UP-GROWING ON-LINE LINEAR DISCREPANCY OF TRIPLE-OPTIMAL PARTIALLY ORDERED SETS

A Thesis
Submitted to the Faculty
of
Washington and Lee University

By

Matthew Kiser

In Partial Fulfillment of the Requirements
for the Degree of
BACHELOR OF ARTS
WITH HONORS

Major Department: Mathematics
Thesis Advisor: Dr. Mitchel T. Keller
Second Reader: Dr. Aaron Abrams

May 2016

ABSTRACT

Whether we acknowledge it as a poset or not, posets arise in many natural contexts, and many also seem to warrant linear extensions (or rankings) of the poset. In some sense, the linear discrepancy of a linear extension L of a poset $\mathcal{P}$ indicates the unfairness of L. We describe triple-optimal posets, a class of posets where there exists at least one linear extension which has linear discrepancy three times the minimum linear discrepancy l. This is the worst case scenario; there is no way to have a worse linear discrepancy than triple the optimal linear discrepancy. Two players, a Builder and an Assigner, play an on-line game to construct a linear extension. The Builder gives the Assigner points from $\mathcal{P}$ that the Assigner subsequently irrevocably places in a linear extension L_A using an algorithm. The Builder's goal is to maximize the linear discrepancy of L_A while the Assigner battles to minimize the linear discrepancy of L_A. Restrictions can be placed on the Builder, such as up-growing where the Builder cannot give points less than those points already given. In the context of up-growing, we play this on-line game using triple-optimal posets and develop an algorithm that caps the linear discrepancy of L_A at $2l$ on triple-optimal posets with linear discrepancy l.

ACKNOWLEDGMENTS

Here's to the best family, friends, mentors and faculty/coaches; I appreciate everything you have helped me achieve. (Note: those sets are not disjoint.) I am lucky to have had the best teachers (in and out of the classroom). You have taught me that I have the ability to accomplish incredible things. I hope that I make you proud. Happy Mother's Day, Mom!

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1. Background

In a *set*, we can create or describe relationships between the *elements*, or *points*. One set is the natural numbers, and as we learned in elementary school, 3 is greater than 2 and 4 is greater than 3. Also, 4 is greater than 2. What we did not learn in elementary school is that our numbers are simply elements of an infinite *totally ordered set*, where every element is *comparable* to each of the other elements in the set. (While we do not have all of the notation to understand the below definition of comparable quite yet, do not fret; you will shortly understand.)

Definition 1.1. Two distinct points in a set $\mathcal{P}$ are *comparable* if either $x < y$ in $\mathcal{P}$ or $y < x$ in $\mathcal{P}$.

Totally ordered sets are pretty straight-forward, however, and a far more interesting, complex topic is a set that is not totally ordered. What does it mean to not be totally ordered? Two elements can now be *incomparable*. Before defining incomparability with mathematical terms, we will build an example to make sure we understand the mechanics behind this concept.

Let's enter the world of an Emergency Room waiting area. Just like a normal ER, people are coming in with various medical afflictions to (hopefully) be treated by our amazing doctors. Jess and Elle are sitting in our luxurious waiting room. Both are right-handed and have broken their right arms in the same location on the same bone. Jess and Elle are clearly different people; however, there is no inherent reason Elle should be treated before Jess or vice versa (other than the doctor only tending to one patient at a time). Also, we should note that neither Elle nor Jess would be terribly distraught if the other were to be called back earlier than themselves. Thus, we deem Elle and Jess to be *incomparable* to each other in our Emergency Room.

Definition 1.2. Two distinct points in a set $\mathcal{P}$ are *incomparable* if the two points are not comparable in $\mathcal{P}$. We write $x \parallel y$ in $\mathcal{P}$ if x and y are incomparable.

Similar to our elementary school numbers, we can still have patients comparable to our friends Elle and Jess. This includes Lydia, a patient with a heart attack, and Riley, a patient with a stubbed toe. Our friends writhing in pain from a broken arm, while incomparable to each other, are not as high priority as our new friend Lydia, but are higher priority than our fumbling patient, Riley.

1.2. Further Background (With More Math Talk)

Now these fun and exciting sets do have a name. Since these sets are not totally ordered, we call them *partially ordered sets*, or *posets*.

Definition 1.3. A *partially ordered set* or *poset*, $\mathcal{P} = (X, P)$ is:

a ground set X whose elements are referred to as *points* and
a binary relation P (also called a *partial order*) on X that is:
reflexive – $(x, x) \in P$, for all $x \in X$
antisymmetric – if $(x, y) \in P$ and $(y, x) \in P$, then $x = y$
transitive – if $(x, y) \in P$ and $(y, z) \in P$, then $(x, z) \in P$

One alternative to writing $(x, y) \in P$, is the notation $x \leq y$ in P, and if we know that x and y are distinct points we can write $x < y$ in P. Sometimes we will drop the "in P" when the poset to which we are referring is understood or clear. Now, when looking at the definitions of comparability and incomparability, we should fully understand the notation as well as the concepts.

Definition 1.4. Given a poset $\mathcal{A}$, a poset $\mathcal{B}$ is called a *subposet* of $\mathcal{A}$ iff the ground set of $\mathcal{B}$ is a subset of the ground set of $\mathcal{A}$ and has the same binary relations for that subset. We can also write $\mathcal{B} \subseteq \mathcal{A}$.

Example 1.5. One poset is the "divides poset" $\mathcal{P}_n$. Given a ground set of integers $\{1, 2, 3, \ldots, n\}$, we say that $a \leq b$ iff a divides b. Otherwise, $a \parallel b$. So for $\mathcal{P}_6$ composed of the ground set $\{1, 2, 3, 4, 5, 6\}$, $2 < 4$ and $2, 3 < 6$, but $2 \parallel 5$. Hence, for $\mathcal{P}_6$ we have:

$$\begin{aligned} \text{the ground set } X &= \{1, 2, 3, 4, 5, 6\} \\ \text{the binary relation } P &= \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 2), \\ &\qquad (2, 4), (2, 6), (3, 3), (3, 6), (4, 4), (5, 5), (6, 6)\} \end{aligned}$$

All of those numbers and parentheses are a little difficult to distill quickly. "So, is there an easier way?!" you ask excitedly. Well of course! Figure 1 shows the *order diagrams* of a few posets. The first poset is a *chain*, where every point is comparable to every other point, also known as a total order. The next poset is an *antichain*, where every point is incomparable to every other point. The third diagram is a $\mathbf{2 + 2}$ poset. The family of this kind has two chains, one of size m and other of size n, both are incomparable to the other and written $\mathbf{m + n}$. Finally, IV is $\mathcal{P}_6$ as described in Example 1.5.

An *order diagram* (sometimes called a diagram or Hasse diagram) displays the elements as points with lines signifying a comparability. In other words, if there is a line connecting two points, then the element higher up in the diagram is greater than the lower one in the poset. If there is no line connecting two points, then they are incomparable. Since drawing lines for all the comparabilities can make for a very messy diagram, we do not draw the lines implied by transitivity. In Example 1.5 we know that $1, 2, 3 \leq 6$. When we draw the diagram for $\mathcal{P}_6$ in III of Figure 1 we do not

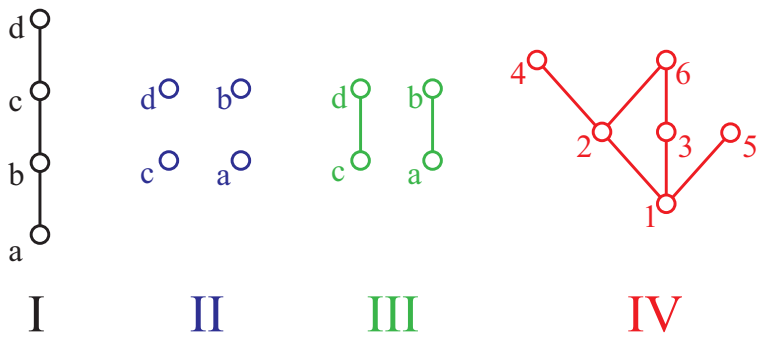


Figure 1. Order Diagram

explicitly draw a line from 1 to 6 because that line is implied by the lines from 1 to 2 and from 2 to 6 (or from 1 to 3 and from 3 to 6).

