Point counting Algorithms

J. Mulik and J. Pathak

February 28, 2005

1 Preliminaries

Suppose

$$V = \{(x_1, x_2 \ldots, x_u) \mid 1 \leq x_i \leq n, x_i \in \mathbb{N}, i = 1 \ldots u\}$$

Define $X_i(I) = i^{th}$ coordinate of I for $I \in V$. A **cost function** $C : V \to \mathbb{R}_+$ is defined as $C((x_1, x_2 \ldots, x_u)) = \sum_{i=1}^{u} C_{x_i}$. for some constants $C_i \in \mathbb{R}_+$ with $C_i \leq C_j$ for $i \leq j$.

Suppos T is any positive constant.

Definition 1.1 *Suppose I, $J \in V$.* **Distance** *d of I, J is $d(I, J) = \sum_{j=1}^{u} |X_j(I) - X_j(J)|$. I, J are called* **neighbors** *if $I \neq J$ and $|X_k(i) - X_k(j)| \leq 1$ for all k, $1 \leq k \leq u$. We say that J is a* **smaller neighbor** *of I if $X_k(J) = X_k(I) - 1$ for exactly one value of k. $(1, 1, \ldots 1)$ does not have a smaller neighbor. A point $P \in V$ is called an* **anchor point** *if $C(P) \geq T$ and if P has a smaller neighbors, then there exist a smaller neighbors P_0 with $C(P_0) < T$. We will say that P_0* **defines** *P. The set of anchor points is denoted by $\mathfrak{A}$. A subset X of V is called* **connected** *if for any $I, I' \in X$, $\exists I_1, I_2, \ldots, I_r \in X$ such that I is a neighbour of I_1, I_r is a neighbour of I' and I_j is a neighbour of I_{j+1} for $1 \leq j < r$. We also say that $\{I_1, \ldots, I_r\}$ is a connected path from I to I'.*

Let $\sigma \in S_u$ be a permutation. We define $\sigma((x_1, \ldots, x_u)) = (x_{\sigma(1)}, \ldots, x_{\sigma(u)})$. We record some elementary properties in the following lemma.

Lemma 1.2 *Suppose $\sigma \in S_u$ and I, J, P, $Q \in V$. Then yhe following hold.*

1. *$C(\sigma(I)) = C(I)$ for all $I \in V$.*
2. *If Q is a neighbor of P then $\sigma(Q)$ is a neighbor of $\sigma(P)$.*
3. *If $P \in \mathfrak{A}$, then $\sigma(P) \in \mathfrak{A}$. In other words, $\mathfrak{A}$ is stable under S_u.*
4. *$d(P, Q) = d(\sigma(P), \sigma(Q))$.*
5. *If $S \subset V$ is connected, then $\sigma(S) = \{\sigma(I) | I \in S\}$ is also connected.*

Proof If $P \in \mathfrak{A}$ and P_0 is a defining small neighbor of P then $\sigma(P_0)$ is a defining small neighbor of $\sigma(P)$. ∎

Theorem 1.3 *If $P \in \mathfrak{A}$ and $P \notin \{(1,\dots,1),(n,\dots,n)\}$. Then at least one neighbor of P is in $\mathfrak{A}$.*

Proof Suppose P_0 defines P. By Lemma **??**, we can assume that $X_1(P_0) = X_1(P)-1$ and $X_i(P_0) = X_i(P)$ for $i > 1$. Set $Q = (X_1(P_0), X_2(P_0)+1, X_3(P_0), \dots, X_u(P_0))$. Clearly Q is a neighbor of P. If $C(Q) \geq T$, then Q is an anchor point as P_0 is the defining smaller neighbor of Q. On the other hand, if $C(Q) < T$, then $Q' = (X_1(P_0)+1, X_2(P_0)+1, X_3(P_0), \dots, X_u(P_0)) \in \mathfrak{A}$ as Q is the defining smaller neighbor of Q'. Note that Q and Q' both are neighbors of P. This completes the proof. ∎

Lemma 1.4 *Suppose $P,\ P' \in \mathfrak{A}$ are such that $X_j(P') = X_j(P) + r_j\ \forall j,\ 1 \leq j \leq u$ for some nonnegative $r_j \in \mathbb{N}$ and suppose Q is in V with $X_j(P) \leq X_j(Q) \leq X_j(P')$, Then at least one $r_j = 0$ and $Q \in \mathfrak{A}$.*

Proof Suppose P_0' defines P'. Then for some i ($1 \leq i \leq u$), $X_i(P_0') = X_i(P') - 1$. Clearly, $r_i = 0$. Define P_1 as $X_j(P_1) = X_j(P_0')$ and $X_i(P_1) = X_i(P_0') + 1$. It is now easy to show that P_0' defines P_1. This completes the proof.

