Black-box Complexity of Local Minimization

Stephen A. Vavasis*

TR 90-1132
June 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Supported in part by the Applied Mathematical Sciences Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25013.A000.

Black-box complexity of local minimization

Stephen A. Vavasis*
Department of Computer Science
Cornell University

June 4, 1990

Abstract

We study the complexity of local minimization in the black-box model, that is, the model in which the objective function and possibly its gradient are available as external subroutines. This is the model used, for example, in all the optimization algorithms in the 1983 book by Dennis and Schnabel. Our first main result is that the complexity grows polynomially with the number of variables n, in contrast to other related black-box problems (global minimization, Brouwer fixed points) for which the worst case complexity is exponential in n.

Our second contribution is the construction of a family of functions that are bad cases for all possible black-box local optimization algorithms.

1 Black box model

Numerical optimization refers to the problem of minimizing a continuous function $f : D \rightarrow \mathbb{R}$ where D is a subset of $\mathbb{R}^n$. Except for convex problems,

*Supported in part by the Applied Mathematical Sciences Program (KC-04-02) of the Office of Energy Research of the U. S. Department of Energy under grant DE-FG02-86ER25013.A000.

most optimization algorithms will not return global minima; instead, they will return (at best) local minima.

It is therefore natural to inquire about the complexity of local minimization for general nonconvex objective functions. In order to make general statements about local optimization, it is necessary to have definitions of valid objective functions and of "approximate" local minima. These definitions will be the subject of most of the rest of this introduction. To our knowledge, this paper is the first attempt to define approximate local minimization.

The remainder of the paper is organized as follows. In Section 2 we present the first main result of this paper, that is, a simple algorithm to find an approximate local minimum. Its running time is polynomial in n (the number of variables) and M/ϵ (see below for an explanation). In Section 3 we present the second main result, a family of functions that constitute a bad case for minimization algorithms. These functions lead to a lower bound that is polynomial in M/ϵ. In Section 4 we give an algorithm with a better bound for some values of the parameters. In Section 5 we compare our bounds to the bounds known for global minimization and Brouwer fixed points (a closely related problem).

The model of computation will be a real-number model. We assume that the algorithm can store and compute exact real numbers. We assume that the objective function f is provided by the user via a subroutine. This subroutine takes as input a vector $\mathbf{x} \in \mathbb{R}^n$ and returns a real number $f(\mathbf{x})$. We assume for this work that f is continuously differentiable. We assume that the gradient ∇f is also available as a subroutine (although see further remarks on this in Section 2). Some of the algorithms that fall into this category are the steepest descent method, the Powell-symmetric-Broyden method, the Broyden-Fletcher-Goldfarb-Shanno method, and the line-search and trust-region modifications of these algorithms. See Dennis and Schnabel [1983] for more information.

This model of computation is known as a "black-box" model, a "function-evaluation" model, or an "oracle" model. The key limiting feature is that global information about f is not available to the minimization algorithm (unlike, for instance, the special case of quadratic programming).

Because our focus is on the objective function rather than the constraints of the problem, we will assume the simple case that the domain of f is the n-dimensional unit cube denoted by I^n (the n-fold Cartesian product of the interval $I = [0,1]$). It would be perhaps easier to assume simply that f is unconstrained (i.e., the domain is $\mathbb{R}^n$), but this leads to difficulties of scale as well as the problem that local minima might not exist. Since I^n is compact, there is always a global (and hence a local) minimum.

An algorithm to find a local minimum takes as input a function f and its gradient ∇f as black-box subroutines. It must repeatedly evaluate f and ∇f at points in I^n until it has found a local minimum. It is easy to see that in the real number function-evaluation model, there will always some uncertainty about the exact position of the local minimum. Accordingly it is useful to define approximate local minima.

Recall that $\mathbf{x}^* \in I^n$ is said to be a *global* minimum of f if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in I^n$. The point $\mathbf{x}^*$ is said to be a *local* minimum of f if there exists an open set N containing $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in N \cap I^n$.

Definition. *A point* $\mathbf{x}^* \in I^n$ *is said to be an* **ϵ-approximate local minimum** *of a continuous function* $f : I^n \to \mathbb{R}$. *if there exists an open set* N *containing* $\mathbf{x}^*$ *such that*

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) + \epsilon \|\mathbf{x} - \mathbf{x}^*\|$$

for all $\mathbf{x} \in N \cap I^n$.

Below we give an alternate characterization of this definition. First, we explain this definition and also pointing out its shortcomings. The motivation of this definition is that while $\mathbf{x}^*$ may not have the smallest function value in the neighborhood N, the value of f decreases slowly (at a rate no faster than ϵ) as one moves away from $\mathbf{x}^*$.

The most obvious shortcoming of this definition is that an interior local maximum or interior saddle point would also qualify as an ϵ-approximate local minimum under this definition. We do not feel that this property is a severe flaw in the definition, however. For example, examining the local minimization algorithms of Dennis and Schnabel, one sees that it is possible for these algorithms in some cases to converge to local maxima. Indeed, distinguishing local minima from local maxima in general is a computationally difficult problem; see, for example, Murty and Kabadi [1987].

We observe that it is required to select a norm in the above definition. For this paper we will assume that the 1-norm is used in that definition. The norms in this paper have been selected to make the analysis simple.

We now give an alternative characterization of an approximate local minima. We will say that $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ is an ϵ-*KKT point* of $f : I^n \to \mathbb{R}$ if

1. For all i such that $x_i^* > 0$, $\partial f/\partial x_i(\mathbf{x}^*) \leq \epsilon$.
2. For all i such that $x_i^* < 1$, $\partial f/\partial x_i(\mathbf{x}^*) \geq -\epsilon$.

(Note that if $\epsilon = 0$ these conditions are the KKT (Karush-Kuhn-Tucker) necessary conditions for local optimality). If $\mathbf{x}^*$ is interior, these conditions are equivalent to the requirement that $\|\nabla f(\mathbf{x}^*)\|_\infty \leq \epsilon$.

Assuming f is differentiable, we claim that if $\mathbf{x}^*$ is an ϵ-approximate local minimum, then it is an ϵ-KKT point. We verify statement 1 above for a particular index i (statement 2 is similar). Assuming $x_i^* > 0$, the point $\mathbf{x}^* - t\mathbf{e}_i$ is feasible for small enough $t > 0$, where $\mathbf{e}_i$ is the ith column of the identity matrix. For small enough t, this point is contained in $N \cap I$, hence $f(\mathbf{x}^*) - f(\mathbf{x}^* - t\mathbf{e}_i) \leq t\epsilon$ by definition of approximate local minimum. Since this holds for all t small enough, by definition of the partial derivative this implies $\partial f/\partial x_i(\mathbf{x}^*) \leq \epsilon$.

Conversely, suppose $\mathbf{x}^*$ is an ϵ-KKT point. Then we claim it is an ϵ'-approximate local minimum for any $\epsilon' > \epsilon$. To see this, recall that the definition of a derivative is that for all $\mathbf{d}$,

$$f(\mathbf{x}^* + \mathbf{d}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T \mathbf{d} + o(\|\mathbf{d}\|).$$

Suppose that $x_i^* > 0$ for some i; then we know

$$f(\mathbf{x}^* - t\mathbf{e}_i) = f(\mathbf{x}^*) - t\frac{\partial f}{\partial x_i}(\mathbf{x}^*) + o(t)$$

so

$$f(\mathbf{x}^* - t\mathbf{e}_i) \geq f(\mathbf{x}^*) - t\epsilon + o(t)$$

so

$$f(\mathbf{x}^* - t\mathbf{e}_i) \geq f(\mathbf{x}^*) - t\epsilon'$$

for all t small enough. This inequality holds not only for $\mathbf{x}^*$ but for every $\mathbf{x}$ in a neighborhood of $\mathbf{x}^*$ since we are assuming that f is continuously differentiable. Then we see that we can get a lower bound on $f(\mathbf{x}^* + \mathbf{d})$ for an arbitrary $\mathbf{d}$ that is small enough by expressing $\mathbf{d}$ as a sum of small steps of the form $t_i\mathbf{e}_i$.

We next ask the question: given a continuously differentiable function $f : I^n \to \mathbb{R}$ and given a number $\epsilon > 0$, what is the complexity of finding an ϵ-approximate local minimum? It turns out that the number of steps required is infinite. In particular, for any finite sequence of test points $x_1, \ldots, x_k$, there exists a continuously differentiable function $f : [0,1] \to \mathbb{R}$ such that $f(x_i) = 0$ and $f'(x_i) = 1$ at all testpoints (except if $x_i = 0$ then $f'(0) = -1$). Moreover, $f'(x) \in [-1, 1]$ for all $x \in [0, 1]$. Figure 1 illustrates an example of a sequence of test points and the bad-case function for these points. To construct this function, put the x_i's into increasing order, and then let f be the correct piecewise cubic function on each interval.

An algorithm trying to find approximate local minima for this family of functions will always completely fail (i.e., it will discover that $f(x) = 0$ and $f'(x) = 1$ at all of its test points) for at least one function in the family after any finite number of steps.

The problem with this family of functions is that the first derivatives can vary too much over short intervals, so that no algorithm can get a handle on the first derivative of the function.

Accordingly, we place additional restrictions on the function. In particular, we require that the first derivative satisfy a *Lipschitz condition*, that is, there exists a constant M such that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_\infty \leq M \|\mathbf{x} - \mathbf{y}\|_\infty$$

for all $\mathbf{x}, \mathbf{y} \in I^n$.

We now ask the question, what is the complexity of finding an ϵ-approximate local minimum for a function in this class? Clearly the answer depends on ϵ, M, and n. In the next section, we give an algorithm for this problem, which yields an upper bound on the complexity.

