Complexity of Minimization

Warren D. Smith*
WDSmith@fastmail.fm

11 May 2000, revised Jan 2005

Abstract — **An important problem in numerical analysis is trying to minimize (or maximize) a real function $F(x_1, x_2, ..., x_d)$ of d real variables. We investigate (and survey) its computational complexity borderlines.**

Results include (precise theorem statements are in the text): Minimizing, e.g., functions specified by polynomial formulas involving both trigonometric and ordinary arguments, is undecidable. Hence consider plain rational functions; we'll show minimizing these (more precisely, deciding whether the minimum lies below some threshhold) is in NP. If the sum of the numerator and denominator degrees of a (multivariate) rational function is ≤ 4, then finding all its local minima to A decimal places of accuracy may be accomplished in time polynomial in A and the bitlengths of the input *and* the output. (Call this "output sensitive polynomial time" P_O.) However, in the $4+0$ case of a quartic polynomial (or its reciprocal) it is NP-complete even to decide whether a *specified* point is a *local* min. Minimizing a linear function in a cube is polynomial, but minimizing a quadratic in a cube is NP-complete. Minimizing a quadratic in a ball is in P, but it is NP-complete for a quartic and unknown-status for a cubic. In all these NP-completeness results, even *approximating* the value of the min is NP-complete, and the problems remain NP-complete even if all integers are input in *unary.* The P minimization algorithms for low degree rational functions involve new techniques of "multi-stage minimization" and "dimension reduction." At the end the power of our techniques is considered abstractly – allowing us to begin to elucidate the class of "efficiently minimizable functions." We suggest some improvements to heuristic minimizers.

Minimizing functions "unimodal on lines," is in P. Solving systems of nonlinear equations $\vec{F}(\vec{x}) = 0$ is in P, if all the nonlinear functions F_j are monotonic on lines. We present evidence that minimizing strictly unimodal functions (with exactly one minimum, and no saddlepoints) is exponentially hard; but if additionally the function obeys certain derivative bounds, then hardness seems to occur *precisely* when there are extremely long "winding valleys."

Keywords — Minimization, maximization, unimodality, local minima, multistage minimization, polynomials, ellipsoid algorithm, handling indefinite Hessians.

1 Introduction

An important problem in numerical analysis is trying to minimize (or maximize) a real function $F(x_1, x_2, ..., x_d)$ of d real variables. Theoretical numerical analysts usually assume that a "black box" is provided for evaluation of that function at any point to any desired number of digits of precision; and one usually also assumes that a similar black box is available for the evaluation of its gradient. Our important new results are theorems 7, 9, 15, 12, 17, 25; some of the other theorems are not new. Along the way we also provide a new lemma 16 giving a best known result on polynomial root separation.

2 Notation & conventions

A "local minimum," or just "a minima" of a continuous function will mean a minimum within some neighborhood. A function f is "Lipshitz with Lipshitz constant L" if $|f(\vec{a}) - f(\vec{b})| \le L|\vec{a} - \vec{b}|$ for all $\vec{a}$, $\vec{b}$. Polynomial degree means total degree, e.g. x^5y^2z has degree 8.

"Wlog" means "without loss of generality."

When stating run time bounds, we will employ a "real computational model" in which we are allowed to add, subtract, multiply, and divide real numbers exactly in one step. However, I've paid attention to issues of roundoff error to sufficient extent that in all cases in which the real model achieves polynomial time, computers with bounded wordsize will be able to do the same job also in polynomial time. When we say a minimization algorithm (or

*Work originally done at the now-defunct NECI, 4 Independence Way, Princeton NJ 08540. Non-electronic correspondence to 21 Shore Oaks Drive, Stony Brook NY 11790.

algorithm to evaluate a function) involving real numbers runs in "polynomial time," we mean that the output is computable with within $\pm 10^{-N}$ in time polynomial in N and the number of bits in the input.

We are going to assume that the functions we're minimizing are either polynomial-time, or at least (if provided by an infinitely fast oracle) have the same kind of continuity properties as do polynomial-time functions – e.g. it is impossible for them to be so sensitive to their input that computing a logarithmic number of decimals of the output requires knowledge of polynomial number of bits of the input. Without such properties, just the overhead of *communicating* with the oracle would be too onerous.

3 Undecidability

Theorem 1 *Given a function $F(\vec{x})$ specified as a formula involving integer constants and π, $+$, $-$, $\times$, $/$, and circular functions, it is Turing-undecidable what its minimum value over $\vec{x} \in \mathbf{R}^d$ is, or what any location within any computable distance of the min's location, is.*

Proof. For example, consider minimizing the smooth function

$$F(\vec{x}) = \sum_{k=1}^{d} \sin^2(\pi x_k) + \sum_{m=1}^{q} P_m(\vec{x})^2, \tag{1}$$

where P_1,..., P_q are specified multivariate polynomials with integer coefficients, over $\vec{x} \in \mathbf{R}^d$. Clearly $F(\vec{x}) \geq 0$ everywhere. It is Turing-undecidable whether there exists $\vec{x}$ such that $F(\vec{x}) = 0$, because any zero of F would necessarily have all coordinates x_k be integers (thanks to the $\sin^2(\pi x_k)$ terms) and also would necessarily obey the polynomial diophantine equations $P_m(\vec{x}) = 0$; and J.Robinson & Y.Matiyasevich's famous solution of Hilbert's 10th problem (this story is recounted in [3]) shows that deciding whether there is a solution to a system of polynomial diophantine equations is undecidable.[1] $\square$

One may also show that polynomial time computable one-dimensional functions (mapping $[0, 1] \to \mathbf{R}$) exist whose minimum is uncomputable [25] [14].

Thus, to make any progress, we must make additional assumptions about F.

4 Minimizing functions unimodal on lines, and solving systems of equations monotonic on lines

We'll call a continuous real-valued $F(\vec{x})$ "unimodal[2]" if it has at most one min (which will be permitted to be a connected "flat region"). We say $F(\vec{x})$ is "strictly unimodal" if it is unimodal but neither flat regions containing more than one point, nor non-min stationary points (where $\vec{\nabla} F = \vec{0}$) are permitted.

Theorem 2 *We can find the minimum* value *of any concave-∪ function $F(\vec{x})$ for $\vec{x}$ in a given d-ball,*[3] *accurate to N significant figures, in a number of gradient evaluations and additional computational work that are both bounded by polynomial functions of N and d. Also, the* location *of the minimum can be confined within a d-ellipsoid of volume 2^N times smaller than the volume of the original ball.*

Examples: on immediate consequence of this is the previously known fact ("fractional linear programming") that optimizing the ratio $L_1(\vec{x})/L_2(\vec{x})$ of two linear functions within a convex polytope (within which L_2 does not change sign) is in P. A second immediate consequence is that "convex quadratic programming" is in P.

This theorem first became evident with the advent of the "Ellipsoid algorithm" [10]. The key properties of concave-∪ functions which make them easy to minimize are as follows.

Theorem 3 *A concave-∪ function $F(\vec{x})$ has the following properties.*

1. *It has a unique local minimum (which is its global minimum) in any convex region. (If the concavity is not strict, then it is possible for the minimum to be a convex "flat area.")*
2. *A concave-∪ function is differentiable almost everywhere in any d-ball.*
3. *The sets $\{\vec{x} \mid F(\vec{x}) \leq B\}$ are convex.*
4. *If a hyperplane H is drawn that contains some point $\vec{x} \in R$, R convex, and H is normal to $\vec{\nabla} F(\vec{x})$ there, then the minimum of $F(\vec{x})$ in R lies on the downward side of H.*

[1] It is also easy (by considering an appropriate preliminary change of variables) to show the undecidability of minimizing functions in this formula class within compact disks. Testing global analyticity of such formulas (or just in disks) is also undecidable.

[2] We admit a better name might have been "uni-minimal."

[3] Or the "ball" may be repalced by an arbitrary convex set, specified by a "separation oracle" [10].

