The Tverberg Number of Small Trees

By

DOMINIC YANG

SENIOR THESIS

Submitted in partial satisfaction of the requirements for Highest Honors for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS

in the

COLLEGE OF LETTERS AND SCIENCE

of the

UNIVERSITY OF CALIFORNIA,

DAVIS

Approved:

Jesús A. De Loera

June 2018

ABSTRACT.

In 1966, Helge Tverberg proved that given a number of classes m and a dimension d, there is a minimum constant $N_T(K_m, d)$ where K_m is the complete graph on m nodes, such that for any point configuration, there is a partition of the configuration into m subsets whose convex hulls have nonempty intersection.

Put another way, we can say that the intersection graph of this particular partition of points is the complete graph. In this thesis, I consider the generalization of Tverberg's Theorem to trees with small number of vertices. I present a computational algorithm which, by enumeration of all order types of small point sets, definitively provides the minimal number of points necessary for this particular graph.

In this senior thesis, I will present the details of my work with Professor Jesús De Loera, and discuss the peculiarities of the various graphs of small numbers of vertices. In particular, I will provide exact values or bounds on the Tverberg numbers for all trees on 5 or fewer vertices.

Contents

CHAPTER 1

Introduction

In this thesis, we will examine a problem from discrete geometry which describes the interaction between the convex hulls of the partitions of finite point sets in the plane. This is a problem which has been of interest since Helge Tverberg proved that when given enough points in the plane $(3n-2)$, it is always possible to partition the points into n classes such that the convex hull of every pair of sets intersects.

To better characterize the nature of the intersections between convex hulls of constituent sets of a partition, we will introduce the notion of the intersection graph which generalizes the concepts which Tverberg studied. Then, after reviewing basic topics from convex geometry, we will establish the main topic of interest, the Tverberg number of a graph, which will state the minimum number of points needed to generate this intersection graph. Later on, we will introduce a computational method by which we can then explicitly enumerate all point sets to establish the Tverberg numbers for small graphs.

1.1. Basic Definitions

To more formally establish this concept, we review some definitions from convex geometry.

DEFINITION 1. A set $K \subset \mathbb{R}^n$ is **convex** if and only if for any $x_1, x_2 \in K$, we have that $(1-\lambda)x_1 + x_2 \in K$.

From this notion we define the convex hull:

DEFINITION 2. Given a set $A \in \mathbb{R}^n$, the **convex hull** of A is defined as

$$conv(A) = \{\lambda_1 x_1 + \ldots + \lambda_k x_k | x_1, \ldots, x_k \in A, \sum_{i=1}^{k} \lambda_i = 1, \lambda_i \geq 0, i = 1, \ldots, k\}.$$

Equivalently, the convex hull is the smallest convex set which contains A.

Central to our study of intersection graphs is the partition which we define here.

DEFINITION 3. Let $X \in \mathbb{R}^d$ be a set of points. A set $A = \{A_1, \ldots, A_n\}$ is a **partition** of X if we have that

(1) For $i = 1, \ldots, n$, $A_i \subset X$.
(2) For $i \neq j$, $A_i \cap A_j = \emptyset$, i.e., the sets are disjoint.
(3) $\bigcup_{i=1}^{n} A_i = X$, i.e., every point in X is included in some A_i.

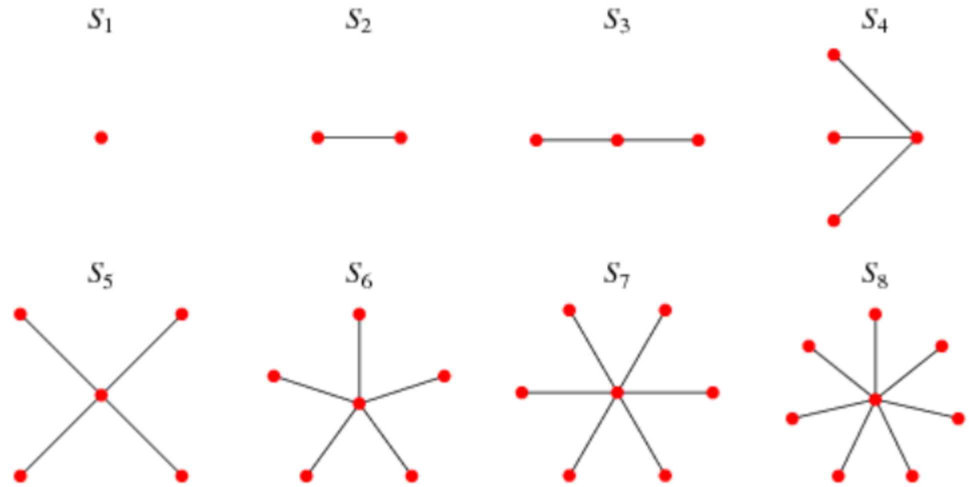

FIGURE 1. The First 8 Star Graphs

In addition, we introduce these concepts which will be of use later on.

DEFINITION 4. A finite set of n points are in **general position** if no three points are collinear. They are in **convex position** if no point is in the interior of the convex hull of all the points, equivalently if the points form the vertices of a convex polygon.

For our purposes, we will only consider sets of points in general position.

We also review some definitions from graph theory which will be referred to throughout the thesis.

DEFINITION 5. Let $G = (V, E)$ be a graph. A sequence of vertices $v_1, \dots, v_n$ is a **path** if v_i is adjacent to v_{i+1} for all $i = 1, \dots, n-1$. It is a **cycle** if $v_1 = v_n$.

Now we define some special types of graphs which will be studied later on.

DEFINITION 6. A **tree** is a connected graph without any cycles.

DEFINITION 7. The **star graph** S_n of order n is the tree on n vertices with 1 "central" vertex having degree $n-1$ and the remaining $n-1$ vertices having degree 1.

For examples of star graphs, see Figure 1.

DEFINITION 8. The **path graph** P_n is a tree with two vertices of degree 1 and the remaining $n-2$ vertices of degree 2.

For examples of path graphs, see Figure 2.