We remark that none of our complexity bounds depend on M or ϵ individually; instead, they all depend on the ratio M/ϵ. This is expected because the problem of finding an ϵ-approximate local minimum for f is the same problem as finding a $c\epsilon$-approximate local minimum for cf (where $c > 0$). Therefore, we would expect the complexity to be unchanged if M and ϵ are scaled by the same amount.

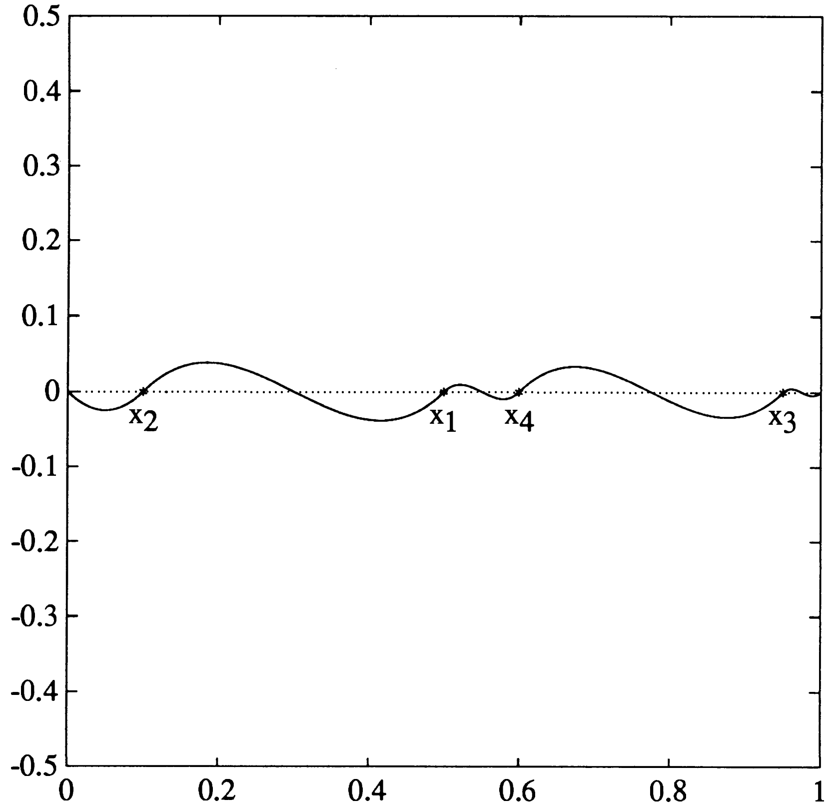

Figure 1: An impossible case for local minimization.

2 An algorithm for local minimization

In this section we propose an algorithm for approximate local minimization, along with a complexity analysis. We assume that ϵ, M and n are given. We call this algorithm LOCAL1. We assume also that M/ϵ is an integer. We are given a starting point $\mathbf{x}^{(0)} \in I^n$, which is assumed to have each coordinate equal to an integer multiple of ϵ/M. If no starting point is given, the origin can be used.

Given a function $f(\mathbf{x})$, we define vector-valued function $\mathbf{g}(\mathbf{x})$ as follows. The ith entry of $\mathbf{g}(\mathbf{x})$ is defined by

$$g_i(\mathbf{x}) = \begin{cases} \min\left(0, \dfrac{\partial f}{\partial x_i}(\mathbf{x})\right) & \text{if } x_i = 0, \\ \dfrac{\partial f}{\partial x_i}(\mathbf{x}) & \text{if } 0 < x_i < 1, \text{ or} \\ \max\left(0, \dfrac{\partial f}{\partial x_i}(\mathbf{x})\right) & \text{if } x_i = 1. \end{cases}$$

Notice that if $\mathbf{x}$ is interior to I^n, then $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$. This function $\mathbf{g}(\mathbf{x})$ could be called the "projected gradient," although this terminology is not standard.

Algorithm LOCAL1 begins by testing whether $\|\mathbf{g}(\mathbf{x}^{(0)})\|_\infty > M$. If this inequality holds, then for some i we know $|\partial f/\partial x_i(\mathbf{x}^{(0)})| > M$. Take the case that $\partial f/\partial x_i(\mathbf{x}^{(0)}) > M$ (the negative case is similar). We claim that $\partial f/\partial x_i(\mathbf{x}) > 0$ for all $\mathbf{x} \in I^n$. This follows from the Lipschitz bound on ∇f.

This means in particular that any local minimum of f must occur on the face $T = \{\mathbf{x} \in I^n : x_i = 1\}$ of I^n. Moreover, if $f_1 : T \to \mathbb{R}$ denotes the restriction of f to T, then it suffices to find an ϵ-approximate local minimum of f_1. Therefore, we can project $\mathbf{x}^{(0)}$ onto T and work on the restricted problem. The restriction operation has the effect of deleting the ith entry from the vector $\mathbf{g}(\mathbf{x})$.

Accordingly, we can continue to reduce the dimensionality of the problem coordinate by coordinate. Therefore, without loss of generality, we can assume our starting point satisfies $\|\mathbf{g}(\mathbf{x}^{(0)})\|_\infty \leq M$.

Let $\mathbf{x}^*$ be a global minimum of f. We can use the upper bound on $\mathbf{g}$ to derive an upper bound on the difference $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)$. Let $s =$

$\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_1$. Then we can construct a path of made up of segments parallel to the coordinate axes from $\mathbf{x}^{(0)}$ to $\mathbf{x}^*$; the length of this path will be exactly s. Assume the path is made up of n segments $P_1, \ldots, P_n$ such that P_i is parallel to $\mathbf{e}_i$.

Then we know that

$$f(\mathbf{x}^*) - f(\mathbf{x}^{(0)}) = \sum_{i=1}^{n} \int_{P_i} \frac{\partial f}{\partial x_i}. \tag{1}$$

We now derive an upper bound on each integral in (1). There are two cases. In the first case, $g_i(\mathbf{x}^{(0)}) = \partial f/\partial x_i(\mathbf{x}^{(0)})$. In this case, we can apply the Lipschitz bound directly. We know that $|g_i(\mathbf{x}^{(0)})| = |\partial f/\partial x_i(\mathbf{x}^{(0)})| \leq M$. Since the distance from $\mathbf{x}^{(0)}$ to any point of P_i is at most 1 in the ∞-norm, we know that the magnitude of $\partial f/\partial x_i$ along P_i is at most $2M$. Therefore, the above integral has magnitude at most $2M$.

In the second case, $g_i(\mathbf{x}^{(0)}) \neq \partial f/\partial x_i(\mathbf{x}^{(0)})$. Examining the definition of $\mathbf{g}$, we see that there are two possible subcases: either $x_i^{(0)} = 0$ and $\partial f/\partial x_i(\mathbf{x}^{(0)}) > 0$, or $x_i^{(0)} = 1$ and $\partial f/\partial x_i(\mathbf{x}^{(0)}) < 0$. We treat the first subcase since the second subcase is analogous. Since $x_i^{(0)} = 0$ and $\partial f/\partial x_i(\mathbf{x}^{(0)}) > 0$, we know that $\partial f/\partial x_i$ cannot drop below $-M$ at any point on P_i. Moreover, we know that P_i is oriented in the positive direction with respect to $\mathbf{e}_i$, because $x_i^{(0)} = 0$. Therefore, the ith integral in the above summation at least $-M$ (this argument does not give an upper bound, but only a lower bound is needed).

We conclude that all the integrals in (1) are at least $-2M$, and hence

$$f(\mathbf{x}^*) - f(\mathbf{x}^{(0)}) \geq -2Mn,$$

i.e.,

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*) \leq 2Mn.$$

This gives an upper bound on how much the objective function can decrease.

We now return to the main part of LOCAL1, under the assumption that $\|\mathbf{g}(\mathbf{x}^{(0)})\|_\infty \leq M$. The algorithm operates on an imaginary grid of nodes spaced ϵ/M apart in each dimension of I^n and aligned with the coordinate axes. By our assumptions made earlier, there are an integer number of mesh cells in every dimension, and the initial point $\mathbf{x}^{(0)}$ is one of the mesh points.

We now use the following iteration. Assume the current iterate is $\mathbf{x}^{(k)}$. We compute $\mathbf{g}(\mathbf{x}^{(k)})$. If $\|\mathbf{g}(\mathbf{x}^{(k)})\|_\infty < \epsilon$, then we halt. The justification for halting is as follows. If $\|\mathbf{g}(\mathbf{x}^{(k)})\|_\infty = \epsilon'$ and $\epsilon' < \epsilon$, then it is easy to verify by definition of $\mathbf{g}$ that $\mathbf{x}^{(k)}$ is an ϵ'-KKT point, and is therefore an ϵ-approximate local minimum.

Otherwise, suppose $\|\mathbf{g}(\mathbf{x}^{(k)})\|_\infty \geq \epsilon$. Then we identify a component, say $g_i(\mathbf{x}^{(k)})$, whose absolute value is at least ϵ. Say, for example, that $g_i(\mathbf{x}^{(k)}) \geq \epsilon$ (the negative case is similar). This means by definition that $\partial f/\partial x_i(\mathbf{x}^{(k)}) \geq \epsilon$ and that $x_i^{(k)} > 0$. Then we set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\epsilon/M)\mathbf{e}_i$. If $g_i(\mathbf{x}^{(k)})$ had been negative then we would have instead added $(\epsilon/M)\mathbf{e}_i$. Notice that, under this definition, $\mathbf{x}^{(k+1)}$ will be a mesh point lying in I^n.

