\infty} dz e^{-z^2/2},$$

$$\left|h(x) - h_1(x)\right| \leq 2\sqrt{2}\Delta_y .$$

These bounds are valid provided $|\Delta_y z_K/k| = O(\varepsilon)$. Then to ensure that $|1/x - h(x)| \leq \varepsilon$, it is sufficient to take

$$\Delta_y = \frac{\varepsilon}{16}\left[\log\left(1+\frac{8\kappa}{\varepsilon}\right)\right]^{-1/2},$$

$$J = \lfloor \frac{16\sqrt{2}\kappa}{\varepsilon} \log\left(1+\frac{8\kappa}{\varepsilon}\right) \rfloor,$$

$$\Delta_z = \frac{2\pi}{\kappa+1}\left[\log\left(1+\frac{8\kappa}{\varepsilon}\right)\right]^{-1/2},$$

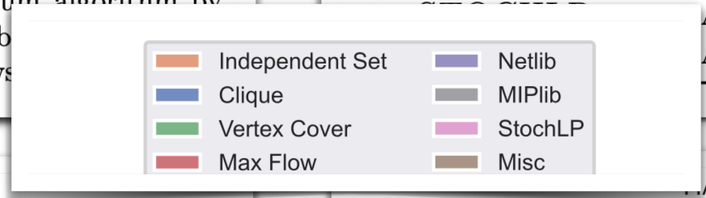
$$K = \lfloor \frac{\kappa+1}{\pi} \log\left(1+\frac{8\kappa}{\varepsilon}\right) \rfloor .$$