Corollary 1.5 *Suppose $I,\ J \in V$ with $X_k(I) > X_k(J)\ \forall k,\ 1 \leq k \leq u$, If $I \in \mathfrak{A}$ then $J \notin \mathfrak{A}$.*

Proof We set $r_j = X_j(I) - X_j(J)$ and apply Lemma 1.4 ∎

Lemma 1.6 *Suppose $I,\ J \in V$ are neighboring points such that $C(I) < T$ and $C(J) \geq T$. Then either $J \in \mathfrak{A}$ or there is $P \in \mathfrak{A}$ which is a neighbor of I and J.*

Proof If $d(I,J) = 1$, then I defines J and so $J \in \mathfrak{A}$. Since $C(I) < C(J)$, we can find the smallest index i such that $X_i(I) < X_i(J)$. Define I_1 as follows: $X_j(I_1) = X_j(I)$ for $j \neq i$ and $X_i(I_1) = X_i(I) + 1 = X_i(J)$. Note that $d(I_1, J) < d(I, J)$. If $C(I_1) \geq T$, then $I_1 \in \mathfrak{A}$ and the proof is complete. If $C(I_1) < T$, then we define I_2 taking I_1 for I in the above construction. Note that still I_2 is a neighbor of I and J. If I_2 is not in $\mathfrak{A}$, we define I_3 and so on. Since $d(I_i, J) < d(I_{i-1}, J)$, this process will stop. ∎

Corollary 1.7 *Suppose X is any connected set that contains points $I,\ J$ such that $C(I) < T$ and $C(J) \geq T$, then X contains an anchor point or X contains a neighbour of an anchor point.*

Proof Let $I_1, I_2, \dots, I_r$ be a connected path from I to J. Since $C(I) < T$ and $C(J) \geq T$, there exists i such that $C(I_i) < T$ and $C(I_{i+1}) \geq T$, where $I_{r+1} \equiv I'$. Now we apply Lemma 1.6. ∎

Theorem 1.8 *The set $\mathfrak{A}$ of anchor points is connected.*

Proof Suppose P and Q are in $\mathfrak{A}$ with $d(P,Q) = d$. We use induction on d. $d = 1$ is clear. Assume that $d > 1$. Suppose P_0 defines P. We can assume that $X_1(P_0) = X_1(P) - 1$.

Case I: $X_1(P) < X_1(Q)$

By Lemma 1.5 $\exists k$ such that $X_k(P) > X_k(Q)$. Set $P' = (X_1(P), \ldots, X_k(P) - 1, \ldots, X_u(P)$. Clearly $d(P', Q) < d$. If $C(P') \geq T$, then $(X_1(P) - 1, \ldots, X_k(P) - 1, \ldots, X_u(P))$ defines P' and the proof is complete. On the other hand, if $C(P') < T$ take $P'' = (X_1(P) + 1, \ldots, X_k(P) - 1, \ldots, X_u(P))$, then P' is a smaller neighbor of P''. If $C(P'') \geq T$, then $P'' \in \mathfrak{A}$. If $C(P'') < T$, then $P_1 = (X_1(P) + 1, \ldots, X_k(P), \ldots, X_u(P)) \in \mathfrak{A}$ and $d(P_1, Q) < d$.

Case I: $X_1(P) > X_1(Q)$

By Lemma 1.5 there exists a k such that $X_k(P) < X_k(Q)$. Set $P' = (X_1(P), \ldots, X_k(P) + 1, \ldots, X_u(P))$. Clearly $C(P') \geq T$ and $d(P', Q) < d$. Let $P'' = (X_1(P) - 1, \ldots, X_k(P) + 1, \ldots, X_u(P))$, then $d(P'', Q) < d$.

If $C(P'') < T$, then $P' \in \mathfrak{A}$. If $C(P'') \geq T$, then $P'' \in A$.