1.3. What do I do with a poset?

Now imagine that our intercom calling out patients' names were to call out all the incomparable points at one time. If you imagined anything other than an inaudible mess, really think about more than five names being called at the exact same time. Thus, we need to put our patients (the ground set) in some order for the nurse to call out in a logical and practical sequence that treats higher priority patients first (an order that does not reverse any comparabilities). In mathematical terminology, we organize our poset into a *linear extension* L.

Definition 1.6. For a poset $\mathcal{P}$, a *linear extension* L of $\mathcal{P}$ is a list (total order) of the elements in an order such that if $x < y$ in $\mathcal{P}$, then x will not be in a higher position than y in the linear extension L. The concept of *in a higher position* will be made precise after Example 1.8 by defining *height*.

Definition 1.7. In a poset $\mathcal{P} = (X, P)$, a comparability between two points x and y, where $x < y$ in $\mathcal{P}$, is *reversed* in a total order T on X if x is placed in a higher position than y in T. Thus, T would not be a linear extension.

Example 1.8. In Figure 2, we see the diagram of $\mathcal{P}_6$ repeated from Figure 1 as well as a few a linear extensions of $\mathcal{P}_6$. This list is not complete; we leave at least one for the reader. Notice that 1,2,5,4,6,3 is not a linear extension since $3 < 6$ in $\mathcal{P}_6$, but 6 is above 3 in this total order.

We do want to mention that counting (not to mention generating) all linear extensions of a poset is computationally overwhelming. Imagine a poset with 100 points and trying to figure out all of the possibilities.

Definition 1.9. The *height* of a point x in a linear extension of a poset $\mathcal{P}$, denoted $h_L(x)$, is one more than the number of points between x and the lowest point in the linear extension. It can also be defined as the number of points in L that are below x in L, $h_L(x) = |\{y \in \mathcal{P} : y < x \text{ in } L\}|$. The height of the bottom point z_0 is zero: $h_L(z_0) = 0$.

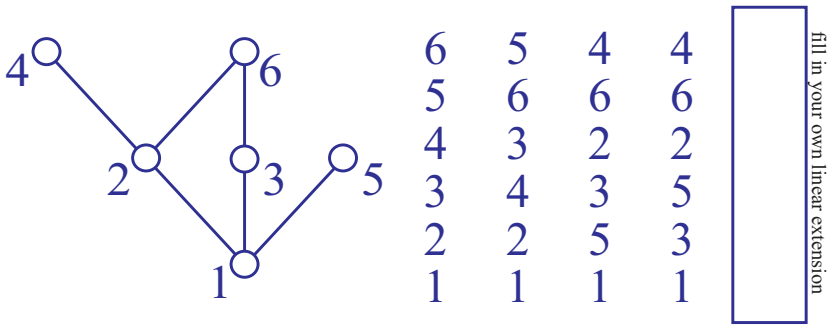


Figure 2. Linear extensions of the diagram in Figure 1

You might be thinking to yourself, "But how do we put incomparable elements in an order? Won't that cause better or worse linear extensions? There are probably some instances of forcing an unfairness, right?" You're right, and you definitely want read the entire thesis! If you weren't asking that, we forgive you, but you will still want to read all of this thesis, you just will not be as excited about everything.

To address the first question, since incomparable points do not need to be in any given order, we can put them in whatever order works. Fundamentally, calling back Elle before Jess is just as acceptable as diagnosing Jess before Elle. However, Lydia absolutely must be seen before both Elle and Jess. Similarly, in Figure 2 the farthest left linear extension can put 3 below 4, and the next linear extension can put 4 below 3. This does not violate any rules and is acceptable.

To address the second question, what the layperson can think of as the unfairness of an order, mathematicians call the *linear discrepancy* of a linear extension L of a poset $\mathcal{P}$, also written $\mathrm{ld}(L, \mathcal{P})$. We can then have an optimal linear discrepancy $\mathrm{ld}(\mathcal{P})$, which is (quite obviously) the smallest linear discrepancy across all possible linear extensions. To define these terms more precisely, we have the following definitions:

Definition 1.10. The *linear discrepancy* $\mathrm{ld}(L, \mathcal{P})$ of a linear extension L of a poset $\mathcal{P}$ is the maximum difference between the heights of any two incomparable points in L.

Definition 1.11. The *optimal linear discrepancy* $\mathrm{ld}(\mathcal{P})$ of the poset $\mathcal{P}$ (or just linear discrepancy of a poset) is the smallest linear discrepancy across all linear extensions. That is

$$\mathrm{ld}(\mathcal{P}) := \min\{\mathrm{ld}(L, \mathcal{P}) : L \text{ is a linear extension of } \mathcal{P}\}.$$

The concept of linear discrepancy was introduced in 2001 by Tanenbaum et al. in [5]. Their paper revolves around four examples: residency selection by two future doctors exiting medical school, the ER problem we touched on earlier in this thesis, project alternatives for a management presentation and salary computation.

To address the third question, we reference research first explored by Kierstead in [3] on coloring graphs in an *on-line* fashion where vertices of graphs were given one at a time and assigned a color based on an algorithm. Our application of on-line

differs from the variety explored in [3] in that we will be examining linear extensions of posets instead of graphs and colors.

Our interests begin when we play a game with our friend Alex. We will give Alex points from a poset $\mathcal{P}$ one at a time and tell Alex all of the relations the point has with the previously given points. She will follow an Algorithm to irrevocably order the new points in some linear extension. Because we are Bullies, we are trying to force the largest unfairness (linear discrepancy) possible. (I know, it's kind of mean; that's life.) However, Alex is not a pushover, so she is going to attempt to limit the linear discrepancy using some Algorithm. Additionally, we, the Bullies, cannot make two points x and y appear to be incomparable when in fact later we give a point z such that $x < z < y$; in other words, transitive relations must be given.

Research surrounding on-line games attempts to create an algorithm such that for a family of posets the linear discrepancy of the linear extension created by the on-line game will be limited to a certain upper bound $f(l)$, where f is a function and l is the optimal linear discrepancy. This is Alex's goal in our brief description. At the same time the Bully (also known as Builder) is trying to force a lower bound $g(l)$ regardless of the algorithm, where g is a function and l is the optimal linear discrepancy. We, as mathematicians, want $g(l) = f(l)$.

Most work in the field of on-line problems for posets has focused on the width and chain partitions of specific classes of posets. We will focus on linear discrepancy instead of width and chain partitioning. Keller, Streib and Trotter first studied on-line linear discrepancy in [2]. The results from that paper conclude that using a certain on-line algorithm, any poset with linear discrepancy l has an upper bound of $3l - 1$. Additionally, for every positive integer l, there is a poset with linear discrepancy l on which a Builder can force for any algorithm a lower bound of $3l - 1$.

This thesis will study *up-growing* on-line linear discrepancy. *Up-growing* restricts the Builder order to only giving points that are greater than or incomparable to points already in the on-line linear extension. The motivation to study up-growing stems from known examples achieving the lower bound $3l - 1$ all relying on Builder orders that are not up-growing. Thus, those lower bounds exploit the ability to force points below previously given points due to comparabilities. Felsner et al. studied on-line chain partitions on semi-orders restricted to up-growing in [1]. Semi-orders are a special class of interesting posets, that we will keep outside of the scope of this thesis.

