Optimal Randomized Algorithm for the Density Selection Problem*

Tien-Ching Lin and D.T. Lee**

Institute of Information Science, Academia Sinica, Taipei, Taiwan
{kero,dtlee}@iis.sinica.edu.tw
**Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan

Abstract. In the paper we consider a generalized version of three well-known problems: Selection Problem in computer science, Slope Selection Problem in computational geometry and Maximum-Density Segment Problem in bioinformatics. Given a sequence $A = (a_1, w_1)$, $(a_2, w_2), \ldots, (a_n, w_n)$ of n ordered pairs (a_i, w_i) of real numbers a_i and $w_i > 0$ for each $1 \leq i \leq n$, two nonnegative real numbers ℓ, u with $\ell \leq u$ and a positive integer k, the Density Selection Problem is to find the consecutive subsequence $A(i^*, j^*)$ over all $O(n^2)$ consecutive subsequences $A(i, j)$ satisfying width constraint $\ell \leq w(i, j) = \sum_{t=i}^{j} w_t \leq u$ such that the rank of its density $d(i^*, j^*) = \sum_{t=i^*}^{j^*} a_t / w(i^*, j^*)$ is k. We will give a randomized algorithm for density selection problem that runs in optimal expected $O(n \log n)$ time.

1 Introduction

Let $A = (a_1, w_1), (a_2, w_2), \ldots, (a_n, w_n)$ be a sequence of n ordered pairs (a_i, w_i) of real numbers a_i and width $w_i > 0$ for each $1 \leq i \leq n$. A *segment* $A(i, j)$ is a consecutive subsequence of A starting with index i and ending with index j. The *width* $w(i, j)$ of segment $A(i, j)$ is $\sum_{t=i}^{j} w_t$. The *density* $d(i, j)$ of segment $A(i, j)$ is $\sum_{t=i}^{j} a_t / w(i, j)$. Given a sequence $A = (a_1, w_1), (a_2, w_2), \ldots, (a_n, w_n)$ of n ordered pairs (a_i, w_i) of real numbers a_i and $w_i > 0$ for each $1 \leq i \leq n$, two nonnegative real numbers ℓ, u and a positive integer k, the Density Selection Problem (DSP) is to find the feasible segment $A(i^*, j^*)$ over all feasible segments such that the rank of its density $d(i^*, j^*)$ is k. We say that a segment $A(i, j)$ is *feasible* if its width satisfies $\ell \leq w(i, j) \leq u$. A sequence A is called *uniform width* if all w_i's are identical for each i, otherwise it is called *non-uniform width*.

The density selection problem for uniform width such that $\ell = 1$, $u = 1$ is the most well-known selection problem in computer science. Hoare [11] and

* Research supported in part by the National Science Council under the Grants No. NSC-94-2213-E-001-004, NSC-95-2221-E-001-016-MY3, and NSC 94-2752-E-002-005-PAE, and by the Taiwan Information Security Center (TWISC), National Science Council under the Grant No. NSC94-3114-P-001-001-Y.

Y. Dong, D.-Z. Du, and O. Ibarra (Eds.): ISAAC 2009, LNCS 5878, pp. 1004–1013, 2009.

Floyd and Rivest [9] gave an optimal expected $O(n)$ time randomized algorithm respectively. Blum, Floyd, Pratt, Rivest, and Tarjan [2] gave an optimal $O(n)$ time deterministic algorithm. The density selection problem such that k is equal to the total number of feasible segments is exactly the extensively studied maximum-density segment problem [4,10,12,14,15,18,20] which arises from the problem of finding the biologically meaningful region, called the most GC-ratio region, in a DNA sequence. When we let the input sequence $A = (a_1, w_1), (a_2, w_2), \ldots, (a_n, w_n)$ correspond to a given DNA sequence with uniform width such that $a_i = 1$ if the corresponding nucleotide in the DNA sequence is G or C, and $a_i = 0$ if the corresponding nucleotide in the DNA sequence is A or T. It is obvious that the output feasible segment then corresponds to the most GC-ratio region of the given DNA sequence. The density selection problem for fixed $\ell = 0, u = \infty$, also known as the slope selection problem [3,5,8,13,16], has received much attention in computational geometry. Cole et al. [5] first gave an optimal $O(n \log n)$ time deterministic algorithm for the slope selection problem by combining an approximate counting scheme, the AKS sorting network and parametric search technique. Brönnimann and Chazelle [3] modified their approximate counting scheme combining ε-net to obtain another optimal algorithm for this problem. Dillencourt et al. [8] and Matoušek [16] both gave an optimal randomized Monte Carlo algorithm respectively using the random sampling technique. Katz and Sharir [13] gave an optimal deterministic algorithm using expander graph and approximation technique. In this paper we will give an optimal randomized Monte Carlo algorithm for the density selection problem, using the random sampling technique [8,16], that runs in $O(n)$ space and optimal expected $O(n \log n)$ time. Therefore, it can solve the slope selection problem in optimal expected $O(n \log n)$ time as well.