By induction, we can find a connected path from I to Q. ∎

Let $X = \{(i, i, \ldots, i) \in V |, 1 \leq i \leq n\}$. Then X is a connected set. In applications we generally expect $C(1, 1, \ldots, 1) < T < C(n, n, \ldots, n)$. Therefore by applying corollary 1.7, we can find an anchor point which is a neighbor to $(i, i, \ldots, i)$ for some i. Now using connectedness of $\mathfrak{A}$ and an exhaustive search, we can find all the anchor points. We describe this in the following algorithm.

Once all $anchorpoint$ anchor points are found is found we can algorithmically find all the points in V with cost greater or equal to T. First we prove the following theorem

Lemma 1.9 *(a) Suppose* $(a_1, a_2, \ldots, a_u) \in \mathfrak{A}$ *is such that*

$$a_k = max\{b_k \mid (a_1, \ldots, a_{k-1}, b_k, b_{k+1}, \ldots, b_u) \in A\}$$

If $(a_1, \ldots, a_{k-1}, a'_k, \ldots, a'_u) \in V$ *and if* $a'_k > a_k$ *then* $C((a_1, \ldots, a_{k-1}, a'_k, \ldots, a'_u)) \geq T$.

(b) Suppose $(b_1, \ldots, b_u)$ *is a point in* V *which is not an anchor point such that* $C((b_1, \ldots, b_u)) \geq T$. *Then* $\exists (a_1, \ldots, a_u) \in \mathfrak{A}$ *such that,*

1. $a_i = b_1$ *for* $i = 1, \ldots, k-1$, $a_k < b_k$ *and,*
2. $a_k = max\{c_k \mid (a_1, \ldots, a_{k-1}, c_k, \ldots, c_u) \in A\}$

Proof (a) Suppose $C(a_1 \ldots, a_{k-1}, a'_k, \ldots, a'_u) < T$. Since $a'_k > a_k$, not all $a'_i \geq a_i$, $i = k+1, \ldots, u$. Assume that $a'_{l_i} < a_{l_i}$ for $i = 1, \ldots, r$, $k+1 \leq l_i \leq u$. Now by adding s to a'_{l_i} and checking the cost of a point $(a_1, \ldots, a_{k-1}, a'_k, \ldots, a'_{l_1} + s, a'_{l_2}, \ldots, a'_u)$ we can eventually find a point $v = (a_1, \ldots, a_{k-1}, a'_k, \ldots, a'_{l_j}, \ldots, a'_u)$, such that $C(v) < T$ but $C((a_1, \ldots, a_{k-1}, a'_k, \ldots, a'_{l_j} + 1, \ldots, a'_u) \geq T$. But then $(a_1, \ldots, a_{k-1}, a'_k, \ldots, a'_{l_j} + 1, \ldots, a'_u) \in A$ with $a'_k > a_k$, which contradicts the maximality of a_k.[9]

(b) We choose $(a_1, \ldots, a_u) \in A$, with the properties,

- $a_i = b_i$ for $i = 1, \ldots, k-1$,

- k is the largest value such that any anchor point has first $k-1$ coordinates equal to $b_1, \ldots, b_{k-1}$,

- $a_k = max\{c_k \mid (a_1, \ldots, a_{k-1}, c_k, \ldots, c_u) \in A \text{ and } a_k < b_k\}$.

We will show that [10] $a_k = max\{b'_k \mid (a_1, \ldots, a_k{-}1, b'_k, \ldots, b'_u) \in A\}$

Suppose this is not true. Then, there exists $(a_1, \ldots, a_{k-1}, b'_k, \ldots, b'_u) \in A$, such that $b'_k > a_k$. Since k is largest so that $a_1, \ldots, a_{k-1}$ are the coordinates of the given point $b'_k \geq b_k$.

Consider $v_1 = (a_1, \ldots, a_{k-1}, b_k, 1, \ldots, 1) \in A$. If $v_1 \in A$, then we have a contradiction (since longest prefix is $a_1, \ldots, a_{k-1}, b_k$ and not $a_1, \ldots, a_{k-1}$.

Case I: $C((a_1, \ldots, a_{k-1}, b_k, 1, \ldots, 1)) < T$

Then we have, $C((a_1, \ldots, a_{k-1}, b_k, 1, \ldots, 1)) < T$ and $C((a_1, \ldots, a_{k-1}, b_k, \ldots, b_u)) \geq T$. Then there exists a stage such that $C((a_1, \ldots, a_{k-1}, b_k, b_{k+1}, \ldots, b_l{-}1, 1, \ldots, 1)) < T$ and $C((a_1, \ldots, a_{k-1}, b_k, b_{k+1}, \ldots, b_l, 1, \ldots, 1)) \geq T$. But then $(a_1, \ldots, a_{k-1}, b_k, \ldots, b_l, 1, \ldots, 1)) \in A$, a contradiction.

