Time-Space Trade-offs for Voronoi Diagrams

Matias Korman* Wolfgang Mulzer† André van Renssen* Marcel Roeloffzen‡ Paul Seiferth†
Yannik Stein†

Abstract

Let S be a planar n-point set. Classically, one can find the Voronoi diagram $\mathrm{VD}(S)$ for S in $O(n \log n)$ time and $O(n)$ space. We study the situation when the available workspace is limited: for $s \in \{1, \ldots, n\}$, an s-workspace algorithm has read-only access to an input array with the points from S in arbitrary order, and it may use only $O(s)$ additional words of $\Theta(\log n)$ bits for reading and writing intermediate data. We describe a randomized s-workspace algorithm for finding $\mathrm{VD}(S)$ in expected time $O((n^2/s) \log s + n \log s \log^* s)$. This almost matches the optimal running times for both constant and linear workspace and provides a continuous trade-off between them.

1 Introduction

Since the beginning of computer science, there has been interest in algorithms that can deal with strong memory constraints. This started in the early 70's [10] when memory was expensive, but now it is motivated by the proliferation of small embedded devices where a lot of memory is neither feasible nor desirable.

Even when space is not an issue, one might want to limit the number of write operations: even though one can read flash memory very fast, writing (or even reordering) data is slow and reduces the lifetime; write-access to removable memory is sometimes limited for technical or security reasons; similar problems occur when concurrent algorithms need to access the same data and create concurrency problems. A way to deal with this is to consider algorithms that do not modify the input, and use as few variables as possible.

The exact setting may vary, but there is a common theme: the input resides in a read-only data structure, the output must be written to some write-only structure, and we can use $O(s)$ additional variables to compute the solution (for a given parameter s). Our aim is to design algorithms whose running time decreases as s grows, giving a *time-space trade-off* [11].

One of the initial problems considered in this model is *sorting* [7, 8]. It is known that the time-space product of any sorting algorithm is $\Omega(n^2)$ [5], and matching upper bounds for the case $b \in \Omega(\log n) \cap O(n/\log n)$ were obtained by Pagter and Rauhe [9] (b denotes the size of the workspace, in bits). Since the sorted list cannot be stored explicitly in memory, we must report the values one by one in order.

The concept of memory constrained algorithms was introduced to computational geometry by Asano et al. [2]. They show how to compute many classic geometric structures with $O(1)$ workspace. Afterwards, several time-space trade-off algorithms have been designed for classic problems within a simple polygon, such as shortest path computation [1], visibility [4], or computing the convex hull of a simple polygon [3].

Problem Setting. We are given a list S of n points in the plane. We assume that the points are in some structure (say, an array) that allows random access to any point. We would like to design an algorithm that computes the Voronoi diagram of S, $\mathrm{VD}(S)$. Since the diagram itself cannot be explicitly stored in memory, the aim is to report its vertices one by one in a write-only data structure in no particular order. In addition to the input, the algorithm can use $O(s)$ variables (for some parameter $s \leq n$). We assume that each variable or pointer contains a data word of $\Theta(\log n)$ bits.

Our aim is an algorithm whose running time decreases as s grows. Ideally, we would like that the trade-off is continuous and that the running times for both extremes of s match with the currently best known algorithms for these cases ($O(n^2)$ and $O(n \log n)$ time for $s = 1$ and $s = n$, respectively). As we will see below, we can almost achieve this goal.

2 Voronoi Diagrams Through Random Sampling

Given a planar n-point set S, we would like to find the vertices of $\mathrm{VD}(S)$. Let $K = \{p_1, p_2, p_3\}$ be a triangle with $S \subseteq \mathrm{conv}(K)$ so that all vertices of $\mathrm{VD}(S)$ are vertices of $\mathrm{VD}(S \cup K)$. We use random sampling to divide the problem of computing $\mathrm{VD}(S \cup K)$ into $O(s)$ subproblems of size $O(n/s)$. First, we show how to take a random sample from S with small workspace.