On the other hand, it was observed that the compositional heterogeneity is highly correlated to the GC content of the genomic sequences [18,21]. The GC-ratios of the DNA sequences in all organisms vary from 25% to 75%. The typical GC-ratios of mammalian genomes stay in 45-50% and the GC-ratios of human DNA in 30-60%, but the GC-ratios have the greatest variations among bacteria's DNA sequences. Therefore, we are also interested in finding the range of the GC-ratios of a DNA sequence for a species. We will consider the Density Range Query Problem (DRQP) as follows. The input consists of a sequence A of n ordered pairs, two width bounds ℓ, u with $\ell \leq u$ and two real numbers d_l, d_r with $d_l \leq d_r$, the reporting mode of the DRQP is to report all feasible segments $A(i, j)$ satisfying $d_l \leq d(i, j) \leq d_r$ and the counting mode is to count the total number of feasible segments $A(i, j)$ satisfying $d_l \leq d(i, j) \leq d_r$. We will show that the reporting mode and counting mode can be solved in optimal $O(n \log m + h)$ and optimal $O(n \log m)$ time respectively, where $m = min\{\frac{u-\ell}{w_{min}}, n\}$ and h is the output size. Clearly, when $u = \ell$, both DSP and DRQP can easily be solved in $O(n)$ time and space. Therefore, from here on we assume $u > \ell$.

The rest of the paper is organized as follows. Section 2 solves the density range query problem. Section 3 gives an algorithm for the density selection problem. Section 4 gives some conclusion.

2 Algorithm for Density Range Query Problem

In this section we consider the density range query problem. Without loss of generality, we may assume $w_i \geq 1$ for each i and $w_{min} = 1$ for DRQP, since the problem for a sequence A of n ordered pairs (a_i, w_i) with respect to width bounds ℓ and u is equivalent to the problem for a sequence B of n ordered pairs $(\frac{a_i}{w_{min}}, \frac{w_i}{w_{min}})$ with respect to width bounds $\frac{\ell}{w_{min}}$ and $\frac{u}{w_{min}}$.

We first transform the DRQP into a geometric slope range query problem in $O(n)$ time as follows. We define the point set $P = \{p_0, p_1, \ldots, p_n\}$ in $\mathbf{R}^2$ according to the prefix sums of the sequence A, where $p_i = (x_i, y_i) = (\sum_{t=1}^{i} w_t, \sum_{t=1}^{i} a_t)$, $i = 1, 2, \ldots, n$ and $p_0 = (0, 0)$. It is easy to see that the slope $m(i, j)$ of the line segment $s(i, j)$ connecting p_i and p_j is equal to the density $d(i+1, j)$ of segment $A(i+1, j)$, so we can define a line segment $s(i, j)$ is feasible if its corresponding segment $A(i+1, j)$ is feasible.

> Given a point set $P = \{p_0, p_1, \ldots, p_n\}$ in $\mathbf{R}^2$, two width bounds ℓ, u and two density bounds d_l, d_r, find all feasible line segments $s(i, j)$ such that $d_l \leq m(i, j) \leq d_r$.

We can further transform this geometric slope range query problem into its dual problem, by transforming points into lines and vice versa. Consider the dual transform that maps the point $p_i = (x_i, y_i)$ into the dual line $l_i : y = x_i x - y_i$. For any two points p_i, p_j, their corresponding dual lines l_i, l_j will intersect at the point with abscissa $x_{ij} = (y_j - y_i)/(x_j - x_i) = m(i, j)$. It means that the abscissa of the intersection point of the two corresponding dual lines l_i, l_j is equal to the slope $m(i, j)$ of line segment $s(i, j)$. Again, we say that an intersection point of two dual lines l_i, l_j is feasible if $\ell \leq x_j - x_i \leq u$.

> Given a set of dual lines $L = \{l_0, l_1, \ldots, l_n\}$ in $\mathbf{R}^2$, where $l_i : y = x_i x - y_i$, two width bounds ℓ, u and two density bounds d_l, d_r, find all feasible intersection points $p_{ij} = (x_{ij}, y_{ij})$ such that their abscissae $x_{ij} \in [d_l, d_r]$.