Case II: $C((a_1, \ldots, a_{k-1}, b_k, 1, \ldots, 1)) \geq T$

Now $(a_1, \ldots, a_{k-1}, b'_k, \ldots, b'_u) \in A$, and $b'_k \geq b_k$. [11] This implies that

$$T \leq C((a_1, \ldots, a_{k-1}, b_k, 1, \ldots, 1)) \leq C((a_1, \ldots, a_{k-1}, b'_k, \ldots, b'_u))$$

Suppose $(a_1, \ldots, a_l - 1, \ldots, a_{k-1}, b'_k, \ldots, b'_u)$ is a small neighbor of the anchor point $(a_1, \ldots, a_{k-1}, b'_k, \ldots, b'_u)$, then since

$$C((a_1, \ldots, a_l{-}1, \ldots, a_{k-1}, b_k, 1, \ldots, 1))) \leq C((a_1, \ldots, a_l{-}1, \ldots, a_{k-1}, b'_k, \ldots, b'_u)) < T,$$

$(a_1, \ldots, a_{k-1}, b_k, 1, \ldots, 1) \in A$ a contradiction.

If $(a_1, a_{k-1}, b'_k{-}1, b'_{k+1}, \ldots, b'_u)$ is a small neighbor of $(a_1, \ldots, a_{k-1}, b'_k, \ldots, b'_u)$, then $(a_1, a_{k-1}, b_k - 1, 1, \ldots, 1)$ is a small neighbor of $(a_1, a_{k-1}, b_k, 1, \ldots, 1)$ again showing that $(a_1, a_{k-1}, b_k, 1, \ldots, 1) \in A$ since,

$$C((a_1, a_{k-1}, b_k, 1, \ldots, 1)) \leq C((a_1, a_{k-1}, b'_k, \ldots, b'_l - 1, \ldots, b'_u))$$

for any $b'_l > 1$. $(a_1, \ldots, a_{k-1}, b'_k,$ [12] $\ldots, b'_u)$ cannot be a smaller neighbor of the following type: $(a_1, \ldots, a_{k-1}, b'_k, \ldots, b'_l - 1, b'_u)$. ∎

2 Counting in a Slice of V

Set $\overline{V} = \{P \in V | X_i(P) \leq X_{i+1}(P),\ 1 \leq i < u\}$. For every point P in V, we denote by $\overline{P}$ a point whose coordintrs are sane the coordintes of P, but are in ascending order. Thus we have a map $\bar{-} : V \to \overline{V}$. Denote the image of a set S under this map by $\overline{S}$.

The following properties are easy to prove;

Proposition 2.1 *1. Foe any $I \in V$, there exists a $\sigma \in S_u$ such that $\sigma(I) = \overline{I}$.*

2. $C(P) = C(\overline{P})$.

3. $\cup_{\sigma \in S_u} \sigma(\overline{V}) = V$.

4. *If P is a neighbor to Q, then then $\overline{P}$ is a neighbor to $\overline{Q}$.*

5. *If S is connected then $\overline{S}$ is also connected.*

6. *If P_0 defines P, then $\overline{P}_0$ defines $\overline{P}$. Coversly if $\overline{P}_0$ defines $\overline{P}$ then there exists $\sigma \in S_u$ such that $\sigma(P_0)$ defines P.*

7. *$P \in \mathfrak{A}$ iff $\overline{P} \in \overline{\mathfrak{A}}$. In other words $\overline{\mathfrak{A}} = \mathfrak{A} \cap \overline{V}$.*

8. $\cup_{\sigma \in S_u} \sigma(\overline{\mathfrak{A}}) = \mathfrak{A}$.