Lemma 1 *We can sample a uniform random subset $R \subseteq S$ of size s in time $O(n + s \log s)$ and space $O(s)$.*

*National Institute of Informatics (NII), Tokyo, Japan; JST, ERATO, Kawarabayashi Large Graph Project. {korman, andre}@nii.ac.jp

†Institut für Informatik, Freie Universität Berlin, Germany. {mulzer,pseiferth,yannikstein}@inf.fu-berlin.de

‡Tohoku University, Japan. marcel@dais.is.tohoku.ac.jp

Proof. We sample a random sequence I of s distinct numbers from $\{1,\dots,n\}$. This is done in s *rounds*. At the beginning of round k, for $k=1,\dots,s$, we have a sequence I of $k-1$ numbers from $\{1,\dots,n\}$. We store I in a binary search tree T. We maintain the invariant that each node in T with value in $\{1,\dots,n-k+1\}$ stores a pointer to a unique number in $\{n-k+2,\dots,n\}$ that is not in I. In round k, we sample a random number x from $\{1,\dots,n-k+1\}$, and we check in T whether $x\in I$. If not, we add x to I. Otherwise, we add to I the number that x points to. Let y be the new element. We add y to T. Then we update the pointers: if $x=n-k+1$, we do nothing. Now suppose $x<n-k+1$. Then, if $n-k+1\notin I$, we put a pointer from x to $n-k+1$. Otherwise, if $n-k+1\in I$, we let x point to the element that $n-k+1$ points to. This keeps the invariant and takes $O(\log s)$ time and $O(s)$ space. We continue for s rounds. Any sequence of s distinct numbers in $\{1,\dots,n\}$ is sampled with equal probability.

Finally, we scan through S to obtain the elements whose positions correspond to the numbers in I. This requires $O(n)$ time and $O(s)$ space. □

We use Lemma 1 to find a random sample $R\subseteq S$ of size s. We compute $\mathrm{VD}(R\cup K)$, triangulate the bounded cells and construct a planar point location structure for the triangulation. This takes $O(s\log s)$ time and $O(s)$ space. Given a vertex $v\in\mathrm{VD}(R\cup K)$, the *conflict circle* of v is the largest circle with center v and no point from $R\cup K$ in its interior. The *conflict set* B_v of v contains all points from S that lie in the conflict circle of v, and the *conflict size* b_v of v is the number of points in B_v. We scan through S to find the conflict size b_v for each vertex $v\in\mathrm{VD}(R\cup K)$: every Voronoi vertex has a counter that is initially 0. For each $p\in S\setminus(R\cup K)$, we use the point location structure to find the triangle Δ of $\mathrm{VD}(R\cup K)$ that contains it. At least one vertex v of Δ is in conflict with p. Starting from v, we walk along the edges of $\mathrm{VD}(R\cup K)$ to find all Voronoi vertices in conflict with p. We increment the counters of all these vertices. This may take a long time in the worst case, so we impose an upper bound on the total work. For this, we choose a *threshold* M. When the sum of the conflict counters exceeds M, we start over with a new sample R. The total time for one attempt is $O(n\log s+M)$, and below we prove that for $M=\Theta(n)$ the success probability is at least $3/4$. Next, we pick another threshold T, and we compute for each vertex v of $\mathrm{VD}(R\cup K)$ the *excess* $t_v=b_v s/n$. The excess measures how far the vertex deviates from the desired conflict size n/s. We check if $\sum_{v\in\mathrm{VD}(R\cup K)} t_v\log t_v\le T$. If not, we start over with a new sample. Below, we prove that for $T=\Theta(s)$, the success probability is at least $3/4$. The total success probability is $1/2$, and the expected number of attempts is 2. Thus, in expected time $O(n\log s+s\log s)$, we can find a sample $R\subseteq S$ with $\sum_{v\in\mathrm{VD}(R\cup K)} b_v=O(n)$ and $\sum_{v\in\mathrm{VD}(R\cup K)} t_v\log t_v=O(s)$.

We now analyze the success probabilities, using the classic Clarkson-Shor method. We begin with the following version of the Chazelle-Friedman bound [6].

