Lower bound techniques for algorithmic problems

Gerhard J Woeginger

RWTH Aachen

4 May 2017 — Ecco-2017.Koper.Slovenia

Introductory example

Problem: Minimum Length Triangulation of Convex Polygon

Input: The vertices $v_1, \ldots, v_n$ of a convex polygon P.
Goal: Find a triangulation of P that minimizes total length.

- Dynamic programming solves this in $O(n^3)$ time
- see for instance: textbook by Corman, Leiserson, Rivest, Stein

Old questions

- How can we reach a better time complexity?
- And if the answer should be negative:
  How can we prove that the cubic running time is best possible?

Structure of talk

In this talk, I will discuss tools for establishing lower bounds on the time complexity of certain (mainly polynomially solvable) problems

1. Algebraic computation tree
2. 3-SUM conjecture
3. Orthogonal vectors
4. All-Pairs-Shortest-Paths
5. Some other related stuff

Algebraic computation tree

Algebraic computation tree (1)

Back in 1983, Michael Ben-Or analyzed the Algebraic Computation Tree model of computation (ACT, for short). An ACT is a binary tree, where each node v has one of the following three roles.

Algebraic computation tree (1)

Back in 1983, Michael Ben-Or analyzed the Algebraic Computation Tree model of computation (ACT, for short). An ACT is a binary tree, where each node v has one of the following three roles.

Computation node

The node has a single child and computes value $f(v)$ as

$$f(v) = f(x) \circ f(y) \qquad \text{or} \qquad f(v) = \sqrt{f(x)}.$$

Here $f(x)$ and $f(y)$ are values of ancestor nodes of v, or input values, or arbitrary real constants. And $\circ \in \{+, -, \times, /\}$.

Algebraic computation tree (1)

Back in 1983, Michael Ben-Or analyzed the Algebraic Computation Tree model of computation (ACT, for short). An ACT is a binary tree, where each node v has one of the following three roles.

Computation node

The node has a single child and computes value $f(v)$ as

$$f(v) = f(x) \circ f(y) \qquad \text{or} \qquad f(v) = \sqrt{f(x)}.$$

Here $f(x)$ and $f(y)$ are values of ancestor nodes of v, or input values, or arbitrary real constants. And $\circ \in \{+, -, \times, /\}$.

Comparison node

The node has two children labeled true and false, and performs test

$$f(x) > 0 \qquad f(x) = 0 \qquad f(x) \geq 0$$

for some ancestor x of v.

Algebraic computation tree (1)

Back in 1983, Michael Ben-Or analyzed the Algebraic Computation Tree model of computation (ACT, for short). An ACT is a binary tree, where each node v has one of the following three roles.

Computation node

The node has a single child and computes value $f(v)$ as

$$f(v) = f(x) \circ f(y) \qquad \text{or} \qquad f(v) = \sqrt{f(x)}.$$

Here $f(x)$ and $f(y)$ are values of ancestor nodes of v, or input values, or arbitrary real constants. And $\circ \in \{+, -, \times, /\}$.

Comparison node

The node has two children labeled true and false, and performs test

$$f(x) > 0 \qquad f(x) = 0 \qquad f(x) \geq 0$$

for some ancestor x of v.

Output node

The node is a leaf, labeled with ACCEPT or REJECT.

Algebraic computation tree (2)

An Algebraic computation tree

- labels every input point $(x_1, \ldots, x_n) \in \mathbb{R}^n$ with ACCEPT/REJECT
- solves membership problem for underlying set A of accepted points

Theorem (Ben-Or, 1983; based on results of Milnor and Thom)

Let $A \subseteq \mathbb{R}^n$, and let $\#A$ be the number of connected components of A. Then any algebraic computation tree for A has worst case height of at least $0.38 \log(\#A) - 0.61n$.

Small (but difficult) technical improvements by Seiferas (1988); A.C.C. Yao (1991, 1995); Grigoriev & Vorobjov (1996); Fleischer (1999)

Algebraic computation tree / Concrete example

Problem: Element Uniqueness

Input: Real numbers $x_1, \ldots, x_n$
Question: Are these numbers pairwise distinct?