A curious coincidence is that every linear extension, from bottom to top, is a sequence in which the Builder can give points to the Assigner in an up-growing restricted setting. Assuming we have been ordering our linear extensions such that lesser elements in the poset go lower in the linear extension, if the Builder gives the points from bottom to top, no point less than the point in the poset being given will remain in the linear extension.

We begin with describing triple-optimal posets and the different subposets of a triple-optimal poset. Necessary conditions for a poset to be in the triple-optimal family are also discussed. Through a few iterations and some problem solving, we

develop an algorithm to limit the linear discrepancy of an on-line linear extension to $2l$ in an up-growing setting. We then prove that this algorithm will have an upper bound of $2l$ and a Bully can force a lower bound of $2l$.

CHAPTER 2. TRIPLE-OPTIMAL POSETS

Now that we are experts in the field of partially ordered sets, we can confidently move forward. Kloks, Kratsch and Müller in [4] found that for any poset with optimal linear discrepancy l the worst possible linear discrepancy across any of the linear extensions of that poset is necessarily less than or equal to $3l$. This means we could never find a poset with a linear extension with quadruple the linear discrepancy of the poset. Let's explore this topic further by describing a *triple-optimal poset*.

2.1. Description of triple-optimal posets

Posets come in all varieties and can be broken into different families or general patterns. Two of the first ones that you learn are *chains*, total orders that look like one long stick (or even possibly a chain) as shown in I in Figure 1, and *antichains*, posets with no comparabilities between any of the points as shown in II in Figure 1. In terms of creating linear extensions out of these posets, chains and antichains are quite mundane, and they do not leave that much to research. There is a class of posets that exhibit a particularly interesting behavior, and we call the class of them *triple-optimal posets*. Every poset with optimal linear discrepancy l in our family of triple-optimal posets has at least one linear extension, which we call the worst linear extension, with linear discrepancy $3l$.

Definition 2.1. A *triple-optimal poset* $\mathcal{P}$ is a poset for which there exists at least one linear extension L such that $\mathrm{ld}(\mathcal{P}, L) = 3\,\mathrm{ld}(\mathcal{P})$.

In Figure 3 we see the diagrams of two posets with the triple-optimal property. An optimal and a worst linear extension are given next to each of them. While the linear discrepancy is a result of the difference between multiple pairs of points in the left diagram, we take particular note of the pair a and j. In L_O j and a are three apart, also j and a are 9 apart in L_W. Similarly in the right diagram, c and a have the same property. Looking at L_O, b and i are also 3 apart. It may be helpful to note or mark each of the incomparabilities in the four linear extensions.

In this section and beyond, we will be looking at two linear extensions: one linear extension of a triple-optimal poset with optimal linear discrepancy l, which we will call L_O, and one linear extension of the same poset with linear discrepancy $3l$, which we will call L_W, since it is a worst linear extension of this poset with respect to linear discrepancy.

We will only look at *l-critical* triple-optimal posets. For ease of reading, throughout this thesis where we write "triple-optimal" poset we actually mean "l-critical triple-optimal" poset.

Definition 2.2. An *l-critical triple-optimal poset* is a triple-optimal poset such that there are no points that can be removed and keep an optimal linear discrepancy l and worst linear discrepancy $3l$.

For some triple-optimal poset $\mathcal{Q}$, if we can remove a point(s) from $\mathcal{Q}$ and L_W still has linear discrepancy $3l$, then $\mathcal{Q}$ is not a l-critical triple-optimal poset. It is also

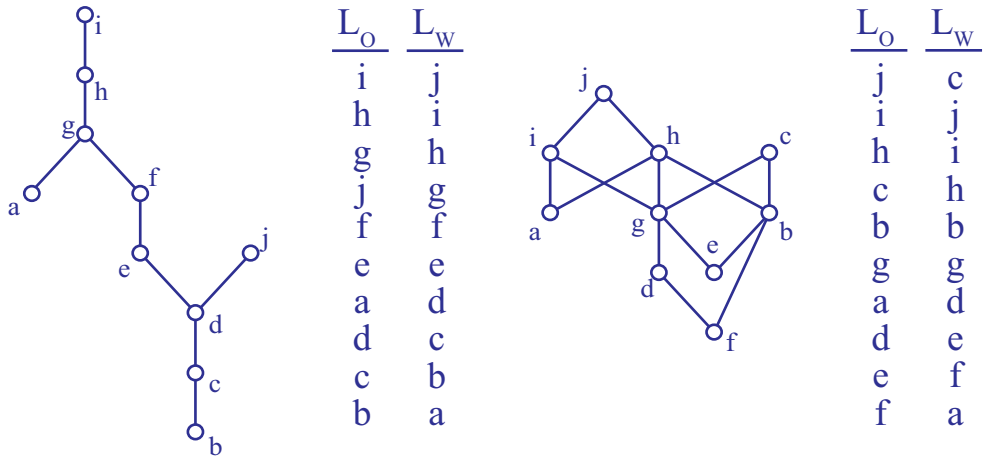


Figure 3. Triple-optimal poset with an optimal linear extension and triple linear extension

worth noting that a critical triple-optimal poset $\mathcal{P}$ does not contain any point z such that $z > g$ for all $g \in \mathcal{P}$ or $z < g$ for all $g \in \mathcal{P}$. This happens when $\mathcal{P}$ is a subposet of some greater poset $\mathcal{R}$ that has the same linear discrepancy as $\mathcal{P}$.

We are careful not to define l-criticality as simply removing points such that we still get a triple-optimal poset. Based on the left diagram in Figure 4, we may think that conception is true. If we remove the red point in the left diagram, we see the remaining poset is still triple-optimal with $l = 3$. Thus, the left poset is not a 3-critical triple-optimal poset. On the flip side, removing the three red points in the right diagram in Figure 4 reveals a triple-optimal poset where $l = 3$. Does this mean the diagram including the three red points is not a 4-critical triple-optimal poset? No, if we were to remove a single point, the remaining poset would not be triple-optimal when it was before we removed the point. Thus, the poset on the right is a 4-critical triple-optimal poset. We can also note that the right diagram where $l = 4$ has 13 points, which allows for a linear extension with linear discrepancy $3l$, but 12 points would not allow that.

Now that we have the necessary foundation to understand a triple-optimal poset, we will dive into more detail on the comparabilities and incomparabilities in a triple-optimal poset $\mathcal{P}$. Let L_O be an optimal linear extension and L_W be a worst linear extension. By criticality, we can assume in L_W, the discrepancy is realized by the top and bottom points because the maximum discrepancy only occurs once in L_W. Thus, we label the top point x and the bottom point y. When we look at L_W, there are $(3l - 1)$ points between x and y, since we know this linear extension has linear discrepancy $3l$. These intermediate points are members of one of the three sets:

$$A = \{a : a < x\}$$

$$B = \{b : b \parallel x \text{ and } b \parallel y\}$$

$$C = \{c : c > y\}$$

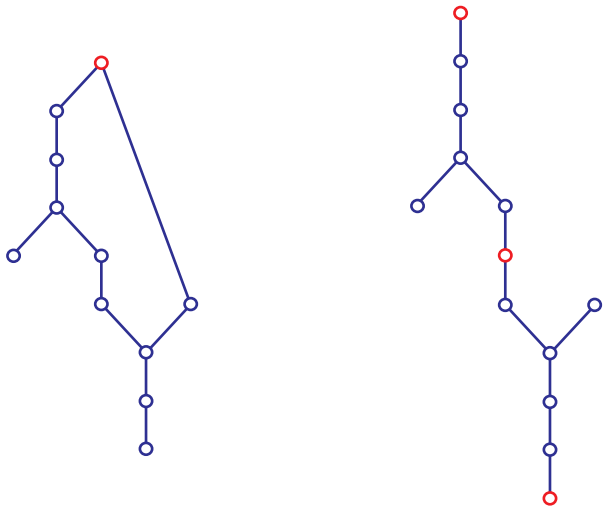


Figure 4. 3-critical vs. 4-critical triple optimal poset

You may be wondering, "Is $A \cap C$ nonempty or can we modify our definitions of A and C to clarify they are disjoint?"