Proof Wewill prove (4). If $\sigma \in S_u$, then $\sigma(P)$ and $\sigma(Q)$ are neighbors. Therefore we can assume that$P = \overline{P}$. Suppose k is an index such that $X_k(Q) = \min\{X_i(Q)$. Put $Q' = (1, k)(Q)$. It is enough to prove that $|X_i(P) - X_i(Q')| \leq 1$ for $i = 1$ and $i = k$. Since $P = \overline{P}$, and Q is a neighbor of P we have, $X_1(P) \leq X_k(P) \leq X_k(Q) + 1$. Therefore $X_1(P) - X_1(Q') \leq 1$. Also by the choice of k, $X_1(P) \geq X_1(P) \geq X_k(Q) - 1$. Thus $|X_1(P) - X_1(Q')| \leq 1$ Again since Q is a neighbor of P and $X_k(Q) \leq X_1(Q)$, $X_k(P) \leq X_k(Q) + 1 \leq X_1(Q) + 1 = X_k(Q;) + 1$. Thus $X_k(P) - X_k(Q') \leq 1$. On the other hand, $X_k(P) \geq X_1(P) \geq X_1(Q) - 1 = X_k(Q;) - 1$ This shows that $|X_k(P) - X_k(Q')| \leq 1$. This completes the proof.

Lemma 2.2 *For any $P \in V$, the cardinality of the set $\{Q \in V \mid \overline{Q} = \overline{P}\}$ is $\binom{u}{r_1, \ldots, r_s}$ where,*

$$\begin{aligned} r_1 &= \text{repetition of } x_1(\overline{P}) \\ r_2 &= \text{repetition of } x_{1+r_1}(\overline{P}) \\ \ldots \end{aligned}$$

Proof See second paragraph on page 16 in [**?**].

Example 2.3 *Let $V = \{(x_1, x_2, x_3, x_4) \mid x_i \in \mathbb{Z}^+, i = 1, \ldots, 4\}$, and $P = (1, 2, 1, 3)$. Then $\overline{P} = (1, 1, 2, 3)$ and $r_1 = 2, r_2 = 1, r_3 = 1$, and*

$$\left|\{Q \in V \mid \overline{Q} = \overline{P}\}\right| = \binom{4}{2, 1, 1} = \frac{4!}{2!1!1!} = 12.$$

Definition 2.4 Slice anchor points *are $A_+ = A \bigcap \bar{V}$, where A is the set of anchor points.*

It is clear that $< A_+ >= A$.

Definition 2.5 *Constant $\binom{u}{r_1, \ldots, r_s}$ associated with any point Q is called the permutation degree (p-degree) of Q.*

Lemma 2.6 *Suppose $Q_1, Q_2 \in \bar{V}$ are such that $x_i(Q_i) \leq x_i(Q_2)$, $1 \leq i \leq u$ and $C(Q_1) < T$, $C(Q_2) \geq T$. Then there exists $P \in \bar{\mathfrak{A}}$, such that $x_i(Q_1) \leq x_i(P) \leq x_i(Q_2)$.*

Proof We use induction on $m = \sum_{i=1}^{u} [x_i(Q_2) - x_i(Q_1)]$. If $m = 1$, then $Q_2 \in \bar{\mathfrak{A}}$ and we are done. Suppose j is a maximum index such that $x_j(Q_1) \neq x_j(Q_2)$. Set $Q = (x_1(Q_1), \ldots, x_j(Q_1)+1, \ldots, x_u(Q_1))$. Now, $x_j(Q_1) < x_j(Q_2) \leq x_{j+1}(Q_2) = x_{j+1}(Q_1)$. So, $x_j(Q_1)+1 \leq x_{j+1}(Q_1)$. Hence $x_j(Q) \leq x_{j+1}(Q)$. Therefore, $Q \in \bar{V}$. If $C(Q) \geq T$, then $Q \in A_+$. If $C(Q) < T$, then since

$$\sum_{i=1}^{u} [x_i(Q_2) - x_i(Q_1)] \leq \sum_{i=1}^{u} [x_i(Q_2) - x_i(Q_1)] + 1$$

by induction there exists a $P \in \bar{\mathfrak{A}}$ with required property.

The above lemma gives us a faster algorithm to compute $\mathfrak{A}$. We first find $\bar{\mathfrak{A}}$ and then take permutations of all points to construct $\mathfrak{A}$. For finding $\bar{\mathfrak{A}}$, we define the following order on the points of V, $I \leq J$ if $|I| < |J|$ or $|I| = |J|$ and $x_u(I) < x_u(J)$ or $|I| = |J|$, $x_i(I) < x_i(J)$ and $x_j(I) = x_j(J)\ \forall j > i$. Then as before, we consider $X = \{(i, i, \ldots, i) | 1 \leq i \leq n\}$ and apply lemma 2.6 to find an $anchorpoint$. The following algorithm uses this technique.