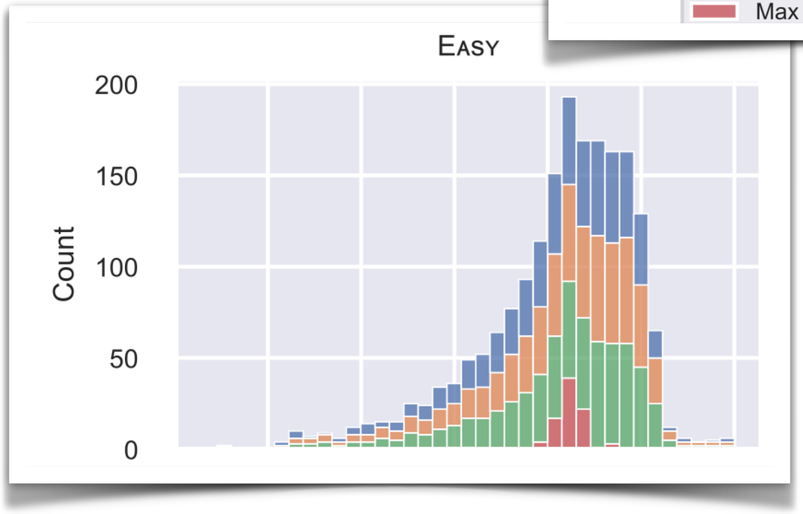
Estimating Quantum Simplex

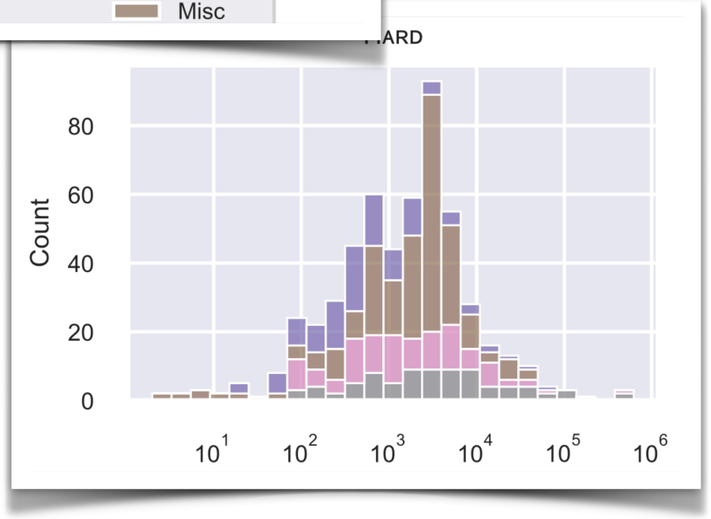
3 Hybrid Benchmarking

In analogy to classical runtime analysis, we are benchmarking the expected computational effort in the standard model for asymptotic quantum algorithm analysis. Thus we are not benchmarking the performance of some assumed quantum device. Instead, we focus on quantifying or estimating (under a very generous set of assumptions) the runtime of a quantum algorithm by counting the minimum expected numb
gates required to run it. This allows

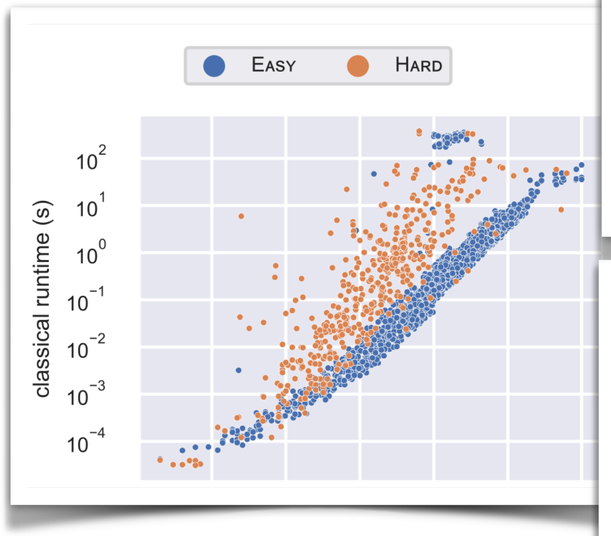
Benchmark	group	# Instances	Source
Random Graphs[5]	EASY	1,711	self generated
DIMACS Graphs[5]	EASY	101	[30]
Maximum Flow	EASY	88	[48] via [29]
NETLIB	HARD	109	[25]
MIPLIB	HARD	84	[27]
	ARD	108	[40]
	ARD	231	[39]