5. *The intersection of convex regions, is convex.*

6. *The set of points to one side of some hyperplane, is convex.*

Proof sketches: All properties except 4, 2, and 3 are well known or trivial. Property 4, follows from the other properties: the level set in 3 is a subset of the halfspace in 4 and then apply 1. Property 2 follows from the fact that limits of decreasing bounded sequences exist (here the slope of a tangent plane is decreasing as we move towards some point) and monotonic functions have at most a countably infinite number of discontinuities (see theorem 4.30 page 96 of [23]). □

The ellipsoid algorithm works roughly as follows. At any time, we have a d-ellipsoid which contains the minimum. The initial ellipsoid is the initial d-ball. We let $\vec{x}$ be the center of the current ellipsoid and ask our gradient oracle for $\vec{\nabla}F(\vec{x})$ (taking advantage of property 2 of theorem 3 to realize that a gradient will exist, with probability 1, e.g. assuming our initial ball had been perturbed randomly by ϵ). We may then use property 4 to split the ellipsoid with a hyperplane through its center into two hemi-ellipsoids – and we know which hemiellipsoid the minimum lies in! Then we replace our current ellipsoid by the smallest-volume ellipsoid containing this hemi-ellipsoid (there is a simple formula describing it) and continue on. The decrease in ellipsoid volume each step is at least a factor of $e^{-1/(2d)}$, which is fast enough to prove theorem 2. (One also has to analyse the number of digits of precision needed to describe all the ellipsoids, but this also may be proven to be polynomially bounded, and we also need to have the ability to make small random perturbations to reach points where the gradient is defined, so that our computational model needs a random number generator.)

An alternative algorithm, originating with Vaidya [26], is based on polytopal surrounding regions rather than ellipsoids. Instead of using the center of the ellipsoid as the query point, we use the so-called "analytic center" of the polytope, which (Vaidya shows) may be found accurately enough in polynomial time. Vaidya also shows cutting with any hyperplane through the analytic center of an F-faced d-polytope will reduce its d-volume by a factor sufficient to assure polynomial worst case time for the overall minimization procedure. Vaidya pointed out that it would have been even better if one could split the polytope by cutting it with a hyperplane through its center of mass (also called the "volumetric center") because that is guaranteed to reduce the volume by a factor of at least $(1 - 1/e) \approx .6321$. Unfortunately, it seems generally impossible to find the volumetric center of a d-polytope in polynomial time. There are numerous variants of these algorithms. One survey is [20].

1D functions: There are also many (well-known) algorithms [22][4] for minimizing continuous functions of *one* variable that work by repeatedly cutting the interval known to contain a min, into fragments, a particular one of which is known to contain a min. These techniques will find the location of a local min to A decimal places in $O(A)$ steps, each step involving a function evaluation. If the function is unimodal, the min found will necessarily be the global min. On the other hand, if the function has multiple maxes and mins (say W of them) in the initial interval, then their locations *all* may be found accurate to $\pm 10^{-A}$ by continuing the process onward after the first min is found, in the subintervals to its right and left, only now looking for a *max*. If one is found, the next level seeks a min, and so on. The entire process will take $O(WA)$ steps. This is our first example of and *output sensitive* polynomial time minimization algorithm, in the class (which we hereby christen) "P_O."

Now although these algorithms are very fine, they only handle concave-∪ or 1D-unimodal functions – a very small class of functions compared to the functions people are often interested in minimizing.

For many years, it has been quite an embarrassment for the entire field of computer science, that concave-∪ functions (with polynomial-time computable gradients), and 1-dimensional functions were essentially the *only* continuous functions people knew how to minimize in polynomial time![4] Surely some larger classes of functions ought to be easy to minimize? We begin by pointing out, essentially, that the true class of functions the ellipsoid algorithm and related algorithms can minimize, are the functions *unimodal on lines.* This result therefore unifies the previous two classes – concave-∪ multivariate and unimodal 1-variate – of easy-to-minimize functions.

Observation 4 *Functions unimodal on lines*

1. *Are differentiable almost everywhere in any d-ball.*

2. *Have at most one local minimum (which is the global minimum). If the unimodality is not strict, then it is possible for the minimum to be a connected "flat area."*

[4] There are some trivial exceptions to this. For example, if T is an efficiently computable diffeomorphism whose inverse map is also efficiently computable, then $F(T(\vec{x}))$ is readily minimized if $F(\vec{x})$ is. The usual numerical methods [4] only come with rigorous *local* convergence proofs, except in the 1D case and can only be regarded as heuristic otherwise. There are techniques guaranteed to find global minima [6] but they may never terminate and/or can require enormous runtimes. It is revealing of the great and unfortunate disconnect between practitioners and rigorous complexity results for algorithms *guaranteed* to succeed in finding a min, that [4][5][6] do not even *mention* the ellipsoid algorithm.

3. *One can reach this minimum simply by starting anywhere and then descending, e.g. by following the path of steepest descent.*

Proof: Property 2 is proven by contradiction – draw a line through the "two" mins.

Property 1 follows from the fact that a function that is monotonic in some interval, is differentiable almost everywhere in that interval. Although this fact is not to be found in most books titled "Real Analysis," it is a standard result in measure theory. According to page 97 of [13], if F is monotonic, then $\lim_{h\to 0}[F(x+h)-F(x)]/h$ (a) converges in L_1 and (b) converges pointwise almost everywhere. Only (a) is proven in [13], and the reader is somewhat vaguely redirected to other sources for the proof of (b), "the best version of Lebesque's density theorem." It is covered better in chapter 22 of volume 2 of [7]. Ultimately (b) traces back to Henri Lebesque in 1902. □

Observation 5 *Given d real-valued functions F_1, F_2, ..., F_d in $\mathbf{R}^d$ which each are monotonic on lines. Then there is at most one solution $\vec{x}$ of the simultaneous equations $F_1(\vec{x}) = 0$, $F_2(\vec{x}) = 0$, ..., $F_d(\vec{x}) = 0$. (If the monotonicity if not strict, then the solution could be a connected "flat area.")*

Proof. Consider a line through the "two" solutions. □

Functions unimodal on lines *can* be minimized in polynomial time, provided their gradients are available in polynomial time and if by "minimize" we mean it suffices if we can confine the minimum inside a d-ellipsoid of exponentially small volume.

Here are the requisite trivialities:

Observation 6 *The functions unimodal on lines, are precisely the functions $F(\vec{x})$ whose "$(\le B)$-sets" $\{\vec{x} \mid F(\vec{x}) \le B\}$ are convex.*

Proof. If a function has convex $(\le B)$-sets, it is clearly unimodal on lines. On the other hand, consider the $(\le B)$-set of a function unimodal on lines, and suppose some such set S is nonconvex. In that case, there is a line L whose intersection with S is nonconvex, i.e. consists of at least 2 disconnected components. But then, the function cannot be unimodal on L, a contradiction. □

Now in both the Ellipsoid algorithm, and Vaidya's algorithms, the actual *values* of $F(\vec{x})$ are never requested (except perhaps once at the end); only the gradient is ever asked for. Furthermore, the gradient does not need to have the right length, since the length of the gradient is never used, only its direction is used (to generate a splitting hyperplane normal to it). Now the gradients of functions are normal to their level sets, and in view of the differentiability properties of our class of functions and of the surfaces of convex bodies, will always (if we make a small random perturbation, with probability 1) exist, Hence, the ellipsoid and Vaidya algorithms simply will work to minimize functions unimodal on lines. There is the caveat that the actual value of the minimum cannot be determined accurately because there could be a huge "spike" near the minimum which we never stumble upon. However, if it is deemed sufficient to confine the minimum within a d-ellipsoid of volume 2^N times smaller than the original ball, then we are OK. Hence:

Theorem 7 *We can find a d-ellipsoid, in which lies the minimum, in any given d-ball B, of any function $F(\vec{x})$ unimodal on lines, in a number of gradient evaluations and additional computational work that are both bounded by polynomial functions of N and d. Here the confining d-ellipsoid has volume 2^N times smaller than the volume of the original ball.* □

Observation 8
1. *The max of unimodal functions, is a unimodal function.*
2. *The square (or the absolute value, or any unimodal function) of a monotonic function, is a unimodal function.*

Therefore,

Theorem 9 *Given d real-valued functions F_1, F_2, ..., F_d in $\mathbf{R}^d$ which each are monotonic on lines, and a d-ball B. We can find a d-ellipsoid, in which lies the solution of the simultaneous equations $F_1(\vec{x}) = 0$, $F_2(\vec{x}) = 0$, ..., $F_d(\vec{x}) = 0$, (if there is such a solution in B) in a number of function and gradient evaluations and additional computational work that are both bounded by polynomial functions of N and d. Here the confining d-ellipsoid has volume 2^N times smaller than the volume of the original ball.*

Proof. Minimize the function (which is unimodal on lines) $F = \max(|F_1|^2, |F_2|^2, ..., |F_d|^2)$. Or the function $F = \max(|F_1|, |F_2|, ..., |F_d|)$. This latter function has the advantage that its gradient is computable by evaluating the gradients of the F_k only (without needing also to evaluate the F_k themselves, except for their signs). (If the value $F = 0$ is achievable, the simultaneous equations are solvable.) □

Remark 10 *Numerical methods books often mention the similar sounding idea of minimizing* $F = |F_1|^2 + |F_2|^2 + \cdots + |F_d|^2$ *to solve simultaneous nonlinear equations. However, the sum of unimodal functions (or the sum of squares of monotonic functions) is not necessarily unimodal. Therefore, our suggestion is better.*[5]

We also mention that applying a continuous-monotonic-increasing function to a unimodal function unimodal on lines, yields a function that still is unimodal on lines. Pretransforming $\vec{x}$ by any projective transformation of $\mathbf{R}^d$ preserves the unimodality-on-lines of $F(\vec{x})$ in the region to one side of the projective transformation's singularity.