- Every permutation $(\pi(1), \pi(2), \ldots, \pi(n))$ of integers $1, \ldots, n$ lies in separate connected component of YES region
- Hence $\#A \geq n!$
- Hence Ben-Or's theorem yields $\Omega(n \log n)$ lower bound in ACT

Algebraic computation tree / More examples

Problem: Sorting

Input: Real numbers $x_1, \dots, x_n$ and $y_1 \le y_2 \le \cdots \le y_n$
Question: Is y-list the sorted version of x-list?

Problem: Set Equality

Input: Real numbers $x_1, \dots, x_n$ and $y_1, \dots, y_n$
Question: Is y-list a re-ordered version of x-list?

- Ben-Or's theorem yields $\Omega(n \log n)$ lower bounds in ACT

Algebraic computation tree / Another example

Problem: Minimum link path

Input: A set of polygonal (pairwise disjoint) obstacles in $\mathbb{R}^2$ with altogether n corners; a start point s and a goal t.

Goal: Find polygonal path from s to t that avoids all obstacles and has minimum number of links.

Theorem (Mitchell, Rote & Woeginger, 1992)

- A minimum link path can be found in $O(n^2\,\alpha(n)\,\log^2 n)$ time.
- Lower bound $\Omega(n \log n)$ for decision version in ACT.

Algebraic computation tree / A final example

Problem: Subset Sum

Input: Positive real numbers $x_1, \ldots, x_n$
Question: Does there exist a subset whose elements add up to 1?

Theorem (Dobkin & Lipton 1978, combined with Ben-Or 1983)

SUBSET SUM has $\Omega(n^2)$ lower bound in ACT.

Theorem (Meyer auf der Heide, 1984)

SUBSET SUM has $O(n^4 \log n)$ algorithm in ACT (even in the more restricted linear decision tree model).

Algebraic computation tree / Remarks

- Note: Floor and ceiling function are not atomic operations in ACT
- Note: ACT does not support indirect addressing

- $\Omega(n \log n)$ lower bounds in ACT often deteriorate to $\Omega(n)$ in RAM
- There exist (fairly natural) problems with $\Omega(n \log n)$ lower bound in ACT and with $O(n)$ algorithm on RAM

- ACT does not require algorithms to be uniform
- Hence: Lower bounds in ACT also hold for non-uniform algorithms
- Hence: Lower bounds in ACT tend to be weak

3-SUM conjecture

3-SUM conjecture (1)

Problem: 3-SUM

Input: Positive real numbers $x_1, \ldots, x_n$
Question: Do there exist three indices i, j, k with $x_i + x_j + x_k = 0$?

Some known facts about 3-SUM on RAM:

- Solvable in $n^2 \log n$ time (very easy)
- Solvable in n^2 time (needs some thought)
- Solvable in $n^2/(\log n/\log\log n)^{2/3}$ time (Grønlund & Pettie, 2014)

Some known fact about 3-SUM in linear decision tree:

- Solvable in $n^{3/2}\sqrt{\log n}$ time (Grønlund & Pettie, 2014)

3-SUM conjecture (2)

3-SUM conjecture

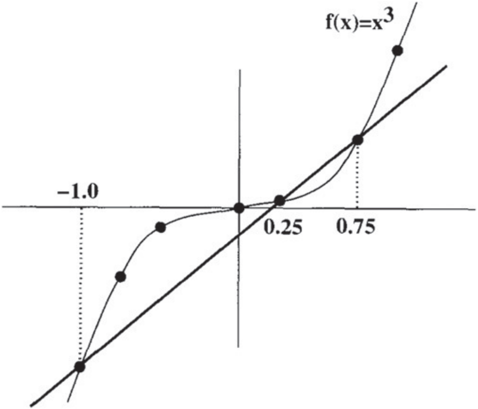
For no real $\varepsilon > 0$,
there exists an $O(n^{2-\varepsilon})$ algorithm for 3-SUM on the RAM.