Lemma 2.3. *The set $A \cap C$ is empty.*

Proof. Suppose there is a $q \in A \cap C$. This would mean $y < q < x$, which means $y < x$, which violates our assumption that $x \parallel y$. Thus, A and C are disjoint! □

So now we clarify our definitions to:

$$A = \{a : a < x \text{ and } a \parallel y\}$$
$$B = \{b : b \parallel x \text{ and } b \parallel y\}$$
$$C = \{c : c > y \text{ and } c \parallel x\}$$

Now that we have sketched out L_W, what does L_O look like? Well, from L_W we know there are $3l+1$ points divided into x, y, A, B and C. Since x is incomparable to y and all the points in B and C, $|B \cup C| \leq 2l-1$. To see why, suppose $|B \cup C| \geq 2l$. Then, in L_O regardless of the order in which the points are put, the difference in height between x and the top or bottom point from $(B \cup C) \cup \{y\}$ would be greater than l, contradicting $\mathrm{ld}(\mathcal{P}) = l$. Similarly, $|A \cup B| \leq 2l-1$ because y is incomparable to x and all points in A and B.

We claim $l \leq |A| \leq 2l-1$ and $l \leq |C| \leq 2l-1$. We know A, B and C are disjoint and contain $3l-1$ points. Since $|B \cup C| \leq 2l-1$, it is necessary for $|A| \geq l$. If we interchange A and C in the preceding argument, we get $|C| \geq l$. To see the upper bound, suppose B is empty. This would mean

$$2l-1 \geq |A \cup B| \geq |A \cup \emptyset| \geq |A|.$$

We can use a similar argument for the upper bound of C. Thus, we get:

$$l \leq |A| \leq 2l-1 \text{ and } l \leq |C| \leq 2l-1.$$

When B is non-empty, the lower bound for the sizes of A and C is unchanged, but the upper bound for $|A|$ and $|C|$ decreases by the size of B. For example, let C have l points and B be empty. This means A has $2l-1$ points. Now increase $|B|$ to one while $|C|$ remains at l. Consequently $|A|$ decreases to $2l-2$. Continuing this until $|B| = l-1$, A will have l points. In every scenario the lower bound for $|A|$ and $|C|$ remains at l, but the upper bound decreases as the size of B increases.

To recap a little, we have a subset A whose elements are all less than x and incomparable to y and another subset C whose elements are all greater than y and incomparable to x, where $|C| \geq l$ and $|A| \geq l$. In Figure 5, we color the points in Figure 3 to more clearly see this distinction.

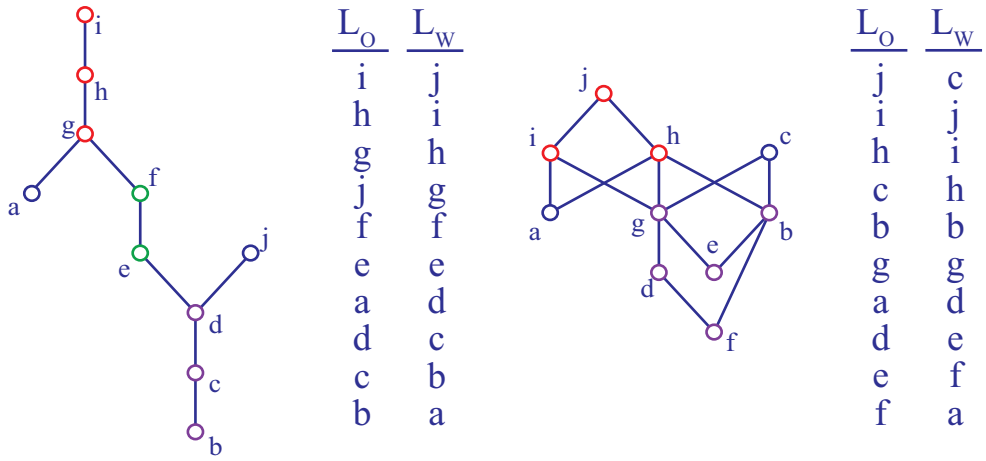


Figure 5. Posets with elements colored as elements of A as purple, B as green, and C as red.

We also want to know the relation between the elements of A and the elements of C. It would be very simple and straightforward if we had $c > a$ for all $a \in A$ and $c \in C$. However, life is not simple and there can be considerable variation. Thus, we now want to define two more sets:

$$A^c = \{a \in A : \text{ there exists a } c \in C \text{ such that } a \parallel c\}$$

$$C^a = \{c \in C : \text{ there exists an } a \in A \text{ such that } a \parallel c\}$$

On the left diagram in Figure 5, $C^a = \emptyset = A^c$. This leads us to note that when B has $l-1$ points C^a and A^c are empty. Whereas, in the right diagram, $C^a = \{i\}$ and $A^c = \{b\}$.

2.2. A discussion of a non-theorem on determining a triple-optimal poset

Without knowing any of the linear extensions, when checking a poset to see if it is triple-optimal, a good place to start is counting the points. Suppose we count k points, we want to make sure that there is an l such that $3l+1 = k$. If this equation is not satisfied, then this poset may not be an l-critical triple-optimal poset, and we

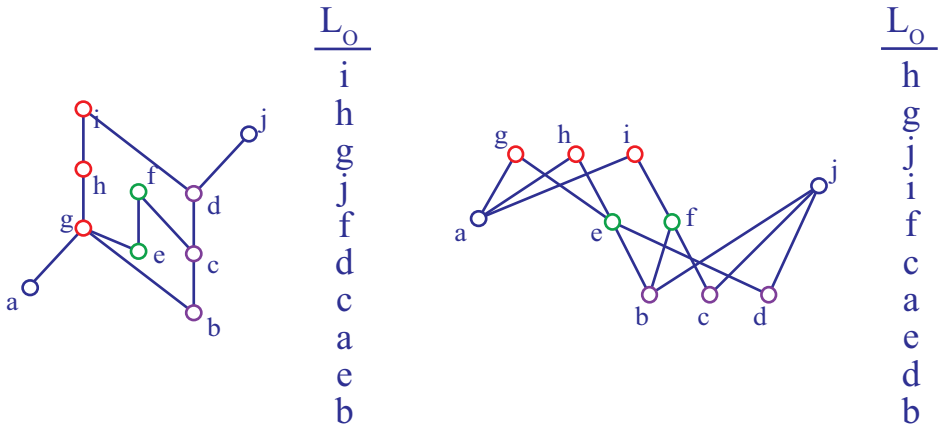


Figure 6. Posets that appear to be triple-optimal, but are not

may want to see if we can remove a point(s) to create an l-critical poset. For advice on which points are good candidates to remove, we look to the definition of l-critical.

If $3l + 1 = k$ is satisfied or we have removed points to satisfy the equation, we want to find the sets A, B and C as well as the points x and y. If we can successfully break the poset into the sets, the complexity of determining possible linear extensions skyrockets. The number of possible combinations of comparabilities is too large to mention all of them here. Our best bet is to attempt to create L_O and see if all the properties hold.

