Optimizing a 2D Function Satisfying Unimodality Properties

Erik D. Demaine [a], Stefan Langerman [b]

[a] *MIT Computer Science and Artificial Intelligence Laboratory, 200 Technology Square, Cambridge, MA 02139, USA*

[b] *Chargé de Recherche FNRS, Département d'informatique, Université Libre de Bruxelles, ULB CP212, Bruxelles, Belgium*

1. Introduction

A general formulation of discrete optimization is to maximize a given function $f : D \rightarrow \mathbb{R}$ over a discrete (finite) domain D. In general, of course, this problem may require $|D|$ probes to f. One approach to making optimization more tractable is to be satisfied with finding a *local* maximum, i.e., a point at which f attains a value larger than all "neighboring" points, for some definition of neighborhoods. In particular, for the standard 1D domain $D = \{1, 2, \ldots, n\}$, Fibonacci search [Kie53] finds a local maximum using $\log_\phi n + O(1)$ probes, where $\phi = (1 + \sqrt{5})/2$ is the golden ratio. Surprisingly, even for a square 2D domain $D = \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$, the problem complexity grows exponentially: Mityagin [Mit03] proved that $\Theta(n)$ probes to such an f are sufficient and sometimes necessary. Thus weakening the problem to finding local maxima does not cause the exponential speedup from 1D in higher dimensions.

Another approach to making optimization more tractable is to add assumptions about the function f. For example, if we assume that f is *unimodal* (denoted "$\odot$ unimodal"), i.e., it has exactly one local maximum, then finding local maxima and finding global maxima are equivalent. One could hope that having this structural information about the function would also help in finding that maximum. Unfortunately, a careful reading of the construction in [Mit03] of 2D functions f requiring $\Theta(n)$ probes are in fact unimodal.

We study the related condition that the 2D function f is unimodal in every row ($\leftrightarrow$ unimodal) and/or in every column ($\updownarrow$ unimodal). (These properties are satisfied by e.g. convex functions.) While seemingly weaker than unimodality, these properties are incomparable to unimodality, and in fact result in exponential speedup for finding local maxima.

Table 1 summarizes all of our results. Our upper bounds all follow from a combination of linear search and/or Fibonacci search in each dimension. Matching local bounds for global optimization follow in some cases from independence of the columns. Some bounds are tight only up to logarithmic factors, leaving intriguing open questions. In the full paper, we provide the omitted proof and prove the comforting fact that a natural random probing algorithm makes $\Omega(\lg m \lg n)$ expected probes even for a convex function, as in our dual Fibonacci search.

Other properties of matrices facilitating optimization have been studied; see [GP92].

2. Lower Bound for $\updownarrow$ Unimodal

Theorem 1 *If $n \geq m^\varepsilon$ where $0 < \varepsilon \leq 1$, there is an adversary that (a) generates $\updownarrow$-unimodal functions each with a unique local optimum, and (b) forces any local or global optimization algorithm to make $(\varepsilon^2/4) \lg m \lg n - O(\lg m \lg \lg n)$ probes.*

The adversary gives the algorithm extra information, which can only help. When-

Assumption	Local optimization		Global optimization	
None	$\leq min(\lg \frac{max}{min} + 4) + O(\lg max)$	[Mit03]	$\leq m \cdot n$	[lin.×lin.]
	$\geq min\{min, max/2\}$	[Mit03]	$\geq m \cdot n$	[indep.]
$\odot$ unimodal	$\leq min(\lg \frac{max}{min} + 4) + O(\lg max)$	[Mit03]	same as local	
	$\geq min\{min, max/2\}$	[Mit03]		
$\updownarrow$ unimodal	$\leq \log_\phi m \lg n + O(\lg m)$	[Fib.×Fib.]	$\leq n \log_\phi m + O(n)$	[Fib.×lin.]
	$\geq (\varepsilon^2/4) \lg m \lg n - O(\lg m \lg \lg n)$ *	[Thm. 1]	$\geq n \log_\phi m + O(n)$	[indep.]
$\updownarrow$, $\leftrightarrow$	$\leq \lg m \lg n / \lg \phi + O(\lg min)$	[Fib.×Fib.]	$\leq min \log_\phi max + O(min)$	[Fib.×lin.]
unimodal	$\geq \lg m + \lg n$	[info. theo.]	$\Omega(min)$	[omitted]
$\odot$, $\updownarrow$	$\leq \log_\phi m \lg n + O(\lg m)$	[Fib.×Fib.]	same as local	
unimodal	$\geq (\varepsilon^2/4) \lg m \lg n - O(\lg m \lg \lg n)$ *	[Thm. 1]		
$\odot$, $\updownarrow$, $\leftrightarrow$	$\leq \lg m \lg n / \lg \phi + O(\lg min)$	[Fib.×Fib.]	same as local	
unimodal	$\geq \lg m + \lg n$	[info. theo.]		