- Folklore conjecture from the 1980s
- Popularized by a 1995 paper of Gajentaan & Overmars
  "On a class of $O(n^2)$ problems in computational geometry"

Problem: 3-points-on-line

Input: A set of n points in the plane.
Question: Is there a line that contains at least three of these points?

3-SUM conjecture (3)



- $a + b + c = 0$ if and only if (a, a^3) and (b, b^3) and (c, c^3) collinear (by considering $x^3 - (a + b + c)x^2 + (ab + ac + bc)x - abc = 0$)
- Hence 3-SUM conjecture implies lower bound for 3-points-on-line

3-SUM conjecture (4)

Theorem (Gajentaan & Overmars, 1995)

The 3-SUM conjecture implies that
there is no $O(n^{2-\varepsilon})$ algorithm for the following problems:

- Deciding whether a planar point set has three points on common line
- Deciding whether a set of line segments in $\mathbb{R}^2$ has a line separator
- Deciding whether a given set of strips covers a given rectangle
- Deciding whether a given set of triangles covers a given triangle
- Deciding whether the union of a set of triangles contains a hole
- Deciding whether a rod can be moved through a set of line segment obstacles from given source to given goal position
- Etc, etc, etc.

Tons of similar results have been derived over the last 20 years.

Orthogonal vectors and SETH

Orthogonal vectors (1)

Problem: Orthogonal Vectors

Input: Two n-element sets $U, V \subset \mathbb{R}^d$
Question: Do there exist $u \in U$ and $v \in V$ with $uv = 0$?

Some known facts about Orthogonal Vectors on RAM:

- Solvable in $n^2 d$ time (trivial)
- The exponent can be lowered from 2 to $2 - const/(\log d - \log\log n)$ (Abboud, Williams & Yu, 2015)

Orthogonal Vectors conjecture (Ryan Williams, 2005)

For no real $\varepsilon > 0$,
there exists an $O(n^{2-\varepsilon} d^{const})$ algorithm for Orthogonal Vectors.

Orthogonal vectors (2)

Strong Exponential Time Hypothesis (SETH)

For no real $\varepsilon > 0$,
the SATISFIABILITY problem with n variables and $m = O(n)$ clauses can be solved in $O(2^{(1-\varepsilon)n})$ time.

Orthogonal vectors (2)

Strong Exponential Time Hypothesis (SETH)

For no real $\varepsilon > 0$,
the SATISFIABILITY problem with n variables and $m = O(n)$ clauses can be solved in $O(2^{(1-\varepsilon)n})$ time.

Theorem (R. Williams, 2005)

SETH implies the Orthogonal Vectors conjecture.

- Consider SATISFIABILITY instance
- m-dimensional vectors; every coordinate corresponds to one clause
- Split variables into two groups U' and V' of size $n/2$
- For every truth assignment for U' create vector:
  If clause c is satisfied, coordinate c is set to 0; otherwise set to 1.
- Do the same for truth assignments for V'
- $2^{n/2}$ vectors for U', and $2^{n/2}$ vectors for V'

Orthogonal vectors (3)

Problem: Frechet distance

Input: Given two polygonal routes P and Q with n corners in $\mathbb{R}^2$.
Goal: Find monotone parametrizations $\alpha : [0,1] \to P$ and $\beta : [0,1] \to Q$, so that the maximum distance $\alpha(x)$ to $\beta(x)$ is minimized.

Some known facts about Frechet distance problem:

- Solvable in n^2 time (Alt & Godau,1995)
- Solvable in $n^2 \log n / \log\log n$ time
  (Agarwal, Avraham, Kaplan & Sharir, 2012)
  (Buchin, Buchin, Meulemans & Mulzer, 2014)
- Orthogonal Vectors conjecture implies that Frechet distance problem has no $O(n^{2-\varepsilon})$ algorithm (Bringmann, 2014)

Orthogonal vectors (4)

Orthogonal vectors yields many other quadratic lower bounds:

- for instance: longest common subsequence
- for instance: dynamic time warping distance
- for instance: string edit distance (Levenshtein distance)

It is unknown

- whether Orthogonal Vectors conjecture implies SETH
- whether Orthogonal Vectors implies 3-SUM conjecture
- whether 3-SUM implies Orthogonal Vectors conjecture
- whether 3-SUM implies SETH
- whether SETH implies 3-SUM

All-Pairs-Shortest-Paths

All-Pairs-Shortest-Paths (1)

Problem: All-Pairs-Shortest-Paths (APSP)

Input: An edge-weighted graph on n vertices.
Goal: Compute lengths of shortest paths between all pairs of vertices.