5 Strictly unimodal functions

Since minimizing functions unimodal on lines turned out to be so easy, what about just minimizing *all* strictly unimodal functions? These functions $F(\vec{x})$ have at most one min, and no matter from where you start, there is always a path on which F decreases, strictly monotonically, until the min is reached.

Naively, it would seem as though these functions F ought to be easy to minimize! But actually, we will now present evidence that (even if F is Lipshitz) minimizing F is still (apparently) exponentially hard.[6] Our example will involve an "exponentially long winding valley." We then show (by constructing an algorithm) that smooth strictly unimodal functions with suitably bounded derivatives are always easy to minimize if the "lengths of their discretized steepest descent paths" are polynomially bounded. Thus we have now reached a complete (albeit conjectural) understanding: there is *only one way* that minimization of such functions can be hard: if they have super-polynomially long "winding valleys."

5.1 Construction of a well behaved strictly unimodal Lipshitz function $F(\vec{x})$ that seems very hard to minimize

Definition 11 *Let $H[N]$ be the number of Hamiltonian paths on the 2^N vertices of a boolean N-cube C^N starting from a particular starting point (say 00000).*

Then[7] $H[1] = 1$, $H[2] = 2$ (clockwise & anticlockwise), $H[3] = 18$, $H[4] = 5712$, $H[5] = 5859364320$. $N!$ divides $H[N]$. By considering gluing together Hamiltonian paths on two opposite cube faces, $H[N] \ge NH[N-1]^2$. Then by realizing the path has $\le N-1$ choices at each node,

$$5859364320^{2^{N-5}+(N-5)\log_2(N/2)} \le H(N) \le (N-1)^{2^N-3}N, \quad N \ge 5. \tag{2}$$

Consider an "oracle" for providing function values. This oracle will call as a sub-oracle, somebody who has chosen one of these Hamiltonian paths on C^N at random. This sub-oracle could assign any monotone decreasing function values he wants on C^N's vertices. Now expand each bit to M bits (like a "concatenated code") and call two complementary (of the 2^M possible) patterns for each expanded bit "legitimate." (The legalities may differ at different bit locations.) These legitimate patterns represent a 0 or 1 (unexpanded) bit.

Then let $F(\vec{X})$, when $\vec{x} \in \{0,1\}^{MN}$ be defined as follows: If all N of the "patterns" (M-tuples of consecutive coordinates in $\vec{x}$) are "legitimate," let F be the integer distance along the Hamiltonian path in C^N. But if any pattern is "illegitimate," assign a higher function value, e.g., add $10MN$ times the Hamming distance to the nearest legitimate bit-word.

We now have a function mapping MN-bit strings to real numbers. If we start at random vertex of C^{MN} and do local searches over neighborhoods of Hamming radius $M/2$ to find the min in that neighborhood, then go to that min and do it again, and so on, then we will first fall down onto the legitimate subset of C^{MN}, which is essentially a C^N, then follow the Hamilton path on C^N to the minimum. Such a procedure will require an expected #steps

[5] It is true that (if the F_k are smooth) our suggestion is generally not going to have the property that it is smooth everywhere in every neighborhood of the solution, while the sum of squares function does have that property. However this would only be relevant if we planned to switch to a locally convergent algorithm, depending on smoothness, once we got close enough to the solution – and if we planned to do that, we would surely prefer to employ Broyden's algorithm [4] or some other local algorithm intended for solving nonlinear systems directly, rather than via a sum of squares minimization problem which requires carrying twice as many digits of precision for no reason.

[6] *Proving* this would require first proving that UP∩co-UP≠P, an open problem equivalent to the existence of 1-way functions.

[7] $H[1], \ldots, H[5]$ were computed by David Vanderschel in 1973; see Scient. Amer. 228,4 (April 1973) 111-112. $H[6]$ is probably also within reach by a means of a better algorithm I'll now sketch. The idea is that it is possible to count Hamiltonian paths in a graph in an amount of time much smaller than the number of those paths. In the particular case where the graph is the N-cube, an additional factor of $N \cdot N!$ may be saved. For example, one could tabulate all self-avoiding paths of half length (actually only tabulate the *counts* of such paths on node subsets), then glue pairs of halflength paths (visiting disjoint vertex sets; the second path is the result of taking a path starting at 000000 and complementing the bits necessary to make its starting point instead be the endpoint of the first path) together. Only "canonical" paths (which "turn on" bits in lexical order) need be considered if we multiply counts appropriately by $N!$. Also, only paths whose complement node-set is connected, with valencies ≥ 2 everywhere but at one node, need be tabulated. It is also possible to glue together subpaths of length 1/3, which allows smaller tables but may increase the runtime.

$\geq 2^{N-1}$ (each step being a local search in a neighborhood of cardinality $\geq \sum_{J=0}^{M/2} \binom{MN}{J}$)) before reaching the min. With appropriate choices of M, N this is an exponential number of steps in a space of polynomial dimension MN.

It is also possible to modify this example to make the path be strict monotone, as opposed to "bumpy" with bumps of "width" M steps each. To do that we now make $M+1$ "legitimate" M-bit patterns instead of just two (namely: the two original complementary patterns, as well as all the $M-1$ intermediate patterns got by changing each bit, one bit at a time in order, to convert the first into the second). Illegitimate patterns now are assigned a cost penalty of M^2N times their hamming distance to the closest legitimate pattern, and the magic "path" of in C^N now contains at least M times more nodes than it originally did, since each bit-alteration is now caused by an M-step changeover, and the distance along this path to the goal is hence about M times larger (and gives the cost of the legitimate states).

Now, we have a function, (provided by an oracle at any C^{NM} vertex on request) which has exactly one local min in the Hamming neighborhood structure − but it seems hard to find. Due to the doubly exponential number (EQ 2) of such paths, it is impossible to tell which path is the correct one that the oracle has in mind, unless one asks at least an exponential expected number ($\approx 2^N$) of 1-bit questions. Even if an exponentially large number of the path's vertices, in order, are known, the path's endpoint could still be any of an exponentially large number of possible vertices, so it seems impossible to "cheat" by just guessing the min and not trying to determine the oracle's whole conception. Due to the exponential cardinality of radius-$M/2$ neighborhoods, it seems exponentially difficult to find a "shortcut" − at least if the underlying plan is not known. Now I admit these arguments are not a proof of exponential hardness, but they do make it plausible that, in general, exponentially long "winding valleys" are possible in high dimensions such that "shortcutting" the valleys seems exponentially difficult.

This example *does* prove that any "simplex-type" (i.e. locally downward-stepping) algorithm (including randomized algorithms) *must* consume exponential time to find the minimum of a (nonlinear, but unimodal) function on the vertices of a cube. In other words, if simplex algorithms are efficient, it is due to the *linear* nature of the objective function, not to its unimodality.

It is easy to interpolate this discrete $F(\vec{x})$ to make a *piecewise linear* function (now $\vec{x} \in \mathbf{R}^{MN}$) which has exactly one local min and which has slope ≥ 1 everywhere (but the slope is always $O(MN)$) which seems to be hard to minimize. Specifically you put piecewise linear "mountain pyramids" making it "undesirable" to be inside or outside the hypercube, or indeed undesirable to be on the interior of any face of dimensions ≥ 2 of the hypercube − you always want to be on a vertex or an edge.[8] Finally, the oracle could transform $\vec{x}$ by any bidirectionally-Lipshitz diffeomorphism, which should effectively render the cubelike logical structure invisible, or at least disguise it very heavily. Also, it is possible to smooth this piecewise linear function without destroying its structure, by, e.g., convolution with any sufficiently sharply peaked Gaussian.