Lemma 2 *Let X be a planar point set of size o, and let $Y\subset\mathbb{R}^2$ with $|Y|\le 3$. For fixed $p\in(0,1]$, let $R\subseteq X$ be a random subset of size po and let $R'\subseteq X$ be a random subset of size $p'o$, for $p'=p/2$. Suppose that $p'o\ge 4$. Fix $\mathbf{u}\in X^3$, and let $v_{\mathbf{u}}$ be the Voronoi vertex defined by $\mathbf{u}$. Let $b_{\mathbf{u}}$ be the number of points from X in the largest circle with center $v_{\mathbf{u}}$ and with no points from R in its interior. Then,*

$$\Pr[v_{\mathbf{u}}\in\mathrm{VD}(R\cup Y)]\le 64e^{-pb_{\mathbf{u}}/2}\Pr[v_{\mathbf{u}}\in\mathrm{VD}(R'\cup Y)].$$

Proof. Let $\sigma=\Pr[v_{\mathbf{u}}\in\mathrm{VD}(R\cup Y)]$ and $\sigma'=\Pr[v_{\mathbf{u}}\in\mathrm{DT}(R'\cup Y)]$. The vertex $v_{\mathbf{u}}$ is in $\mathrm{VD}(R\cup Y)$ precisely if $\mathbf{u}\subseteq R\cup Y$ and $B_{\mathbf{u}}\cap(R\cup Y)=\emptyset$, where $B_{\mathbf{u}}$ are the points from X in the conflict circle of $v_{\mathbf{u}}$. If $Y\cap B_{\mathbf{u}}\neq\emptyset$, then $\sigma=\sigma'=0$, and the lemma holds. Thus, assume that $Y\cap B_{\mathbf{u}}=\emptyset$. Let $d_{\mathbf{u}}=|\mathbf{u}\setminus Y|$, the number of points in $\mathbf{u}$ not in Y. There are $\binom{o-b_{\mathbf{u}}-d_{\mathbf{u}}}{po-d_{\mathbf{u}}}$ ways to choose a po-subset from X that avoids all points in $B_{\mathbf{u}}$ and contains all points of $\mathbf{u}\cap X$, so

$$\begin{aligned}\sigma&=\binom{o-b_{\mathbf{u}}-d_{\mathbf{u}}}{po-d_{\mathbf{u}}}\Big/\binom{o}{po}\\&=\frac{\prod_{j=0}^{po-d_{\mathbf{u}}-1}(o-b_{\mathbf{u}}-d_{\mathbf{u}}-j)}{\prod_{j=0}^{po-d_{\mathbf{u}}-1}(po-d_{\mathbf{u}}-j)}\Big/\frac{\prod_{j=0}^{po-1}(o-j)}{\prod_{j=0}^{po-1}(po-j)}\\&=\prod_{j=0}^{d_{\mathbf{u}}-1}\frac{po-j}{o-j}\cdot\prod_{j=0}^{po-d_{\mathbf{u}}-1}\frac{o-b_{\mathbf{u}}-d_{\mathbf{u}}-j}{o-d_{\mathbf{u}}-j}\\&\le p^{d_{\mathbf{u}}}\prod_{j=0}^{po-d_{\mathbf{u}}-1}\left(1-\frac{b_{\mathbf{u}}}{o-d_{\mathbf{u}}-j}\right).\end{aligned}$$

Similarly, we get

$$\sigma'=\prod_{i=0}^{d_{\mathbf{u}}-1}\frac{p'o-i}{o-i}\prod_{j=0}^{p'o-d_{\mathbf{u}}-1}\left(1-\frac{b_{\mathbf{u}}}{o-d_{\mathbf{u}}-j}\right),$$

and since $p'o\ge 4$ and $i\le 2$, it follows that

$$\sigma'\ge\left(\frac{p'}{2}\right)^{d_{\mathbf{u}}}\prod_{j=0}^{p'o-d_{\mathbf{u}}-1}\left(1-\frac{b_{\mathbf{u}}}{o-d_{\mathbf{u}}-j}\right).$$

Therefore, since $p'=p/2$,

$$\begin{aligned}\frac{\sigma}{\sigma'}&\le\left(\frac{2p}{p'}\right)^{d_{\mathbf{u}}}\prod_{j=p'o-d_{\mathbf{u}}}^{po-d_{\mathbf{u}}-1}\left(1-\frac{b_{\mathbf{u}}}{o-d_{\mathbf{u}}-j}\right)\\&\le 64\left(1-\frac{b_{\mathbf{u}}}{o}\right)^{po/2}\le 64\,e^{pb_{\mathbf{u}}/2}.\end{aligned}$$

□

We can now bound the total expected conflict size.

Lemma 3 *We have* $\mathbf{E}\left[\sum_{v\in \mathrm{VD}(R\cup K)} b_v\right] = O(n)$.