1.2. Intersection graphs

Given a set of n points $X = \{x_1, \dots, x_n\}$ and a partition of X into m classes, we can consider geometrically how each of the components of the partition interact, in particular

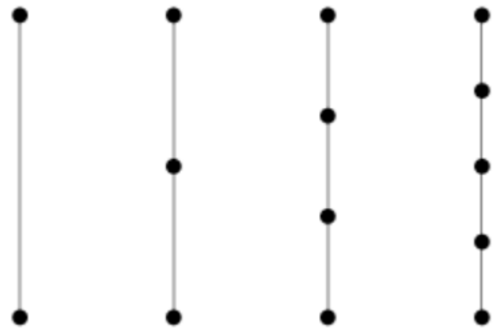

FIGURE 2. The First 4 Path Graphs

whether two given sets intersect. To study this behavior, we define the notion of an intersection graph:

DEFINITION 9. Let $X = \{x_1, \ldots, x_n\}$ be a finite set of n points and $A = \{X_1, \ldots, X_m\}$ be a partitioning of the points into m classes. Let G be a graph on m vertices $v_1, \ldots, v_m$. We say that G is the **intersection graph** of the partition A on X if for $i \neq j$, we have that v_i is adjacent to v_j if and only if $conv(X_i) \cap conv(X_j) \neq \emptyset$, i.e., the convex hull of the respective sets intersect. We denote the intersection graph of a given set of points X and a partition A on X by $IG(X, A)$.

For an example of a point configuration in the plane, a partition on the points, and the intersection graph of the partition, observe the colored point set and its induced graph in Figure 3. We plot the convex hulls of each set to better demonstrate the intersection of the sets. Note how the presence of the singleton set that is disjoint from the convex hulls of the other sets leads to a disconnected graph.

1.3. Early results

The study of these types of partitions has led to some results regarding the minimal number of points to be guaranteed a partition inducing certain intersection graphs. The simplest of these results comes from Johann Radon.

THEOREM 1. ***(Radon's Theorem)****: Any set of $d+2$ points in $\mathbb{R}^d$ can be partitioned into two disjoint sets whose convex hulls intersect.*

PROOF. Let A be a set of $d+2$ points in $\mathbb{R}^d$. For $a \in A$, we let $\begin{pmatrix} a \\ 1 \end{pmatrix}$ denote the vector with the first d components equal to those of a and the $d+1$-th component equal to 1. If we consider the set $\left\{ \begin{pmatrix} a \\ 1 \end{pmatrix} | a \in A \right\}$, we note that this is a set of $d+2$ vectors in $\mathbb{R}^{d+1}$

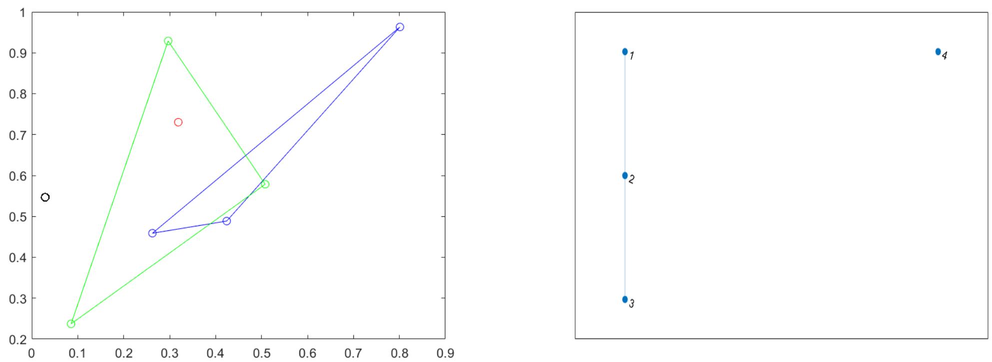

FIGURE 3. A point configuration of 8 points with a given coloring and its corresponding intersection graph. In the intersection graph, vertex 1 corresponds to the blue points, vertex 2 to the green points, vertex 3 to the red point, and vertex 4 to the black point.

meaning the set is not linearly independent. Hence, there exists α_i not all 0 such that

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \sum_{i=1}^{d+2} \alpha_i \begin{pmatrix} a_i \\ 1 \end{pmatrix}.$$

Equality of the last component of the vectors tells us that $\sum \alpha_i = 0$ which implies that there exist positive and negative values of α_i. Let I be the set of indices i for which α_i is positive and J the set of indices j where α_j is negative. Then we can write

$$\sum_{i \in I} \alpha_i \begin{pmatrix} a_i \\ 1 \end{pmatrix} = \sum_{i \in J} (-\alpha_i) \begin{pmatrix} a_i \\ 1 \end{pmatrix}.$$

Then let $t = \sum_{i \in I} \alpha_i = \sum_{i \in J}(-\alpha_i)$ and dividing both sides by t, we can write

$$z := \sum_{i \in I} \frac{\alpha_i}{t} a_i = \sum_{i \in J} \frac{-\alpha_i}{t} a_i.$$

If we choose our partition as $X = \{a_i, i \in I\}$ and $Y = A \setminus X$, we see that the left hand side of the above is in $\operatorname{conv}(X)$ as by definition of t, $\sum \frac{\alpha_i}{t} = 1$, and similarly the right hand side is in $\operatorname{conv}(Y)$. Letting $z = \sum_{i \in I} \frac{\alpha_i}{t} a_i = \sum_{i \in J} \frac{-\alpha_i}{t} a_i$, we see z is a point which is in the intersection of the convex hulls of partitioned sets and so we have a partition with nonempty intersection.

□

In particular, in 2 dimensions only 4 points are needed to have a guaranteed partition such that the convex hulls of a partition into two sets intersect. The two cases which may occur are shown in Figure 4. In terms of intersection graphs, we can say that the

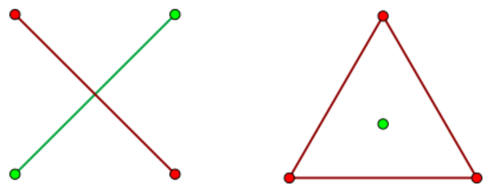

FIGURE 4. Two Radon Partitions of 4 Points

connected graph on two vertices can always be induced by a set of at least $d+2$ points in d dimensions.