5.2 Functions with suitably bounded derivatives and short descent paths are easy to minimize

Let a "subgradient descent path" for a differentiable function $F(\vec{x})$ be any curve $\vec{x}(t)$ such that $\vec{x}'(t)$ points in a direction at most 45° away from $\vec{\nabla} F(\vec{x}(t))$.

Theorem 12 *Let $F(\vec{x})$ mapping $\mathbf{R}^d \to R$ be (1) strictly unimodal, (2) differentiable, (3) each element of its gradient $\vec{G}$ is locally Lipshitz with Lipshitz "constant" K, (4) Either $|\vec{G}| \geq$the distance to the minimum, or $|\vec{G}| \geq 1$, everywhere, (5) F's minimum is known to lie inside an initially given ball of radius r, (6) suppose every subgradient steepest descent path, starting from any point in the ball, is short (total length $\leq S$).*

Then there exists an algorithm running in $O(S/K)$ gradient-F evaluations and $O(Sd/K)$ additional work, to reduce the nonoptimality of F (and the distance to the min) at least halfway.

Proof sketch. Consider a ball $B_{\vec{x}}$ of radius $.5|\vec{G}|/K$ about the current $\vec{x}$. Due to (3) it cannot contain the global min (where $\vec{G} = \vec{0}$) and indeed $\vec{G}$ cannot change direction by more than 45° (versus its value at the center $\vec{x}$ of the ball) anywhere within $B_{\vec{x}}$. Hence the global minimum *within* $B_{\vec{x}}$ must lie on its surface somewhere on a spherical cap of angular radius 45° centered in the $-\vec{G}$ direction. The algorithm will step to the center of this spherical cap (which must be a downward step and will drop F's value by at least a constant fraction of the way to its value at the global min within $B_{\vec{x}}$, and the distance to that min is also reduced by at least a constant fraction). We then replace $\vec{x}$ by this new point and continue on.

Due to the lower bound (4) on the slope of F, the steps will never be short, allowing us to bound the number of steps taken by the minimizer by $O(S/K)$. □

[8] For example, in 2D to make it undesirable to be on the interior of the square $|x| < 1$, $|y| < 1$, you could add a penalty term proportional to $\min\{1 - |x|, 1 - |y|\}$. Such piecewise linear functions are available in every dimension d to prevent being on the interior of any k-face for any k with $2 \leq k \leq d$.

6 Minimization of polynomials in or on cubes

Wlog, we will assume all d of our discrete variables are *boolean* choices. Then, if we had a "black box" available to compute F exactly, we could minimize F by exhaustion over all 2^d possibilities.

However, it may be possible to do better if F is assumed to have some particular form. For example, suppose $F(\vec{x})$ is linear:

$$F(\vec{x}) = a_0 + \sum_{k=1}^{d} a_k x_k. \tag{3}$$

In that case, if the a_k are known it is trivial to find the $\vec{x}$ which minimizes F; simply choose $x_k = 1$ if and only if $a_k < 0$ (for $k = 1, 2, ..., d$); otherwise $x_k = 0$.

Even if the d coefficients (ignoring the irrelevant constant term a_0) of $F(\vec{x})$ are not initially known, but a black box is available for computing F, then they may be deduced by evaluating F at a suitable set of d points $\vec{x}$ (an especially good choice of d points are the d rows of a $d \times d$ Hadamard matrix) then deducing the coefficients from the values by solving a linear system. (The advantage of using a Hadamard matrix is that it is orthogonal, so that we need not perform a matrix inversion and we will automatically get least-squares fitting in the presence of noise.)

What about if F is quadratic:

$$F(\vec{x}) = a_0 + \sum_{k=1}^{d} (a_k x_k + \sum_{j \le k} b_{jk} x_j x_k) \ ? \tag{4}$$

In that case, it is NP-hard to find the minimum of F. This follows by a reduction from "max 2-SAT" [8].

Indeed, one may even show that it is hard to *approximate* the minimum values of quadratics and cubics on (or in[9]) a hypercube.

Theorem 13 *It is NP-complete to find the minimum value of a quadratic function of d variables, each lying in the set $\{0, 1\}$. Indeed, we construct an infinite set of quadratics with bounded integer coefficients, so that even approximating the minimum value to within a factor of* 1.01*, or to within an additive constant, is NP-hard.*

Proof. Wlog, the quadratic only has terms of the forms $x_i x_j$, $(1 - x_i)x_j$ $x_i(1 - x_j)$, and $(1 - x_i)(1 - x_j)$, $1 \le i < j \le n$, with coefficients in $\{0, 1\}$, There are no linear terms (x_i). Also wlog the hypercube we are speaking of is $\{0, 1\}^n$. In that case, maximizing the quadratic is exactly the same problem as entirely general MAX-2SAT: maximizing the number of clauses $x_i \wedge x_j$, $\bar{x}_i \wedge x_j$, etc. from a list of order S specified clauses, which are satisfied by n boolean variables x_1, ..., x_n. MAX-2SAT is known to be MAX-SNP-hard and indeed it is known that approximating the maximum achievable number of clauses to within 1% is NP-hard [1]. QED.

Remark 14 *It is certainly not a new observation that quadratic programming is NP-hard, e.g. see [18], and see [19] for an NP-hardness result about local optimality. But the present result is for the most part stronger than previous ones.*

7 Continuous minimization of rational functions; unconstrained and in balls

In general if F is a multivariate polynomial (or rational function more generally), then minimizing it is in the class NP in the sense that the min may be described in a polynomial number of bits (and the fact that the value of this min lies below some given rational number, may be verified in a polynomial number of operations):

Theorem 15 (In NP) *Given a polynomial $F(\vec{x})$ of degree m in d real variables, with integer coefficients, and a d-ball B with rational center and radius. (This radius is allowed to be ∞.) In a number of bits bounded by a polynomial of the number of bits in the input (assuming the polynomial is represented "densely" with the coefficients as binary integers) it is possible to write down coordinates for the center and the radius of a ball B^* guaranteed to contain a global minimum $\vec{x}^*$ of F in B (if one exists – the min could also be $-\infty$ and occur on the sphere at ∞), and such that F is concave-$\cup$ within B^* so that any additional number N of decimal places of accuracy may then be obtained in time polynomial in m, d, and N.*

Proof. On any rational line, F is a 1-variable polynomial of degree d with integer coefficients. Employing lemma 16 below (item 2), we see that the separation s between any two inflection points of F along a rational line defined by coefficients that are $\le N$ decimals long, obeys $|\log s| < P$ where P is a polynomial in N, d, m, and the number of bits in the input. This proves that the ball size may be described while only using a polynomial number of bits, and the ball center, by claim 1 of the same lemma, is also concisely describable. □

[9] Subtract $K|\vec{x}|^2$ from the function for K sufficiently large that its minima are forced to lie at the vertices of the hypercube (here assuming the origin is at the center of the cube) rather than in its interior.

Lemma 16 (Polynomial root bounds and root separation) *A 1-variable monic polynomial of degree m with integer coefficients $\le B$ in absolute value, obeys*

1. *The absolute value of the largest root is $\le 1 + B$,*
2. *If all the roots lie in a ball of radius R, then the separation s between the two closest nonidentical roots obeys $s \ge R^{-(m-2)(m+1)/2}(m-1)^{-(m-1)/2}2^{-(m-2)}$.*

Proof of lemma. Claim 1 follows from Gershgorin's circle theorem. Specifically: Gershgorin's theorem (6.1.1 of [11]) states that the spectrum of matrix M is contained within the union of the circular disks $|z - c - i| \le r_i$ in the complex plane with centers $c_i = M_{ii}$ and radii $r_i = \sum_{j \ne i} |M_{ij}|$.

The companion matrix M of a polynomial $P(z) = z^m - \sum_{i=0}^{m-1} a_i z^i$ is the $m \times m$ matrix with the a_i in order down its rightmost column, 1s on its subdiagonal ($m-1$ of them in all), and 0s elsewhere. Because $P(z) = \det(M - zI)$ (as may be shown inductively by Laplace's expansion by minors), the spectrum of M is the same as the roots of $P(z)$. By applying Gershgorin's theorem to the companion matrix we see that all roots x of $P(z)$ obey $|x| \le \max_i |a_i|$.

To prove the second claim (which improves upon previous results), we wlog consider the GCD of the polynomial and its derivative in order to get rid of multiple roots. We consider the "discriminant" D of this polynomial, which may be written in two ways, first as the product $D = \prod_j \prod_{i \ne j}(z_i - z_j)$ of all the root separations, and secondly as a determinant. Since this determinant has integer entries and $D \ne 0$, $|D| \ge 1$. On the other hand, the largest that the product form of $|D|$ could be, subject to the constraint that all the z_i lie in the disc of radius R, $R \le 1 + B$, is when the z_i are the minimal energy locations of m free $+1$ point charges with logarithmic potential energies (i.e. inverse linear force laws) in that disc, which occurs when all the charges lie on the boundary of the disc and form a regular m-gon.