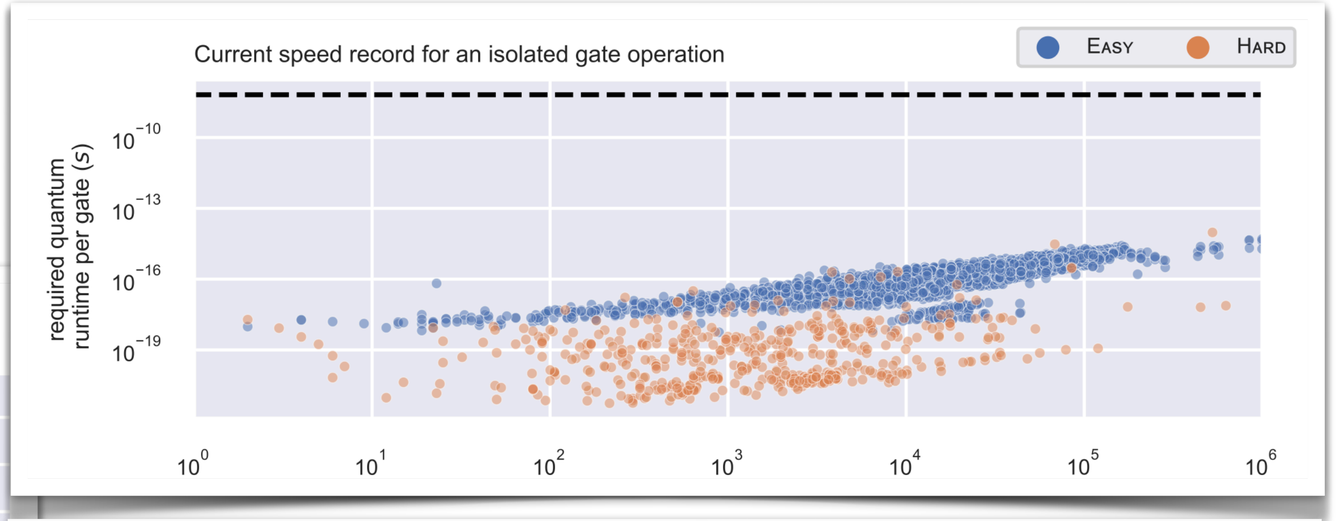


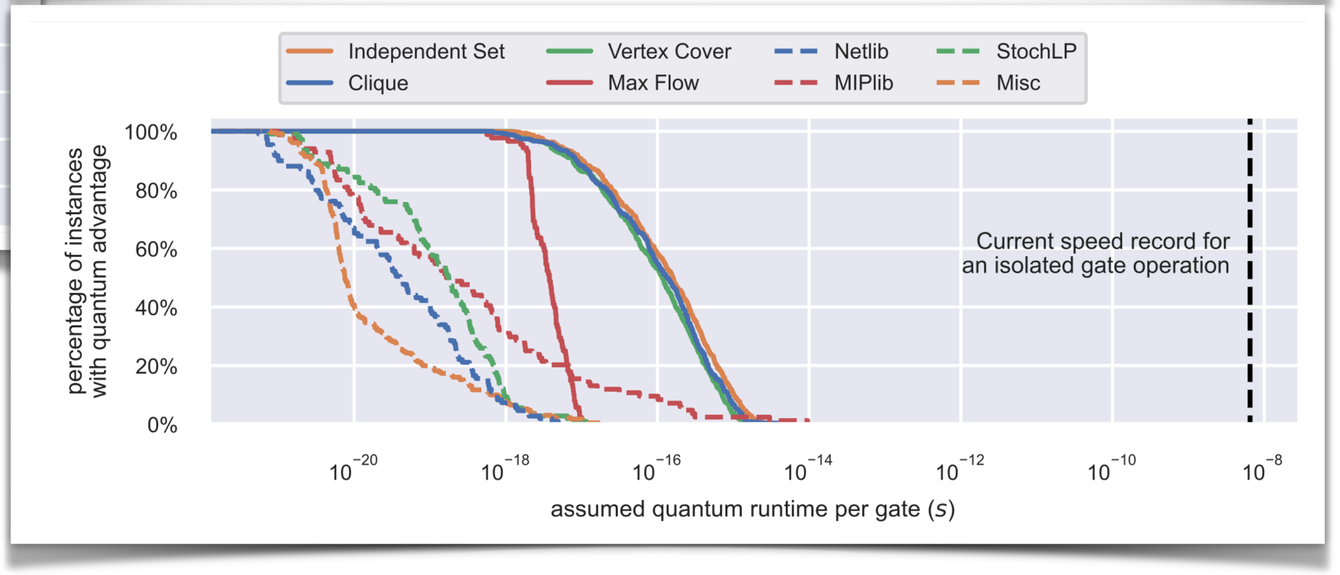




Bad News: Estimating Quantum Simplex







Good News: Knapsack Problems

npj | quantum information **Article**

Published in partnership with The University of New South Wales

https://doi.org/10.1038/s41534-025-01097-8

A quantum algorithm for solving 0-1 Knapsack problems

Check for updates

Sören Wilkening[1,2] ✉, Andreea-Iulia Lefterovici[1], Lennart Binkowski[1], Michael Perk[3], Sándor P. Fekete[3] & Tobias J. Osborne[1]