Let $L_{a,b}$ denote the subset $\{l_a, l_{a+1}, \ldots, l_b\}$ of L starting with *left index a* and ending with *right index b*. For each dual line l_j we have a set of feasible dual lines $L_{c_j,d_j} = \{l_{c_j}, l_{c_j+1}, \ldots, l_{d_j}\}$, such that each $l_i \in L_{c_j,d_j}$ satisfies $\ell \leq x_j - x_i \leq u$. Without confusion we shall for simplicity denote L_{c_j,d_j} as L_j. Since the slope sequence $\{x_j\}_{j=1}^{n}$ of L is monotonically increasing, the left and right index sequences $\{c_j\}_{j=1}^{n}$ and $\{d_j\}_{j=1}^{n}$ are monotonically increasing respectively. Therefore, we can obtain sequences $\{c_j\}_{j=1}^{n}$ and $\{d_j\}_{j=1}^{n}$ by a linear scan of the sequence $\{x_j\}_{j=1}^{n}$. To solve the dual problem, it suffices to iterate on each j finding all feasible intersection points $p_{ij} = (x_{ij}, y_{ij})$ of L_j and l_j such that their abscissae $x_{ij} \in [d_l, d_r]$.

Instead of solving the dual problem directly we will further transform the dual problem into an orthogonal range query problem in computational geometry. For each dual line $l_i : y = x_i x - y_i$ in L, we let $q_i = (u_i, v_i) = (x_i d_l - y_i, x_i d_r - y_i)$ be the point with abscissa u_i defined by the intercept of l_i at $x = d_l$ and ordinate

v_i defined by the intercept of l_i at $x = d_r$. Let $Q = \{q_0, q_1, \ldots, q_n\}$ and $Q_j = \{q_i \in Q \mid \ell \leq x_j - x_i \leq u\}$. By the monotonically increasing property of the slope sequence $\{x_j\}_{j=0}^{n}$, we know that the slope of l_j is larger than the slope of l_i for each $l_i \in L_j$. Therefore, a dual line l_i in L_j will intersect l_j in $[d_l, d_r]$ if and only if $u_i \geq u_j$ and $v_i \leq v_j$. To solve the dual problem, it is now equivalent to making an orthogonal range query of the form $R_j = [u_j, \infty) \times (-\infty, v_j]$ to report all the points of Q_j which lie in R_j for each $j = 1, 2, \ldots, n$.

We first develop a reporting mode algorithm for the DRQP. Our reporting mode algorithm for the DRQP will iterate from $j = 1$ to n. At any iteration j, we will maintain a data structure $\zeta(Q_j)$ in the current window Q_j such that we can make an orthogonal range query of the form $R_j = [u_j, \infty) \times (-\infty, v_j]$, and then we delete points $q_{c_j}, q_{c_j+1}, \ldots, q_{c_{j+1}-1}$ from $\zeta(Q_j)$ and insert points $q_{d_j+1}, q_{d_j+2}, \ldots, q_{d_{j+1}}$ into $\zeta(Q_j)$ to obtain $\zeta(Q_{j+1})$. We will use a data structure called *priority search tree* to support the above orthogonal range query. A priority search tree [17] is a hybrid of a heap and a balanced binary search tree used for orthogonal range query where at least one of sides of the query range is unbounded. We will make the priority search tree $\zeta(Q_j)$ dynamic to support insertion and deletion operations as well. The priority search tree $\zeta(Q_j)$ can be constructed by using any balanced binary search tree and the performance of the priority search tree is summarized in the following lemma.

Lemma 1 ([7, Theorem 10.9, Page 221]). *The priority search tree $\zeta(S)$ for a set S of n points in $\mathbf{R}^2$ can be constructed in $O(n \log n)$ time and $O(n)$ space. Using the priority search tree we can report all points in a query range of the form $R = [u, w] \times (-\infty, v]$ in $O(\log n + h)$ time, where h is the number of reported points that lie in R.*

McCreight [17] shows that a balanced priority search tree can be made dynamic to support both insertion and deletion operations in $O(\log n)$ time if the number of rotations per updating operation can be bounded by a constant. Tarjan [22] shows that a class of balanced binary trees can be updated in $O(1)$ rotations. For example, a red-black tree belongs to the class. Therefore, if we use a red-black tree as our balanced binary search tree to implement dynamic priority search tree, then both insertion and deletion operations can be updated in $O(\log n)$ time. Since the reporting mode algorithm for the DRQP needs to do totally n times range queries, insertions and deletions on the window Q_j with $|Q_j| \leq m$, the overall running time is therefore $O(n \log m + h)$ by Lemma 1, where $m = min\{u - \ell, n\}$ and h is the output size. We can also develop a counting mode algorithm by using the order-statistics tree data structure similarly. Due to page limitation, we omit it here. Thus, we obtain the following theorem.