This is by considering $\ln |D|$. When m is finite, the fact that the potential energy seen by one charge when all the others are held fixed, is harmonic and tending to $+\infty$ if two charges approach one another, implies that its minimum occurs on the boundary of the enclosing disc, hence all the charges migrate to the boundary. Once they are there, it seems "obvious" that the best configuration is a regular m-gon. This "obvious" fact is provable by using the fact that the logarithmic potential function $\log |\csc \theta|$ is *concave-*$\cup$ because its second θ derivative is $\csc^2 \theta > 0$. Consequently, the energies associated with nearest-neighbor charge interactions ("nearest" angularly, that is) are minimized when the charges are *equally spaced.* The same argument may be used for the interaction energies between (angularly) second-nearest neighbor charges, between third-nearest-neighbors, and so on. At every level, an energetically best configuration is a regular m-gon, and at the first level this optimum is unique. Consequently, this configuration is also uniquely optimum for the whole level-combined energy function.

This shows $|D| \le R^{(m-1)m} m^m$ with the upper bound being attained if and only if the polynomial is $z^m - R^m$. Now even if the product of all the root separations among $m-1$ of the m roots were as large as possible (e.g. meeting the upper bound $R^{(m-2)(m-1)}(m-1)^{m-1}$) and even if the remaining mth point were at maximal distance $2R$ from all the others except for one, then in order to keep $|D| \ge 1$ the remaining separation s could not be too small. Specifically $(2R)^{2(m-2)} R^{(m-2)(m-1)}(m-1)^{m-1} s^2 \ge 1$, leading to the claimed bound. □

Claim 1 of the above lemma is not new and, e.g., may be found at the beginning of theorem 4.2 page 146 of [16], which goes on to prove stronger bounds. Claim 2 is new and, e.g., gives a bound stronger in both terms than the one in theorem 4.6 page 166 of [16], which nevertheless would still have been strong enough for our purposes. More such bounds are in [17].

Regarding the question of how tight these bounds are: the first is tight to within a factor of 2 for the polynomials $z^m - 1$. Mignotte's polynomials $P(z) = z^m - 2(az-1)^2$ with integer $a \ge 10$, $m \ge 3$ and where we consider the regime $a, m \to \infty$ with $a = m^{O(1)}$, have all roots within a disk, centered at 0, of radius R with

$$\max\left\{2^{1/m},\ (4a/m)^{1/(m-1)},\ (\frac{4a^2}{(m-1)m})^{1/(m-2)}\right\} < R < ([2 + o(1)]a^2)^{1/(m-2)} \tag{5}$$

(and hence $R \to 1$) and has two real roots both within distance $h = a^{-(m+2)/2}$ of $1/a$, i.e. the minimum separation s between two roots obeys $s < 2h$. These polynomials are irreducible over the rational numbers.[10] Our lemma's bound gives

$$s > R^{-(m-2)(m+1)/2}(m-1)^{-(m-1)/2}2^{-(m-2)} \tag{6}$$

but our upper bounds on s and R for Mignotte's polynomials give

$$s < R^{-(m-2)(m+2)/4}[2 + o(1)]^{(m+6)/4} \tag{7}$$

[10] The irreducibility is proven with Eisenstein's test. The fact that $P(1/a) = a^{-m} > 0$ while $P(a^{-1} \pm h) = 2a^{-m} - 2a^2h^2 = 0$ proves the root separation claims. The upper bound on R arises from balancing the two highest degree terms in $P(z)$ while neglecting the others (or directly from one of the cases of Mignotte's theorem 4.6 page 166 [16]) and $R > 2^{1/m}$ is since, e.g., the product of P's m roots is 2.

In other words, the *logarithm* of claim 2's bound is, infinitely often, tight to within a multiplicative factor arbitrarily near 2.

Theorem 17 (Complexity of minimizing rational functions) *Let $F(\vec{x})$ be a rational function of d real variables $\vec{x}$ with rational numbers as coefficients. Let the sum of the total degrees of the numerator and denominator be D. For example $R(x,y) = (9x^3 + xy^4 + 1)/(17 + x^2y)$ would have $D = 5 + 3 = 8$.*

If $D \le 4$ then there is an algorithm running in time polynomial in d, D, A, and the total number of decimals in the input, to determine all the local minima of F to A decimal places of accuracy. The polynomial is individually linear in the number W of local minima, and the number A of desired decimal places. Even more strongly, not only can we handle minimizing rational functions $P(\vec{x})/Q(\vec{x})$ where P and Q are polynomials with $\max\{\deg P, \deg Q\} \le 3$, but in fact, our algorithms work almost unchanged to minimize $P(\vec{x})^p/Q(\vec{x})^q$ where the powers p and q are any fixed rational numbers (these rational numbers must have odd denominators if we plan to power negative real numbers). In the cases $3+0$, $2+1$, $2+0$, $1+1$, $1+0$ the number W of local minima obeys $W \le 1$.

However, there is an exception. In the cases $D = 4+0$ and $D = 0+4$ (of a quartic polynomial and its reciprocal) it is NP-hard to tell whether the global min of F (or the global min within a specified ball) lies below a given value. Indeed it is NP-complete even to decide whether $\vec{x} = \vec{0}$ is a local *minimum of F (versus a "saddlepoint"). Indeed even determining the global minimum value to within a sufficiently small additive or multiplicative constant is NP-complete.*

Theorem 17 actually just summarizes a large number of individual theorems we'll now state and prove.

1+0: Obviously a linear function has no local minima.

1+1: If $F(\vec{x})$ is a nonconstant ratio of two linear polynomials $L_1(\vec{x})/L_2(\vec{x})$, then it has no finite minimum. Proof: If $\vec{\nabla}L_1$ and $\vec{\nabla}L_2$ are parallel, then this is just the 1-dimensional problem of minimizing $(ax+b)/(cx+d)$ which one can easily see has no minimum since its derivative is $(ad-bc)/(cx+d)^2$ which is everywhere nonzero (or everywhere zero, but we have disallowed constant F). If they are nonparallel, then at a "minimum" of $F(\vec{x})$, we could perturb $\vec{x}$ in a direction perpendicular to $\vec{\nabla}L_2$. Either this perturbation or its negative would decrease L_1, a contradiction. □

2+2, 2+0:

Theorem 18 *In time polynomial in N and d one can determine whether a given quadratic function of d real variables (with rational coefficients) has a unique minimum, and if so, find it to N decimals of accuracy. Also in polynomial time one can determine the minimum of such a quadratic on (or in) a given d-dimensional sphere (whose center and radius are given rationals), or indeed on any surface described by a quadratic equation.*

Proof. The quadratic has a unique unconstrained minimum if and only if it is positive definite, i.e. all the eigenvalues of its Hessian matrix are > 0. (This can be checked in polynomial time, by, e.g. verifying that all determinants of diagonal minors are nonnegative.) Letting the quadratic be $F_0 + \vec{a}\cdot\vec{x} + \vec{x}^T H \vec{x}$, the minimum then occurs at the solution $\vec{x}$ of the d simultaneous linear equations $(H + H^T)\vec{x} + \vec{a} = \vec{0}$ (saying that the gradient is $\vec{0}$) which may be solved (solution $\vec{x} = (H + H^T)^{-1}\vec{a}$) by Cholesky's method [9].

For the problem of minimizing a quadratic on a sphere $|\vec{x} - \vec{c}| = R$, the location $\vec{x}$ of the minimum must satisfy

$$(H + H^T)\vec{x} + \vec{a} = (\vec{c} - \vec{x})\kappa, \qquad |\vec{c} - \vec{x}|^2 = R^2 \tag{8}$$

for some positive scalar Lagrange multiplier κ. Indeed, more generally minimizing a quadratic $Q_1(\vec{x})$ subject to *any* desired quadratic equality constraint $Q_2(\vec{x}) = 0$, leads to

$$\vec{\nabla}Q_1(\vec{x}) + \kappa\vec{\nabla}Q_2(\vec{x}) = \vec{0}, \qquad Q_2(\vec{x}) = 0. \tag{9}$$

In both cases we may solve the first equation (which is a *linear* system) for $\vec{x}(\kappa)$, getting a rational function of κ of numerator and (common) denominator degree $\le d$. We may then find the κ that works in the second equation by solving [21] a univariate polynomial equation of degree $\le 2d-1$ (arising by setting the derivative of our rational function to 0), then examining all the $\le 2d-1$ possibilities.