Lemma 2.7 *Suppose $(a_1, \ldots, a_r, \ldots, a_u) \in \bar{\mathfrak{A}}$ has the following property:*

$$a_r = max\{x_r(P) \mid x_i(P) = a_i, i = 1, \ldots, r-1, P \in \bar{\mathfrak{A}}\}$$

Suppose $Q \in \bar{V}$ is such that, $x_i(Q) = a_i$ for $1 \leq i < r$ and $x_r(Q) > a_r$, Then $C(Q) > T$.

Proof Suppose $(a_1, \ldots, a_r, \ldots, a_u) \in A_+$ had the property given in the hypothesis and
$(b_1, \ldots, b_u) \in \bar{V}$ with $a_i = b_i$ for $1 \leq i < r$ and $b_r > a_r$. We claim that $C(b_1, \ldots, b_u) \geq T$.

We prove this by contradiction. Suppose $C(b_1, \ldots, b_u) < T$. Set $Q = (a_1, \ldots, a_{r-1}, b_r, c_{r+1}, \ldots, c_u)$ with $c_i = \max(a_i, b_i)$, $r+1 \leq i \leq u$. Note that $b_r = \max(a_r, b_r) \leq \max(a_{r+1}, b_{r+1})$ since $a_r \leq a_{r+1}$ and $b_r \leq b_{r+1}$. Thus $Q \in \bar{V}$. Further, $x_i(Q) \geq b_i\ \forall i$ and $C(Q) \geq C(a_1, \ldots, a_u) \geq T$. Hence by Lemma 2.6, there exist $P \in A_+$ with $x_i(Q) \geq x_i(P) \geq b_i$. But for $1 \leq i < r$, $x_i(Q) = b_i = a_i$ and $x_r(Q) = b_r$. Thus we have an anchor point P with $x_i(P) = a_i$, $1 \leq i < r$ and $x_r(P) > a_r$. This is a contradiction. Thus we must have $C(b_1, \ldots, b_r) \geq T$.

Conversely,

Lemma 2.8 *For any $Q \in \{\bar{V} \setminus \bar{A}\}$ with $C(Q) > T$, there exists $(a_1, \ldots, a_r, \ldots, a_u) \in \bar{\mathfrak{A}}$, such that, $x_i(Q) = a_i$, for all $1 \leq i \leq r-1$ and $x_r(Q) > a_r$.*

Proof Suppose $(b_1, \ldots, b_u) \in \{\bar{V} \setminus \bar{\mathfrak{A}}\}$ with $C(b_1, \ldots, b_u) \geq T$. Choose a point $(a_1, \ldots, a_r, \ldots, a_u) \in \bar{\mathfrak{A}}$ such that r is a maximum integer such that $a_i = b_i$, $1 \leq i < r$ and $a_r \neq b_r$.This condition is vacuous if $r = 1$. We claim $a_r < b_r$.

We prove by contradiction. Suppose $a_r > b_r$. Set $Q = (a_1, \ldots, a_{r-1}, b_r, c_{r+1}, \ldots, c_u)$ with $c_i = \min(a_i, b_i)$. It is easy to show that $Q \in \bar{V}$. Since $C(Q) \leq C(a_1, \ldots, a_u)$, for any j,

$$C(x_i(Q), \ldots, x_j(Q) - 1, \ldots, x_u(Q)) \leq C(a_1, \ldots, a_j - 1, \ldots, a_u)$$

If $(a_1, \ldots, a_u) \in \bar{\mathfrak{A}}$, then there exists j such that $C(a_1, \ldots, a_j - 1, \ldots, a_u) < T$ by definition of anchor point. Hence, $C(x_i(Q), \ldots, x_j(Q) - 1, \ldots, x_u(Q)) < T$. Thus if $C(Q) \geq T$, then Q is an anchor point, which contradicts the minimality of r. Thus $C(Q) < T$. Also $x_i(Q) \leq b_i$. Now we use Lemma 2.6 to find $P \in \bar{\mathfrak{A}}$, such that $x_i(Q) \leq x_i(P) \leq b_i$. Since $x_i(Q) = b_i = a_i$, $1 \leq i \leq r$ and $x_r(Q) = b_r$, we found $P \in \bar{\mathfrak{A}}$ such that $x_i(P) = b_i$ for $1 \leq i \leq r$. This violates the maximality of rs. Thus we must have $a_r < b_r$.

