WHY BENCHMARKING IS AN (ASYMPTOTICALLY) OPTIMAL APPROACH TO NUMERICAL METHODS: A GEOMBINATORIC PROOF

Vladik Kreinovich and Scott A. Starks
NASA Pan-American Center for Earth and Environmental Studies
University of Texas at El Paso
El Paso, TX 79968, USA
emails {vladik,sstarks}@utep.edu

Abstract. *In numerical mathematics, one of the most frequently used ways of gauging the quality of different numerical methods is benchmarking. Specifically, once we have methods that work well on some (but not all) problems from a given problem class, we find the problem that is the toughest for the existing methods. This problem becomes a benchmark for gauging how well different methods solve problems that previous methods could not. Once we have a method that works well in solving this benchmark problem, we repeat the process again – by selecting, as a new benchmark, a problem that is the toughest to solve by the new methods, and by looking for a new method that works the best on this new benchmark. At first glance, this idea sounds like a heuristic, but its success in numerical mathematics indicates that this heuristic is either optimal or at least close to optimality. In this paper, we use the geombinatoric approach to prove that benchmarking is indeed asymptotically optimal.*

What is benchmarking and how is it usually done. In many areas such as numerical mathematics, computer architecture, Artificial Intelligence, etc., we are interested in solving problems from a certain class. In order to gauge how good is a given method in solving these problems, researchers usually select several *benchmark* problems on which different methods are tested.

We would like to select the benchmark problems in such a way that the method's performance on these problems serves as a good indication of its performance on all other problems.

Usually, we start with a problem (or problems) that really need

to be solved, so these problem become our first benchmarks. We then find a method (or several methods) that solves all these benchmark problems, and start using this method to solve other problems as well. After some time, it turns out that this method (or methods) does not work well on some of the problems. So, we select the toughest of these difficult-to-solve problems as a new benchmark, and look for a method that works well not only for the old benchmarks, but for the new benchmark as well. After such a method is designed, we start using it. Again, it usually turns out that this method is not working well for some problems, so we select the roughest of them as a new benchmark, etc.

Formulation of the problem. The above procedure looks like a heuristic. However, its success in many areas such as numerical mathematics indicates that this heuristic is probably either optimal or at least close to optimality.

In this paper, we use the geombinatoric approach to prove that benchmarking is indeed asymptotically optimal.

Towards geometric formulation of the problem. The main idea behind benchmarking is that if two problems p and q are close, then the behavior of a method on one of these problem is a good indicator of how well this method performs on another problem. A natural way to describe closeness is by a distance $d(p, q)$: the smaller this distance, the better the behavior of the method on the problem p can predict how this method behaves on the problem q.

Thus, if we have a single benchmark problem p_1, and a method works well on this problem, then this method's behavior on a problem $p \neq p_1$ can be gauged by the distance $d(p, p_1)$.

Suppose now that we have *several* benchmark problems $p_1, \ldots, p_n$, we have a method that works well on all of them, and we want to estimate this method's behavior on a problem $p \neq p_i$. The closer p to one of the benchmarks p_i, the better the predicted behavior. Thus, as a gauge of how well the method behaves, we can take the distance

$$d(p, \{p_1, \ldots, p_n\}) \stackrel{\text{def}}{=} \min_i d(p, p_i). \qquad (1)$$

For each benchmark set $\{p_1, \ldots, p_n\}$, this formula leads to different quality on different problems p. A natural numerical characteristic of the overall quality of a benchmark set is its worst-case behavior, i.e., the value

$$Q(\{p_1, \ldots, p_n\}) \stackrel{\text{def}}{=} \max_p d(p, \{p_1, \ldots, p_n\}). \qquad (2)$$

Thus, we can reformulate the problem of selecting optimal benchmarks as follows:

Let a metric space be given; given an integer n, find n points $p_1^{\text{opt}}, \ldots, p_n^{\text{opt}}$ for which the value $Q(\{p_1, \ldots, p_n\})$ is the smallest possible, i.e., for which

$$Q(\{p_1^{\text{opt}}, \ldots, p_n^{\text{opt}}\}) = Q_{\text{opt}}(n) \stackrel{\text{def}}{=} \min_{p_1, \ldots, p_n} Q(\{p_1, \ldots, p_n\}).$$