In Figure 6, the posets may appear to be 3-critical triple-optimal: "A" and "C" have $l = 3$ points and "B" has $l - 1 = 2$ points; however, the linear discrepancy of both posets is 5, which is clearly not l. In the left poset, we see that the points h, c, j and e are incomparable to another point at distance 5 away from itself. There is also no way to rearrange the linear extension to lower the distance between every pair of points 5 apart from each other because lowering one would increase another. The diagram on the right has linear discrepancy 5 between i and d, j and e and h and c. Attempting to improve one would either increase the distance between another pair of points or leave another pair unaffected. Thus, even though both of these diagrams appear to be triple-optimal at first glance, neither are triple-optimal due to the comparabilities and incomparabilities within the poset.

CHAPTER 3. GREEDY-SELFISH ALGORITHM ON TRIPLE OPTIMAL POSETS

At the end of Section 1.3 we mentioned the concept of an on-line game between ourselves (the Bully or Builder) and Alex (an Algorithm). In Section 2.1 we examined the class of posets where the optimal linear discrepancy is l and the worst linear extension has linear discrepancy $3l$. We will now explore and develop an algorithm that aims to cap the linear discrepancy of the linear extension created by the Algorithm and the Builder in the up-growing on-line game of any triple-optimal poset $\mathcal{P}$ at $2l$.

We want to remember that we refer to x as the top point in the worst linear extension L_W of a poset $\mathcal{P}$, and y as the bottom point in L_W. Also, height is defined as the number of points in L that are below w in L, $h_L(w) = |\{z \in \mathcal{P} : z < w \text{ in } L\}|$. It is also important to remember $h_{L_O}(x) - h_{L_O}(y) = l$. Additionally, we are operating in an up-growing setting, where the order in which the Builder can give points to the Algorithm is restricted in that the order must be a linear extension of the poset. Thus, each new point cannot be less than any previously-presented point.

3.1. Quick Review of Algorithms

We will first define a few terms that are essential to understanding an on-line algorithm.

Definition 3.1. A *legal position* in a linear extension L for a point x in the context of an on-line setting is any location in L into which x may be inserted without reversing any comparabilities. Other positions are called *illegal*.

Definition 3.2. A *qualifying position* for a point in the context of an on-line setting is any legal position in a linear extension that complies with a given algorithm.

Also, we establish the following notation for the rest of the chapter:

$\mathcal{P}$	is some triple-optimal poset
L_A	is the linear extension created by the Algorithm and Builder
L_A^t	is the linear extension created after the t^{th} point is given while playing the on-line game
L_O	is a linear extension with the optimal linear discrepancy
L_W	is a linear extension with the worst linear discrepancy

There is a large variety of algorithms that go along a range of complexity. Some of the simplest algorithms are putting the new point in the highest legal position or alternately the lowest legal position. Due to the complexity of the posets, we want to start with a more complex algorithm. Thus, we begin by using the *Greedy Algorithm*.

3.2. Greedy Algorithm

In the greedy algorithm **G**, a qualifying position for the t^{th} point z_t is the highest position that minimizes the linear discrepancy of L_A^t, $\text{ld}(\mathcal{P}, L_A^t)$.

Thus, the goal of this condition is to keep $\text{ld}(\mathcal{P}, L_A^t) = \text{ld}(\mathcal{P}, L_A^{t-1})$. We may think that if this goal cannot be met, then $\text{ld}(\mathcal{P}, L_A^t) = 1 + \text{ld}(\mathcal{P}, L_A^{t-1})$; however this is false when you consider $\mathbf{1} + \mathbf{10}$ where the **10** chain comes first in the builder order. Thus, $\text{ld}(\mathcal{P}, L_A^t)$ is as small as possible subject to the algorithm's previous decisions that formed L_A^t.

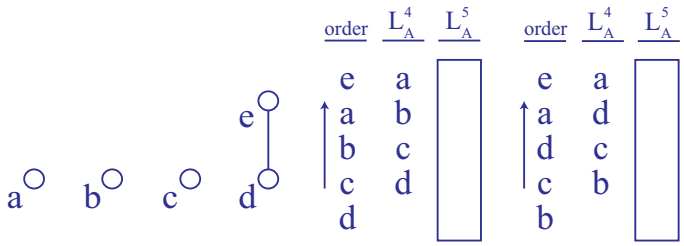


Figure 7. Diagram, builder order and L_A's for Example 3.3

Example 3.3. Figure 7 shows a diagram for a poset $\mathcal{Q}$ with ground set $\{a, b, c, d, e\}$. Two orders are given for the Builder to give the Algorithm. The linear extension L_A^4 is given for both orders, fill in the box with L_A^5 and calculate the linear discrepancies of the two L_A's. The correct answer is in the footnote[1].

3.3. Greedy-Selfish Algorithm

When we look at Example 3.4 below, we see how the greedy algorithm is not a sufficient algorithm for triple-optimal posets if we wish to keep $\text{ld}(L_A) \leq 2l$.

Example 3.4. Using the greedy algorithm on the triple-optimal poset $\mathcal{R}$ shown in Figure 8 subject to the Builder's order given next to the diagram, we get a linear discrepancy of 9 between the points e and i. However, $\text{ld}(\mathcal{R}) = 4$, and thus $\text{ld}(\mathcal{R}, L_A) > 2\,\text{ld}(\mathcal{R})$. The correct L_A is in the footnote[2].

We must now give a new definition to improve our Algorithm.

Definition 3.5. The *point discrepancy* of a point z in a linear extension L of a poset $\mathcal{P}$ is the maximum distance between z and a point w such that $w \parallel z$. We write $\text{pd}(z, \mathcal{P}, L)$.

We will name our altered algorithm the greedy-selfish algorithm **GS**. The qualifying position using **GS** for the t^{th} point z_t is the highest position that minimizes $\text{pd}(z_t, \mathcal{P}, L_A^t)$ after minimizing the linear discrepancy of L_A^t, $\text{ld}(\mathcal{P}, L_A^t)$. A more explicit version of **GS** is below.

[1]Left: d, c, b, a, e and $\text{ld}(\mathcal{P}) = 3$; Right: b, c, d, a, e and $\text{ld}(\mathcal{P}) = 4$.

[2]From bottom to top: $y, a, b, e, c, d, f, g, j, x, k, h, i$

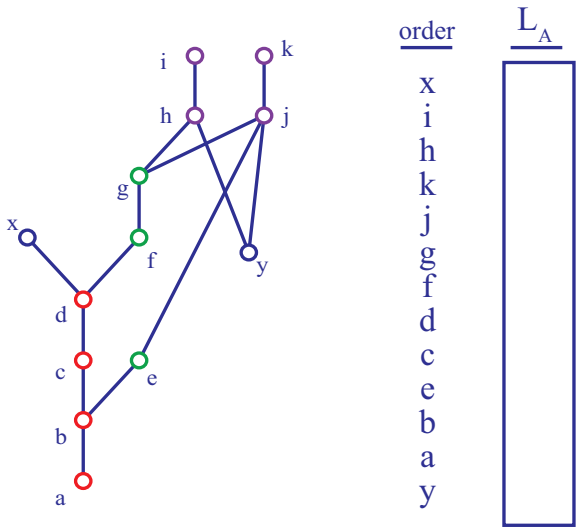


Figure 8. Diagram, builder order and L_A's for Example 3.4

1. Determine the legal positions for z_t.
2. Exclude the legal positions that, assuming z_t were put in that position, would result in a non-minimal linear discrepancy for L_A^t.
3. For each of the remaining positions, exclude the positions that, assuming z_t were put in that position, would result in a non-minimal point discrepancy for z_t in L_A^t.
4. The qualifying position for z_t is the highest of the remaining positions.

In Example 3.6, we will see how this condition fixes the problem in Figure 8.