Definition 2.9 *Let $(a_1, \ldots, a_r, \ldots, a_u) \in \bar{\mathfrak{A}}$ be an anchor point of Lemma 2.7. Consider,*

$$\tau(a_1, \ldots, a_r) = \{P \in V \mid x_i(\overline{P}) = a_i, 1 \leq i < r, \text{ and } x_r(\overline{P}) > a_r\}$$

Lemma 2.10 *Suppose $P_1, P_2 \in \bar{\mathfrak{A}}$ are such that,*

$$x_r(P_1) = \max\left\{x_r(Q) \mid x_i(Q) = x_i(P_1), 1 \leq i < r, Q \in \bar{\mathfrak{A}}\right\}$$

$$x_s(P_2) = \max\left\{x_s(Q) \mid x_i(Q) = x_i(P_2), 1 \leq i < s, Q \in \bar{\mathfrak{A}}\right\}$$

Then $\tau(x_1(P_1), \ldots, x_r(P_1))$ and $\tau(x_1(P_2), \ldots, x_s(P_2))$ are disjoint or identical.

Proof We will show that if $(c_1, \ldots, c_u)$ is in,

$$\tau(x_1(P_1), \ldots, x_r(P_1)) \bigcap \tau(x_1(P_2), \ldots, x_s(P_2))$$

then $x_i(P_1) = x_i(P_2)$ for $r = s$ and $1 \leq i \leq r$.

We can assume that $r \leq s$. If $r < s$, then we have $x_r(P_1) \geq x_r(P_2)$ since $x_i(P_1) = x_i(P_2) = c_i$ for $1 \leq i < r$. However by Lemma 2.8 we know that $x_r(P_1) < c_r = x_r(P_2)$. This is a contradiction. Hence we must have $r = s$ and $x_r(P_1) = x_r(P_2)$.

Lemmas 2.7 and 2.10 show that for the right choices of $\{a_1, \ldots, a_r\}$, $1 \leq r \leq u$,

$$\left\{Q \in \bar{V} \mid C(Q) \geq T\right\} \equiv \bar{\mathfrak{A}} \dot{\bigcup_{(a_1, \ldots, a_r)}} \tau(a_1, \ldots, a_r)$$

Lemma 2.11 *Let $1 \leq a_1 \leq a_2 \cdots \leq a_r \leq n$ with $r \leq u$. Define,*

$$\tau(a_1, \ldots, a_r) = \{P \in \bar{V} \mid x_i(P) = a_i, 1 \leq i \leq r \text{and} x_r(P) > a_r\}$$

and $f(r_1, \ldots, r_s)$ is the number of points P with $x_1 P$ repeated r_1 times, $x_{r_1+1}(P)$ repeated r_2,...$x_{r_1+r_2+\cdots+r_{s-1}+1}(P)$ repeated r_s times, then

$$\binom{f(r_1, \ldots, r_s) = k_1}{s\binom{k_2}{s-1}}$$

$k_1 = u - r$ *and* $k_2 = u - n$.

3 Counting Points in Slice

Algorithm 1 illustrates the algorithm used to compute the points with cost $\geq T$ in V using only $\bar{\mathfrak{A}}$. Below we give some definitions and formulae used in the Algorithm 1.

Definition 3.1 *If $part_u = \{$paritions of $u\}$, then for $E \in part_u$, $E = s_1^{p_1} s_2^{p_2} \dots s_r^{p_r}$. Also,*

$$E_n = \sum_{i=1}^{r} p_r \tag{1}$$

$$mult_E = \binom{E_n}{p_1, \dots, p_r} \tag{2}$$

$$perm_E = \binom{u}{\underbrace{s_1, s_1 \dots}_{p_1} \cdots \underbrace{s_r, s_r \dots}_{p_r}} \tag{3}$$

Let,

$$F_u = \sum_{E \in part_u} mult_E \times perm_E \times D_F \qquad D_F = \begin{cases} 1 & E_n = 1, 2 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$R_u = \sum_{E \in part_u} mult_E \times perm_E \times D_R \qquad D_R = \begin{cases} 1 & E_n = 2, 3 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$Q_{j,E_n} = \begin{cases} 0 & E_n < 3 \\ \dfrac{1}{(E_n - 3)!} & E_n = 3 \\ \dfrac{1}{(E_n - 3)!} \displaystyle\prod_{m=0}^{E_n - 4} (j - E_n + 3 + m) & \text{otherwise} \end{cases} \tag{6}$$

Algorithm 1 *Anchors2PointsSlice*(S, n, r)