Comment 1. The value $Q_{\text{opt}}(n)$ is related to a so-called ε-entropy H_ε of a metric space (see, e.g., [Lorentz 1966] and [Lorentz 1976]). Specifically, for a metric space M and a real number ε:

- an *ε-net* is a subset $S \subseteq M$ such that for every $p \in M$, there exists an $s \in S$ for which $d(p, s) \leq \varepsilon$;
- N_ε is defined as the smallest possible number of elements in an ε-net;
- the *ε-entropy* is defined as $\log(N_\varepsilon)$.

It can be shown (see, e.g., [Kreinovich 1974] and [Kreinovich 1975]) that the function $Q_{\text{opt}}(n)$ is an inverse to N_ε:

$$Q_{\text{opt}}(n) = \min\{\varepsilon \mid N_\varepsilon = n\}.$$

Comment 2. It makes sense to only consider *compact* metric spaces, because only for them, $Q_{\text{opt}}(n) \to 0$ as n increases.

Traditional benchmarking approach reformulated in geometric terms. For a given metric space M and a given integer n, we sequentially select the points $p_1^{\text{alg}}, \ldots, p_n^{\text{alg}}$ as follows:

- we arbitrarily select the first point $p_1^{\text{alg}} \in M$;
- once we have selected the points $p_1^{\text{alg}}, \ldots, p_{k-1}^{\text{alg}}$ and $k \le n$, we select, as p_k^{alg}, the point $p \in M$ for which the distance $d(p, \{p_1^{\text{alg}}, \ldots, p_{k-1}^{\text{alg}}\})$ is the largest possible; if there are several such points, we select one of them.

Let us denote the corresponding value of Q by $Q_{\text{alg}}(n)$.

Comment. From the algorithmic viewpoint, this method can be viewed as a "greedy" algorithm (see, e.g., [Cormen et al. 2001]) in the sense that at each step, we select the (locally) best next point. It is known that in many cases, greedy algorithms are asymptotically optimal; this is true in our case as well:

Proposition. $Q_{\text{opt}}(n) \le Q_{\text{alg}}(n) \le 2 \cdot Q_{\text{opt}}(n-1)$.

Proof. The first inequality follows from the fact that $Q_{\text{opt}}(n)$ is the quality of the optimal n-element set, so it is sufficient to prove the second inequality.

To prove this inequality, let us first prove that for any two different points p_j^{alg} and p_k^{alg} ($j \ne k$) from sequence $p_1^{\text{alg}}, \ldots, p_n^{\text{alg}}$, we have $d(p_j^{\text{alg}}, p_k^{\text{alg}}) \ge Q_{\text{alg}}(n)$. (W.l.o.g., we can assume that $j < k$.)

We will prove this auxiliary statement by reduction to a contradiction. Suppose that $Q_{\text{alg}}(n) > d(p_j^{\text{alg}}, p_k^{\text{alg}})$. By definition of $Q_{\text{alg}}(n)$, this means that there exists a point p for which, for all i from 1 to n, we have $d(p_i^{\text{alg}}, p) \ge Q_{\text{alg}} > d(p_j^{\text{alg}}, p_k^{\text{alg}})$. In particular, this is true for all $i < k$, thus,

$$\min_{i=1,\ldots,k-1} d(p_i^{\text{alg}}, p) > d(p_j^{\text{alg}}, p_k^{\text{alg}}).$$

Since $j < k$, we can thus conclude that

$$\min_{i=1,\ldots,k-1} d(p_i^{\text{alg}}, p) > \min_{i=1,\ldots,k-1} d(p_i^{\text{alg}}, p_k^{\text{alg}}).$$

This inequality contradicts to our selection of p_k^{alg} as the point p for which the value

$$d(p, \{p_1^{\text{alg}}, \ldots, p_{k-1}^{\text{alg}}\}) = \min_{i=1,\ldots,k-1} d(p_i^{\text{alg}}, p)$$

is the largest possible. This contradiction shows that our original assumption that $Q_{\text{alg}}(n) > d(p_j^{\text{alg}}, p_k^{\text{alg}})$ is impossible, hence $d(p_j^{\text{alg}}, p_k^{\text{alg}}) \geq Q_{\text{alg}}(n)$.