Theorem 1. *The reporting and counting mode of the density range query problem can be solved in $O(n)$ space and optimal $O(n \log m + h)$ time and optimal $O(n \log m)$ time respectively, where $m = min\{\frac{u-\ell}{w_{min}}, n\}$ and h is the output size.*

Now, we show that both reporting and counting algorithms of the DRQP are optimal in the worst case. It is known that the Element Uniqueness Problem, i.e., to determine if a set of n real numbers $y_1, y_2, ..., y_n$ are all distinct,

has a lower bound of $\Omega(n \log n)$ time in the algebraic decision tree model of computation [1]. We can transform an instance of element uniqueness problem to an instance of the DRQP with $\ell = 0$, $u = \infty$, $d_l = d_r = 0$ and $w_i = 1$ for each i in $O(n)$ time by letting $a_1 = y_1$, $a_i = y_i - y_{i-1}$ for $i = 2, \ldots, n$. The output of the reporting mode of the DRQP is an empty set (or The output of the counting mode of the DRQP is 0) if and only if $y_1, y_2, ..., y_n$ are all distinct. Therefore, both the reporting and counting mode of the DRQP has a lower bound of $\Omega(n \log n)$ time in the algebraic decision tree model of computation.

3 Algorithm for Density Selection Problem

In this section we give an optimal randomized Monte Carlo algorithm for the DSP based on three subroutines, random sampling subroutine, reporting mode and counting mode algorithms for the DRQP. The DSP is equivalent to the following problem.

> Given a set of lines $L = \{l_0, l_1, \ldots, l_n\}$ in $\mathbf{R}^2$, where $l_i : y = x_i x - y_i$, find the feasible intersection point $p_{i^*j^*} = (x_{i^*j^*}, y_{i^*j^*})$ such that its abscissa $x_{i^*j^*}$ is the k-th smallest among all feasible intersection points.

For convenience we shall without confusion use the intersection point p_{ij} and its abscissa x_{ij} interchangeably. We first develop a random sampling subroutine running in expected $O(n \log n)$ time to randomly generate $\frac{nN_f}{2N}$ to $\frac{3nN_f}{2N}$ feasible intersection points allowing duplicates such that they all lie in a given interval $[d_l, d_r]$, where N and N_f are the total numbers of intersection points and feasible intersection points in $[d_l, d_r]$ respectively. Note that N and N_f can be obtained by the counting algorithm for the DRQP. Dillencourt et al. [8] developed a random sampling subroutine running in $O(n \log n)$ time by merge sort technique to randomly generate n intersection points such that they all lie in a given interval $[d_l, d_r]$. We summarize it in the following lemma.

Lemma 2 (Dillencourt et al. [8]). *Let $L = \{l_0, l_1, \ldots, l_n\}$ be a set of lines in $\mathbf{R}^2$ and $[d_l, d_r]$ be a given interval. We can obtain a random sampling S by randomly generateing n intersection points of L allowing duplicates in $O(n \log n)$ time such that all points of S are in $[d_l, d_r]$.*

We carefully analyze their random sampling subroutine and find that it can be used to randomly generate $\frac{nN_f}{2N}$ to $\frac{3nN_f}{2N}$ feasible intersection points allowing duplicates such that they all lie in a given interval $[d_l, d_r]$ with high probability by using the well-known Chebyshev's inequality in probability theory.

Whenever we select a random intersection point in $[d_l, d_r]$, it has a probability $\frac{N_f}{N}$ such that it is feasible. Consider such an event as a "success" in performing n independent Bernoulli trials, each with a probability $\frac{N_f}{N}$. Let X_i be the random variable, attaining value 1 with probability $p_x = \frac{N_f}{N}$ and value 0 if otherwise. Let $X = X_1 + X_2 + \cdots + X_n$ be the total number of feasible intersection points for a random sampling S obtained by Lemma 2. The expected value of X is

$\mu = np_x = \frac{nN_f}{N}$ and the standard deviation of X is $\sigma = \sqrt{np_x(1-p_x)} = \sqrt{\frac{nN_f}{N}(1-\frac{N_f}{N})} \leq \sqrt{\frac{nN_f}{N}}$. By Chebyshev's inequality, for any $\lambda \geq 0$ we have $Pr[|X-\mu| \geq \lambda\sigma] \leq \frac{1}{\lambda^2}$, so the probability $Pr[\mu + \lambda\sigma \geq X \geq \mu - \lambda\sigma] \geq 1 - \frac{1}{\lambda^2}$. Therefore, if we choose $\frac{nN_f}{2N} \geq 2\lambda^2 = 4$, we have $Pr[\frac{3nN_f}{2N} \geq \frac{nN_f}{N} + \lambda\sqrt{\frac{nN_f}{N}} \geq \mu + \lambda\sigma \geq X \geq \mu - \lambda\sigma \geq \frac{nN_f}{N} - \lambda\sqrt{\frac{nN_f}{N}} \geq \frac{nN_f}{2N}] \geq \frac{1}{2}$. Hence, if $N_f \geq \frac{8N}{n}$, we can obtain a random sampling S of n intersection points in $[d_l, d_r]$ such that it contains $\frac{nN_f}{2N}$ to $\frac{3nN_f}{2N}$ feasible intersection points with probability no less than $\frac{1}{2}$. Otherwise, if $N_f < \frac{8N}{n}$ we can solve the density selection problem directly by using reporting algorithm for DRQP to enumerate N_f feasible intersection points in $O(n \log m + N_f) = O(n \log m + \frac{8N}{n}) = O(n \log m + n) = O(n \log n)$ time and then select the k-th smallest feasible intersection point d^* from those feasible intersection points by using any standard selection algorithm in $O(n)$ time. Thus, we can assume $N_f \geq \frac{8N}{n}$ from now on. Therefore, we have the following random sampling subroutine.