Bryan John Birch considered a generalization of this result which showed that for any $3N$ points in the plane, there is a way to partition the points into N 3-point sets such that all the triangles have a point in common [**8**]. He later was able to strengthen this result and tighten the bound by showing that any $3N-2$ in the plane can be partitioned into N sets whose convex hulls each intersect. Birch conjectured the following result in higher dimensions which was later proven by Helge Tverberg in 1966 [**7**]:

THEOREM 2. ***(Tverberg's Theorem)****: Let $d \geq 1, k \geq 2$. For any set of N points, $N \geq (d+1)(k-1)+1$, there exists a partition of the points into k subsets such that all the convex hulls of the subsets intersect at a single point.*

In terms of intersection graphs, we can understand this as stating that the complete graph on k vertices is always possible to create with $(d+1)(k-1)+1$ points in $\mathbb{R}^d$. Note that this reduces to Birch's amount when considering points in the plane.

1.4. Tverberg numbers

We now define the central motivation for the thesis.

DEFINITION 10. Let G be any simple graph. We define the **Tverberg number** $N_T(G,d)$ of this graph to be the minimal number of points in $\mathbb{R}^d$ such that for all $n \geq N_T(G,d)$, for any set X of n points in general position, there exists a partition A on X such that $IG(X,A) = G$. If no such number exists then the Tverberg number of the graph is not defined.

In this thesis, we are only considering points in the plane, and so any reference to a Tverberg number for a given graph G will refer to $N_T(G,2)$ which will be written $N_T(G)$.

From this definition, we immediately have the following two results, the first from Radon's Theorem, and the second from Tverberg's Theorem:

LEMMA 1. *The Tverberg number of the path on two vertices is 4.*

LEMMA 2. *The Tverberg number of the complete graph on n vertices is $3n-2$.*

We note that it may be possible that for a certain graph, for any point configuration on n points, there exists a partition which induces the intersection graph. However, for

$m > n$, there may exist a point configuration for which the graph is impossible to induce with any partition. For these graphs we would have $N_T(G) > n$. We do not know of the existence of such graphs, but we do not doubt their existence.

Finally, we allude to the existence of graphs which are impossible to induce by any partition on any point set. There are various examples for which no point configuration can induce these particular graphs. Theorem 1.1 in [**6**] implies the existence of such graphs. There also exist graphs for which certain point configurations are guaranteed not to be able to induce these graphs. These also do not have well-defined Tverberg numbers, although perhaps under a restricted class of point configurations, these would realize a finite number.

1.5. Bounds on Tverberg numbers

For certain simple graphs, some results have been determined which can provide a bound on the Tverberg number.

In particular, for any given tree on n vertices, we have the following result from Professor Jesús De Loera, Tommy Hogan, and Professor Deborah Oliveros:

THEOREM 3. *Given a tree G on n vertices. Then if we have $2n$ points in $\mathbb{R}^2$ in convex position, there exists a partition of the points which will induce the tree G. Furthermore, for any other point in $\mathbb{R}^2$, there exists a subset in the partition which the point can be added to without changing the intersection graph.*

With this result, one only needs enough points to guarantee that $2n$ points are in convex position. From Erdös and Szekeres [**4**], there is a number $f(n)$ for any given n which ensures the existence of a subset of the points with n points in convex position. These results together imply that the Tverberg number for any tree is less than $f(2n)$. Unfortunately, this bound is very weak as $f(n) \geq 1 + 2^{n-2}$ [**4**].

CHAPTER 2

Enumeration of point sets

In this section, we present our method for enumeration of all possible point configurations to determine the Tverberg Numbers of given small graphs.

2.1. Order types

We are able to find lower bounds for the Tverberg Number of small graphs via an enumeration of all point sets, and determining for each point set which partitions can be induced. Obviously, as there are infinitely many point configurations in the plane, an explicit enumeration is impossible; however, there is a way to classify the combinatorial properties of small point sets (whether line segments through points cross, possible triangulations, etc.) into a finite number of point configurations. One way of classifying point configurations is through the **order type** of a set.

DEFINITION 11. The **order type** of a set of points in general position $\{x1, \ldots, x_n\}$ in $\mathbb{R}^2$ is mapping that assigns each ordered triple i, j, k in $\{1, \ldots, n\}$ the orientation of the points x_i, x_j, x_k. We then say that two point sets are equivalent if there is a bijection between the sets that preserves the order type.

For example, on four points, there are two order types, one with one point contained in the interior of the convex hull of the other three and the other with all four points in convex position. One of these configurations is shown in Figure 1. Its order type can be computed as assigning the triples $(1,2,3), (1,3,4), (1,4,2), (2,3,1), (2,1,4), (2,3,4), (3,4,2), (3,1,2), (3,4,1), (4,2,1), (4,2,3)$, and $(4,1,3)$ to -1 for counter-clockwise orientation and the remaining ordered triples to 1 for clockwise orientation.

The order type encodes various combinatorial properties of point sets. If two point sets have the same order type, then if two segments cross in one set, their corresponding segments in the other order set should cross. If we have a valid triangulation in one set, it remains valid when mapped onto the other set. More importantly for our purposes, if two point sets have the same order type, then the intersection graphs induced by the two sets remain the same. This can be observed by the following proof from Tommy Hogan. This proof uses Carathéodory's theorem which we state here as well.

THEOREM 4. *(Carathéodory's Theorem): If a point $x \in \mathbb{R}^n$ is in the convex hull of a set P, then x can be written as a convex combination of at most $n + 1$ points of P.*

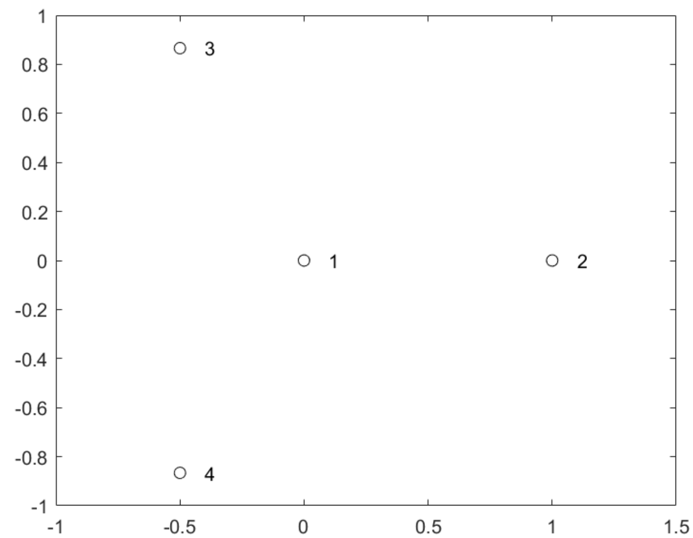

FIGURE 1. This point configuration has one of two order types on four points

Now we present the proof that the intersection graph on a given partition for an order type is the same for all point sets of that order type.