Some known facts about APSP:

- Floyd-Warshall algorithm (from 1962) solves APSP in $O(n^3)$ time.
- Small speed-up to $n^3/2^{\sqrt{\log n}}$ by Williams (2014)
- So far, no $O(n^{3-\varepsilon})$ algorithm has been found for APSP.

APSP conjecture (Vassilevska-Williams & Williams, 2010)

For no real $\varepsilon > 0$,
there exists an $O(n^{3-\varepsilon})$ algorithm for APSP.

All-Pairs-Shortest-Paths (2)

Theorem (Vassilevska-Williams and Williams, 2010)

The APSP conjecture is equivalent to the conjecture that
there is no $O(n^{3-\varepsilon})$ algorithm for the following problems:

- Detecting a negative weight triangle in an edge-weighted graph
- Finding min weight cycle in graph with non-negative edge weights
- Finding the second-shortest s-t-path in an edge-weighted graph
- The replacement paths problem in an edge-weighted digraph
- Verifying whether a given matrix defines a metric
- Verifying a matrix product over $(\min, +)$ semiring
- Etc, etc, etc.

TSP and k-opt

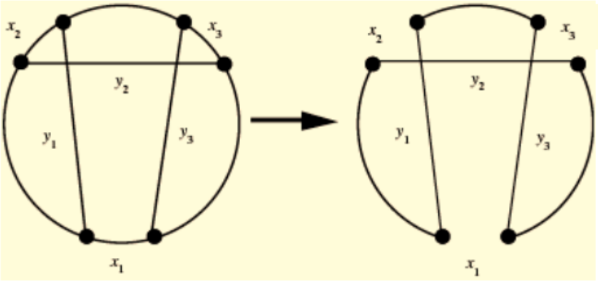
TSP and k-opt

Problem: Travelling Salesman Problem (TSP)

Input: n cities, plus all the distances $d(\cdot,\cdot)$ between city pairs
Goal: Find shortest round-trip (TSP tour) through these cities.

k-opt is a popular local search neighborhood for the TSP:

- To improve a given tour, remove k edges and reconnect the pieces



Fact sheet for k-opt

- Goes back to Croes (1958), Flood (1956), Lin (1965), Bock (1958)

- Local optima can be hard to find
  (exponential lower bound for path following version)
- Computing local optima for 20-opt is PLS-complete (Krentel, 1989)

- Local optima for k-opt may be sub-optimal (even if $k \approx 3n/8$)
- Local optima for k-opt may be very bad (unbounded worst case ratio)
- Local optima for 2-opt may be very bad (worst case ratio $\Omega(\sqrt{n})$)
- Local optima for 2-opt may be very bad, even in Euclidean plane
  (worst case ratio $\Omega(\log n / \log \log n)$)

- Experimental: local optima for Euclidean 2-opt within 5% of optimal
- Experimental: local optima for Euclidean 3-opt within 2% of optimal

Local optima (1)

Problem: k-opt detection

Input: a TSP-instance; a tour T
Question: does the k-opt neighborhood of T contain a shorter tour?

Local optima (1)

Problem: k-opt detection

Input: a TSP-instance; a tour T
Question: does the k-opt neighborhood of T contain a shorter tour?

Problem: k-opt optimization

Input: a TSP-instance; a tour T
Question: find the shortest tour in the k-opt neighborhood of T

Local optima (1)

Problem: k-opt detection

Input: a TSP-instance; a tour T
Question: does the k-opt neighborhood of T contain a shorter tour?

Problem: k-opt optimization

Input: a TSP-instance; a tour T
Question: find the shortest tour in the k-opt neighborhood of T

A trivial observation:

For fixed values of k,
both problems can be solved in $O(n^k)$ time.