Even if the $(d+1)d/2+1$ coefficients of the quadratic $F(\vec{x})$ are not initially known, but a black box is available for computing F, then they may be deduced by evaluating F at a suitable set of $\vec{x}$'s and interpolating. □

Remark. By an affine transformation of d-space it is possible to simultaneously diagonalize the Hessian matrices of the two quadratics Q_1 and Q_2. In the sphere case where $Q_2(\vec{x}) = |\vec{x} - \vec{c}|^2$, the Hessian of Q_2 is already diagonal (indeed it is the identity matrix) and the required affine transformation is merely a rotation of the basis to become the orthogonal eigenvectors of $H + H^T$. The appropriate affine in the general case may be found in the same way after first affining Q_2 so that it becomes spherical; unfortunately if Q_2 is indefinite, solution of a "generalized eigenproblem" will be required.[11]

The advantage of using the resulting transformed version of the problem, is that the linear system then becomes trivial to solve.

Now it is possible to optimize the ratio of two quadratics by minimizing the numerator with the denominator held fixed (say set equal to K) using the method of the above theorem. Then, we get a 1-dimensional "curve of minima" as a function of K (or a set of $\le d$ such curves). We can then perform 1D minimization by, e.g. golden section search [22] along these curves to find the global minimum.

2+2′: An alternate approach, which also works, is to find the (generically unique) min or saddlepoint of $Q_1(\vec{x}) - \mu Q_2(\vec{x})$ as a function of the Lagrange multiplier μ, which is a different parameterization of the "curve of minima." This method has the advantage that no eigenproblems need to be solved, only linear system solving is required (set $\vec{\nabla} = \vec{0}$).

The situation is even simpler in the **2+1** case of a quadratic divided by a linear. Setting the linear equal to K restricts us to a hyperplane; we minimize the quadratic within the hyperplane (a $d-1$-dimensional problem). Then we get a "curve of minima" $\vec{x}(K)$ which we can optimize over. It is also possible to solve the **2+1** problem by pre-transforming (by a suitable affine) to a basis in which the quadratic has diagonal Hessian. In that case, the curve of minima will simply be a *line* in the $\vec{\nabla}L$ direction. Then by un-affining, we see this curve of minima is *always* just a line; and the minimization problem restricted to this line is simply minimizing a degree-$(2+1)$ univariate rational function.

In the **2+2** case the curve is a rational function with numerator and (common) denominator degrees $\le d$, and the 1D minimization thus may be performed by solving a univariate polynomial equation of degree $\le 2d-1$.
4+0: On the other hand, quartics are hard to optimize.

Theorem 19 *Determining whether the global minimum value of a quartic multivariate polynomial with integer coefficients lies below given some threshold K (or even determining the minimum value to within an additive amount ± 1, or to within a multiplicative constant factor $1+\delta$, for some sufficiently small absolute constant $\delta > 0$) is strongly*[12] *NP-complete.*

Proof. $p(x) = (x^2-1)^2$ has 2 local minima at $x = \pm 1$, with value 0. So

$$\text{quartic}(x) = p(x_1) + p(x_2) + \ldots + p(x_d) \tag{10}$$

has 2^d local minima at the vertices "$(\pm 1)^d$" of a d-hypercube. Now, add $\epsilon\text{quadratic}(x)$ to this quartic where ϵ is very small. We use the quadratic of theorem 13. The heights of the minima will be perturbed by $\epsilon\text{quadratic}(x) + O(\epsilon^2)$ and their locations will be perturbed by $-\frac{\epsilon}{8}\vec{\nabla}\text{quadratic}(x) + O(\epsilon^2)$. It will now suffice to take $\epsilon = 0.001/S$ where S is the sum of the absolute values of the (bounded integer) coefficients of quadratic(x), and then rescaling all the coefficients to integers by multiplying by $1000S$ still leaves the mean square coefficient bounded by an absolute constant, if $S = \Omega(d^2)$. □

Indeed,

Theorem 20 *Determining whether the origin is a local minimum of a multivariate quartic polynomial (as opposed to a "saddlepoint") is strongly NP-complete.*

Remark 21 *It is easy to modify theorem 19 and its proof to see that minimizing a quartic on a sphere or in a ball also is NP-hard.*

Proof. Regard the vectors of $(d+1)$-dimensional real space as 2-tuples $(\vec{x}, z)$ where $\vec{x} \in \mathbf{R}^d$. Let $Q(\vec{x})$ be the d-variate quartic from theorem 19. Then

$$z^4 Q(\frac{\vec{x}}{z}) \tag{11}$$

is a $(d+1)$-variate quartic such that the origin $(\vec{x}, z) = (\vec{0}, 0)$ is a local minimum if and only if the global minimum of $Q(\vec{x})$ is positive. □

[11] To be precise, if the two symmetric Hessian matrices of our quadratics are A and B, then if the symmetric matrix $B^{-1/2}AB^{-1/2}$ has eigendecomposition $Q^{-1}\Lambda Q$ where the eigenvalues are the entries of the diagonal matrix Λ and the (orthonormal) eigenvectors are the rows of $Q = Q^{-T}$, then the linear coordinate transformation $\vec{x} \to QB^{1/2}\vec{x}$ will cause the two quadratics (as re-expressed in the new coordinates) to have Hessians Λ and I respectively – both diagonal. $B^{1/2}$ is real if B is semidefinite; but if B is indefinite, this won't work. However, even then a suitable linear coordinate change always exists which will simultaneously diagonalize A and B; this is the problem of "diagonalizing a regular symmetric matrix pencil" discussed in §8.7.1 of [9]. Theorem 8.7.1 there shows how to find such a coordinate change if A is positive definite, *regardless* of the definiteness of B. Indeed even if A also is indefinite, provided *some* convex combination $\mu A + (1-\mu)B$, $0 \le \mu \le 1$, is definite, we can find such a coordinate change.

[12] That is, still NP-complete even if the coefficients of the polynomial are required to be written in unary.

3+0: With theorem 18 showing that optimizing quadratics is easy and 19, 20 showing that optimizing quartics is hard, the obvious question then is: what about cubics? Well, since every nondegenerate cubic achieves unboundedly large positive and negative values, it is only interesting either to talk about local minima, or to minimize the cubic in a compact set, for example on a d-dimensional sphere. The following causes one to suspect that local minimization of cubics might be tractable.

Lemma 22 *A cubic polynomial in d real variables has at most one local minimum and at most one local maximum.*

Proof. Suppose not. Draw a line containing the "two" local minima. Restricted to this line we have a 1-variable cubic, which cannot have 2 minima! $\square$

In fact, this suspicion of tractability is correct; the appropriate converse of theorem 20 is

Theorem 23 *Given a cubic function of d real variables (with rational coefficients) there is an algorithm whose running time is polynomial in N, d, and the number of bits in the input, which will find the local minimum of the cubic (if it exists, otherwise it will report nonexistence) to A decimals of accuracy.*

Proof. The Hessian matrix (2nd derivatives) of the cubic has *entries* which are *linear* functions of the d coordinates. A function (in particular our cubic) is "locally concave-∪" at a point $\vec{x}$ if its Hessian is positive semidefinite at $\vec{x}$. The set $\mathbf{R}^{d\times d}_{\mathrm{sym}\geq 0}$ of $d \times d$ symmetric matrices which are positive semidefinite is a convex subset of the set $\mathbf{R}^{d\times d}_{\mathrm{sym}}$ of all $d \times d$ symmetric matrices.[13] If we want to find the set of coordinates in $\mathbf{R}^d$ at which the cubic is locally convex, this is the intersection of an affine version of $\mathbf{R}^d$ inside $\mathbf{R}^{d\times d}_{\mathrm{sym}}$ (arising from the linear functions of $\vec{x} \in \mathbf{R}^d$ that arise from our particular cubic) with this semidefinite set, hence (since the intersection of 2 convex sets is convex) is also a convex set. So we've shown

Lemma 24 *The locus of points where a cubic is locally concave-∪ is a convex set.* $\square$

Indeed the locus of points $\vec{x} \in \mathbf{R}^d$ where the Hessian $H(\vec{x})$ of a cubic has minimum eigenvalue $\lambda_{\min}(H(\vec{x})) \geq K$, is a convex set for any real K, and indeed $\lambda_{\min}(H)$ is a concave-∩ function of H ([2] theorem 30 p.85; [15]; there is however a much simpler proof of this by simply employing the Courant-Fischer characterization of $\lambda_{\min}(H)$ as the minimum value of of $\vec{x}^T H \vec{x}$ over all $\vec{x}$ with $|\vec{x}| = 1$, and considering a convex linear combination of two symmetric matrices pre-scaled to have equal $\lambda_{\min}$).