$\{S$ is the $k \times U$ matrix of k of slice anchor points$\}$
$\{n$ is the max value of any element in an anchor point$\}$
$\{r$ is a 2×2 matrix where $[rs\ cs; re\ ce]$ define the upper left hand corner (rs, cs) and (re, ce) define the lower right hand corner of the space in $\bar{\mathfrak{A}}$ that is currently being processed$\}$
$result$ = Anchor2PointsSliceRecur(S, n, r)
for all p, slice anchor point in S **do**
  $result$ = $result$ + permutation degree of p
end for
return $result$

Anchors2PointsSliceRecur
f
MakeGroups

Algorithm 2 *Anchors2PointsSliceRecur*(S, n, r)

$\{S$ is the $k \times U$ matrix of k of slice anchor points$\}$
$\{n$ is the max value of any element in an anchor point$\}$
$\{r$ is a 2×2 matrix where $[rs\ cs; re\ ce]$ define the upper left hand corner (rs, cs) and (re, ce) define the lower right hand corner of the space in A that is currently being processed$\}$
$\{$If the final value of a summation variable is smaller than its initial value, let that summation be zero.$\}$
if S is empty **then**
  return 0
end if
$[rs\ cs; re\ ce] \Leftarrow r$
if $ce > cs$ **then**
  $r1 = makegroups(A, r)$
  for all ri such that ri is a 2×2 group specification in $r1$ **do**
    $result = result + Anchors2PointsSliceRecur(S, n, ri)$
  end for
end if
$femax$ =max(cs^{th} column from rows rs to re in A)
$u = ce - cs + 1$
while $fi <= n$ **do**
  $fsum = fsum$+f(fi,u,n)
  $fi = fi + 1$
end while
$k = n - femax - 1$
$\{F_u, R_u$ and Q_{j,E_n} below are from Equation (4), (5) and (6) respectively.$\}$

$$\begin{aligned} fsum &= 1 + kF_u + \frac{(k-1)k}{2}R_u + \\ &\sum_{E \in part_u} \left[mult_E \times perm_E \times \sum_{j=2}^{k-1} \left(\frac{(k-j)(k+1-j)}{2} Q_{j,E_n} \right) \right] \end{aligned}$$

d = denominator of multinomial coefficient of $S(re, 1), S(re, 2), \ldots, S(re, cs-1)$
$rp = \frac{U!}{d \times u!} \times fsum$
return $result + rp$

Algorithm 3 *f*(fi,u, n)

{fi is coordinate value to process}
{u is number of axes after and including fi}
{n maximum coordinate value}
{In the computation of $count$, if the starting condition of the product is greater than the ending condition, then $count = 0$.}
$k = n - fi, result = 0$
if $k \leq 0$ **then**
  return 0
end if
for all $E = s_1^{p_1} s_2^{p_2} \dots s_r^{p_r}$, partition of u **do**
  $mul = \binom{r}{p_1, \dots, p_r}$
  $perm = \binom{u}{\underbrace{s_1, s_1 \dots}_{p_1} \cdots \underbrace{s_r, s_r \dots}_{p_r}}$
  $t = p_1 + p_2 + \cdots + p_r$

$$count = \begin{cases} 1 & \text{if } t = 1; \\ k & \text{if } t = 2; \\ \sum_{l=t-2}^{k-1} \left[\left(\frac{1}{t-3} \prod_{m=0}^{t-4} (l - t + 3 + m) \right) (k - l) \right] & \text{otherwise.} \end{cases}$$

  $result = result + (mul \times perm \times count)$
end for
return $result$

Algorithm 4 *MakeGroups*(A, r)

{A is the $k \times U$ matrix of k of anchor points}
{r is a 2×2 matrix where $[rs\ cs; re\ ce]$ define the upper left hand corner (rs, cs) and (re, ce) define the lower right hand corner of the space in A within which groups are to be made}
$[rs\ cs; re\ ce] \Leftarrow r$
$ri \Leftarrow rs$
while $ri \leq re$ **do**
  Record $[ri\ (cs + 1)]$ as beginning of group
  $fe \Leftarrow A(ri, cs + 1)$
  while $ri \leq re$ **do**
    if $fe = A(ri, cs + 1)$ **then**
      $rf \Leftarrow ri$
      $ri \Leftarrow r1 + 1$
    end if
  end while
  Record $[rf\ ce]$ as end of group
end while
return all groups