- try all $\binom{n}{k}$ possibilities for removing k edges
- try all $2^k k!$ ways of reflecting and ordering the k resulting tour pieces

Local optima (2)

Theorem (Marx, 2008)

Problem k-opt detection with parameter k is W[1]-complete.

Theorem (Guo, Hartung, Niedermeier & Suchý, 2013)

Under the Exponential Time Hypothesis (ETH),
in the running time of an algorithm for k-opt detection
the exponent of n must grow at least like $k/\log k$.

- by fpt-reduction from k-Partitioned Subgraph Isomorphism
- delicate and tedious handling of the parameter

The case k=2

Observation

Any algorithm for 2-opt detection on n cities needs $\Omega(n^2)$ time in the worst case.

Proof: the algorithm must read & analyze all the input data

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Find the hidden zero!

The case k=3

Johnson and McGeoch (1997) write in a survey chapter on local search:

> To complete our discussion of running times, we need to consider the time per move as well as the number of moves. This includes the time needed to *find* an improving move (or verify that non exists), together with the time needed to *perform* the move. In the worst case, 2-opt and 3-opt require $\Omega(n^2)$ and $\Omega(n^3)$ time respectively to verify local optimality, assuming all possible moves must be considered.

The case k=3, continued (1)

Theorem (De Berg, Buchin, Jansen, Woeginger, 2016)

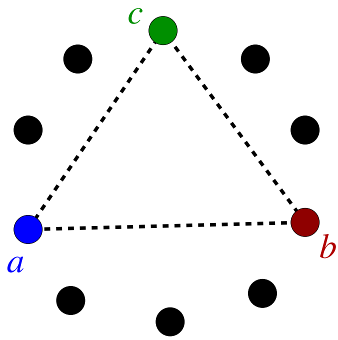
Under the All Pairs Shortest Paths conjecture (APSP),
3-opt detection cannot be solved in $\Omega(n^{3-\varepsilon})$ time.

Sketch of proof:

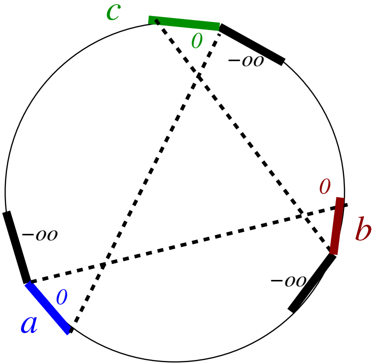
- Take an instance of detecting negative triangle in edge-weighted graph
- Order the vertices in a circle
- Translate every vertex into two edges, of cost 0 and $-\infty$
- The resulting cycle with $2n$ edges is the starting tour T
- The edge-weights (in graph) become distances between cost 0 edges

Remark: We also have a reduction in the other direction

The case k=3, continued (3)



Graph



TSP

The case $k = 4$

Johnson and McGeoch (2002) write in a survey chapter on the TSP:

Currently, 2-opt and 3-opt are the main k-opt heuristics used in practice, introduced respectively by Flood and Croes (1958) and by Bock. In Shen Lin's influential 1965 study of 3-opt, he concluded that the extra time required for 4-opt was not worth the small improvement in tour quality it yielded, and no results have appeared since then to contradict this conclusion.

The case $k = 4$, continued (2)

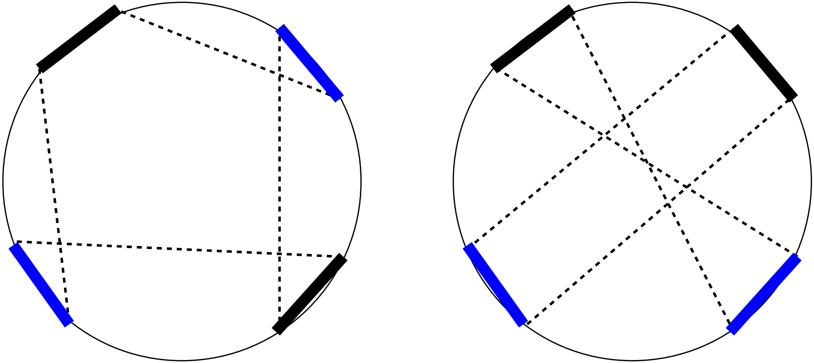
Theorem (De Berg, Buchin, Jansen, Woeginger, 2016)

The 4-opt optimization problem for n cities can be solved in $O(n^3)$ time.