With this formula for $\mathbf{x}^{(k+1)}$, we claim that $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}) - 0.5\epsilon^2/M$. To see this, observe that in the case that $\partial f/\partial x_i(\mathbf{x}^{(k)})$ is positive,

$$\begin{aligned} f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) &= \int_{-\epsilon/M}^{0} \frac{\partial f}{\partial x_i}(\mathbf{x}^{(k)} + t\mathbf{e}_i)\,dt \\ &\geq \int_{-\epsilon/M}^{0} (\epsilon + Mt)\,dt \\ &\geq 0.5\epsilon^2/M \end{aligned}$$

To derive the second line we used the fact that $\partial f/\partial x_i(\mathbf{x}^{(k)}) \geq \epsilon$ as well as the Lipschitz bound on $\partial f/\partial x_i$.

We conclude that the objective function decreases by at least $0.5\epsilon^2/M$ per iteration. As noted earlier, the most that the objective function can decrease is $2Mn$. Therefore, the maximum number of iterations is $4n(M/\epsilon)^2$. Let us state this as a theorem.

Theorem 1. *Let $f : I^n \to \mathbb{R}$ be a C^1 function whose gradient satisfies a Lipschitz condition with bound M. Then an ϵ-approximate local minimum can be found with at most $4n(M/\epsilon)^2$ function and gradient evaluations.*

We remark that if gradient values are not available, the first part of the algorithm (the restrictions to subproblems) can be carried out by estimating the gradient via finite differences. A bound can be derived on the accuracy of finite difference approximations to the gradient using the hypothesis that the gradient is Lipschitz-bounded.

If gradient values are not available, then the main local-search step of the algorithm can be replaced with a comparison of the objective value at

$\mathbf{x}^{(k)}$ to the objective values at the neighboring grid points. This requires $2n$ function evaluations per local search step.

3 A lower bound for local minimization

In the last section we saw polynomial dependence on both n and M/ϵ. The polynomial dependence on n is to be expected in general (since f depends on n variables, it presumably takes at least n operations merely to evaluate f). The polynomial dependence on M/ϵ is clearly unavoidable with that algorithm since the step size is ϵ/M.

It is natural to inquire whether the polynomial dependence on M/ϵ is actually necessary for all algorithms. Could an algorithm with large steps (say, steepest descent combined with line-search) be more effective?

The purpose of this section is to give a lower bound showing that polynomial dependence on M/ϵ is inherent in the problem of black-box local minimization. The lower bound applies to all algorithms based on the function evaluation model (not merely to the algorithm of the last section). The lower bound is based on a family of functions that could fool any algorithm until it has made at least $\Omega\left(\sqrt{M/\epsilon}\right)$ function and gradient evaluations. The construction of this family has two parts: an algebraic/geometric part and a combinatorial part. Notice that because we are trying to provide a "bad case" (lower bound) for all possible information-based algorithms, we need a whole family of bad-case functions rather than a single function.

These functions are bad cases in the sense that an algorithm for local minimization will require many steps. There are other senses in which a local optimization example could be bad (for instance, it may be that local minima are easily found but have large objective function values with respect to the global minimum).

We focus on the $n = 2$ case since the interest here is the dependence on M/ϵ. This lower bound is very reminiscent of a lower bound for Brouwer fixed points in two dimensions due to Hirsch, Papadimitriou and Vavasis [1989]. We assume that M and ϵ are given. In this section we work with $\|\cdot\|_2$ norms because there are two rotated coordinate systems (other norms could lead to confusion).

The lower bound is based on how much information any algorithm could get about f. We argue in this section informally about what the algorithm "knows" from its function evaluations, but the information model can be cast into formal terms. See, for example, Traub, Wasilkowski and Woźniakowski [1982].

We divide the unit square I^2 into $K \times K$ subsquares, where K is an integer on the order of $\sqrt{M/\epsilon}$ (the exact value will be selected below). Besides K, we also have the parameters δ and δ', which are both on the order of ϵ (the exact formulas are below). Number the subsquares with ordered pairs $\langle u, v\rangle$, $u, v = 0, \ldots, K-1$. Two subsquares are said to be *adjacent* if they have a common edge.

We will embed I^2 in the plane diagonally, i.e., with vertices at $(0,0)$, $(\sqrt{2}/2, \pm\sqrt{2}/2)$ and $(\sqrt{2}, 0)$. The relationship between the subsquare numbering and coordinate system is as follows. The vertex of subsquare $\langle u, v\rangle$ with minimum x coordinate is at

$$\frac{1}{J}(u+v, v-u)$$

where $J = K\sqrt{2}$. The embedding along with some numbered subsquares is indicated in Figure 2.

A *southeast track* is a sequence of adjacent subsquares with increasing first coordinates, and a *northeast track* is a sequence of adjacent subsquares with increasing second coordinates.

The *west subsquare* is subsquare $\langle 0, 0\rangle$. Define a *riverbed* to be a sequence of adjacent subsquares starting at the west subsquare, proceeding along a northeast track, and then following a sequence of alternating southeast and northeast tracks, and ending somewhere inside the square. This terminology is used because the mesh plot of function f based on this construction resembles the top view of a riverbed on a hillside. An example of a riverbed is indicated in Figure 3. Note that the riverbed will have at most $2K-1$ subsquares. The last (closest to the east) subsquare of the riverbed is called the *sink*. The subsquares of I^2 not in the riverbed are called *hillside* subsquares.

Our functions will be defined based on K, δ and δ' (i.e., based on M and ϵ) and on a particular choice of riverbed. The function f will be constructed below so that all ϵ-approximate local minimum lie in the sink subsquare.

Notice that there is a large but finite set of possible riverbeds for each particular value of K. The functions on I^2 in our family will be in correspondence with choices of riverbeds. The particular riverbed to choose

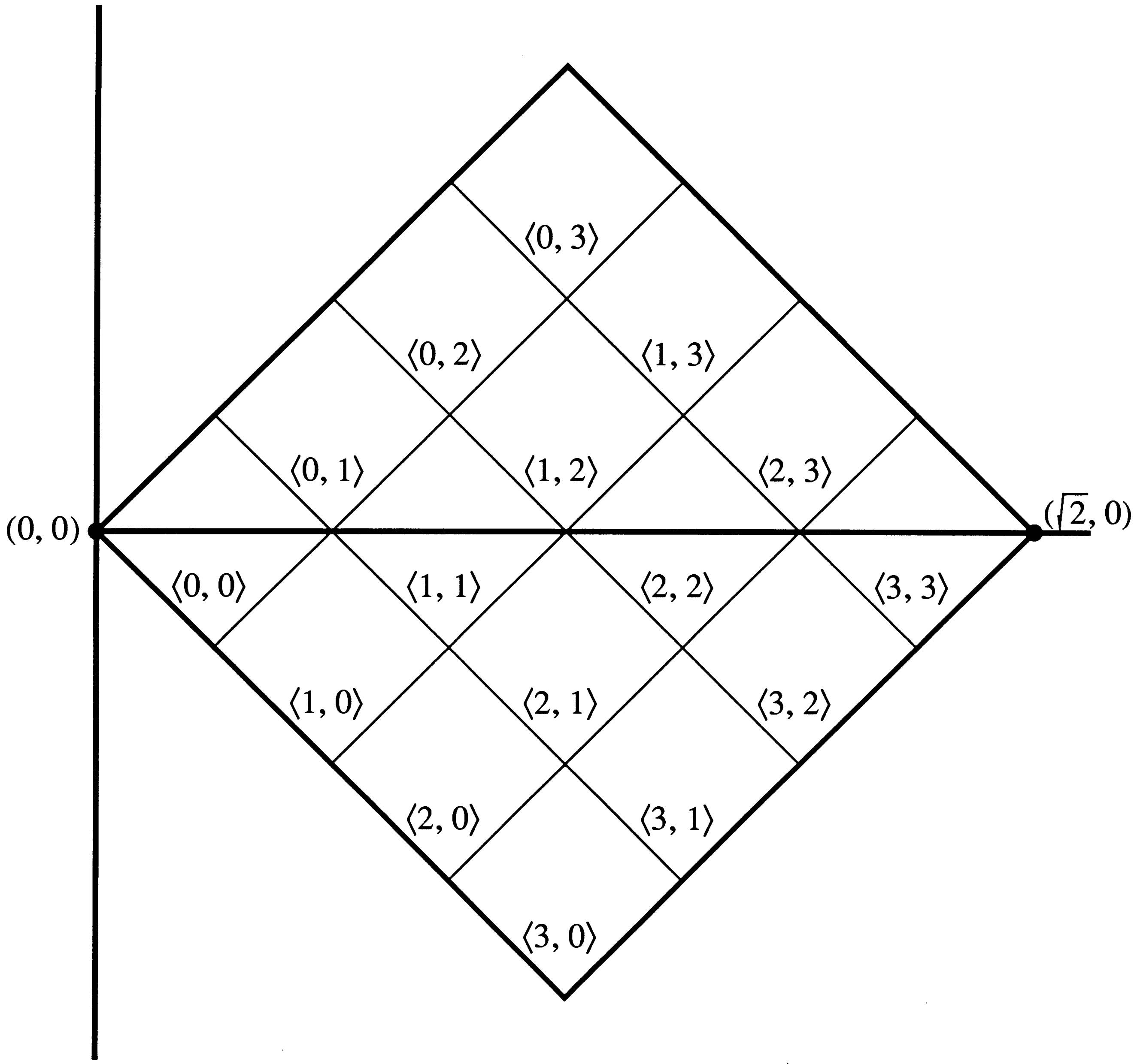

Figure 2: Embedding I^2 in the plane with subsquares indicated.