August 05, 2025

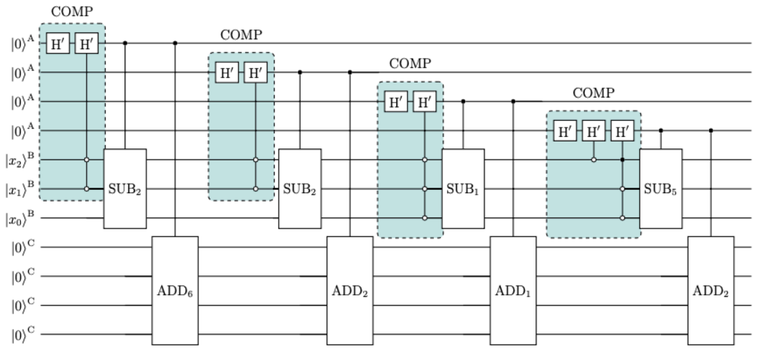
Box 1: Quantum Tree Generator

Input
Knapsack instance $\mathrm{KP}_n(\boldsymbol{p}, \boldsymbol{z}; Z)$, feasible assignment $\boldsymbol{y}$

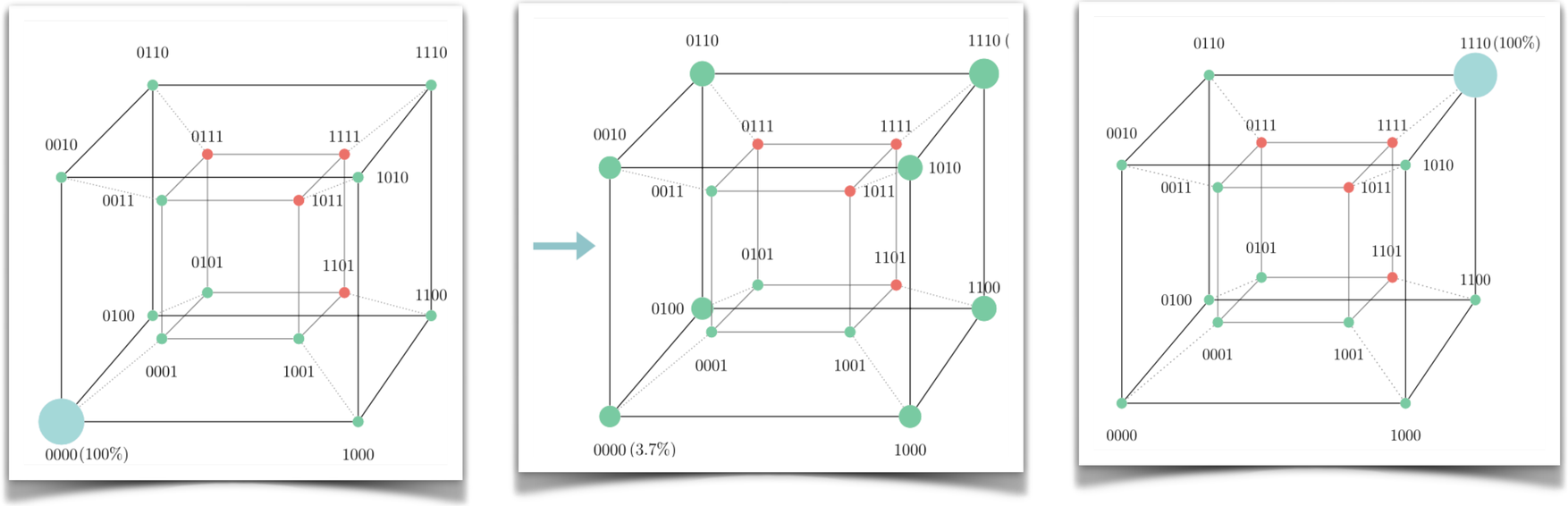
1. Initialize:
Initialize the three registers in the state $|0\rangle^{\mathrm{A}} |Z\rangle^{\mathrm{B}} |0\rangle^{\mathrm{C}}$.