Table 1

Worst-case bounds on the number of probes required to maximize a function $f : \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\} \rightarrow \mathbb{R}$. In the bounds, $max = \max\{m, n\}$ and $min = \min\{m, n\}$. * indicates that the bound holds only when $n \geq m^\varepsilon$ for some $0 < \varepsilon \leq 1$.

ever the algorithm probes the value at a particular point (x, y), the adversary reveals not only that value, but also the slope of that value in that column, i.e., whether the mode in that column x is above or below that point (x, y). Furthermore, if the mode of column x is above the probe point (x, y), then the adversary reveals all values in the column x below the point (x, y); symmetrically, if the mode is below the probe point, the adversary reveals all values above the point in its column. If the algorithm discovers the mode of column x, the adversary reveals all values in the column x. Thus we maintain the invariant that every column that is not totally revealed has some revealed values in the topmost few rows, some revealed values in the bottommost few rows, and the algorithm knows that the mode of the column is somewhere in between.

If the unrevealed region ever becomes disconnected, the adversary reveals all values in all connected components except the largest connected component. Thus we maintain the invariant that the unrevealed region is connected. We also maintain the invariant that the algorithm cannot discover the unique local optimum until every value has been revealed. Together these two invariants make the goal of the algorithm to disconnect the unrevealed region; otherwise, the algorithm must make at least one probe per column, for a total of at least n probes.

The main task of the adversary is to decide whether a probe point is above or below the mode of that column, and then to choose the revealed values below or above the probe point. The adversary bases its decision on matching the "nearest" previous decision, according to a particular distance function. Define the *(biased) distance* between two points (x_1, y_1) and (x_2, y_2) to be $|x_1 - x_2| + |y_1 - y_2|/m^{1-\varepsilon/2}$. Also define the (biased) distance between a point (x, y) and the top horizontal wall to be $y/m^{1-\varepsilon/2}$, and similarly define the biased distance to the bottom wall to be $(m + 1 - y)/m^{1-\varepsilon/2}$.

Suppose that the algorithm probes the point (x, y). If point (x, y) is closer to a horizontal wall than every revealed point, then the adversary reveals all values in column x between (x, y) and the nearest wall, specifying that the mode is in the other direction. Otherwise, the adversary specifies (x, y) to be above or below the mode in its column x according to whether the revealed point (x^*, y^*) nearest to (x, y) is above or below the mode in its column x^*. Then the adversary reveals all unrevealed values starting from (x, y) in the opposite direction to the mode in column x. (In the special case described below that the algorithm discovers the mode among these revealed values, the specification that the mode is above or below (x, y) is false; in this case the adversary reveals all values in column x.)

The adversary chooses the revealed values as follows. Suppose that the algorithm probes (x, y) and say that the adversary decides that probe point (x, y) is above the mode in its column x. If the to-be-revealed points keep the unrevealed region connected, then the adversary repeatedly reveals that the bottommost unrevealed value in column x is one more than the largest previously revealed value, until reaching point (x, y). In this way the revealed values increase in an integer sequence from the bottommost unrevealed value to (x, y). Equivalently, the adversary reveals every unrevealed point (x, y') below (x, y) in column x to have value $m - d$ more than the largest previously revealed value, where $d = y - y'$ is the Manhattan distance between the unrevealed point (x, y') and the probe point (x, y).

On the other hand, if the to-be-revealed points disconnect the unrevealed region, then we isolate a point $(x, \hat{y})$ in column x that is adjacent to the largest resulting connected component, and assign that point $(x, \hat{y})$ to be the mode of column x. (As mentioned above, this assignment contradicts the recent decision of the adversary that the mode is above (x, y); this situation is the only one in which such a contradiction arises.) Then the adversary reveals every unrevealed point (x, y') in column x to have value $m + n - d$ more than the largest previously revealed value, where $d = |y' - \hat{y}|$ is the Manhattan distance between unrevealed point (x, y') and the assigned mode $(x, \hat{y})$ of column x. Simultaneously, we reveal every point (x', y') in every connected component except the largest to have value $m + n - d$ more than the largest previously revealed value, where d is the Manhattan distance between point (x', y') and the assigned mode $(x, \hat{y})$ of column x. Thus point $(x, \hat{y})$ indeed becomes the mode of column x, with $d = 0$.