Example 3.6. Using **GS** on the diagram and Builder order in Figure 8 we get the following linear extension of linear discrepancy 8:

$$y, a, b, c, d, f, x, g, e, h, i, j, k$$

3.4. Greedy-Selfish-Aware Algorithm

Our work on the algorithm to keep the up-growing on-line linear discrepancy less than $2l$ is foiled when the Builder uses a certain poset in a certain order, this builder order is explored in Example 3.7.

Example 3.7. We look to the diagram of the triple-optimal poset with the maximum quantity of incomparabilities $\mathcal{P}_W$ (Skeptical? Try adding an incomparability.) shown in Figure 10. Using **GS** on $\mathcal{P}_W$ and the order shown to the right of the diagram, we get a L_A with linear discrepancy 9 between f and h, which is greater than $2l$. Fill in L_A using the order on the left in the figure. The correct on-line linear extension is in the footnote[3].

[3]From bottom to top: $y, a, b, f, e, c, g, d, x, h, j, k, i$

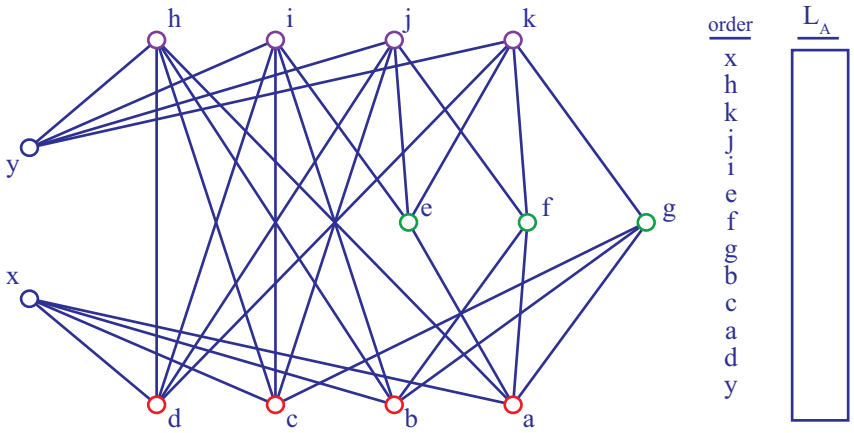


Figure 9. Diagram, builder order and L_A's for Example 3.7.

Therefore, we alter the algorithm to make the greedy-selfish-aware algorithm **GSA**. The aware portion of the title captures the sense that the algorithm is aware of the possibility that the builder order may use the placement of a new point in order to take advantage of the point discrepancy of another point. Since the algorithm is aware this event might occur, the algorithm avoids it.

We define $z_0 \in \mathcal{P}$, where $h_{L_A^t}(z_0) = 0$. (We will later show that z_0 is fixed for all t). Fix $e^\star \parallel z_0$. Now, let $d^\star$ be such that, in L_A^t, $d^\star$ is the highest point incomparable to $e^\star$. We also define $m(e^\star)$ to be the number of points greater than $e^\star$ in $\mathcal{P}$ but below $d^\star$ in L_A, and $k(e^\star)$ to be the number of points incomparable to $e^\star$ in $\mathcal{P}$ and below $e^\star$ in L_A. That is,

$$k(e^\star) = |\{v : v \parallel e^\star \text{ in } \mathcal{P} \text{ and } h_{L_A}(v) < h_{L_A}(e^\star)\}|$$

$$m(e^\star) = |\{u : u > e^\star \text{ in } \mathcal{P} \text{ and } h_{L_A}(u) < h_{L_A}(d^\star)\}|.$$

In addition to the rules from **GS**, a qualifying position using **GSA** for the t^{th} point z_t is the highest position that keeps $m(e^\star) \leq k(e^\star)$ for every applicable $e^\star$ while also minimizing $\text{pd}(z_t, \mathcal{P}, L_A^t)$ and $\text{ld}(\mathcal{P}, L_A^t)$. If a point $e \in \mathcal{P}$ is comparable to z_0 then this e will not be an $e^\star$. In other words the only candidates for $e^\star$ are points incomparable to z_0. For ease of understanding, the more explicit steps for determining the qualifying position of the t^{th} point z_t using **GSA** are:

1. Determine the legal positions for z_t.
2. Exclude any position such that placing z_t in that position would make $m(e^\star) > k(e^\star)$ for at least one $e^\star$.
3. Exclude the legal positions that, assuming z_t were put in that position, would result in a non-minimal linear discrepancy for L_A^t.
4. For each of the remaining positions, exclude the positions that, assuming z_t were put in that position, would result in a non-minimal point discrepancy for z_t in L_A^t.

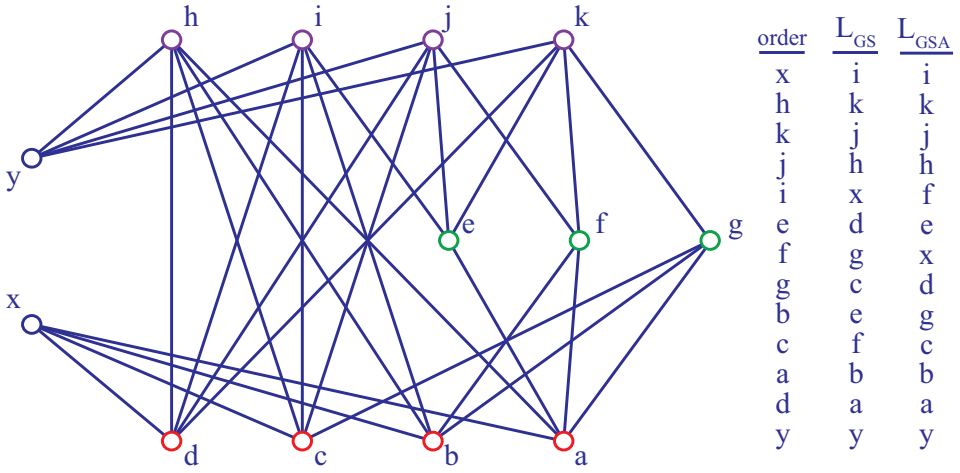
5. The qualifying position for z_t is the highest of the remaining positions.

It is also worth noting that our poset in Figure 10, where $l = 4$ can be generalized for all values of l. The sizes of A and C would be l, and the size of B would be $l - 1$. The pattern of comparabilities would also be continued to follow the pattern in the figure. Additionally, we can vary the size of B and consequently the sizes of A and C.

Example 3.8. Using **GSA** on the diagram and Builder order in Figure 10 we get the following linear extension of linear discrepancy 8:

$$y, a, b, c, g, d, x, e, f, h, j, k, i$$

The intricacies of the algorithm and the poset warrant a breakdown of the steps to get this up-growing on-line extension as $L_{\mathbf{GSA}}$. We begin by repeating the figure below paired with the Builder order, $L_{\mathbf{GS}}$ and $L_{\mathbf{GSA}}$.



order	L_{GS}	L_{GSA}
x	i	i
h	k	k
k	j	j
j	h	h
i	x	f
e	d	e
f	g	x
g	c	d
b	e	g
c	f	c
a	b	b
d	a	a
y	y	y

Figure 10. Diagram, builder order, $L_{\mathbf{GS}}$ and $L_{\mathbf{GSA}}$ for Example 3.8.

The first five points $\{y, d, a, c, b\}$ are an antichain. Thus, we simply perform **GS** on them to get y, a, b, c, d.

The introduction of g forces us to consider the values of $m(e^\star)$ and $k(e^\star)$ for $e^\star = a, b, c$. We get the values in Figure 11. Coincidentally, $d = d^\star$.

	$m(a)$	$k(a)$	$m(b)$	$k(b)$	$m(c)$	$k(c)$
g placed above $d = d^\star$	0	1	0	2	0	3
g placed below $d = d^\star$	1	1	1	2	1	3

Figure 11. Values for $m(e^\star)$ and $k(e^\star)$ when placing g in $L_{\mathbf{GSA}}$.