2. Tree traversing:
For each item $m = 1, \ldots, n$, perform the following steps:
(a) Create a superposition. Apply the biased Hadamard H′ on the m-th qubit in register A, controlled on whether z_m does not exceed the remaining capacity stored in register B.
(b) Update the remaining costs. Subtract z_m from the state of register B, controlled on the m-th qubit in register A.
(c) Update the total profit. Add p_m to the state of register C, controlled on the m-th qubits in register A.

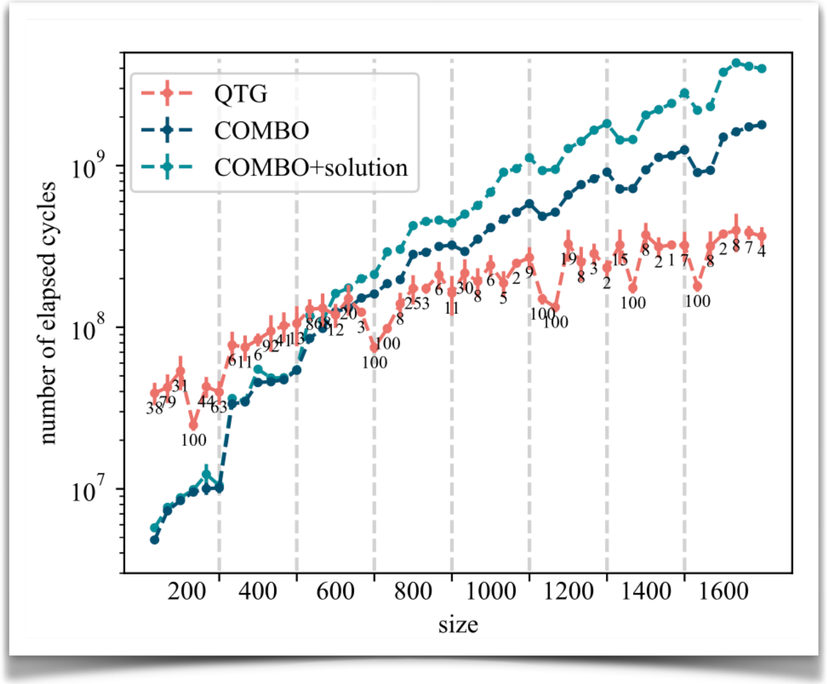
Example
Knapsack instance $\mathrm{KP}_4(\boldsymbol{p}, \boldsymbol{z}; 7)$ with costs $\boldsymbol{z} = (2, 2, 1, 5)$ and profits $\boldsymbol{p} = (6, 2, 1, 2)$