Lemma 3. *Let $L = \{l_0, l_1, \ldots, l_n\}$ be a set of lines in $\mathbf{R}^2$. Let N and N_f (assuming $N_f \geq \frac{8N}{n}$) be the total numbers of intersection points and feasible intersection points of L in a given interval $[d_l, d_r]$ respectively. We can randomly generate in expected $O(n \log n)$ time a set of n intersection points allowing duplicates in $[d_l, d_r]$ such that they contain M to $3M$ feasible points, where $M = \frac{nN_f}{2N}$.*

We now start to solve the DSP. We shall consider a more general problem, called Density Selection Range Query Problem (DSRQP) defined as follows. Given an interval $[d_l, d_r]$ which contains $N = N_f + N_i$ intersection points where N_f and N_i are the total number of feasible and infeasible intersection points in $[d_l, d_r]$ respectively, we would like to find the k-th smallest feasible intersection point among the N_f feasible intersection points in the interval $[d_l, d_r]$. Let d^* denote the k-th smallest feasible intersection point in the interval $[d_l, d_r]$. Note that the DSP is just a special case of this problem such that $N = \frac{n(n-1)}{2}$, $N_f = O((u-\ell)n)$ and $[d_\ell, d_r] = (-\infty, \infty)$.

The randomized algorithm for the DSRQP will contract the interval $[d_l, d_r]$ into a smaller subinterval $[d_{l'}, d_{r'}]$ such that it also contains d^* and the new subinterval $[d_{l'}, d_{r'}]$ contains at most $O(N_f/\sqrt{M})$ feasible intersection points. It will repeat to contract the interval several times until the interval $[d_{l'}, d_{r'}]$ contains not only d^* but also at most $O(n)$ feasible intersection points. It then outputs all the feasible intersection points in $[d_{l'}, d_{r'}]$ by the reporting mode algorithm for the DRQP and finds the feasible intersection point d^* with an appropriate rank by using any standard selection algorithm.

Our randomized algorithm for the DSRQP runs as follows: We first use our random sampling subroutine to randomly generate a set of feasible intersection points $S' = \{s_1, s_2, \ldots, s_F\}$ in $[d_l, d_r]$. If F is smaller than M or greater than $3M$ we repeat our random sampling subroutine again. From Lemma 3 the probability that the set of n randomly generated intersection points contains M to $3M$ feasible intersection points is no less than $1/2$, so we would perform the random

sampling subroutine at most twice on average. Assume that we have obtained a random sampling S' which contains M to $3M$ feasible points. We then try to use this random sampling $S^{'}$ to obtain a smaller subinterval $[d_{l'}, d_{r'}]$ as follows. For each of the selected random feasible intersection point in $S^{'}$, it has a probability $\frac{k}{N_f}$ such that it is smaller than or equal to d^*. Consider such an event as a "success" in performing F independent Bernoulli trials, each with a probability $\frac{k}{N_f}$. Let X_i be the random variable, attaining value 1 with probability $p_x = \frac{k}{N_f}$ and value 0 with probability $p_x = 1 - \frac{k}{N_f}$. Let $X = X_1 + X_2 + \cdots + X_F$ be the total number of sample feasible intersection points falling before d^*. The expected value of X is $\mu_x = Fp_x = \frac{Fk}{N_f}$ and the standard deviation of X is $\sigma_x = \sqrt{Fp_x(1-p_x)}$. It means that the average number of feasible intersection points in $S^{'}$ which is smaller than or equal to d^* is $\frac{Fk}{N_f}$. Hence we expect that the w-th smallest element in $S^{'}$, where $w = \lfloor Fp_x \rfloor = \lfloor \frac{Fk}{N_f} \rfloor$ should be a good approximation for the k-th smallest feasible intersection point d^*. Let $l^{'} = \max\{1, \lfloor \frac{Fk}{N_f} - t\frac{\sqrt{F}}{2} \rfloor\}$ and $r^{'} = \min\{F, \lceil \frac{Fk}{N_f} + t\frac{\sqrt{F}}{2} \rceil\}$, for some constant t to be determined later. Therefore, after we get a successful random sampling $S^{'}$, we can find the $l^{'}$-th smallest element $d_{\ell'}$ and the $r^{'}$-th smallest element $d_{r'}$ in $S^{'}$ by any standard selection algorithm in $O(|S^{'}|)$ time to obtain a subinterval $[d_{\ell'}, d_{r'}]$. The key step of our randomized algorithm for the DSRQP is to check whether the subinterval $[d_{\ell'}, d_{r'}]$ satisfies the following two conditions by the counting algorithm for the DRQP:

(1) The density d^* of the k-th smallest feasible intersection point lies in the subinterval $[d_{\ell'}, d_{r'}]$.
(2) The subinterval $[d_{\ell'}, d_{r'}]$ contains at most $\frac{t^2 N_f}{(t-1)\sqrt{M}}$ $(< \frac{2tN_f}{\sqrt{M}})$ feasible intersection points and contains at most $\frac{3t^2 N}{2(t-1)\sqrt{M}}$ intersection points.

If either (1) or (2) is violated, we repeat our randomized algorithm for the DSRQP from scratch again until both (1) and (2) are satisfied: i.e. we need to randomly select F, where $M \leq F \leq 3M$, feasible intersection points with replacement in the interval $[d_\ell, d_r]$ by running the random sampling algorithm again to obtain a new subinterval $[d_{\ell'}, d_{r'}]$ and then check the above two conditions (1) and (2) for the new subinterval $[d_{\ell'}, d_{r'}]$. Let k_1 and k_2 be the total number of feasible intersection points lying in $[d_\ell, d_{\ell'})$ and $[d_\ell, d_{r'}]$ respectively. Note that d^* lies in the subinterval $[d_{\ell'}, d_{r'}]$ if and only if $k_1 < k$ and $k_2 \geq k$. If both of these conditions hold, we replace the current interval $[d_\ell, d_r]$ by the subinterval $[d_{\ell'}, d_{r'}]$ and let $k^{'} = k - k_1$.

Note that the density selection algorithm starts with $N = \frac{n(n-1)}{2}$ intersection points and $N_f = O((u-\ell)n)$ feasible intersection points in the initial interval $[d_\ell, d_r] = (-\infty, \infty)$. Therefore, after the first successful random sampling which satisfies conditions (1) and (2) we have an interval $[d_{\ell'}, d_{r'}]$ which contains the $k^{'}$-th smallest feasible intersection point d^* and it contains $O(\frac{N_f}{\sqrt{M}})$ feasible intersection points and $O(\frac{N}{\sqrt{M}})$ intersection points. That is in each iteration we try to

prune the numbers of intersection points and feasible intersection points roughly by a factor of $O(\sqrt{M}) = O(\sqrt{\frac{nN_f}{N}})$ respectively, so after one successful random sampling we still maintain the ratio of the number of feasible intersection points and the number of intersection points in $[d_{\ell'}, d_{r'}]$ to be $O(\frac{N_f}{N})$. Therefore, we can repeat the same procedure for the next iteration. After the second successful random sampling which satisfies conditions (1) and (2) we have an interval $[d_{\ell''}, d_{r''}]$ which contains the k''-th smallest feasible intersection point d^* and which has $O(\frac{N_f}{\sqrt{M}}/\sqrt{M}) = O(\frac{N_f}{M}) = O(\frac{N}{n}) = O(n)$ feasible intersection points. We can then enumerate all feasible intersection points from this interval $[d_{\ell''}, d_{r''}]$ by the reporting mode algorithm for the DRQP and select the k''-th smallest feasible intersection point d^* from those feasible intersection points by using any standard selection algorithm. We now show that with a high probability the key step of our randomized algorithm for the DSRQP is satisfied.

Lemma 4. *Let N and N_f be the total numbers of intersection points and feasible intersection points in $[d_l, d_r]$ respectively. For a random choice of F independent feasible intersection points with replacement in the interval $[d_\ell, d_r]$ where $M \leq F \leq 3M$, we can find a subinterval $[d_{\ell'}, d_{r'}]$ containing at most $t\sqrt{F}$ sample feasible intersection points such that the probability that it contains at least $\frac{t^2 N_f}{(t-1)\sqrt{M}}$ feasible intersection points is at most $e^{-\sqrt{M}/2(t-1)}$ and the probability that it contains at least $\frac{3t^2 N}{2(t-1)\sqrt{M}}$ intersection points is at most $e^{-\sqrt{M}/2(t-1)}$.*