Since g can be placed either above or below d, we follow the **GS** steps of the algorithm to place g below d to yield y, a, b, c, g, d.

Next, the point f also affects the values of $m(e^\star)$ and $k(e^\star)$ for $e^\star = a, b, c$ and $d = d^\star$. We get the below values in Figure 12.

	$m(a)$	$k(a)$	$m(b)$	$k(b)$	$m(c)$	$k(c)$
f placed above $d = d^\star$	1	1	1	2	1	3
f placed below $d = d^\star$	2	1	2	2	2	3

Figure 12. Values for $m(e^\star)$ and $k(e^\star)$ when placing f in $L_{\mathbf{GSA}}$.

Since the value for $m(a)$ when f is placed below d is greater than $k(a)$, we must put f above d. When considering the point e going below or above d results in the same values of $m(e^\star)$ and $k(e^\star)$ as f does in Figure 12. Thus, we place e between d and f in $L_{\mathbf{GSA}}$ to get y, a, b, c, g, d, e, f.

The point i goes either above or below f due to the comparability to e. This means the relevant candidates for $e^\star$ are c and d. Our on-line linear extension has two points incomparable to c and d below c and d. Hence, i can go either above or below f based on the aware criteria. Thus, we perform the greedy and selfish steps to put i above f to get $y, a, b, c, g, d, e, f, i$.

Introducing j affects the values for $m(e)$ and $m(f)$, but not the points $\{a, b, c, d\}$ nor the point g because $j \parallel g$ and k has not been given yet. The value for $k(f)$ and $k(e)$ are both 5, which gives $m(e^\star)$ lots of room. Since $m(f)$ and $m(e)$ stays below 2 in either legal position of j, we decide that j follows the rules of the greedy and selfish steps to go below i to yield $y, a, b, c, g, d, e, f, j, i$.

The values for $m(e^\star)$ and $k(e^\star)$ where $e^\star = g, f, e$ when placing the point k are shown in Figure 13. This means that k can be placed either above or below i. The greedy and selfish steps dictate that k go below i. Additionally, h does not effect the values of $m(e^\star)$ and $k(e^\star)$. Thus, we get the linear extension $y, a, b, c, g, d, e, f, h, j, k, i$.

	$m(g)$	$k(g)$	$m(f)$	$k(f)$	$m(e)$	$k(e)$
k placed above $i = d^\star$	0	1	0	5	0	5
k placed below $i = d^\star$	1	1	1	5	1	5
k placed 2 below $i = d^\star$	2	1	2	5	2	5

Figure 13. Values for m and k when placing k in $L_{\mathbf{GSA}}$.

Finally, when we place the point x we only consider the greedy and selfish steps, since the placement of x cannot introduce any $k(e^\star)$ value. Thus, we complete the linear extension to get

$$y, a, b, c, g, d, x, e, f, h, j, k, i.$$

CHAPTER 4. MAXIMUM LINEAR DISCREPANCY ON TRIPLE OPTIMALS

4.1. Review of Some Vital Definitions

We assume that there is some underlying triple-optimal poset $\mathcal{P}$ with optimal linear discrepancy l and ground set X. A reminder of the notation for a few important linear extensions:

L_A is the linear extension created by the Algorithm and Builder
L_A^t is the linear extension created after the t^{th} point is given while playing the on-line game
L_O is a linear extension with the optimal linear discrepancy
L_W is a linear extension with the worst linear discrepancy

We will also define six sets below that rely on the two points x and y, which are the points that lead to the linear discrepancy l in L_O and $3l$ in L_W. We know this pair x and y exhibits this property based on the minimum sizes of A and C (as defined below) and the required locations of x and y in L_O to satisfy a linear discrepancy of l. Since x is incomparable to at least l points, x cannot be placed farther than l away from the top point. Similarly, y cannot be placed farther than l away from the bottom point due to the size of C. Thus, altogether, x and y will be l away from each other in L_O.

$$
\begin{aligned}
A &= \{a : a < x \text{ and } a \parallel y\} \\
B &= \{b : b \parallel x \text{ and } b \parallel y\} \\
C &= \{c : c > y \text{ and } c \parallel x\} \\
D &= \{d : d \text{ is above } x \text{ in } L_O\} \\
E &= \{e : e \text{ is between } x \text{ and } y \text{ in } L_O\} \\
F &= \{f : f \text{ is below } y \text{ in } L_O\}
\end{aligned}
$$

We also note that $D \subseteq C$, $E \subseteq A \cup (B \cup C)$, and $F \subseteq A$.

4.2. A Lemma and Corollary First

Lemma 4.1. *The point y in a triple-optimal poset $\mathcal{P}$ will be forced to the bottom of L_A by the Builder restricted to up-growing.*

Proof. Suppose that the sets A, B and C are chains in themselves. (Recall that chain is defined in Section 1.2 and the poset is similar to the poset on the left in Figure 3.) Since y is incomparable to $2l$ points in $A \cup B \cup \{x\}$, placing y in a non-bottom

position would result in a linear discrepancy less than $2l$. Thus, our goal of reaching $2l$ is defeated. Additionally, putting y above some a will not cause the difference in height between a and some c to get $2l$ apart.

We also note that due to the algorithm qualifying the highest position and only allowing Builder to use up-growing orders, no point will be placed below the bottom point.

□

Lemma 4.2. *For a triple-optimal poset $\mathcal{P}$ with optimal linear discrepancy l, regardless of the up-growing order in which the Builder gives points to the Algorithm using* **GSA**, *all $c \in C = \{c : c > y \text{ and } c \parallel x\}$ will always be above $a \in A = \{a : a < x \text{ and } a \parallel y\}$ and $b \in B = \{b : b \parallel x \text{ and } b \parallel y\}$ in the linear extension created by playing the on-line game L_A.*

Proof. Suppose a $c \in C$ is given before a $b \in B$. When the Builder gives one of the remaining elements of B to the Algorithm, **GSA** assigns that b below the existing c in L_A because $h_{L_A}(y) = 0$ and $b \parallel y$. The same logic applies when a c is given after a b, keeping the c above the b.

Since, the placement of elements of B below elements of C does not alter the inequality $m(e^\star) < k(e^\star)$, **GSA** does not derail this lemma. Thus, the greedy aspect of **GSA** will be the only part that affects this part of the poset.

□

Corollary 4.3. *The Builder gains no advantage by giving any element of B before an element of C.*

4.3. The Big Theorem

Theorem 4.4. *Let $\mathcal{P}$ be an l-critical triple-optimal poset. A Builder, when restricted to up-growing, cannot force an on-line linear extension with linear discrepancy greater than $2l$ on $\mathcal{P}$.*

Proof. We will prove this with two parts. We first show that the incomparabilities and comparabilities of a given point are restricted as a result of the triple-optimal property. Then, we show the cleverness of the algorithm will also not force any two points farther than $2l$ apart based on results from part 1 and the said cleverness of the algorithm.

Part 1:

Let z_0 be the point such that $h_{L_A}(z_0) = 0$. Fix $e^\star \in E$. Let i be the number of $f \in F$ such that $e^\star > f$ and let j be the number of $d \in D$ be such that $e^\star \parallel d$. In notation we write,

$$i = |\{v : v \parallel z_0 \text{ in } \mathcal{P} \text{ and } v < e^\star \text{ in } \mathcal{P}\}|$$

$$j = |\{u : u > z_0 \text{ in } \mathcal{P} \text{ and } u \parallel e^\star \text{ in } \mathcal{P}\}|.$$

Let $d^\star$ be the d in D such that, $d^\star$ is the highest point in L_A^t incomparable to $e^\star$.