Sketch of proof:

- Distinguish $2^3 \cdot 3! = 48$ cases for possible orderings of tour-pieces
- Two leaving edges collide, if connected by entering edge
- Every leaving edge collides with exactly two other leaving edges
- In every case: some pair of leaving edges is not colliding
- Check all $O(n^2)$ possibilities for the other leaving edges
- Optimize positions of not colliding pair in $O(n)$ time

The case $k = 4$, continued (3)



Final remarks (1)

To summarize the results on k-opt for $k \leq 4$:

- 2-opt in $O(n^2)$ time (and that's best possible)
- 3-opt in $O(n^3)$ time (and that's best possible)
- 4-opt in $O(n^3)$ time (and that's best possible)

Final remarks (1)

To summarize the results on k-opt for $k \leq 4$:

- 2-opt in $O(n^2)$ time (and that's best possible)
- 3-opt in $O(n^3)$ time (and that's best possible)
- 4-opt in $O(n^3)$ time (and that's best possible)

Theorem (De Berg, Buchin, Jansen, Woeginger, 2016)

For any fixed $k \geq 2$,
the k-opt optimization problem for n cities
can be solved in $O(n^{\lfloor 2k/3 \rfloor + 1})$ time.

Theorem (Cygan, Kowalik, Socala, 2017)

For any fixed $k \geq 2$,
the k-opt optimization problem for n cities
can be solved in $O(n^{(1/4+\varepsilon_k)k})$ time, where $\varepsilon_k \to 0$.

- In particular: 5-opt in $O(n^{3.4})$ time

Final remarks (2)

Unfortunately:

- There are no tools available for proving $\Omega(n^4)$ lower bounds
- There are no tools available for proving $\Omega(n^5)$ lower bounds
- Etc.

Note that:

- 3-SUM is $k = 3$ special case of $W[1]$-hard k-SUM problem
- Negative Weight Triangle is related to $W[1]$-hard k-CLIQUE
- Orthogonal Vectors is $k = 2$ special case of something $W[1]$-hard

Possible research direction, perhaps worthwhile:

- Establish cross-connections between small-parameter cases of various $W[1]$-hard problems
- Perhaps get $\Omega(n^4)$ lower bound technique along similar lines

Final remarks (3)

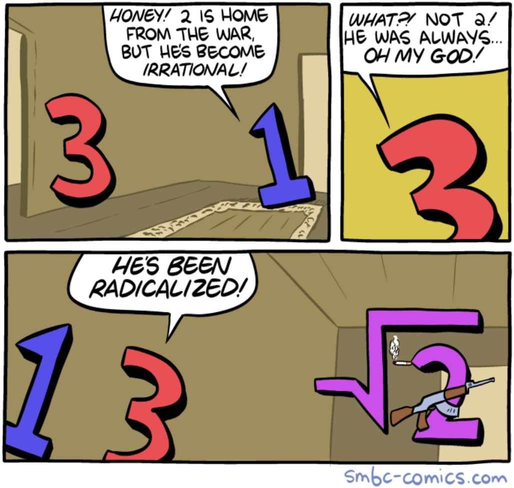
Problem: Minimum Length Triangulation of Convex Polygon

Input: The vertices $v_1, \ldots, v_n$ of a convex polygon P.
Goal: Find a triangulation of P that minimizes total length.

Minimum Length Triangulation of Convex Polygon:

- APSP will not help us in getting $\Omega(n^3)$ lower bound

Thank you!


HONEY! 2 IS HOME FROM THE WAR, BUT HE'S BECOME IRRATIONAL!
WHAT?! NOT 2! HE WAS ALWAYS... OH MY GOD!
HE'S BEEN RADICALIZED!
smbc-comics.com