PROPOSITION 1. *Suppose S_1 and S_2 are planar sets in general position with the same order type, and let σ be a bijection from S_1 to S_2 that preserves the orientation of any triple in S_1. Then any partition of S_1 and the corresponding partition of S_2 via σ have the same intersection graph.*

PROOF. It suffices to show that if $\mathrm{conv}(x_1, \ldots, x_n) \cap \mathrm{conv}(y_1, \ldots, y_m) \neq \emptyset$ for some $x_1, \ldots, x_n, y_1, \ldots, y_m \in S_1$ then $\mathrm{conv}(\sigma(x_1), \ldots, \sigma(x_n)) \cap \mathrm{conv}(\sigma(y_1), \ldots, \sigma(y_m)) \neq \emptyset$. By Carathéodory's theorem we may assume (after taking subsets) that $n = m = 3$. Then $\mathrm{conv}(x_1, x_2, x_3)$ and $\mathrm{conv}(y_1, y_2, y_3)$ are intersecting triangles. If two edges of the triangles intersect the corresponding edges in $\mathrm{conv}(\sigma(x_1), \sigma(x_2), \sigma(x_3))$ and $\mathrm{conv}(\sigma(y_1), \sigma(y_2), \sigma(y_3))$ also intersect. Otherwise one triangle contains the other in it's strict interior. This situation is equivalent to the following: for each edge of the exterior triangle all four of the remaining points in $x_1, x_2, x_3, y_1, y_2, y_3$ lie on the same side of that edge. This property is preserved under σ since two points c_1, c_2 are on the same side of an edge ab if and only if a, b, c_1 and a, b, c_2 have the same orientation. □

Oswin Aichholzer, et. al., have provided a catalog of representative point configurations for all order types up to $n = 10$ [**2**]. Their method of generating each order type is done by generating a list of candidates of "pseudo order types" and then group these candidates into equivalence classes based on their order types. Then, they realize an actual point set

Number of Points	Number of Sets
3	1
4	2
5	3
6	16
7	135
8	3315
9	158817
10	14309547
11	2334512907

TABLE 1. Number of Sets with a Given Order Type

for each possible class of order type until they reach the required number of order types, which is known from the literature. This catalog of point configurations can be found in [**1**].

They have used these point configurations to test small cases for determining questions of isomorphisms between triangulations and the number of triangulations as well as the crossing number of complete graphs and problems related to finding Hamiltonian cycles on complete graphs. As can be seen in Table 1, the number of order types grows exponentially, so enumeration of order types for $n > 10$ is out of the question (in part because Aichholzer has not provided so large a quantity in his database).

2.2. Partitions

Given a point set of n points we are interested in all the ways which we are able to partition this point set into m unlabeled subsets. The amount of ways in which we can partition this set in such a manner is known as the **Stirling number of the second kind** and we denote this value by $\left\{ {n \atop m} \right\}$. For example all the ways to partition the set of 4 elements $\{1, 2, 3\}$ into two parts are $\{\{1\}, \{2, 3\}\}, \{\{2\}, \{1, 3\}\}, \{\{3\}, \{1, 2\}\}$ and so $\left\{ {3 \atop 2} \right\} = 3$.

One simple property that can be derived regarding this quantity is that

$$\left\{ {n+1 \atop m} \right\} = m \left\{ {n \atop m} \right\} + \left\{ {n \atop m-1} \right\}.$$

To see this, we note that to count the number of ways to partition $n + 1$ objects, we have two choices for where to put the $n+1$-th object: we can place it in its own singleton set, in which case we have to partition the remaining n objects into $m-1$ sets, or we partition the n objects into k sets and then we have m choices for where to include the $n + 1$-th object. In the first case we have $\left\{ {n \atop m-1} \right\}$ partitions and in the second we have $m\left\{ {n \atop m} \right\}$ partitions. With the base cases $\left\{ {0 \atop 0} \right\} = 1$ and $\left\{ {n \atop 0} \right\} = \left\{ {0 \atop n} \right\} = 0$ for $n > 0$, we can define Stirling Numbers for all $n, m \geq 0$. We include a table of all the values up to $n = k = 10$ in Table 2.

TABLE 2. Stirling Numbers up to $n = m = 10$. Each row represents a value for n and each column a value for m

	0	1	2	3	4	5	6	7	8	9	10
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0	0	0
3	0	1	3	1	0	0	0	0	0	0	0
4	0	1	7	6	1	0	0	0	0	0	0
5	0	1	15	25	10	1	0	0	0	0	0
6	0	1	31	90	65	15	1	0	0	0	0
7	0	1	63	301	350	140	21	1	0	0	0
8	0	1	127	966	1701	1050	266	28	1	0	0
9	0	1	255	3025	7770	6951	2646	462	36	1	0
10	0	1	511	9330	34105	42525	22827	5880	750	45	1

As n grows large, it was shown in [**5**] that we have a rough estimate for fixed m that $\left\{{n \atop m}\right\} \sim \frac{m^n}{m!}$. This shows that the growth is exponential for fixed n, so we should expect that the explicit enumeration of partitions quickly becomes computationally intractable.

2.3. Enumerating partitions over each of the possible order types

Using the classification from [**2**] of all order types, it is relatively straight forward to determine all possible different graphs which can be induced by partitioning a given point set by brute force enumeration. It is just a matter of listing all possible partitions into k subsets and then determining the intersection graph from checking the intersection of convex hulls. An explicit algorithm for producing the intersection graph of a point set X and a partition σ into k sets is given in Figure 2.

We also present the method used for determining if two given convex sets intersect. The method used to compute the convex hull and for determining if a point is contained in a polygon or if polygons intersect are MATLAB routines.

An alternative implementation based on checking the feasibility of the linear program defined by adjoining the representations of the convex polygons by inequalities was also considered and implemented. This was deemed to run slower than the MATLAB functions for determining the intersection of polygon edges however.