We claim that $j \leq i$. Since $e^\star$ is greater than i elements of F, $e^\star$ is incomparable to $l-i$ elements of F because of the positioning of $e^\star$ and y in L_O. Likewise, since $e^\star$ is incomparable to j elements of D, $e^\star$ is less than $l-j$ elements of D.

Now, suppose $j > i$. This would mean that we would consider the options for constructing L_O pictured in Figure 14.

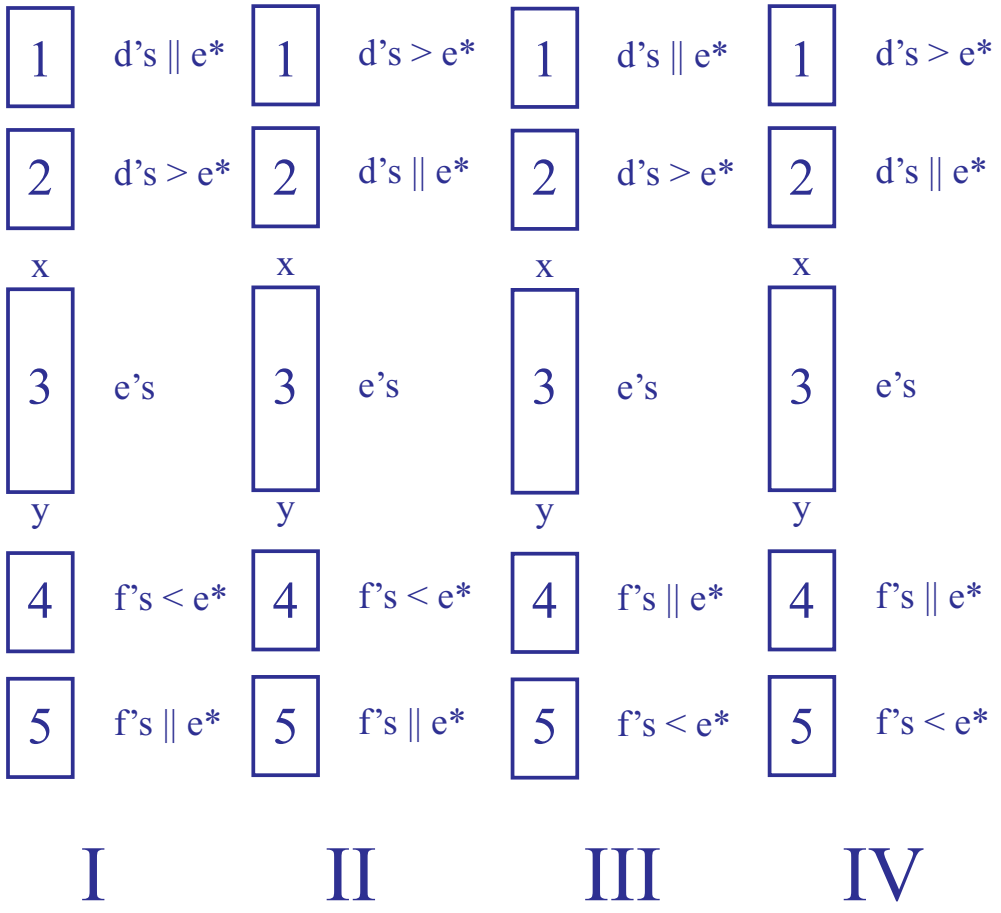


Figure 14. Possible L_O's

Clearly, Scenarios I, II and III in Figure 14 create non-optimal linear extensions. The difference between the highest and lowest points incomparable to $e^\star$ in these scenarios is greater than $2l$. In any legal position for $e^\star$, the distance between $e^\star$ and one of these points will be greater than l. Scenario IV also has a linear discrepancy greater than l. Counting the points between the top point in Grouping 2, d_j, and the bottom point in Grouping 4, f_{i+1}, we are put in a terrible position. Here is the counting of points:

Grouping 2 has j points
Grouping 3 has $l-1$ points
Grouping 4 has $l-i$ points
x is 1 point
y is 1 point

This simplifies to $2l+j-i+1$, but since $j > i$ this total count is greater than $2l+1$ which means $e^\star$ causes a linear discrepancy greater than l in L_O.

Part 2:
Keeping $e^\star$ and $d^\star$ as they were in Part 1, we will now define:

$$w = |\{w_p : w_p \parallel e^\star \text{ in } \mathcal{P} \text{ and } h_{L_A}(w_p) < h_{L_A}(e^\star)\}|$$

$$v = |\{v_q : v_q > e^\star \text{ in } \mathcal{P} \text{ and } h_{L_A}(v_q) < h_{L_A}(d^\star)\}|.$$

We claim that $i + w \geq j + v$.
Since $i \geq j$, we know that $i+w \geq j+w$. Remember that by **GSA** $k(e^\star) \geq m(e^\star)$. We note that $w = k(e^\star)$ and $v = m(e^\star)$, meaning we know that $w \geq v$. Thus, $i + w \geq j + v$.

Thus, when we count the points between $e^\star$ and $d^\star$ in L_A, we get:

$$\begin{aligned} j+v &: \text{ the number of points in } D \text{ below and including } d^\star \\ 1 &: \ x \\ l-2 &: \text{ the number of points in } E \text{ except } e^\star \\ l-(i+w) &: \text{ the number of relevant points in } F \end{aligned}$$

The number of points in these four sets is thus

$$2l + (j+v) - (i+w) - 1.$$

Since $i + w \geq j + v$,

$$2l + (j+v) - (i+w) - 1 \leq 2l - 1.$$

This means the difference between $h_{L_A}(e^\star)$ and $h_{L_A}(d^\star)$ is at most $2l-1$, which will be the linear discrepancy of L_A should this be the pair of points with the greatest distance.

The difference between $h_{L_A}(x)$ and $h_{L_A}(y)$ will be less than $2l$ because x will be put, in the worst case scenario, right above the highest element of E due to the selfish and greedy portions of **GSA**. □

4.4. Conclusion

The concepts we cover in this thesis have a wide range of applications: missile defense funding (forming fair linear extensions), ER callback schedule (on-line linear extension creation), salary computation (forming fair linear extensions) and many more. We look at the game restricted to up-growing, such that a better bound can be found for an algorithm. The algorithm we developed successfully limits the upper bound of the on-line linear discrepancy to double the optimal linear discrepancy as opposed to one less than triple the optimal linear discrepancy as seen in the not up-growing case. We accomplish this by incorporating the point discrepancy of each new point as well as trackable characteristic of the poset (keeping $m(e^\star) \leq k(e^\star)$). A natural progression from this thesis would be extending or adapting **GSA** to a larger class of posets, ideally to establish an upper bound of $2l$ for all up-growing posets of linear discrepancy l.

REFERENCES

[1] Stefan Felsner, Kamil Kloch, Grzegorz Matecki, and Piotr Micek, *On-line chain partitions of up-growing semi-orders*, Order **30** (2013), no. 1, 85–101.

[2] Mitchel T. Keller, Noah Streib, and William T. Trotter, *Online linear discrepancy of partially ordered sets*, An irregular mind, Bolyai Soc. Math. Stud., vol. 21, János Bolyai Math. Soc., Budapest, 2010, pp. 343–357.

[3] Hal A. Kierstead, *Online algorithms: The state of the art*, vol. 1442, ch. Coloring graphs on-line, pp. 281–305, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[4] Ton Kloks, Dieter Kratsch, and Haiko Müller, *Approximating the bandwidth for asteroidal triple-free graphs*, Journal of Algorithms **32** (1999), no. 1, 41–57.

[5] Paul J. Tanenbaum, Ann N. Trenk, and Peter C. Fishburn, *Linear discrepancy and weak discrepancy of partially ordered sets*, Order **18** (2001), no. 3, 201–225.