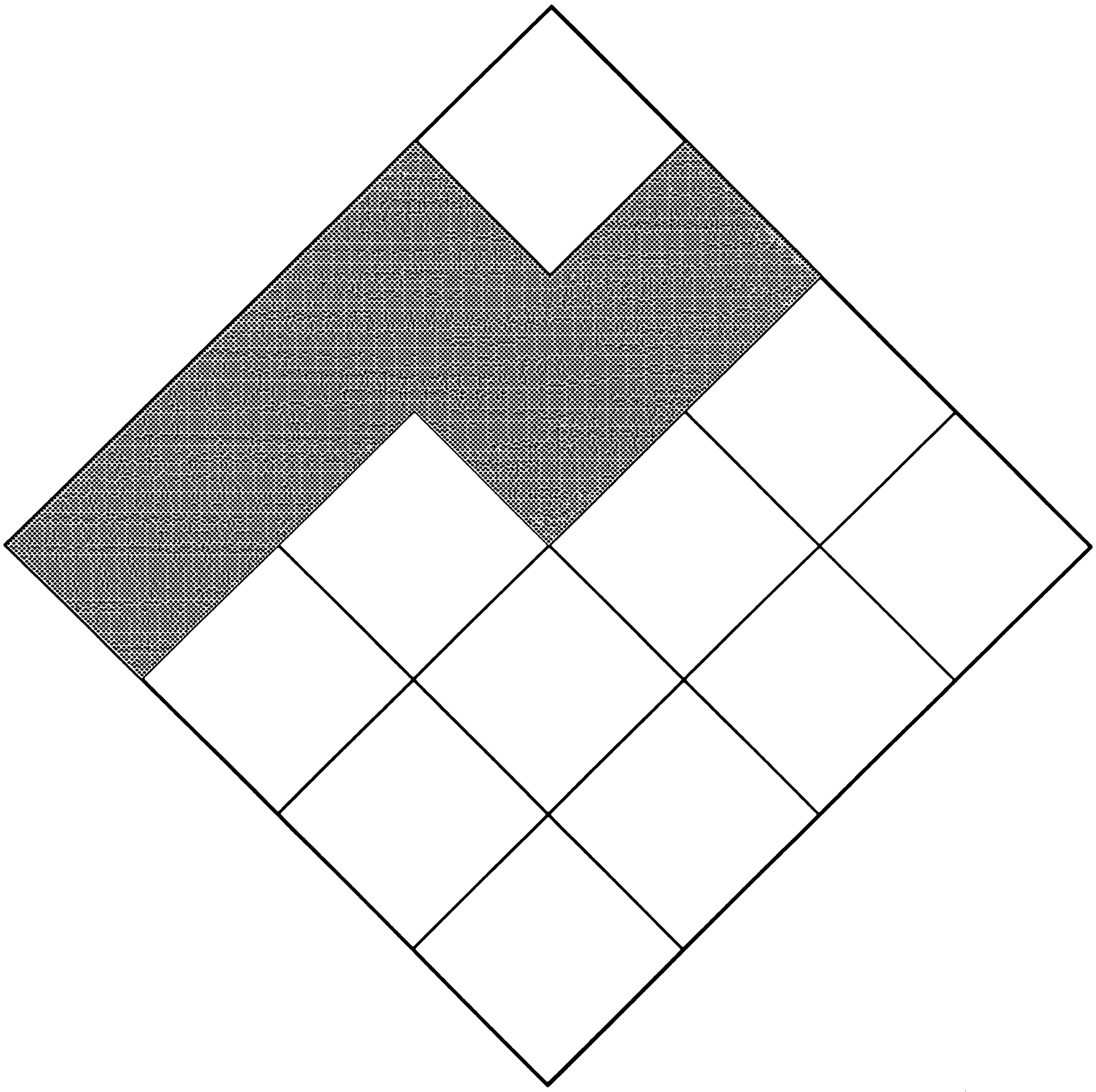

Figure 3: An example of a riverbed for $K = 4$.

will depend on the algorithm at hand—this is the combinatorial part of the construction described below.

For now, we assume that a particular riverbed is selected, and we proceed with the construction of f. All the properties that f should have are stated in the lemmas below. The reader uninterested in the geometric details can skip ahead to Figure 7 and read the lemmas.

The first part of the construction is the function $s(x)$ that traces the shape of the riverbed. The path defined by $(x, s(x))$ as x varies from 0 to x_e passes through all the subsquares of the riverbed (x_e is defined below). It enters and leaves each subsquare through the midpoint of the edge between adjacent subsquares of the riverbed.

In particular, $s(x)$ is defined piecewise on intervals of the form $[(i + 0.5)/J, (i+1.5)/J]$ where i is an integer. If x is the endpoint of one of these intervals, $s'(x) = \pm 1$. The pieces are matched so that $s(x)$ is continuously differentiable. The formulas for $s(x)$ are as follows. In the west subsquare, for x between 0 and $1.5/J$, we define

$$s(x) = \frac{4}{27J}(Jx)^3.$$

We notice that this function leaves the west square through the point $(3/(2J), 1/(2J))$, i.e., the midpoint of the edge between subsquare $\langle 0,0 \rangle$ and $\langle 0,1 \rangle$. This means that the subsquare after $\langle 0,0 \rangle$ in the riverbed will always be $\langle 0,1 \rangle$ (as mentioned above, a riverbed is defined to start with a northeast track). Also, we can check that $s'(3/(2J)) = 1$.

In a subsquare $\langle u,v \rangle$ that is interior to a northeast track, $s(x)$ is a linear function with slope 1. Specifically, $(x, s(x))$ linearly joins the point

$$\frac{1}{J}(u + v + 0.5, v - u - 0.5)$$

to the point

$$\frac{1}{J}(u + v + 1.5, v - u + 0.5).$$

Similarly, in a subsquare interior to a southeast track, $s(x)$ is linear with slope -1.

In a subsquare $\langle u,v \rangle$ in which the riverbed makes a turn, say, from northeast to southeast, the formula is

$$s(x) = \frac{1}{J}(Jx - u - v - 0.5)(u + v + 1.5 - Jx) + v - u - 0.5$$

This function starts at

$$\frac{1}{J}(u+v+0.5, v-u-0.5),$$

ends at

$$\frac{1}{J}(u+v+1.5, v-u-0.5).$$

Function $s(x)$ has slope $+1$ at the first point and -1 at the second. A turn from southeast to northeast is analogous.

In the sink subsquare, $s(x)$ is defined by the linear function of slope 1 if the sink subsquare is the terminal of a northeast track, else $s(x)$ is a linear function of slope -1.

The end x_e of the domain of definition of $s(x)$ is the x-coordinate of the midpoint of edge of the sink square where the riverbed terminates. If $\langle u_e, v_e\rangle$ is the sink subsquare, this coordinate is

$$x_e = \frac{1}{J}(u_e + v_e + 1.5).$$

An example of this construction with $K = 4$ is plotted in Figure 4. Here, the riverbed is given by $\langle 0,0\rangle$, $\langle 0,1\rangle$, $\langle 0,2\rangle$, $\langle 1,2\rangle$, $\langle 1,3\rangle$.

We observe that $s(x)$ has the following properties. It is C^1 and piecewise C^2. The maximum value of $|s'(x)|$ is 1, and the maximum value of $|s''(x)|$ (where defined) is $2J$.

We have now defined a function to specify the shape of the riverbed. The next step is to define the two functions controlling the value of function f on the riverbed portion of I^2. The first function indicates how f varies in the direction across the riverbed, and the second indicates how f varies parallel to the riverbed. The first function is defined by

$$c(w) = \begin{cases} 0 & \text{for } w \leq -1, \\ -w^4 + 2w^2 - 1 & \text{for } w \in [-1, 1], \\ 0 & \text{for } w \geq 1. \end{cases}$$

which is plotted in Figure 5.

It is easily checked that this function has the following properties: c is C^1, $c(-1) = c(1) = 0$, $c(0) = -1$, and $c'(-1) = c'(0) = c'(1) = 0$. Also, the maximum value of $|c(w)|$ is 1, of $|c'(w)|$ is 4, and of $|c''(w)|$ (which is undefined at ± 1) is 8.

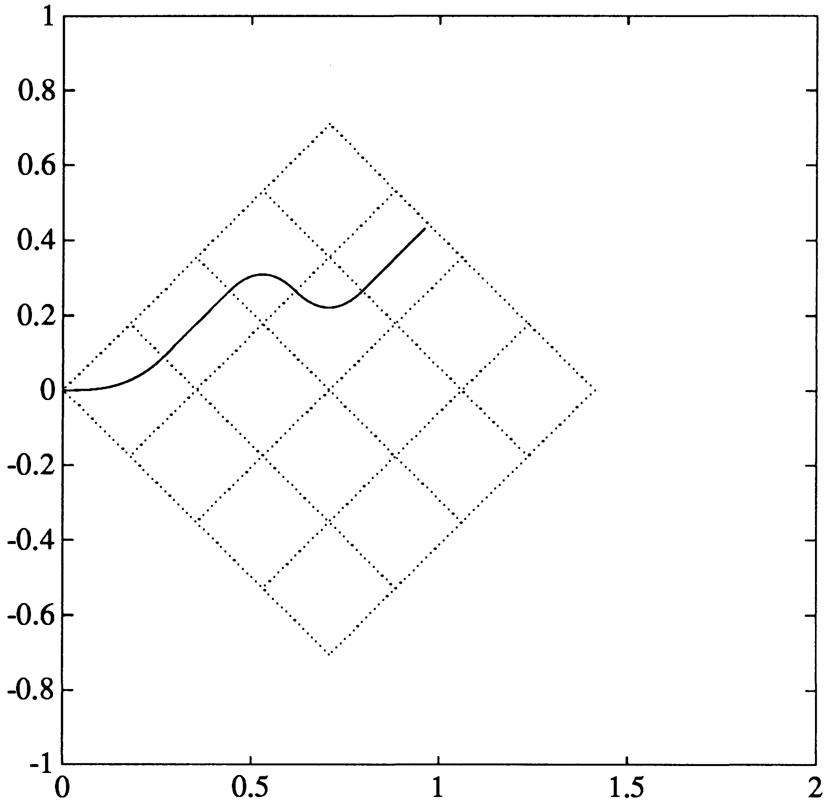

Figure 4: An example of $s(x)$.