Now, we are ready to prove the proposition itself. By definition of $Q_{\text{opt}}(n-1)$, there exists an $(n-1)$-element set $\{p_1^{\text{opt}}, \ldots, p_{k-1}^{\text{opt}}\}$ that is a $Q_{\text{opt}}(n-1)$-net, i.e., for which, for every $p \in M$, there exists an i for which $d(p, p_i^{\text{opt}}) \leq Q_{\text{opt}}(n-1)$. In particular, for every j from 1 to n, this is true for $p = p_j^{\text{alg}}$, i.e., for every such j, there exists an i from 1 to $n-1$ for which $d(p_j^{\text{alg}}, p_i^{\text{opt}}) \leq Q_{\text{opt}}(n-1)$. Since we have n different j's and only $n-1$ different i's, inevitably, there exist $j \neq k$ for which the corresponding value i is the same, i.e., for which $d(p_j^{\text{alg}}, p_i^{\text{opt}}) \leq Q_{\text{opt}}(n-1)$ and $d(p_k^{\text{alg}}, p_i^{\text{opt}}) \leq Q_{\text{opt}}(n-1)$ for the same i. Due to the triangle inequality, we thus have

$$d(p_j^{\text{alg}}, p_k^{\text{alg}}) \leq d(p_k^{\text{alg}}, p_i^{\text{opt}}) + d(p_j^{\text{alg}}, p_i^{\text{opt}}) \leq 2 \cdot Q_{\text{opt}}(n-1).$$

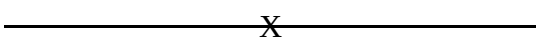
Since we have proven that $d(p_j^{\text{alg}}, p_k^{\text{alg}}) \geq Q_{\text{alg}}(n)$, we thus get the desired inequality. The proposition is proven.

Example 1: linear segment. To illustrate the situation, let us start with the simplest possible example in which the metric space M is a closed interval on a straight line – namely, the interval $[0, 1]$ on the real line:

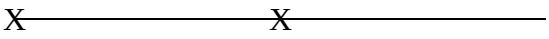
In this example, as one can easily see, for every n,

$$p_i^{\text{opt}} = \frac{i - 1/2}{n},$$

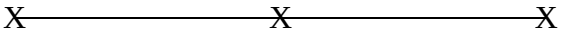
and $Q_{\text{opt}}(n) = 1/(2 \cdot n)$. Let us select the midpoint 1/2 as p_1^{alg}:



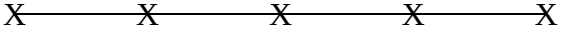
Here, $Q_{\text{alg}}(1) = 1/2$. There are two points that are the farthest from p_1^{alg}: the left endpoint 0 and the right endpoint 1. W.l.o.g., let us select $p_2^{\text{alg}} = 0$:



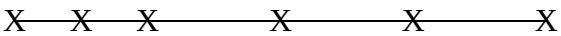
Here, $Q_{\text{alg}}(2) = 1/2$. Now, 1 is the point with the largest value of $Q(p, \{p_1^{\text{alg}}, p_2^{\text{alg}}\})$, so we take $p_3^{\text{alg}} = 1$:



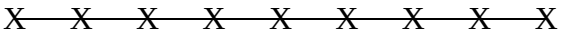
At this step, $Q_{\text{alg}}(3) = 1/4$; the midpoints between 0 and 1/2 and between 1/2 and 1 are the farthest, so, after two steps, we add them both:



So, we have $Q_{\text{alg}}(4) = 1/4$ and $Q_{\text{alg}}(5) = 1/8$. At the next step, we add one of the points in between the existing ones, e.g., the first one (1/8):



After three more steps, we add all midpoints, so we arrive at the following configuration:



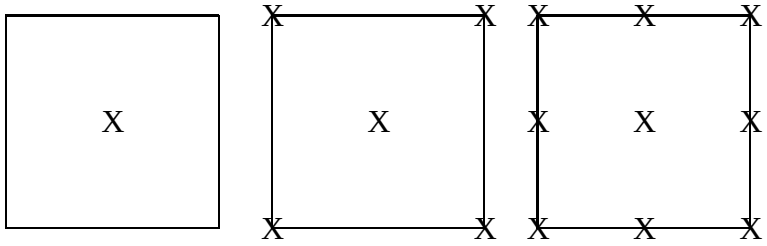
so that $Q_{\text{alg}}(6) = Q_{\text{alg}}(7) = Q_{\text{alg}}(8) = 1/8$ and $Q_{\text{alg}}(9) = 1/16$, etc.

In general, for $n = 2^k$, we have $Q_{\text{alg}}(n) = 1/n$, while $Q_{\text{opt}}(n-1) = 1/(2 \cdot (n-1))$. Hence, the ratio

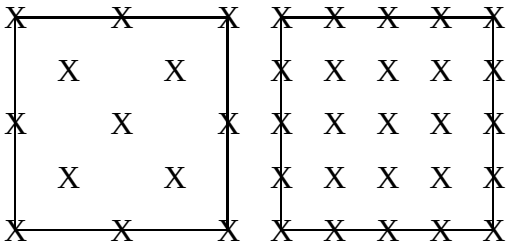
$$\frac{Q_{\text{alg}}(n)}{Q_{\text{opt}}(n-1)} = \frac{2 \cdot (n-1)}{n}$$

tends to 2 as $n \to \infty$. Thus, in the above Proposition, the factor 2 is optimal – in the sense that it cannot be replaced by any smaller number.

Example 2: square. For a square, we get a similar picture:



First, let us pick a midpoint as p_1^{alg} (the left picture). Then, the next four benchmarks are the vertices (see middle picture), after which the next four as the midpoints of the four edges (right picture). Here, we have, in effect, four sub-squares. One the next stage, the same procedure is repeated for each sub-square, etc.



Example 3: equilateral triangle. In this case, if we select the center as p_1^{alg}, we first get the vertices as the next 3 benchmarks, then the midpoints of the 3 edges, then the midpoints of the lines connecting the vertices with the center, then the 1/4 and 3/4 points on the edges, etc. At each stage, we have sub-triangles of repeating shape, and the process repeats for these sub-triangles.

Open problems. Since this geometric problem is of potential importance to numerical methods, it is desirable:

- to describe optimal points for different geometric shapes;
- to describe what points the "greedy" algorithm returns;
- to look for better (closer to optimal) methods of selecting benchmark problems.

Acknowledgments. This work was supported in part by NASA under cooperative agreement NCC5-209 and grant NCC2-1232, by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-00-1-0365, by NSF grants CDA-9522207, EAR-0112968, EAR-0225670, and 9710940 Mexico/Conacyt, and by the IEEE/ACM SC2001 and SC2002 Minority Serving Institutions Participation Grants.

References

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, and Mc-Graw Hill Co., N.Y., 2001.

V. Kreinovich, "Constructivization of the notions of epsilon- entropy and epsilon-capacity," *Proceedings of the Leningrad Mathematical Institute of the Academy of Sciences*, 1974, Vol. 40, pp. 38–44 (in Russian), English translation: *Journal of Soviet Mathematics*, 1977, Vol. 8, No. 3, pp. 271–276.

V. Kreinovich, "On A. F. Timan's inverse problem, for ϵ-entropy of compacts," *Soviet Mathematics Uspekhi*, 1975, Vol. 30, No. 1, pp. 241–242 (in Russian).

G. G. Lorentz, *Approximation of functions*, Halt, Reinhart, and Winston, N.Y., 1966.

G. G. Lorentz, "The 13-th problem of Hilbert", in: F. E. Browder (ed.), *Mathematical Developments Arising from Hilbert's Problems*, American Math. Society, Providence, RI, 1976, Part 2, pp. 419–430.