Proof. To show the subinterval $[d_{\ell'}, d_{r'}]$ contains at most $\frac{t^2 N_f}{(t-1)\sqrt{M}}$ feasible intersection points with high probability $1 - e^{-\sqrt{M}/2(t-1)}$, we just need to show $[d_{\ell'}, d_{r'}]$ contains at most $\frac{t^2 N_f}{(t-1)\sqrt{F}}$ ($\leq \frac{t^2 N_f}{(t-1)\sqrt{M}}$) feasible intersection points with high probability $1 - e^{-\sqrt{M}/2(t-1)}$. Assume that a successful random sampling $S' = \{s_1, s_2, \ldots, s_F\}$ with replacement in $[d_\ell, d_r]$ in the random sampling subroutine for the DSRQP gives a subinterval $[d_{\ell'}, d_{r'}]$ containing at most $t\sqrt{F}$ sample feasible intersection points. Let N' and N'_f be the total numbers of intersection points and feasible intersection points in $[d_{\ell'}, d_{r'}]$ respectively. Assume that $N'_f \geq \frac{t^2 N_f}{(t-1)\sqrt{F}}$. Hence, whenever we select a random feasible intersection point s_i in $[d_\ell, d_r]$, it has probability larger than $\frac{t^2 N_f/(t-1)\sqrt{F}}{N_f} = \frac{t^2}{(t-1)\sqrt{F}}$ such that s_i lies in $[d_{\ell'}, d_{r'}]$. We again think such an event as a "success", each with a probability of success equal to $p \geq \frac{t^2}{(t-1)\sqrt{F}}$. Let X_i be the random variable, attaining value 1 with probability $p \geq \frac{t^2}{(t-1)\sqrt{F}}$ if the i-th selected feasible intersection point falls in $[d_{\ell'}, d_{r'}]$ and value 0 with probability $1 - p$ if otherwise. Let $X = X_1 + X_2 + \cdots + X_F$ be the total number of selected feasible intersection points falling in $[d_{\ell'}, d_{r'}]$. The expectation of the random experiment is $\mu = Fp \geq \frac{t^2 F}{(t-1)\sqrt{F}} = \frac{t^2\sqrt{F}}{t-1}$. By the Chernoff bound, we have $Pr[X \leq t\sqrt{F}] \leq Pr[X \leq (1 - \frac{1}{t})\mu] \leq e^{-\mu/2t^2} \leq e^{-\sqrt{F}/2(t-1)} \leq e^{-\sqrt{M}/2(t-1)}$. Therefore, we have

the joint probability $Pr[(N_f^{'} \geq \frac{t^2 N_f}{(t-1)\sqrt{F}}) \cap (X \leq t\sqrt{F})] \leq e^{-\sqrt{M}/2(t-1)}$. On the other hand, note that $d_{\ell'}$ and $d_{r'}$ are the $l^{'}$-th and $r^{'}$-th smallest elements in the random sampling $S^{'}$ respectively. It means that the random sampling $S^{'}$ contains exactly $r^{'} - \ell^{'}$ ($\leq t\sqrt{F}$) sample feasible intersection points lying in $[d_{\ell'}, d_{r'}]$. Therefore, we have $Pr[(N_f^{'} \geq \frac{t^2 N_f}{(t-1)\sqrt{F}}) \cap (S^{'}$ contains exactly $r^{'} - \ell^{'}$ sample feasible intersection points in $[d_{\ell'}, d_{r'}])] \leq Pr[(N_f^{'} \geq \frac{t^2 N_f}{(t-1)\sqrt{F}}) \cap (X \leq t\sqrt{F})] \leq e^{-\sqrt{M}/2(t-1)}$. The first part of the lemma follows. Due to page limitation we omit the proof of the second part here.

Lemma 5. *Let N and N_f be the total numbers of intersection points and feasible intersection points in $[d_l, d_r]$ respectively. For a random choice of F independent feasible intersection points with replacement in the interval $[d_\ell, d_r]$ where $M \leq F \leq 3M$, we can find a subinterval $[d_{\ell'}, d_{r'}]$ containing at most $t\sqrt{F}$ sample feasible intersection points such that the probability that the k-th smallest feasible intersection point d^* not lying in the subinterval $[d_{\ell'}, d_{r'}]$ is at most $2e^{-t^2/2}$.*

Proof. Let Y_i be the random variable, attaining value 1 with probability $p = \frac{k}{N_f}$ if the i-th sample feasible intersection point is no greater than d^* and value 0 with probability $1-p$ if otherwise. If the $r^{'}$-th smallest feasible intersection point $d_{r'}$ in S' is smaller than d^*, it means that at least $r^{'}$ among the F randomly sample feasible intersection points fall before d^*. Let $Y = Y_1 + Y_2 + \cdots + Y_F$ be the total number of sample feasible intersection points falling before d^*. By the Chernoff bound, we have $Pr[Y \geq r^{'}] = Pr[Y \geq \mu + t\frac{\sqrt{F}}{2}] \leq e^{-t^2/2}$. Similarly, by the Chernoff bound we have $Pr[Y \leq l^{'}] = Pr[Y \leq \mu - t\frac{\sqrt{F}}{2}] \leq e^{-t^2/2}$.