Lemma 2 *The only point to become a local maximum according to the adversary is the mode of the final column to become completely revealed.*

PROOF. The key property is that the values revealed by the adversary are strictly larger as we proceed from one probe to the next, because we always add a positive number to the largest previously revealed value. Thus, within a column, every point except the mode has an adjacent point with larger revealed value, and therefore only the mode could be a local maximum. But in the final phase of a column, the mode is chosen so that it is adjacent to the largest connected component, and all values to be revealed in that component are strictly larger. Thus even the mode of the column cannot be a local maximum, unless the largest connected component is in fact empty, i.e., the last column has been completely revealed. □

Lemma 3 *Biased distance satisfies the triangle inequality.*

Lemma 4 *The algorithm must make* $\min\{n, (\varepsilon/2) \lg m\}$ *probes before the unrevealed region first disconnects into multiple connected components.*

PROOF. To disconnect the unrevealed region horizontally, the algorithm must make at least n probes. We claim that disconnecting the unrevealed region vertically requires at least $(\varepsilon/2) \lg m$ probes. Consider the minimum (biased) distance D between a revealed point above the mode in its column (or the top wall) and a revealed point below the mode in its column (or the bottom wall). Initially D is $m/m^{1-\varepsilon/2} = m^{\varepsilon/2}$. If the unrevealed region disconnects, D must have become at most 1 because $\varepsilon \leq 1$.

Whenever the algorithm probes a point (x, y) that does not disconnect the unrevealed region, we claim that D can decrease by at most a factor of 2. Suppose that (x, y) is closest to a previously revealed point above the mode in its column (or the top wall). Then every newly revealed point (x, y') is also closest to a previously revealed point above the mode in its column. By the triangle inequality (Lemma 3), the sum of the distances from every such point (x, y') to the nearest point above the mode in its column (or the top wall) and to the nearest point below the mode in its column (or the bottom wall) is at least D. The former distance is smaller, so the latter distance is at least $D/2$. Therefore the new value of D after this probe is at least $D/2$.

In conclusion, for D to reduce from $m^{\varepsilon/2}$ to at most 1, the algorithm must make at least $\lg m^{\varepsilon/2} = (\varepsilon/2) \lg m$ probes. □

Lemma 5 *The nearest point or horizontal wall to a point (x, y) is in a column x' such that $|x - x'| \leq m^{\varepsilon/2}$.*

Finally we conclude the proof of Theorem 1. Consider an algorithm that makes fewer than $\lg n \lg m$ probes. As mentioned above, the algorithm must disconnect the unrevealed region or else it is doomed to make at least n probes. Lemma 4 says that the algorithm must make at least $\min\{n, (\varepsilon/2) \lg m\}$ probes for the first disconnection. Consider the final probe that caused the disconnection. By the pigeonhole principle, the $(\lg n \lg m) m^{\varepsilon/2}$ consecutive columns including and to the right of this final probe must have a gap of at least $m^{\varepsilon/2}$ consecutive empty columns, because there are at most $\lg n \lg m$ probes total. We remove columns starting from the final probe up to but not including this gap of $m^{\varepsilon/2}$ consecutive empty columns. Similarly, we remove at most $(\lg n \lg m) m^{\varepsilon/2}$ columns to the left of the final probe up to but not including a gap of $m^{\varepsilon/2}$ consecutive empty columns. Thus we obtain two subproblems (one left and one right) that by Lemma 5 act completely independently from each other and from the probes causing the disconnection, as far as probes made so far. We recursively consider the subproblem corresponding to the larger connected component that remains. This recursive subproblem is a rectangle with m rows and at least $\lfloor n/2 \rfloor - (\lg n \lg m) m^{\varepsilon/2}$ columns. The recursive subproblem may have already been probed, but we can consider such probes as happening after this subproblem, because Thus the recursion applies until $n'/2 < (\lg n \lg m) m^{\varepsilon/2}$.

Therefore we obtain the lower bound of $\min\{n', (\varepsilon/2) \lg m\}$ probes, where $n' \geq 2(\lg n \lg m) m^{\varepsilon/2}$, recursively $\lg(n/(2(\lg n \lg m) m^{\varepsilon/2}))$ times. In total we obtain a lower bound of $(\varepsilon/2)(\lg m)(\lg n - 1 - \lg \lg n - \lg \lg m - (\varepsilon/2) \lg m)$ which simplifies to the bound stated in Theorem 1.

References

[GP92] Zvi Galil and Kunsoo Park. Dynamic programming with convexity, concavity and sparsity. *Theoret. Computer Sci.*, 92(1):49–76, January 1992.

[Kie53] J. Kiefer. Sequential minimax search for a maximum. *Proc. Amer. Math. Soc.*, 4:502–506, 1953.

[Mit03] Anton Mityagin. On the complexity of finding a local maximum of functions on discrete planar subsets. In *Proc. 20th Sympos. Theoret. Aspects Comput. Sci.*, volume 2607 of *Lecture Notes Comput. Sci.*, pages 203–211, February 2003.