Proof. By expanding the expectation, we get

$$\mathbf{E}\left[\sum_{v\in \mathrm{VD}(R\cup K)} b_v\right] = \sum_{\mathbf{u}\in S^3} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R\cup K)] b_{\mathbf{u}},$$

$v_{\mathbf{u}}$ being the Voronoi vertex of $\mathbf{u}$ and $b_{\mathbf{u}}$ its conflict size. By Lemma 2 with $X = S$, $Y = K$ and $p = s/n$,

$$\leq \sum_{\mathbf{u}\in S^3} 64e^{-pb_{\mathbf{u}}/2} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)] b_{\mathbf{u}},$$

where $R' \subseteq S$ is a sample of size $s/2$. We estimate

$$\leq \sum_{t=0}^{\infty} \sum_{\substack{\mathbf{u}\in S^3 \\ b_{\mathbf{u}}\in[\frac{t}{p},\frac{t+1}{p})}} \frac{64e^{-t/2}(t+1)}{p} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)]$$

$$\leq \frac{1}{p} \sum_{\mathbf{u}\in S^3} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)] \sum_{t=0}^{\infty} 64e^{-t/2}(t+1)$$

$$= O(s/p) = O(n),$$

since $\sum_{\mathbf{u}\in S^3} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)] = O(s)$ is the size of $\mathrm{VD}(R'\cup K)$ and $\sum_{t=0}^{\infty} e^{-t/2}(t+1) = O(1)$. □

By Lemma 3 and Markov's inequality, there is an $M = \Theta(n)$ with $\Pr[\sum_{v\in \mathrm{VD}(R\cup K)} b_v > M] \leq 1/4$.

Lemma 4 $\mathbf{E}\left[\sum_{v\in \mathrm{VD}(R\cup K)} t_v \log t_v\right] = O(s)$.

Proof. By Lem. 2 with $X = S$, $Y = K$, and $p = s/n$,

$$\mathbf{E}\left[\sum_{v\in \mathrm{VD}(R\cup K)} t_v \log t_v\right]$$

$$= \sum_{\mathbf{u}\in S^3} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R\cup K)]\, t_{\mathbf{u}} \log t_{\mathbf{u}}$$

$$\leq \sum_{\mathbf{u}\in S^3} 64e^{-pb_{\mathbf{u}}/2} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)] t_{\mathbf{u}} \log t_{\mathbf{u}}$$

$$\leq \sum_{t=0}^{\infty} \sum_{\substack{\mathbf{u}\in S^3 \\ b_{\mathbf{u}}\in[\frac{t}{p},\frac{t+1}{p})}} 64e^{-\frac{t}{2}}(t+1)^2 \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)]$$

$$\leq \sum_{t=0}^{\infty} 64e^{-t/2}(t+1)^2 \sum_{\mathbf{u}\in S^3} \Pr[v_{\mathbf{u}} \in \mathrm{VD}(R'\cup K)]$$

$$= O(s).$$

□

By Markov's inequality and Lemma 4, there is a $T = \Theta(s)$ with $\Pr[\sum_{v\in \mathrm{VD}(R\cup K)} t_v \log t_v \geq T] \leq 1/4$. This finishes the first phase of the sampling.

Let $\alpha > 0$ be a sufficiently large constant. The next goal is to sample for each vertex v with $t_v \geq 2$ a random subset $R_v \subseteq B_v$ of size $\alpha t_v \log t_v$ (recall that B_v is the conflict set of v).

Lemma 5 *In total time $O(n \log s)$, we can sample for each vertex $v \in \mathrm{VD}(R\cup K)$ with $t_v \geq 2$ a random subset $R_v \subseteq B_v$ of size $\alpha t_v \log t_v$.*

Proof. First, we perform $O(s)$ rounds to sample for each vertex v with $t_v \geq 2$ a sequence I_v of $\alpha t_v \log t_v$ distinct numbers from $\{1, \ldots, b_v\}$. For this, we use the algorithm from Lemma 1 in parallel for each relevant vertex from $\mathrm{VD}(R\cup K)$. Since $\sum_v t_v \log t_v = O(s)$, this takes total time $O(s \log s)$ and total space $O(s)$.