Now, we can choose t large enough such that $2e^{-t^2/2} \leq \frac{1}{4}$ and choose M large enough such that $2e^{-\sqrt{M}/2(t-1)} \leq \frac{1}{4}$, i.e. choose $N_f \geq \frac{2cN}{n}$ for some large enough constant c such that $e^{-\sqrt{M}/2(t-1)} \leq e^{-\sqrt{c}/2(t-1)} \leq \frac{1}{8}$. For example, we can choose $t = 2.1$ and $c = 21$ respectively. Therefore, we just need to repeat the key step at most twice on the average in the randomized algorithm for the DSP, otherwise we can solve the DSP directly by using reporting algorithm for DRQP and any standard selection algorithm.

Theorem 2. *The* Density Selection Problem *can be solved in $O(n)$ space and expected $O(n \log n)$ time.*

4 Conclusion

In the paper we considered an interesting density selection problem. It is a generalization of three well known problems, the maximum density segment problem, slope selection problem and selection problem. We have presented a randomized algorithm for this problem running in expected $O(n \log n)$ time. But whether the density selection problem can be solved by a deterministic algorithm within the same time bound remains to be seen.

References

1. Ben-Or, M.: Lower bounds for algebraic computation trees. In: Proc. 15th Annu. ACM Sympos. Theory Comput., pp. 80–86 (1983)
2. Blum, M., Floyd, R.W., Pratt, V., Rivest, R.L., Tarjan, R.E.: Time bound for selection. Journal of Computer and System Sciences 7(4), 448–461 (1973)
3. Brönnimann, H., Chazelle, B.: Optimal slope selection via cuttings. Computational Geometry — Theory and Applications 10(1), 23–29 (1998)
4. Chung, K.-M., Lu, H.-I.: An optimal algorithm for the maximum-density segment problem. SIAM Journal on Computing 34(2), 373–387 (2004)
5. Cole, R., Salowe, J.S., Steiger, W.L., Szemeredi, E.: An optimal-time algorithm for slope selection. SIAM Journal on Computing 18(4), 792–810 (1989)
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introdution to algorithms. MIT Press, Cambridge (1998)
7. De Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, Berlin (1997)
8. Dillencourt, M.H., Mount, M.H., Netanyahu, N.S.: A randomized algorithm for slope selection. International Journal of Computational Geometry and Applications 2(1), 1–27 (1992)
9. Robert, W.F., Ronald, L.R.: Expected time bounds for selection. Communications of the ACM 18(3), 165–172 (1975)
10. Goldwasser, M.H., Kao, M.-Y., Lu, H.-I.: Linear-time algorithms for computing maximum-density sequence segments with bioinformatics applications. Journal of Computer and System Sciences 70(2), 128–144 (2005)
11. Hoare, C.A.R.: Algorithm 63 (partition) and algorithm 65 (find). Communications of the ACM 4(7), 321–322 (1961)
12. Huang, X.: An algorithm for identifying regions of a DNA sequence that satisfy a content requirement. Comp. Applications in the Biosciences 10(3), 219–225 (1994)
13. Katz, M.J., Sharir, M.: Optimal slope selection via expanders. Information Processing Letters 47(3), 115–122 (1993)
14. Kim, S.K.: Linear-time algorithm for finding a maximum-density segment of a sequence. Information Processing Letters 86(6), 339–342 (2003)
15. Lin, Y.-L., Huang, X., Jiang, T., Chao, K.-M.: Locating non-overlapping maximum average segments in a given sequence. Bioinformatics 19(1), 151–152 (2003)
16. Matoušek, J.: Randomized optimal algorithm for slope selection. Information Processing Letters 39(4), 183–187 (1991)
17. McCreight, E.M.: Priority search trees. SICOMP 14(2), 257–276 (1985)
18. Nekrutenko, A., Li, W.-H.: Assessment of compositional heterogeneity within and between eukaryotic genomes. Genome Research 10, 1986–1995 (2000)
19. Ohler, U., Niemann, H., Liao, G., Rubin, G.M.: Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. Bioinformatics 17, 199–206 (2001)
20. Rice, P., Longden, I., Bleasby, A.: Emboss: The European molecular biology open software suite. Trends Genet. 16, 276–277 (2000)
21. Stojanovic, N., Florea, L., Riemer, C., Gumucio, D., Slightom, J., Goodman, M., Miller, W., Hardison, R.: Comparison of five methods for finding conserved sequences in multiple alignments of gene regulatory regions. Nucleic Acids Research 27, 3899–3910 (1999)
22. Tarjan, R.E.: Updating a balanced search tree in O(1) rotations. Information Processing Letters 16, 253–257 (1983)