In fact, by means of "convex programming" we can in polynomial time find a point in this locus, if one exists (in fact this particular case is called "semidefinite programming" and has its own literature [28]).

Now indeed we claim we can find the point, within this locus, which minimizes, to A decimal places, the value of any particular concave-∪ function that we desire, in time polynomial in d and A. In particular – within this locus the cubic itself is concave-∪ so we can minimize it with semidefinite programming! The result will be the local min of the cubic, if one exists (since of course, this min must lie in this locus). To accomplish this, several approaches are possible, including finding the "analytic center" (or volumetric center) of the locus and then splitting it along a hyperplane orthogonal to the gradient of the cubic at that centerpoint.

By a theorem originally by P.Vaidya [26] (see also the review [20]) this reduces the volume quickly enough to assure polynomial time.

A different way of looking at the minimization algorithm for cubics $F(\vec{x})$ is that we minimize a *different* function, namely

$$\tilde{F}(\vec{x}) = \mathrm{Pos}(-\lambda_{\min}(H(\vec{x}))) + \epsilon\, F(\vec{x}) \tag{12}$$

over $\vec{x}$, where $\mathrm{Pos}(t) \equiv \max(t, 0)$, in the limit $\epsilon \to 0^+$. This function is concave-∪. This may also be thought of as "minimizing" a 2-tuple-valued function

$$\Big(\mathrm{Pos}(-\lambda_{\min}(H(\vec{x}))),\ F(\vec{x})\Big) \tag{13}$$

with lexicographic ordering among real 2-tuples. (This function is "unimodal on lines" with this ordering notion.)

3+1: To minimize the ratio $F(\vec{x}) = C(\vec{x})/L(\vec{x})$ of a cubic and a linear (or indeed $F(\vec{x}) = C(\vec{x})^p/L(\vec{x})^q$ for suitable fixed rational numbers p, q), we consider the hyperplanes $L(\vec{x}) = K$. We minimize $C(\vec{x})$ within each such hyperplane, getting an algebraic "curve of minima" $\vec{x}(K)$. Any point on this curve is polytime computable. Then again by using a 1D minimization procedure we can find all the minima.

Actually, 1D minimizers such as derivative-aided interval bisection can assure finding *a* local minimum to A decimal places in polynomial(A) time, but cannot assure finding *all* local minima quickly, so our present proof is not quite complete. However, one can find *all* the local minima and maxima of any differentiable function $f(x)$ on any

[13] A matrix M is positive definite if $\vec{x}^T M \vec{x} > 0$ for all $\vec{x} \neq \vec{0}$. Hence plainly the sum, or any convex linear combination, of two positive definite matrices is positive definite, hence $\mathbf{R}^{d\times d}_{\mathrm{sym}\geq 0}$ is a convex subset of $\mathbf{R}^{d\times d}_{\mathrm{sym}}$.

real interval by using bisection to find one local min (or max), then applying the same procedure recursively to find the other kind of extrema within the two subintervals with endpoint at that min or max, and so on. Provided there is an a priori upper bound W on the number of minima and maxima total ("wiggles"), this method will terminate after making $O(AW)$ function-derivative calls, having located all W extrema to A decimals of accuracy.

Unfortunately, we have no a priori bound on W in our case (except when $q = 0$, a pure-cubic polynomial, or when $q = 3p$, when we have a pure-cubic polynomial function of a projective transformation of $\vec{x}$; in both cases there is at most one local minimum of the function, as you may see by considering its restriction to a line). Unfortunately the "curve of minima" might have exponentially large algebraic degree since each point on it is defined by the solution of simultaneous quadratic equations, and hence naive degree-based bounds yield only very weak algorithm runtime bounds. Indeed, for all we presently know there in fact might be a huge number of local minima of the objective function, in which case at least that many wiggles must occur. We can reassure ourselves in that case that our algorithm is *output sensitive*, i.e. runs in time polynomial in the input size, *and* the output size (reckoning both the number of local minima, and the number of decimals carried, in the latter). That is because (and please realize this) every 1D local min of the objective function on the curve of minima necessarily corresponds to a full-multidimensional local min.

In practice, however, W often is small.

The reciprocated cases $\mathbf{a} + \mathbf{b}$ are essentially the same as the cases **b+a** so need not be considered separately. □

8 Abstract classes of efficiently minimizable functions

Although we developed our techniques for minimizing multivariate rational functions, they have more general applicability. For example, our method for finding all the minima of the *quotient* of two cubics also could be used to minimize their *product*.

Let us call a class of almost-everywhere-differentiable continuous d-variate functions (with black-box function and gradient availability) "efficiently minimizable" if an algorithm, running in time polynomial in W, A, d, exists to find all W minima of the function (and W is necessarily finite) accurate (in the sense that th enclosing ellipsoid volume is $\le 10^{-Ad}$) to A decimal places.

Theorem 25 *The following summarizes the abstract results attainable via the minimization techniques in this paper.*

1. Functions unimodal on lines are efficiently minimizable within convex sets specified by separation oracles by the ellipsoid or Vaidya methods.

2. 1D continuous functions are efficiently minimizable by interval-subdivision methods (output sensitive runtime, all mins in a bracketing interval found).

3. If $F(\vec{x}, y)$ is efficiently minimizable (and with at most a polynomially large number of minima) when y is held fixed, then F is efficiently minimizable by 1D minimization along the curve of minima (output sensitive runtime).

4. If $F(a, b)$ is a continuous function individually monotonic in each of its two arguments if the other is held fixed,[14] if $G(\vec{x})$ is efficiently minimizable on hyperplanes and with at most a polynomially large number of minima,[15] and if $H(\vec{x})$ has hyperplanes as level sets,[16] then $F(G(\vec{x}), H(\vec{x}))$ is efficiently minimizable by our Lagrange multiplier with "curve of minima" methods.

5. If $F(a, b)$ is a continuous function individually monotonic in each of its two arguments if the other is held fixed, and $G(\vec{x}) + \mu H(\vec{x})$ is efficiently minimizable regardless of the value of the constant μ and with at most a polynomially large number of minima, and which either cannot have saddlepoints or can only have at most a polynomially large number of them all of which are easily found,[17] then $F(G(\vec{x}), H(\vec{x}))$ is efficiently minimizable by our Lagrange multiplier with "curve of minima" methods.

6. If the set S of $\vec{x}$ within which $F(\vec{x})$ is concave-∪, is convex, and if there exists some efficiently minimizable function $G(\vec{x})$, efficiently constructible if $F(\vec{x})$ is known, such that S is also describable as the set of $\vec{x}$ with $G(\vec{x}) \le 0$, then: $F(\vec{x})$ is efficiently minimizable by our "two-stage minimization" method. (This includes: not only cubic multivariate polynomials, but also, any function of $\vec{x}$ whose Hessian matrix is a ratio of a linear symmetric-matrix-valued function of $\vec{x}$, divided by a linear scalar-valued function of $\vec{x}$, provided we are optimizing over a convex region in which the denominator linear function stays positive.)

[14] For example, this class includes: $a \pm b$, ab, a/b, $\max(a, b)$, $\min(a, b)$, and various power-combinations.

[15] For example, this includes the functions unimodal on lines.

[16] This class includes not only the linear functions, but also such things as $\arg(x + iy)$ and more generally all functions that are the arrival times of a hyperplane traveling through space (in domains where these are uniquely defined).

[17] For example, if G and H both are quadratics.

So while at the beginning of this paper we said it was an "embarrassment" that Computer Science knew how to minimize so few functions, the situation now is somewhat better: we now are starting to have a fairly large and juicy set of efficiently minimizable functions, and indeed the recognition of such functions from their formulae, and the consequent design of the minimization algorithm, are now fairly mechanical tasks. But on the other hand, we know this class cannot be extended too far because for quartic polynomials, we cannot even confirm a given point is a local min efficiently (unless P=NP), and we've provided fairly convincing evidence (although not a proof) that unimodal Lipshitz functions are a larger class of functions than the efficiently minimizable ones.

Finally, let us argue that our algorithms may be useful. Many heuristic numerical algorithms [4][5][22] work by somehow approximating the function locally by a quadratic, then going toward the minimum of that quadratic. We now see that approximation by (and minimization of) higher degree rational functions also would be feasible.