After that, we scan through S. For each vertex v, we have a counter c_v, initialized to 0. For each $p \in S$, we find the conflict vertices of p, and for each conflict vertex v, we increment c_v. If c_v appears in the corresponding set I_v, we add p to R_v. The total running time is $O(n \log s)$, as we do one point location for each input point and the total conflict size is $O(n)$. □

We next show that for a *fixed* vertex $v \in \mathrm{VD}(R\cup K)$, with constant probability, all vertices in $\mathrm{VD}(R_v)$ have conflict size n/s with respect to B_v.

Lemma 6 *Let $v \in \mathrm{VD}(R\cup K)$ with $t_v \geq 2$, and let $R_v \subseteq B_v$ be the sample for v. The expected number of vertices v' in $\mathrm{VD}(R_v)$ with at least n/s points from B_v in their conflict circle is at most $1/2$.*

Proof. Recall that $t_v = b_v s/n$. We have

$$\mathbf{E}\left[\sum_{\substack{v'\in \mathrm{VD}(R_v) \\ b'_{v'}\geq n/s}} 1\right] = \sum_{\substack{\mathbf{u}\in B_v^3 \\ b'_{\mathbf{u}}\geq n/s}} \Pr[v'_{\mathbf{u}} \in \mathrm{VD}(R_v)],$$

where $b'_{\mathbf{u}}$ is the conflict size of $v'_{\mathbf{u}}$ with respect to B_v. Using Lemma 2 with $X = B_v$, $Y = \emptyset$, and $p = (\alpha t_v \log t_v)/b_v = \alpha(s/n)\log t_v$, this is

$$\leq \sum_{\substack{\mathbf{u}\in B_v^3 \\ b'_{\mathbf{u}}\geq n/s}} 64e^{-pb'_{\mathbf{u}}/2} \Pr[v'_{\mathbf{u}} \in \mathrm{VD}(R'_v)]$$

$$\leq 64e^{-(\alpha/2)\log t_v} \sum_{\mathbf{u}\in B_v^3} \Pr[v'_{\mathbf{u}} \in \mathrm{VD}(R'_v)]$$

$$= O(t_v^{-\alpha/2} t_v \log t_v) \leq 1/2,$$

for α large enough (remember that $t_v \geq 2$). □

By Lemma 6 and Markov's inequality, the probability that all vertices from $\mathrm{VD}(R_v)$ have at most $2n/s$ points from B_v in their conflict circles is at least $1/2$.

If so, we call v *good*. Scanning through S, we can identify the good vertices in time $O(n \log s)$ and space $O(s)$. The expected number of good vertices is $s'/2$, where s' is the size of $\mathrm{VD}(R \cup K)$. For the remaining vertices, we repeat the process with new random samples, but this time we take two samples per vertex, decreasing the failure probability to $1/4$. We repeat the process, taking in each round the maximum number of samples that fit into the work space. In general, if we have s'/a_i active vertices in round i, we can take a_i samples per vertex, resulting in a failure probability of 2^{-a_i}. Thus, the expected number of active vertices in round $i+1$ is $s'/a_{i+1} = s'/(a_i 2^{a_i})$. After $O(\log^* s)$ rounds, all vertices are good. To summarize:

Lemma 7 *In total expected time $O(n \log s \log^* s)$ and space $O(s)$, we can find sets $R \subseteq S$ and $R_v \subset B_v$ for each vertex $v \in \mathrm{VD}(R' \cup K)$ such that (i) $|R| = s$: (ii) $\sum_{v \in \mathrm{VD}(R \cup K)} |R_v| = O(s)$; and (iii) for every R_v, all vertices of $\mathrm{VD}(R_v)$ have at most $2n/s$ points from B_v in their conflict circle.*

We set $R_2 = R \cup \bigcup_{v \in \mathrm{VD}(R \cup K)} R_v$. By Lemma 7, $|R_2| = O(s)$. We compute $\mathrm{VD}(R_2 \cup K)$ and triangulate its bounded cells. For a triangle Δ of the triangulation, let $r \in R_2 \cup K$ be the site whose cell contains Δ, and v_1, v_2, v_3 the vertices of Δ. We set $B_\Delta = \{r\} \cup \bigcup_{i=1}^{3} B_{v_i}$. One can show that $|B_\Delta| = O(n/s)$.