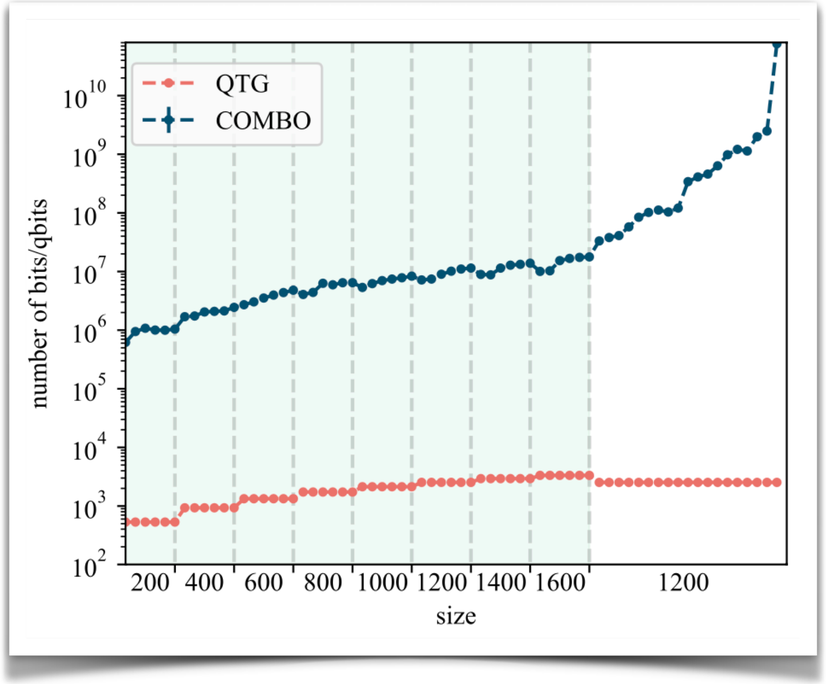


Good News: Knapsack Problems

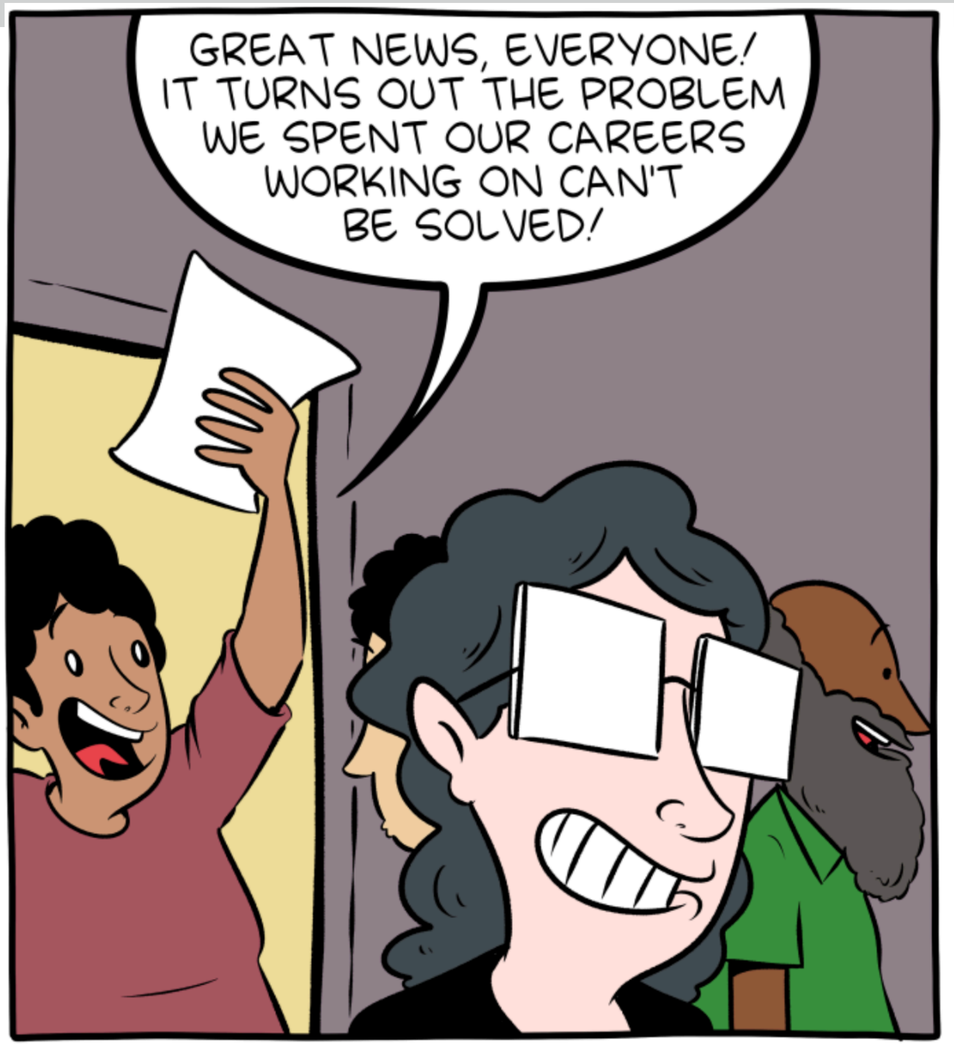


Good News: Knapsack Problems





Thank you!



Mathematicians are weird.