Furthermore, many heuristic numerical unconstrained minimization algorithms [4][5][22] have *difficulty* with the case where their local quadratic model of the objective function, has an *indefinite Hessian*. The BFGS quasi-Newton method [4] has an ingenious positivity-preserving update formula which simply refuses to allow the model's Hessian to stop being positive definite. That "solution," however, is really just closing one's eyes to the problem. It presumably would be superior to *admit* the Hessian is indefinite wherever it is, and then either try to walk down the steepest descent curve (which initially goes in the negated gradient direction and ultimately goes in a direction of an eigenvector with maximally negative eigenvalue) – or to seek a different location at which the Hessian *is* positive definite! The latter goal could be accomplished by making a linear (or even ratio of two linears) model of the Hessian, then walking up the steepest ascent curve for the minimum eigenvalue of that model (which could be found by using a subsidiary convex-$\cap$ function maximization within balls).

9 Open questions

1. A top deficiency of the present paper is that we do not know how many local mins the ratio of two multivariate cubics (or a cubic and a quadratic, or even a cubic and a linear) can have. The question of whether minimizing a cubic $F(\vec{x})$ on a sphere or in a ball is in P is still open. An equivalent or simpler question is: what is the ray along which F decreases asymptotically maximally quickly toward $-\infty$ as $|\vec{x}| \to \infty$? A simpler question is merely to determine (or get good bounds on) $M_3(d)$, where

Definition 26 *$M_3(d)$ is the maximum number of absolute local minima (that is, minima which are in fact the global minimum in some neighborhood) that a cubic function of d real variables can have on $|\vec{x}| = 1$.*

Here are the best bounds I know (but asymptotically, they are exceedingly weak):

Theorem 27 *$M_3(0) = 1$, $M_3(1) = 2$, $M_3(2) = 3$, $M_3(3) \geq 4$. If k is a non-negative integer, $M_3(2k) \geq 3k$ and $M_3(k) \leq 8 \cdot 4^k$.*

Proof. The lower bound $M_3(2k) \geq 3k$ follows by consideration of $p(x, y) = 3xy^2 - x^3 = -\mathrm{Re}[(x + iy)^3]$ when $k = 1$, and $p(x_1, x_2) + p(x_3, x_4) + ... + p(x_{2k-1}, x_{2k})$ when $k > 1$. Indeed, all the $3k$ minima in this example have equal heights. The fact that this lower bound is tight when $k = 1$ may be shown by analysing 2D cubics on the unit circle as 1D rational functions on the real line (by stereographic projection). The locations of the extrema then correspond to the real roots of a polynomial of degree 6, hence there are at most 6 of them, and at most 3 of these can be minima. The fact that $M_3(3) \geq 4$ arises from $p(x, y, z) = 3xy^2 - x^3 - 5xz^2$. This is a modification of $p(x, y)$ by the additional term $-5xz^2$, which does not affect the two minima at $x + iy = \exp(2\pi i/3)$, but splits the minimum at $x + iy = 1$ into two minima lying on opposite sides of the xy equatorial plane.

We will use a theorem of Hugh Warren [29] that the maximum number of connected components of a level set of a polynomial of degree n in R^m is $\leq 2n^m$. Hence, consider the sum of the squares of the k quadratics constituting the gradient of a cubic in $\mathbf{R}^k$, plus the $(k + 1)$th quadratic $|\vec{x}|^2 - 1$ whose 0-set defines the unit sphere. This sum is zero at a local min or max. We apply Warren's bound with $m = k + 1$ and $n = 4$ to get $M_3(k) \leq 8 \cdot 4^k$. $\Box$

2. I conjecture [27][24][12] that RP$\neq$PLS$\cap$UP$\cap$co-UP; indeed that the promise problem of minimizing polynomial time functions of N-bit strings, promised to have exactly 1 local min (which is also the global min) in the Hamming (bit flip) neighborhood structure, is not soluble by an RP algorithm. Completeness results about this problem are open.

3. A natural generalization (if $k > 1$) of "unimodal on lines" is "having at most k minima in any translate of a k-dimensional subspace." Functions with this property have at most k minima (Proof: if they had $k + 1$, consider a k-space spanning them). Under what conditions can *these* functions be minimized quickly?

10 Acknowledgements

Thanks to H. Cejtin, L. Gurvits, D.W.Jacobs, and S.M. Omohundro for helpful conversations.

References

[1] S.Arora: Hardness of Approximations, in Approximation Algorithms for NP-hard Problems, PWS Publishing 1996.

[2] E.F.Beckenbach & R.Bellman: Inequalities, (2nd revised printing) Springer 1965.

[3] G.Chaitin: Algorithmic information theory, Cambridge Univ.Press 1987.

[4] J.E.Dennis & R.B.Schnabel: Numerical methods for unconstrained optimization and nonlinear equations, Prentice-Hall 1983.

[5] R.Fletcher: Practical methods of optimization, 2 volumes, J.Wiley 1980; second edition 1987.

[6] Christodoulos A. Floudas: Deterministic global optimization, Kluwer Academic 1999.

[7] D.H. Fremlin: Measure Theory, multivolume book available electronically from `http://www.essex.ac.uk/maths/staff/fremlin/mt.htm`. Also published (ISBN=0-9538129-2-8) by Torres Fremlin. (Contact M.A.Fremlin, 25 Ireton Road, Colchester CO3 3AT, England; `maria@fremlin.de`.)

[8] M.R.Garey, D.S.Johnson, L.Stockmeyer: Some simplified NP-complete graph problems, Theor.Comput.Sci 1 (1976) 117-129.

[9] G.H.Golub & C.Van Loan: Matrix computations (3rd ed), Johns Hopkins University Press (3rd ed.) 1996.

[10] M.Grötschel, L.Lovasz, A.Schrivjer: Geometric algorithms and combinatorial optimization, Springer 1988.

[11] R.A. Horn & C.R. Johnson: Matrix Analysis, Cambridge Univ. Press 1985.

[12] D. S. Johnson, C. H. Papadimitriou, M. Yannakakis: How easy is local search? J. Computer & System Sciences 37,1 (Aug 1988) 79-100.

[13] J.L.Kelley & T.P.Srinivasan: Measure and Integral, Springer (GTM 116) 1988.

[14] Ker-I Ko: Complexity Theory of Real Functions, Birkhauser, 1991.

[15] P.D.Lax: Differential equations, difference equations, and matrix theory, Commun. Pure & Applied Math. 10 (1958) 175-194.

[16] Maurice Mignotte: Mathematics for computer algebra, Springer-Verlag 1992.

[17] Maurice Mignotte, Doru Stefanescu: Polynomials: an algorithmic approach Springer 1999.

[18] Katta G. Murty & Santosh M. Kabadi: Some NP-complete problems in quadratic and nonlinear programming, Mathematical Programming 39 (1987) 117-129.

[19] P.M.Pardalos & G.Schnitger: Checking local optimality in constrained quadratic programming, Oper. Res. Lett. 7,1 (1988) 33-35.

[20] Y.E.Nesterov & A. Nemirovsky: Interior point polynomial methods in convex programming, SIAM (vol. 13 of "Studies in applied mathematics") 1994.

[21] Victor Y. Pan: Solving a Polynomial Equation: Some History and Recent Progress, SIAM Review 39,2 (1997) 187-220.

[22] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 1992. Second edition 1997.

[23] W.Rudin: Principles of Mathematical Analysis, McGraw-Hill 1976.

[24] A. A. Schäffer & M. Yannakakis: Simple local search problems that are hard to solve, SIAM J. Comput. 20,1 (1991) 56-87.

[25] Ernst P. Specker: Der satz vom Maximum in der rekursiven Analysis, pp. 254-267 in "Constructivity in Math." (ed. A.Heyting) Stud.Logic Found. Math. North-Holland Amsterdam 1959.

[26] Pravin M. Vaidya: A new algorithm for minimizing convex functions over convex sets (extended abstract). Pages 338-343 in 30th Annual IEEE Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, 30 October-1 November 1989.

[27] L.G.Valiant & V.V.Vazirani: NP is as easy as detecting unique solutions, Theor. Comput. Sci. 47 (1986) 85-93.

[28] L.Vandenberghe & S.Boyd: Semidefinite programming, SIAM Review 38,1 (1996) 49-95.

[29] H.Warren: Lower bounds for approximation by nonlinear manifolds, Trans. Amer. Math. Soc. 133 (1968) 167-178.