Lemma 8 *For every triangle Δ in the triangulation of $\mathrm{VD}(R_2 \cup K)$, we have $\mathrm{VD}(S \cup K) \cap \Delta = \mathrm{VD}(B_\Delta) \cap \Delta$.*

Proof. Let v_1, v_2, v_3 be the vertices of Δ and let $r \in R_2 \cup K$ be the point whose cell contains Δ. Fix a point $x \in \Delta$, and let C be a circle with center x and no points from B_Δ in its interior. It suffices to show that C contains no points from $S \setminus B_\Delta$ in its interior. Suppose there exists a point $p \in S \setminus B_\Delta$ that lies inside of C. Consider the bisector B of p and r. Since C contains p but not r, we have $d(x,p) < d(x,r)$, so x lies on the same side of B as p. Since $x \in \Delta$, at least one of v_1, v_2, v_3, is on the same side of B as p; say v_1. This means that $d(v_1,p) < d(v_1,r)$, so p lies inside the circle around v_1 with r on the boundary. This is precisely the conflict circle of v_1. □

Theorem 9 *Let S be a planar n-point set. In expected time $O((n^2/s) \log s + n \log s \log^* s)$ and space $O(s)$, we can compute all Voronoi vertices of S.*

Proof. We compute a set R_2 as above. This takes $O(n \log s \log^* s)$ time and space $O(s)$. We triangulate the bounded cells of $\mathrm{VD}(R_2 \cup K)$ and compute a point location structure for the result. Since there are $O(s)$ triangles, we can store the resulting triangulation in the workspace. Now, the goal is to compute simultaneously for all triangles Δ the Voronoi diagram $\mathrm{VD}(B_\Delta)$ and to output all Voronoi vertices that lie in Δ and are defined by points from S. By Lemma 8, this gives all Voronoi vertices of $\mathrm{VD}(S)$.

Given a planar m-point set X, the algorithm by Asano et al. finds all vertices of $\mathrm{VD}(X)$ in $O(m)$ scans over the input, with constant workspace [2]. We can perform a simultaneous scan for all sets B_Δ by determining for each point in S all sets B_Δ that contain it. This takes total time $O(n \log s)$, since we need one point location for each $p \in S$ and since the total size of the B_Δ's is $O(n)$. We need $O(\max_\Delta |B_\Delta|) = O(n/s)$ such sweeps, so the second part of the algorithm needs $O((n^2/s) \log s)$ time. □

Acknowledgments. This work began while W. Mulzer, P. Seiferth, and Y. Stein visited the Tokuyama Laboratory at Tohoku University. We would like to thank Takeshi Tokuyama and all members of the lab for their hospitality and for creating a conducive and stimulating research environment. W. Mulzer is partially supported by DFG grants MU 3501/1 and MU 3501/2. P. Seiferth is partially supported by DFG Grant MU 3501/1. Y. Stein is supported by the DFG within the research training group 'Methods for Discrete Structures' (GRK 1408).

References

[1] T. Asano, K. Buchin, M. Buchin, M. Korman, W. Mulzer, G. Rote, and A. Schulz. Memory-constrained algorithms for simple polygons. *Comput. Geom.*, 46(8):959–969, 2013.

[2] T. Asano, W. Mulzer, G. Rote, and Y. Wang. Constant-work-space algorithms for geometric problems. *J. of Comput. Geom.*, 2(1):46–68, 2011.

[3] L. Barba, M. Korman, S. Langerman, K. Sadakane, and R. Silveira. Space-time trade-offs for stack-based algorithms. *Algorithmica*, pages 1–33, 2014.

[4] L. Barba, M. Korman, S. Langerman, and R. I. Silveira. Computing the visibility polygon using few variables. *Comput. Geom.*, 47(9):918–926, 2013.

[5] A. Borodin and S. Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM J. Comput.*, 11:287–297, 1982.

[6] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3):229–249, 1990.

[7] J. I. Munro and M. Paterson. Selection and sorting with limited storage. *Theoret. Comput. Sci.*, 12:315–323, 1980.

[8] J. I. Munro and V. Raman. Selection from read-only memory and sorting with minimum data movement. *Theoret. Comput. Sci.*, 165(2):311–323, 1996.

[9] J. Pagter and T. Rauhe. Optimal time-space trade-offs for sorting. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 264–268, 1998.

[10] I. Pohl. A minimum storage algorithm for computing the median. Technical Report RC2701, IBM, 1969.

[11] J. E. Savage. *Models of computation—exploring the power of computing.* Addison-Wesley, 1998.