Checking if a graph has a Tverberg Number of n now simply is a matter of enumerating all order types and checking for every single one if there is a single partition which induces that graph.

```
function IntersectionGraph(X,σ, k)
    Let A ∈ {0,1}^{k×k}
    for i = 1,...,k do
        for j = i,...,k do
            if i = j then
                A_{i,i} = 0
            else
                if Intersects(σ^{-1}(i), σ^{-1}(j)) then
                    A_{i,j} = A_{j,i} = 1
                else
                    A_{i,j} = A_{j,i} = 0
                end if
            end if
        end for
    end for
end function
```

FIGURE 2. Algorithm for Computing the intersection graph of a given partition σ of a point set X onto k subsets.

```
function Intersects(X, Y)
    C_X ← conv(X), C_Y ← conv(Y)
    if |C_X| < |C_Y| then
        Swap C_X and C_Y
    end if
    if |C_X| = 2 and |C_Y| = 2 then
        L_X ← Line(C_X), L_Y ← Line(C_Y)
        return SegmentIntersect(L_X, L_Y)
    else if |C_X| > 2 and |C_Y| = 1 then
        return InPolygon(C_Y, C_X)
    else
        return EdgeIntersect(C_Y, C_X) or InPolygon(C_Y, C_X) or InPolygon(C_X, C_Y)
    end if
end function
```

FIGURE 3. Algorithm for determining if the convex hulls of two given sets X and Y intersect

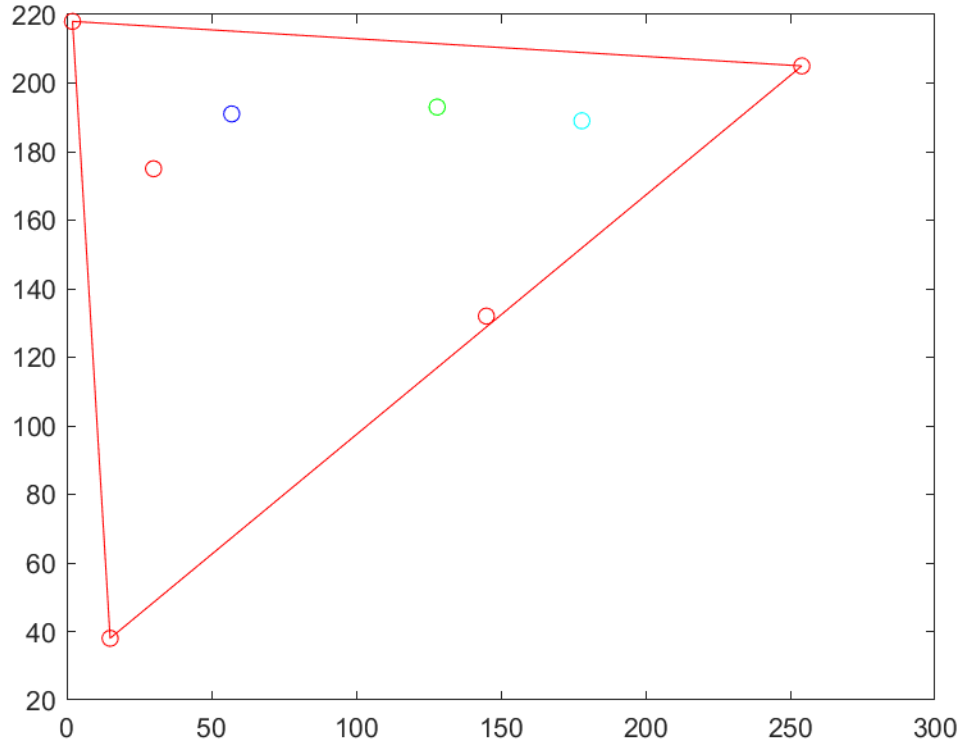

FIGURE 4. Way to Partition a Set to Induce the Graph S_4

2.4. Heuristics for decreasing enumeration time

Enumerating all possible order types is a gargantuan task so in the case of certain graphs, some heuristics can be applied to decrease the large amount of cases to check.

2.4.1. Eliminating order types. For some graphs, certain point configurations can be very easy to partition into subsets which induce the graph. On a given star graph S_n, if we are considering m points $\{x_i\}$ and if the convex hull of those points has $m - n + 1$ vertices, then we can immediately give a partition where we take any $n - 1$ points on the interior of the convex hull and create $n - 1$ different singleton sets $A_1, \ldots, A_{n-1}$ and have the rest of the points belong to one set A_n. We see that $conv(A_n) = conv(\{x_1, \ldots, x_n\})$ and so it intersects every singleton set, but no singleton set intersects any other, so we get S_n. See Figure 4 for an example of how this partition would appear on 8 points whose convex hull has 3 vertices.

With this fact in mind, we can first check if the convex hull of a set has fewer than a $m-n+1$ vertices and then we immediately have a partition. Performing this check allows for the elimination of over half of the order types to check.

2.4.2. Eliminating partition types. On certain graphs, we can further reduce the amount of work to be done by eliminating partitions for which it is impossible to actually induce that graph with. This allows for some minor reductions in the amount of cases to consider as well.

For example, suppose we are attempting to generate the path graph P_4 using 8 points. We can enumerate all the partitions of 8 points into 4 sets to get the following list:

$$\begin{aligned} 8 &= 5+1+1+1 \\ &= 4+2+1+1 \\ &= 3+3+1+1 \\ &= 3+2+2+1 \\ &= 2+2+2+2 \end{aligned}$$

We argue that we have the following result:

Proposition 2. *The only possible partitions of eight points which can induce P_4 are 3 + 3 + 1 + 1, 3 + 2 + 2 + 1, and 2 + 2 + 2 + 2.*

Proof. To eliminate certain partitions, we argue first that to induce a tree graph, any singleton set must be mapped to a vertex of degree 1, as it must be contained in the interior of another set. If it were mapped to a vertex of degree greater than 1, then that singleton set must be contained in convex hulls of two separate subsets in the partition which would then imply that the convex hulls of those two subsets intersect as well and our graph would have a 3-cycle as a subgraph contradicting that it is a tree. Since there are only two vertices of degree 1 in P_4, and three singleton sets in the partition 5 + 1 + 1 + 1, we can eliminate this possibility.