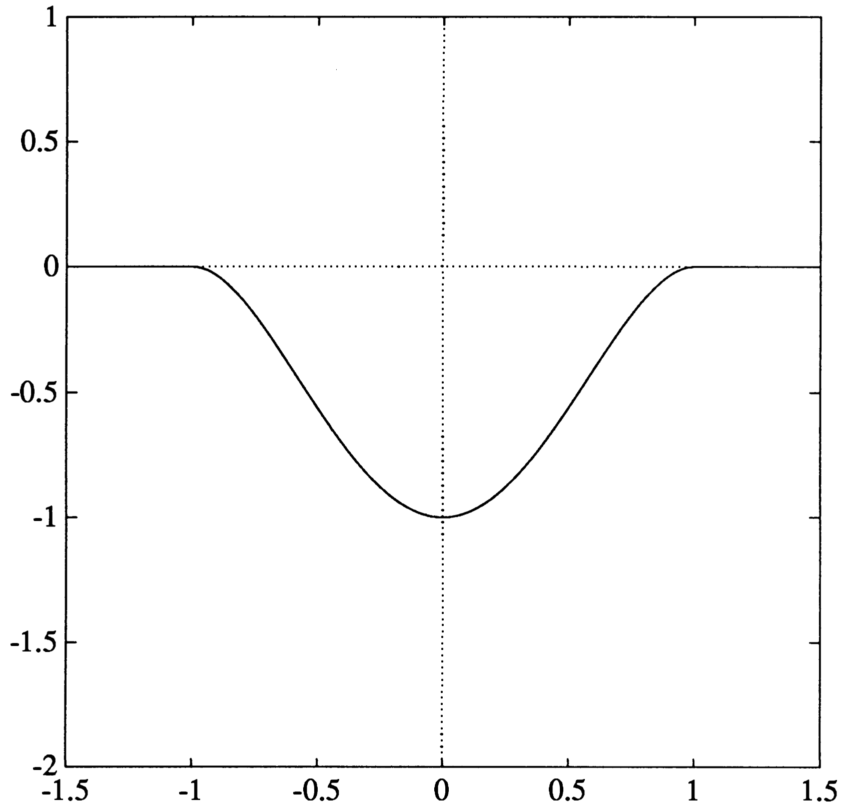

Figure 5: The graph of $c(x)$.

Next we define the function $p(x)$, which determines the how f varies as the riverbed is followed. The value of $p(x)$ depends on the position of the sink. Specifically, suppose $\langle u_e, v_e\rangle$ is the sink subsquare. Then the formulas for $p(x)$ are as follows. Let $x_b = (u_e + v_e + 0.5)/J$, that is, the x-coordinate of the point where path $(x, s(x))$ enters the sink square. Let $x_c = (u_e + v_e + 0.75)/J$, and $x_d = (u_e + v_e + 1.0)/J$. Let $b_0 = \delta + \delta x_b$. Then

$$p(x) = \begin{cases} \delta + \delta x & \text{for } x \in [0, x_b], \\ -2\delta J(x - x_b)^2 + \delta(x - x_b) + b_0 & \text{for } x \in [x_b, x_c], \\ (b_0 + \delta/(8J))\left[2\left(4J(x - x_c)\right)^3 \right. & \\ \qquad \left. - 3\left(4J(x - x_c)\right)^2 + 1\right] & \text{for } x \in [x_c, x_d], \text{ and} \\ 0 & \text{for } x \geq x_d. \end{cases}$$

It can be checked that $p(x)$ is C^1. In particular, $p(x_b) = b_0$, $p(x_c) = b_0 + \delta/(8J)$, and $p(x_d) = 0$. Also, $p'(x_b) = \delta$ and $p'(x_c) = p'(x_d) = 0$. The maximum value of $|p(x)|$ is $b_0 + \delta/(8J)$, which is at most 3δ. Also, it can be checked that $|p'(x)|$ is at most $18\delta J$, and $|p''(x)|$ (where defined) is at most $288\delta J^2$. An example of $p(x)$ is plotted in Figure 6.

From $c(w)$, $p(x)$ and $s(x)$ we now assemble the function $f(x, y)$, which is defined as follows:

$$f(x, y) = p(x) \cdot c(2K(y - s(x))) + \delta' x.$$

A MATLAB mesh plot of $f(x, y)$ is illustrated in Figure 7

We now establish some properties of this function.

Lemma 1. *If (x, y) lies in the hillside, then $f(x, y) = \delta' x$.*

Proof. We must show that the first term vanishes outside the riverbed. If (x, y) is not in the riverbed, either $x > x_d$ or $|y - s(x)| \geq 3/(4J)$. This latter inequality arises from the fact that the distance from $(x, s(x))$ in the y-direction to the boundary of the riverbed is always at least $3/(4J)$ by definition of $s(x)$. If $x > x_d$ then $p(x) = 0$ so the claim is true. Similarly, if $|y - s(x)| \geq 3/(4J)$ then $2K|y - s(x)| \geq (3K)/(2J) \geq 1$, hence $c(2K(y - s(x))) = 0$. ∎

At this point, we choose an ϵ' to be slightly larger than ϵ, and we let

$$\delta = 32\sqrt{2}\epsilon'$$

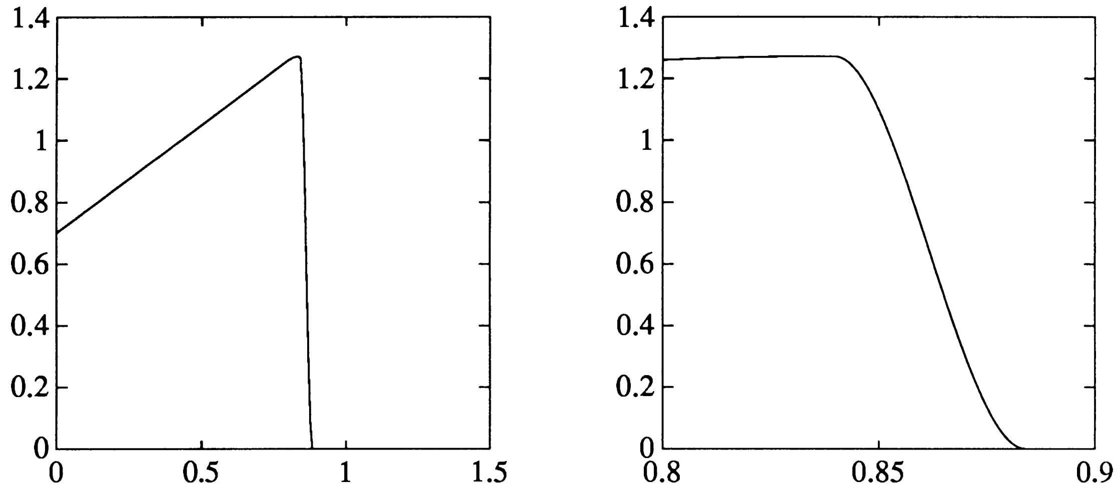

Figure 6: An example of $p(x)$; the right plot shows a detail.

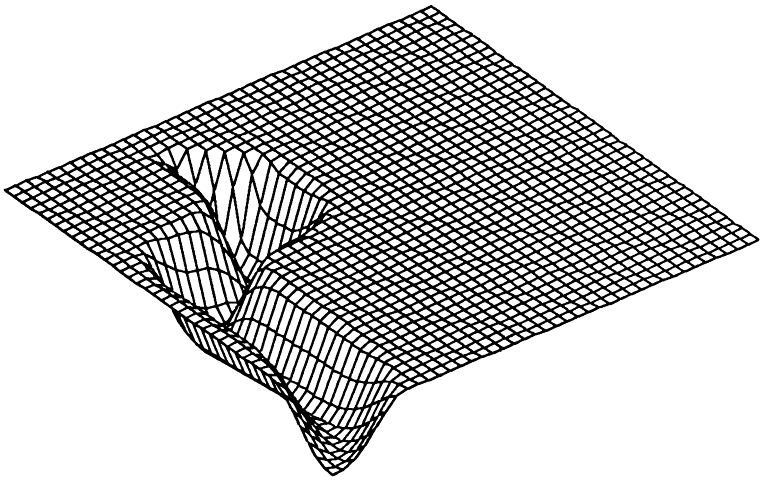

Figure 7: An example of $f(x,y)$.

and

$$\delta' = 19\sqrt{2}\epsilon'.$$

These choices are made so that we can prove the following lemma.

Lemma 2. *Let (x, y) be a point not in a sink square. Then $\|\nabla f(x, y)\|_2 \geq \sqrt{2}\epsilon'$.*

Proof. Let w denote $2K(y - s(x))$. We compute

$$\nabla f(x, y) = (p'(x)c(w) - 2Kp(x)c'(w)s'(x) + \delta', 2Kp(x)c'(w)).$$

We now take cases to prove a lower bound on the size of $\nabla f(x, y)$. The first case is that we are not in the riverbed, which was handled by the previous lemma and the fact that $\delta' \geq \sqrt{2}\epsilon'$. This is the case that $|w| \geq 1$ or $x \geq x_d$. For the other cases we assume $|w| \leq 1$ and $x \leq x_b$. (We can assume $x \leq x_b$ since (x, y) is not in the sink.) We now take subcases. The first subcase is that $|w| \in [1 - 1/(64K), 1]$. In this case, $|c(w)|$ and $|c'(w)|$ are at most $1/(8K)$. Then, we observe that for x not in the sink, $|p(x)| \leq 2\delta$, $|p'(x)| \leq \delta$ and $|s'(x)| \leq 1$. Thus, the term $|p'(x)c(w)|$ above is at most $\delta/16$, and the term $|2Kp(x)c'(w)s'(x)|$ is at most $\delta/2$. The third term of the first entry of ∇f is exactly δ', therefore the first entry of the derivative has magnitude at least $\delta' - (9/16)\delta$. Using the above formulas for δ, δ', this is a magnitude of at least $\sqrt{2}\epsilon'$.