We can also eliminate the partition 4 + 2 + 1 + 1, as this would require in the path graph that a doubleton set would be adjacent to a singleton set. This means that the point in the singleton set would be located on the line segment connecting the points in the doubleton set which is not possible for points in general position. Therefore we can rule out this possibility.

The only remaining possibilities are 3 + 3 + 1 + 1, 3 + 2 + 2 + 1, and 2 + 2 + 2 + 2. $\square$

With these improvements, we can reduce the number of partitions to check by slightly under half in most cases.

CHAPTER 3

The Tverberg number for all trees with 5 or fewer vertices

In this chapter, we hope to enumerate the Tverberg number or a bound on the Tverberg Number for all trees with 5 or fewer vertices. In certain cases, this will be done by appealing to basic theorems, in others by giving an explicit construction, and others by exhaustion of the possibilities through enumeration of all possible order types.

3.1. A lower bound on $N_T(G)$

First we wish to give a lower bound on the Tverberg number $N_T(G)$ when G is a tree.

Proposition 3. *For a connected graph G on $n > 1$ vertices, we have that the Tverberg number $N_T(G) \geq 2n$.*

Proof. To prove this result, we simply need to show that for any number of points less than $2n$, we can come up with a point configuration for which inducing the graph is impossible.

Suppose we can only use m points with $m < 2n$. We will arrange these points in convex position. By the pigeonhole principle, if we partition our m points into n disjoint subsets, there must be at least one subset that is a singleton set. Since the graph is connected, the vertex representing this point is adjacent to at least one other vertex implying that this singleton is contained within the convex hull of another subset. However, this is a contradiction as we have put the points in convex position. Therefore we must have that $N_T(G) \geq 2n$. $\square$

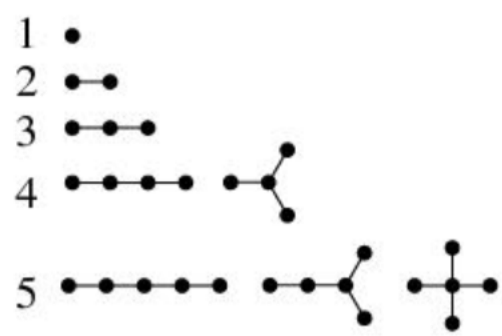

Figure 1. All trees on five vertices or less

In particular, this result must hold for any tree as well. We will find that this result is tight for certain trees and not tight for other trees.

3.2. Star Graphs

Using a simple argument based only on Radon's Theorem, we can arrive at an upper bound for star graphs.

PROPOSITION 4. $T(S_n) \leq 3n - 2$.

PROOF. We prove this by induction on n. For $n = 1$, the partition of 1 point to get S_1 is obvious.

Now assume the result is true for n. We need to show that it is possible to construct a partition for any $3n+1$ points. Given $3n+1$ points, we order them by their x-coordinate. By rotating the axes, we can assume that no two points have the same x-coordinate. We take the $3n-2$ points with smallest x-coordinates and using the induction hypothesis, we construct a partition $\{A_1, \ldots, A_n\}$ which induces S_n. Without loss of generality, assume that A_1 is the central vertex of the star graph.

Let $x \in A_1$ and let x_1, x_2, x_3 be the three remaining points with largest x-coordinate. By Radon's theorem there is a way to partition these four points into two sets with intersecting convex hulls. Let X, Y be such a partition and let $x \in X$. Y intersects $A_1 \cup X$ but does not intersect any of A_i, $2 \leq i \leq n$ as every point in Y has larger x coordinate than any point in A_i. Then we see $\{A_1 \cup X, A_2, \ldots, A_n, Y\}$ is a partition which will induce the graph S_n. $\square$

Using this argument, we have a constructive algorithm for producing a partition inducing S_n when given $3n-2$ points. An example partition on 19 points is shown in Figure 2.

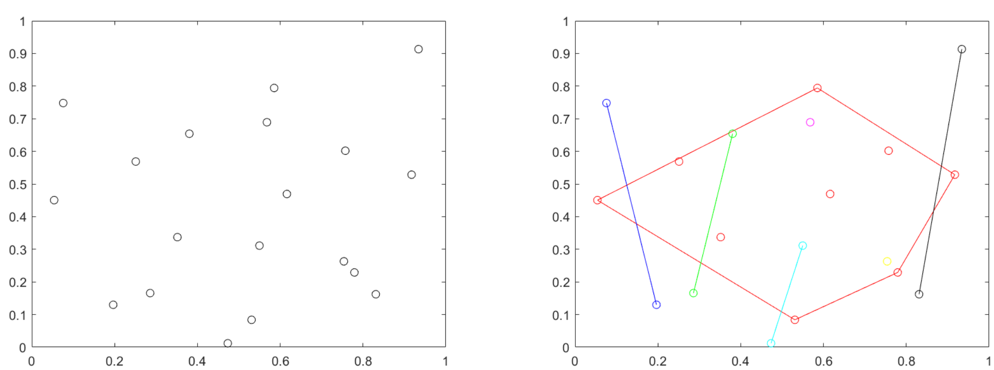

FIGURE 2. A partition of 19 points to give S_7

We can improve this bound to show that in fact $2n$ points is enough to be guaranteed a partition inducing S_n. We give an explicit construction of such a partition in the following proof.

PROPOSITION 5. *For any star graph S_n with $n > 1$, we have that $N_T(S_n) = 2n$.*

Prior to proving this statement, we will find the following lemma to be useful.

LEMMA 3. *Given $2n+2$ points $X \in \mathbb{R}^2$ in general position in the plane. Let ℓ_{pq} denote the line segment between points p and q. If there exists $p, q \in X$ that divide the remaining points into two sets A, B each of size n such that for $a \in A, b \in B$, we have that ℓ_{ab} intersects ℓ_{pq}, then it is possible to pair off elements $a_i \in A$, $b_i \in B$, such that for $i, j = 1, \ldots, n$, $i \neq j$, $\ell_{a_ib_i}$ does not intersect $\ell_{a_jb_j}$.*

PROOF. Suppose we have points p_1 and p_2 and partition the remaining points into A and B. Let ℓ be the line between p_1 and p_2. To pair off the points, we consider the vertices of $conv(A \cup B)$. Since ℓ separates the points of A and B, we must have that there are a pair of adjacent vertices of $conv(A \cup B)$ such that one, a_1, is a member of A and the other b_1, a member of B. The segment between this pair cannot intersect the segment between any other pair of points as this segment forms the boundary of the convex hull. We pair off these two points and then consider $conv(A \setminus \{a_1\} \cup B \setminus \{b_1\})$. We see that ℓ separates $A \setminus \{a_1\}$ and $B \setminus \{b_1\}$, so we can repeat this argument to pair off a_2 and b_2. Continuing in this fashion until we have paired off all the elements, we will have a pairing $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$ where $\ell_{a_ib_i}$ does not intersect $\ell_{a_jb_j}$ for $i \neq j$. □

Now we prove the proposition.

PROOF. Let $A \in \mathbb{R}^2$ be a collection of $2n$ points in general position in the plane. Our goal will be to find a pair of points which can separate the remaining points into two sets of equal sets so we can apply the above lemma. This will not always be possible, so we will try to make the size of the two sets as close as possible.

To do this, we will consider the vertices of the convex hull of A. We pick arbitrarily a vertex p_1 of $conv(A)$ and order the remaining vertices $p_2, \ldots, p_k$ in counter-clockwise order where k is the number of vertices. For $i = 2, \ldots, k$, we divide the remaining points of A into two sets B_i, C_i where B_i is the set of points in A to the left of $\ell_{p_1p_i}$ and C_i is the set of points to the right of $\ell_{p_1p_i}$. We note that the size of B_i decreases from $2n-2$ to 0 as i increases and the size of C_i increases from 0 to $2n-2$.

We consider two cases. The first case is that there exists i such that $|B_i| = |C_i| = n-1$ and then we can apply the above lemma as the line segment between every pair of points in $B_i \times C_i$ intersects $\ell_{p_1p_i}$ since $\ell_{p_1p_i}$ separates B_i and C_i. Then we have a pairing $(b_1, c_1), \ldots, (b_{n-1}, c_{n-1})$ where for any two pairs the segments do not intersect, but each intersects $\ell_{p_1p_i}$. Then the partition $\{\{b_1, c_1\}, \ldots, \{b_{n-1}, c_{n-1}\}, \{p_1, p_i\}\}$ is a partition which induces the star graph S_n. For an example of this case and how to partition the points, see Figure 3.

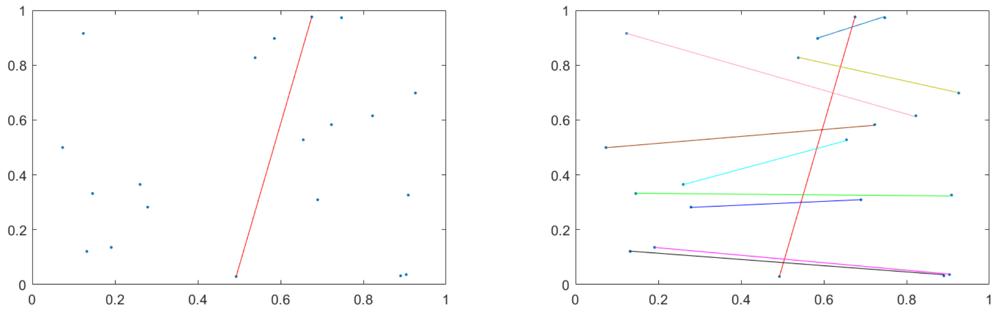

FIGURE 3. In the first case, there is a partition which divides the remaining points into two sets of equal size. Then we can pair off points such that the segment connecting them intersects the dividing line, but no other segment.

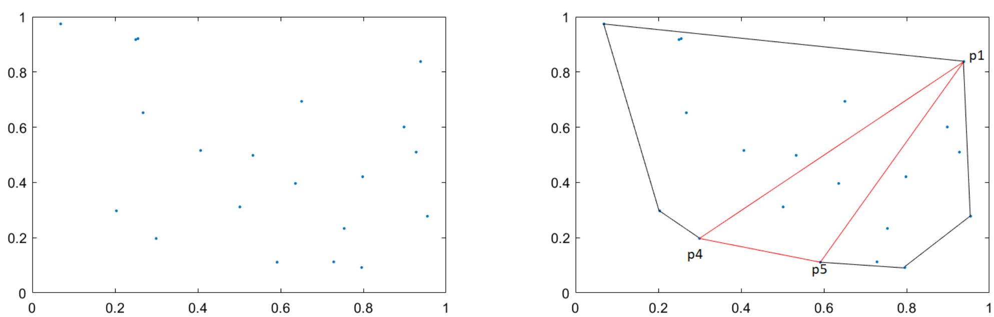

FIGURE 4. Finding a central dividing set of a given point configuration. In the right image, the right region indicates B_{i+1}, the central region represents I and the left region is C_i. We see $|B_{i+1}| = 7$, $|I| = 2$, and $|C_i| = 8$.

The second case is that there does not exist such an i. In this case, we find i such that $|B_i| > |C_i|$ and $|B_{i+1}| < |C_{i+1}|$. We will choose as a subset which will form the center vertex of our star graph to be $D = \{p_1, p_i, p_{i+1}\}$. See Figure 4 for a depiction of this central triangle.

To construct the remaining subsets in our partition, we first count the number of points in each of the B_i and C_i. If we let I denote the interior of $B_i \setminus B_{i+1}$, then we see that $|B_{i+1}| = |B_i| - |I| - 1$ and $|C_{i+1}| = |C_i| + |I| + 1$. This is because we are moving the points from the interior of $conv(\{p_1, p_i, p_{i+1}\})$ and the point p_i from B_i to C_{i+1}. Now we can pair off points from B_{i+1} and C_i to form disjoint segments which will intersect $conv(D)$ using the above lemma, and every point in the interior I can be a singleton set which will intersect $conv(D)$ but not any of the segments since the points are in general position.