In the second subcase, $|w| \in [1/(64K), 1 - (1/64K)]$. In this case, we observe that $|c'(w)| \geq 1/(10K)$. This means that the second entry $|2Kp(x)c'(w)|$ of $\nabla f(x, y)$ is at least $|p(x)/5|$, i.e., at least $\delta/5$. This quantity is greater than $\sqrt{2}\epsilon'$.

In the third subcase, $|w| \in [0, 1/(64K)]$. In this case, $|c(w)| \geq 7/8$, whereas $|c'(w)| \leq 1/(16K)$. Therefore, the first term $|p'(x)c(w)|$ is at least $(7/8)\delta$. The second term $|2Kp(x)c'(w)s'(x)|$ is at most $(1/4)\delta$. The last term is exactly δ'. Thus, the first entry has magnitude at least $(7/8)\delta - (1/4)\delta - \delta'$, which is $\sqrt{2}\epsilon'$. ∎

Lemma 3. *Let (x, y) be a point not in a sink square. Then (x, y) is not an ϵ-approximate local minimum of f.*

Proof. This follows from the previous lemma, with special attention paid to the boundaries. Region I^2 has four boundaries and four corners. We have to check that if (x, y) is on a boundary then the projection of $\nabla f(x, y)$ onto the boundary is at least ϵ'. This is easy to see, as follows. Along a boundary we know that $\nabla f(x, y) = (\delta', 0)$ from the construction, except in the west subsquare. The gradient $(\delta', 0)$ has magnitude at least ϵ when projected on every boundary except the point $(0, 0)$. This takes care of the whole boundary outside the west square.

Therefore, we only have to examine the exterior boundary of the west subsquare. A calculation shows that every point has a projected gradient of size at least ϵ'. ∎

Note that we have not established the existence of an ϵ-approximate local minimum in the sink square. We know by compactness, however, that such a point exists, and the previous lemma forbids its existence anywhere else.

It is now time to select the value of K, which will be

$$K = \left\lfloor \frac{1}{310}\sqrt{\frac{M}{\epsilon'}} \right\rfloor .$$

The reason for this value of K is to establish the following lemma.

Lemma 4. *The gradient $\nabla f(x, y)$ for f defined above is continuous and has Lipschitz constant at most M.*

Proof. The gradient exists everywhere and is continuous because f is assembled from C^1 functions of one variable. Because f is continuously differentiable and piecewise C^2, then the following inequality holds:

$$\|\nabla f(x_1, y_1) - \nabla f(x_2, y_2)\|_2 \leq \int_P \|D^2 f(x, y)\|_2 \cdot \|(x_2, y_2) - (x_1, y_1)\|_2 \, dP$$

where P is a straight-line path from (x_1, y_1) to (x_2, y_2), and D^2 denotes the second derivative. This inequality holds for almost all pairs of points (the only exception being the case when P intersects a continuum of points where f'' fails to exist). Thus, to get a Lipschitz bound on $\nabla f(x, y)$ it suffices to establish an upper bound on the 2-norm of the second derivative wherever it is defined. Since the 2-norm of a matrix is hard to work with, we instead put an upper bound on the infinity norm, and then multiply it by $\sqrt{2}$.

We compute the second derivative, entry by entry:

$$\begin{aligned}
\frac{\partial^2 f(x,y)}{\partial x^2} &= p''(x)c(w) - 4Kp'(x)c'(w)s'(x) \\
&\quad + 4K^2p(x)c''(w)s'(x)^2 - 2Kp(x)c'(w)s''(x), \\
\frac{\partial^2 f(x,y)}{\partial x \partial y} &= 2Kp'(x)c'(w) - 4K^2p(x)c''(w)s'(x), \\
\frac{\partial^2 f(x,y)}{\partial y^2} &= 4K^2p(x)c''(w).
\end{aligned}$$

We can go through each term and use the crude estimates made earlier to get an upper bounds of $1116K^2\delta$ on $\partial^2 f/\partial x^2$, $312K^2\delta$ on $\partial^2 f/\partial x \partial y$, and $96K^2\delta$ on $\partial^2 f/\partial y^2$. Estimating $\delta \leq 45.3\epsilon'$ gives an upper bound of about $6.5 \cdot 10^4 K^2\epsilon'$ for the ∞-norm of the second derivative, which translates to an upper bound of about $9.2 \cdot 10^4 K^2\epsilon'$ for the 2-norm. Therefore, with the above choice of K we are guaranteed to have a Lipschitz bound of at most M. Note that the true Lipschitz bound for our construction grows proportionally to $K^2\epsilon$ but with a much smaller constant. ∎

Lemma 5. *Suppose (x,y) lies in subsquare $\langle u,v\rangle$. From $f(x,y)$ and $\nabla f(x,y)$ it is not possible to determine any information about the riverbed except possibly: whether or not $\langle u,v\rangle$ lies in the riverbed, and if so, what the positions of the two neighboring riverbed squares are.*

Proof. This follows from the definition of $f(x,y)$. If $x \geq x_d$ or $|y - s(x)| \geq 1/2K$ then we cannot determine anything about the riverbed except that $\langle u,v\rangle$ is not in the riverbed. If $|y - s(x)| \leq 1/2K$ and $x \leq x_d$ then we might be able to determine the values of $s(x)$, $s'(x)$, and $p(x)$ from $f(x,y)$ and $\nabla f(x,y)$. This means we can determine that the particular square is in the riverbed, and we can determine what kind of turn the riverbed makes. Nothing else can be determined. ∎

We now prove a general lower bound for finding approximate local minima for this family of functions. We imagine an algorithm A that makes function and gradient evaluations. We want to find a pair of functions $f(x,y)$, $f'(x,y)$ in our family with disjoint sets of approximate local minimum such that algorithm A cannot distinguish them until K function/gradient evaluations (i.e., $\Omega\left(\sqrt{M/\epsilon}\right)$) have been made. Notice that the only approximate local minima for functions in our family occur in the sink square, and therefore f and f' will have different sinks. We start by assuming that the algorithm knows M and ϵ (and therefore, K).

The combinatorial argument that constitutes the remainder of this section is identical to the argument of Hirsch, Papadimitriou, and Vavasis, but we present it again here for the sake of completeness.

To construct f and f' we need to specify riverbeds R, R'. The riverbeds for these two functions will be almost identical. The riverbeds are constructed "adaptively." In particular, we fix more and more of R, R' as we observe the testpoints made by A. The idea is the f and f' will agree at all test points, so R and R' will agree almost until the end.

We assume that A makes a deterministic sequence of testpoints, and that at each testpoint it evaluates f and ∇f. The sequence of testpoints is denoted (x_i, y_i). Each one may depend on previous testpoints in any way possible. Thus, A has unlimited computational power. Note that there is no advantage for A to make a testpoint exactly on a boundary of a subsquare for our family (i.e., no more information about the riverbed can be gleaned from a boundary than from a nearby interior point), so we assume that all testpoints lie in a unique subsquare. Once a testpoint has been made in a subsquare, we assume that A has complete information about the subsquare (i.e., all the values of $f(x, y)$ are known to A for (x, y) lying in the subsquare).

The riverbeds R, R' are constructed as a sequence of tracks alternating northeast and southeast. They are built up as the limit of the sequence $R_0, R_1, R_2, \ldots$, where each $R_i \subset R_{i+1}$, and R_i denotes the partial riverbed that is determined after i testpoints from A (note that $R_0 = \{\langle 0, 0\rangle\}$ since this subsquare is in every riverbed).

We know that R_i starts from $\langle 0, 0\rangle$; denote its last square as $\langle u_i, v_i\rangle$. In our construction R and R' will agree with R_i all the way up to subsquare $\langle u_i, v_i\rangle$, and moreover, that R, R' will make a bend in subsquare $\langle u_i, v_i\rangle$. We let T_i denote the track starting from $\langle u_i, v_i\rangle$ following the direction of the bend and proceeding to the border of I^2. The invariant property of the upcoming construction is that no testpoints have been made in T_i on iterations 1 up to i. Note that T_0 is the northeast track of subsquares with first coordinates equal to 0.

We now give the rules for extending R_{i-1} to R_i. There are three cases for testpoint i. In the first case, (x_i, y_i) lies in the part of I^2 that is already determined. To be specific, suppose, for example, that R_{i-1} ends at subsquare $\langle u_{i-1}, v_{i-1}\rangle$ and that T_{i-1} is a northeast track emerging from this subsquare. Suppose that (x_i, y_i) lies in subsquare $\langle u', v'\rangle$. If $u' < u_{i-1}$ or $v' \leq v_{i-1}$ then the behavior of R is entirely known in $\langle u', v'\rangle$, and hence f, f' are determined already. In this case, we set $R_i = R_{i-1}$, $T_i = T_{i-1}$, and $\langle u_i, v_i\rangle = \langle u_{i-1}, v_{i-1}\rangle$.

The second case is that (x_i, y_i) lies in a subsquare of I^2 not in T_{i-1} but through which R or R' might eventually pass. This is the case that $u' > u_{i-1}$ and $v' > v_{i-1}$. In this case, we mark all subsquares with first coordinate equal to u' as "forbidden" and the same with subsquares with second coordinate equal to v'. In this case again we set $R_i = R_{i-1}$, $T_i = T_{i-1}$ and $\langle u_i, v_i\rangle = \langle u_{i-1}, v_{i-1}\rangle$. We also set $f(x, y) = \delta' x$ in this subsquare (i.e., we "tell" the algorithm that the riverbed does not pass through this subsquare).

In the third case, (x_i, y_i) lies in T_{i-1}. In this case (assuming as above that T_{i-1} is a northeast track), we let v_i be the smallest integer coordinate greater than v_{i-1} that has not yet been forbidden in the construction procedure described above. Then we let R_i be the union of R_{i-1} and the portion of T_{i-1} connecting $\langle u_{i-1}, v_{i-1}\rangle$ to $\langle u_{i-1}, v_i\rangle$. We let $u_i = u_{i-1}$, and T_i be the southeast track starting at $\langle u_i, v_i\rangle$ and including subsquares with increasing first coordinates. Finally, we assign values to $f(x, y)$ based on R_i in the subsquare that contained the testpoint. If T_{i-1} had been a southeast track, then T_i would have been a northeast track.

Figure 8 shows the three possible locations for a testpoint. Notice the rule for forbidding northeast and southeast tracks keeps the whole procedure consistent, i.e., each R_i is a valid riverbed that is consistent with all the testpoints up to (x_i, y_i).

How long can this construction proceed? We notice that if R_i terminates at subsquare $\langle u_i, v_i\rangle$, then $u_i \leq i$ and $v_i \leq i$ because we never pass to a higher value of u unless all lower integer values of u had testpoints associated with them. The same holds for the second coordinate.

Therefore, we can continue extending R_i until K testpoints have been made. Until the $K - 1$st testpoint, there are at least two subsquares in T_i, and therefore, there are at least two subsquares in which the riverbed could end. Therefore, we let R be R_{K-1} terminated with one of these sinks, and R' be R_{K-1} terminated with the other. Then the algorithm cannot distinguish f from f' until $K - 1$ testpoints have been made.

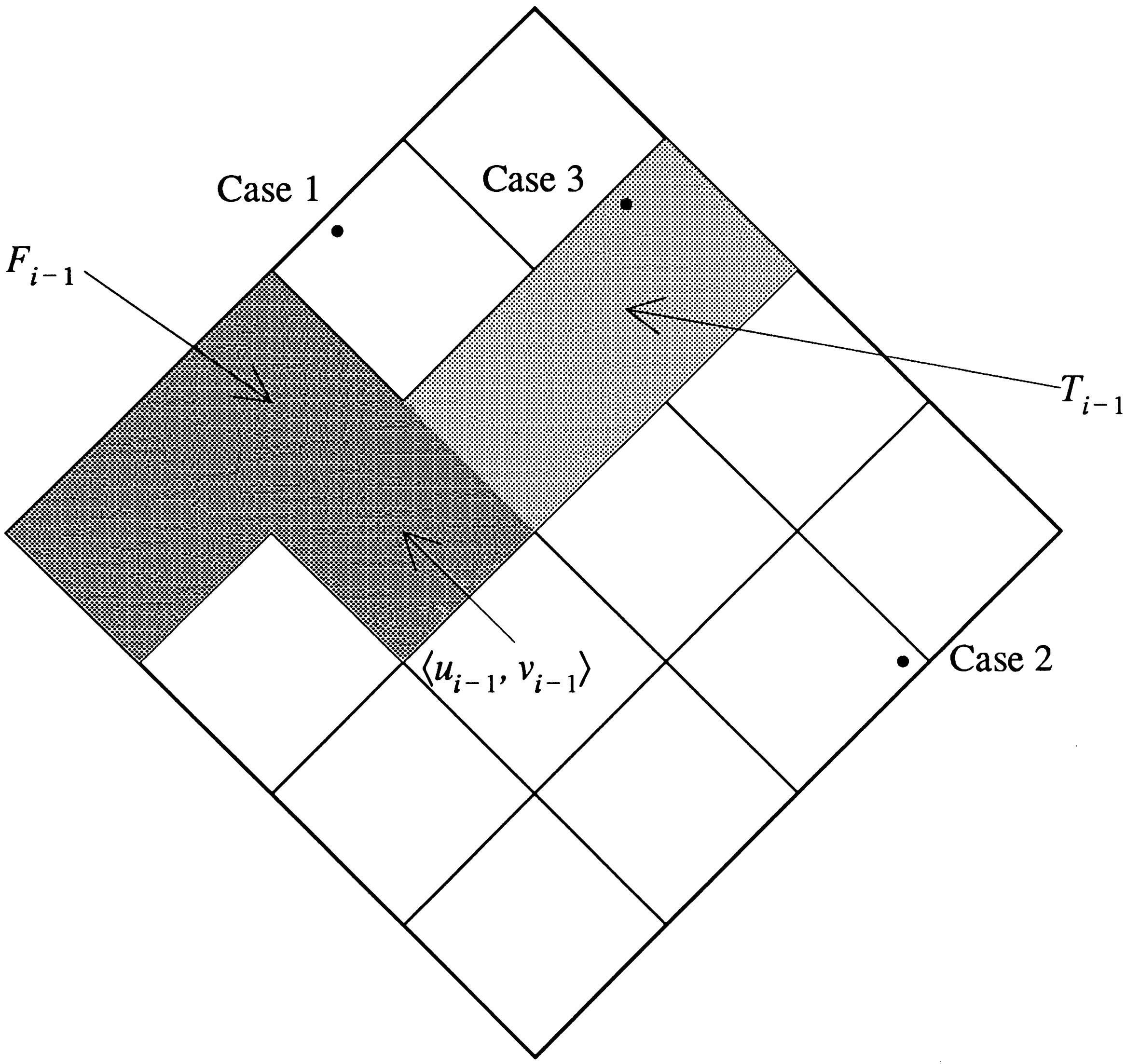

Figure 8: The three cases for testpoints.

We state this as a theorem:

Theorem 2. *Let A be any deterministic algorithm to find ϵ-approximate local minima of functions $f : I^2 \to \mathbb{R}$ whose gradients satisfy Lipschitz conditions with constant M. Assume the algorithm is limited to using function and gradient evaluations. Then, in the worst case, algorithm A requires $\Omega\left(\sqrt{M/\epsilon}\right)$ function and gradient evaluations.*

4 An improved algorithm when n is large

We notice that the upper bound on Algorithm LOCAL1 Section 2 grows like $(M/\epsilon)^2$, whereas the lower bound grows only like $\sqrt{M/\epsilon}$. Is it possible to bring these bounds in closer agreement? In this section we propose an improvement on the algorithm of Section 2 in the case that n is very small with respect to M/ϵ. The new algorithm will be called LOCAL2.

The main point of LOCAL2 is to pick the initial point $\mathbf{x}^{(0)}$ for LOCAL1 in an intelligent manner. We make a mesh with points spaced $1/k$ in every dimension (the "coarse" grid), where k is an integer determined below. Then we evaluate f at every one of these $(k+1)^n$ mesh points. We let $\mathbf{x}^{(0)}$ be the coarse grid point with the minimum value of f. We begin the local improvement algorithm (on the "fine mesh," that is, the mesh with spacing ϵ/M) from this $\mathbf{x}^{(0)}$. Assume that k is an integer divisor of M/ϵ, so that all the coarse grid points are also fine grid points.

Now we reanalyze the number of steps to find a local minimum. Let $\mathbf{x}^{(0)}$ be as in the previous paragraph. Let $\mathbf{x}^*$ be a global minimum of f. In Section 2 we established an upper bound on $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)$ without any special knowledge about $\mathbf{x}^{(0)}$. In this section we want a better bound on this difference.

To establish this bound, let $\mathbf{x}'$ be the coarse grid point closest to $\mathbf{x}^*$. First, we establish the claim that $\|\mathbf{g}(\mathbf{x}')\|_\infty \leq M/(2k)$. Suppose not; suppose, e.g., that $g_i(\mathbf{x}') > M/(2k)$. This means $\partial f/\partial x_i(\mathbf{x}') > M/(2k)$ and $x_i' > 0$. Since $\mathbf{x}'$ is the closest coarse grid point to $\mathbf{x}^*$, the difference between x_i' and x_i^* is at most $1/(2k)$. In particular, $x_i^* > 0$ since $x_i' \geq 1/k$. We have the bound $\|\mathbf{x}' - \mathbf{x}^*\| \leq 1/(2k)$, so the Lipschitz bound implies that $\partial f/\partial x_i(\mathbf{x}^*) > 0$. This, combined with the fact that $x_i^* > 0$, contradicts the minimality of $\mathbf{x}^*$.

Thus, $\|\mathbf{g}(\mathbf{x}')\|_\infty \leq M/(2k)$. Now we put an upper bound on the difference $f(\mathbf{x}') - f(\mathbf{x}^*)$. We use the same reasoning as in Section 2, namely we form a path with n segments between the two points and express $f(\mathbf{x}^*) - f(\mathbf{x}')$ as the integral of partial derivatives along the path. Each path segment has length at most $1/(2k)$, and each integrand is bounded below by $-2M/(2k)$. Therefore, the total difference is at least $-nM/(2k^2)$. This gives an upper bound on $f(\mathbf{x}') - f(\mathbf{x}^*)$. Since $f(\mathbf{x}^{(0)}) \leq f(\mathbf{x}')$ (because $\mathbf{x}^{(0)}$ is the coarse grid point with the smallest value of the objective function), we conclude that

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*) \leq \frac{nM}{2k^2}.$$

Starting from $\mathbf{x}^{(0)}$, we apply the same local search algorithm LOCAL1 as used in the Section 2. We now get a new bound on the number of steps. Since each step decreases the objective function by at least $0.5\epsilon^2/M$, and the maximum possible decrease is given above, we get a bound of $nM^2/(\epsilon^2 k^2)$ on the number of search iterations.

Thus, the algorithm requires a total of

$$(k+1)^n + \frac{nM^2}{\epsilon^2 k^2}$$

function and gradient evaluations. We want to choose k to be an integer that minimizes this total. A good choice is to pick k between

$$\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}} - 2 \leq k \leq \left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}} - 1.$$

With these choices, we can estimate

$$(k+1)^n \leq \left(\frac{nM^2}{\epsilon^2}\right)^{\frac{n}{n+2}}.$$

To analyze the other term, we assume that $M^2/\epsilon^2 \geq 4^{n+2}/n$ (recall that the method of this section is meant to be applied when n is small with respect to M/ϵ). If this holds, then $(nM^2/\epsilon^2)^{1/(n+2)} \geq 4$. Thus,

$$\frac{nM^2}{\epsilon^2 k^2} \leq \frac{nM^2}{\epsilon^2}\left[\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}} - 2\right]^{-2}$$

$$\leq \frac{nM^2}{\epsilon^2}\left[\frac{1}{2}\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}}\right]^{-2}$$
$$\leq 4\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{n}{n+2}}.$$

Thus, we see that the total time for LOCAL2 is at most

$$O\left(\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{n}{n+2}}\right).$$

In the special case of $n = 2$ (the case covered in the previous section), this gives a bound of $O(M/\epsilon)$, which is closer but still not equal to the lower bound.

The above choice of k will in general not be an integer divisor of M/ϵ. This can be addressed with further analysis, which we omit.

5 Local minima compared to global minima and fixed points

In Section 2 we came up with a bound like $O(nM^2/\epsilon^2)$ for finding ϵ-approximate local minima. The purpose of this section is to compare this result to information bounds for global minima and Brouwer fixed points. As we will see, these two other problems both depend on n exponentially.

Define an *ϵ-approximate global minimum* of a function $f : I^n \rightarrow \mathbb{R}$ to be a point $\mathbf{x}$ such that if $\mathbf{x}^*$ is a global minimum, then $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \epsilon$. It turns out the reasonable assumption to make for this problem is that f has Lipschitz bound L (rather than assuming a Lipschitz bound on ∇f). It is fairly straightforward to prove upper and lower bounds on this problem of the form $(cL/\epsilon)^n$, where c is a constant. This result is implicit in work by Sikorski [1984]. Thus, we see an exponential instead of polynomial dependence on n.

Local minima are more closely connected to Brouwer fixed points than to global minima. In fact, as we will show, local minima may be regarded as a special case of Brouwer fixed points. Let $\mathbf{u} : I^n \rightarrow I^n$ be a continuous function. Then Brouwer's fixed point theorem states that there exists an $\mathbf{x} \in I^n$ such that $\mathbf{u}(\mathbf{x}) = \mathbf{x}$. Such an $\mathbf{x}$ is called a *fixed point.*

Define an *ϵ-approximate fixed point* to be a vector $\mathbf{x} \in I^n$ such that $\|\mathbf{u}(\mathbf{x}) - \mathbf{x}\|_\infty \leq \epsilon$. It turns out that the reasonable assumption to make is that function $\mathbf{u}(\mathbf{x}) - \mathbf{x}$ has Lipschitz bound K. In this case, Hirsch, Papadimitriou, and Vavasis showed that the worst-case for Brouwer fixed points in the information model behaves roughly like $(cK/\epsilon)^n$, again, exponential in n.

We claim that local minima can in fact be couched as Brouwer fixed point problems. In particular, given a continuously differentiable function $f : I^n \rightarrow \mathbb{R}$ with a Lipschitz bound of M on the gradient, we define a vector-valued function $\mathbf{u}(\mathbf{x})$ as follows. For the purpose of this discussion, it is convenient to assume that $I^n = [-1/2, 1/2]^n$ so that the origin is the center of the domain. For $c > 0$ let $p_c(x)$ be the function that projects onto the interval $[-c, c]$ (i.e., $p_c(x) = \text{median}\{-c, x, c\}$), and let $\mathbf{p}_c$ be the coordinatewise projection onto $[-c, c]^n$, i.e., $\mathbf{p}_c(\mathbf{x}) = (p_c(x_1), \ldots, p_c(x_n))$.

Let ϵ' be slightly larger than ϵ. We define a new domain U to be $[-1/2 - \epsilon', 1/2 + \epsilon']^n$. Notice that $\mathbf{p}_{1/2}$ maps U to I^n. Then we define $\mathbf{u}(\mathbf{x})$ on U as follows.

$$\mathbf{u}(\mathbf{x}) = \mathbf{p}_{1/2}(\mathbf{x}) - \mathbf{p}_{\epsilon'}(\nabla f(\mathbf{p}_{1/2}(\mathbf{x})).$$

The image of $\mathbf{u}$ lies in U (because the first term has ∞-norm at most $1/2$, and the second term at most ϵ'), hence $\mathbf{u}(\mathbf{x})$ satisfies the conditions of Brouwer's theorem.

The first claim is that $\mathbf{u}(\mathbf{x}) - \mathbf{x}$ has Lipschitz constant equal to M. If $\mathbf{x} \in I^n$ then $\mathbf{u}(\mathbf{x}) = \mathbf{x} - \mathbf{p}_\epsilon(\nabla f(\mathbf{x}))$, hence $\mathbf{u}(\mathbf{x}) - \mathbf{x} = -\mathbf{p}_{\epsilon'}(\nabla f(\mathbf{x}))$. This right-hand side has Lipschitz constant of M. The other case is is handled in a similar manner.

Now, suppose $\mathbf{x} \in U$ is an ϵ-approximate fixed point of $\mathbf{u}$. Let $\mathbf{y} = \mathbf{p}_{1/2}(\mathbf{x})$; we claim $\mathbf{y}$ is an ϵ'-approximate local minimum of f. Let $\mathbf{d} = \mathbf{y} - \mathbf{x}$. For each i, if $d_i > 0$ then $y_i = -1/2$, and if $d_i < 0$ then $y_i = 1/2$. With this notation,

$$\mathbf{u}(\mathbf{x}) - \mathbf{x} = \mathbf{d} - \mathbf{p}_{\epsilon'}(\nabla f(\mathbf{y})). \tag{2}$$

The left-hand side of (2) is assumed to have ∞-norm at most ϵ. Consider an index i such that $y_i > -1/2$. In this case, $d_i \leq 0$, so (2) implies that ith entry of $\mathbf{p}_{\epsilon'}(\nabla f(\mathbf{y}))$ is at most ϵ. This means that $\partial f / \partial x_i(\mathbf{y}) \leq \epsilon$. Analogous reasoning applies to the case that $y_i < 1/2$. This shows that $\mathbf{y}$ satisfies the conditions for being an ϵ-KKT point, and hence an ϵ'-approximate local minimum.

Conversely, suppose $\mathbf{y}$ is an ϵ-approximate local minimum of f. For each i such that $y_i = 1/2$, define $d_i = \min(0, p_{\epsilon'}(\partial f/\partial x_i(\mathbf{y})))$. For each i such that $y_i = -1/2$, define $d_i = \max(0, p_{\epsilon'}(\partial f/\partial x_i(\mathbf{y})))$. For other i, let $d_i = 0$. Then it can be checked that the point $\mathbf{y} - \mathbf{d}$ will be an ϵ-approximate fixed point of $\mathbf{u}$.

Notice that the size of U is slightly larger than the size of I^n. The size of U can be brought to 1 in every dimension by scaling. This would have an effect on the value of M.

Thus, we see that approximate local minimization can be expressed as a special case of Brouwer fixed points. Finally, we remark that approximate local minimization is related to complexity classes PLS and PDLF designed for combinatorial problems, the first having to do with local minima and the second with Brouwer fixed points. See Papadimitriou [1990] for more information.

6 Conclusions

We have presented a simple local search algorithm whose running time is polynomial in the dimension of the problem. We have also presented a family of problems for which finding a local minimum would be time-consuming for any information-based algorithm.

There are many questions left unanswered by this work. What happens when more complicated domains than I^n are used? How can the gap between the lower bound of Section 3 and the upper bound of Section 4 be closed?

We have assumed that the functions under consideration are C^1. What if we assumed that they are C^2 with a Lipschitz bound on the second derivative? This would open up the possibility of using Newton-type methods. Would these Newton-type methods be provably more efficient than gradient-based methods?

Our algorithms LOCAL1 and LOCAL2 were designed mainly with ease of analysis in mind. Can a more practical algorithms (say, an algorithm with long steps and a line search) be put into the context of this paper and analyzed?

Finally, is there a good explanation for the fact that approximate local minima can be found in time polynomial in n but not Brouwer fixed points?

Acknowledgment

We are grateful to Michael Todd of Cornell for pointing out a key improvement to an earlier version of algorithms LOCAL1 and LOCAL2.

References

J. E. Dennis and R. B. Schnabel [1983], *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey.

M. D. Hirsch, C. H. Papadimitriou, and S. A. Vavasis [1989], Exponential lower bounds for finding brouwer fixed points, *J. Complexity* 5:379-416.

K. G. Murty and S. N. Kabadi [1987], Some NP-complete problems in quadratic and nonlinear programming, *Math. Progr.* 39:117-129.

C. H. Papadimitriou [1990], On graph-theoretic lemmata and complexity classes, preprint.

K. Sikorski [1984], Optimal solution of nonlinear equations satisfying a Lipschitz condition, *Numer. Math.* 43:225-240.

J. F. Traub, G. W. Wasilkowski, H. Woźniakowski [1983], *Information, Uncertainty, Complexity*, Addison-Wesley, Reading, MA.