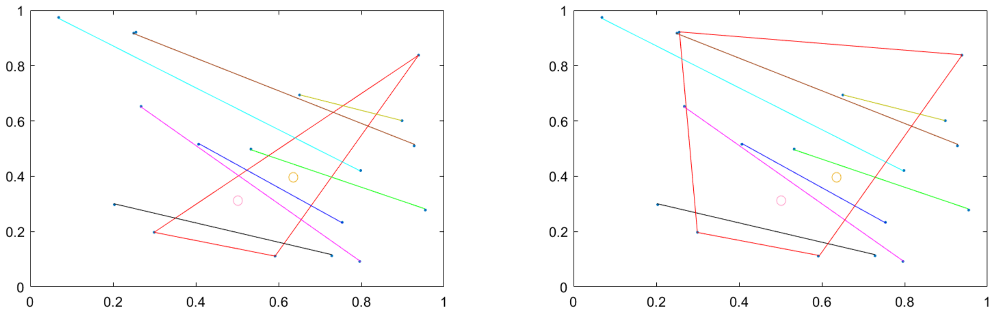

FIGURE 5. Coloring the points to induce the star partition. In the left image, we pair up points on either side of the triangle and add singleton sets on the interior to get 9 sets which intersect with the central triangle. However, there still remains one point not assigned which can be added to the central set without introducing new intersections.

If we let $m = \min(|B_{i+1}|, |C_i|)$, we pair off the points, $\{\{b_1, c_1\}, \ldots, \{b_m, c_m\}\}$ so that the segments between any two points do not intersect but each intersects the central triangle. Note that this set may be empty as one of $|B_{i+1}|$ and $|C_i|$ may be 0. Then to fill out the remaining subsets, we add singleton sets $\{\{x_1\}, \ldots \{x_{n-1-m}\}\}$ from the interior I. After constructing these sets, there may be points remaining Y, which are still unassigned. These can be added to the set D without introducing new intersections since D already intersects every set. Then our final partition is $\{D \cup Y, \{b_1, c_1\}, \ldots, \{b_m, c_m\}, \{x_1\}, \ldots, \{x_{n-1-m}\}\}$. For a depiction of this method of constructing the partition, see Figure 5. This procedure will work provided that there are enough points on the interior and on either side of the triangle to have $n-1$ intersecting sets. Therefore, if we can show that $m + |I| = \min(|B_{i+1}|, |C_i|) + |I| \geq n-1$, we will be done.

Note for any i, we have $|B_i| + |C_i| + 2 = 2n$ just by counting the points in each set. Then since $|B_{i+1}| = |B_i| - |I| - 1$ and $|B_i| > |C_i|$, we can write

$$|B_{i+1}| > |C_i| - |I| - 1 = 2n - 3 - |B_i| - |I|.$$

Substituting in again $|B_i| = |B_{i+1}| + |I| + 1$ and rearranging, we get

$$|B_{i+1}| > n - 2 - |I| \geq n - 1 - |I|.$$

Using that $|C_{i+1}| > |B_{i+1}|$, $|B_{i+1}| = |B_i| - |I| - 1$ and that $|C_{i+1}| = |C_i| + |I| + 1$ we get

$$|C_i| > |B_i| - 2|I| - 2.$$

Since $|B_i| + |C_i| + 2 = 2n$, we can write the above as

$$|C_i| > n - 2 - |I| \geq n - 1 - |I|.$$

Then we have that the number of intersections we can have is $\min(|B_{i+1}, |C_i|) + |I| \geq n-1$ which is exactly enough to form the star graph on n vertices. $\square$

3.3. Path Graphs

We can use same type of argument as used to derive an upper bound on the Tverberg Number of star graphs to get an upper bound on path graphs although a little more care is required to ensure that two sets do not intersect by mistake.

PROPOSITION 6. $N_T(P_n) \leq 3n-2$.

PROOF. As before we will proceed by induction, but we will use a slightly stronger induction hypothesis. For any $3n-2$ points we can partition the points to induce the path graph on n vertices. Let x_1, x_2, x_3 denote the points with the three largest x-coordinates, and let A_n and A_{n-1} denote the sets in the partition corresponding to the endpoint of the path and the vertex adjacent to the endpoint, respectively. Then $A_n \subset \{x_1, x_2, x_3\}$ and $\{x_1, x_2, x_3\} \subset A_{n-1} \cup A_n$.

For $n = 1$, the partition to induce P_1 is obvious. Now we assume our induction hypothesis is true for n. Given $3n+1$ points, we can take the $3n-2$ points with smallest x-coordinates and partition the points into sets $\{A_1, A_2, \ldots, A_n\}$ to have the path graph P_n. We label the sets so that they are in the same order as the vertices on the path, i.e., $A_1 \cap A_2 \neq \emptyset, \ldots, A_{n-1} \cap A_n \neq \emptyset$. By our induction hypothesis, of the $3n-2$ points, the points with three largest x-coordinates, are in either A_{n-1} or A_n with at least one of them, say $x \in A_n$. Then with the remaining 3 points and x, by Radon's Theorem, we can construct a partition into two sets X, Y so that $X \cap Y = \emptyset$. Assume $x \in X$.

Now we claim that the partition $\{A_1, \ldots, A_{n-1}, A_n \cup X, Y\}$ satisfies the claims of the induction hypothesis. By induction hypothesis, the first n sets induce P_n and since $X \cap Y \neq \emptyset$, we have that P_{n+1} is at least a subgraph of the intersection graph. We just need to check the addition of new points do not intersect any unwanted sets. Y only intersects $A_n \cup X$ as it is comprised of points from the three largest x-coordinates. Similarly, since $A_n \cup X$ is among the 6 points with the largest x-coordinates and of the 6 points, those not in $A_n \cup X$, are in the sets A_{n-1} or Y. So we can only have that $A_n \cup X$ intersects A_{n-1} and Y. Therefore we have a partition given P_{n+1}. □

This proof gives a constructive algorithm for finding a partition inducing P_n given $3n-2$ points. For an example of these constructions see Figure 6.

We can further generalize this argument to a special type of graph which is essentially a "path of stars."

DEFINITION 12. A **Star Path Graph** is a tree for which all vertices are degree 1 except for those on a given path of vertices.

Note that there may be multiple paths which serve as the path for a given star graph. For an example star path graph, see Figure 7.

PROPOSITION 7. *For any star path graph G on n vertices, $N_T(G) \leq 3n-2$.*

PROOF. We proceed by induction on the length of the path of the graph. We use a slightly stronger induction statement where if we order the points by x coordinate and v

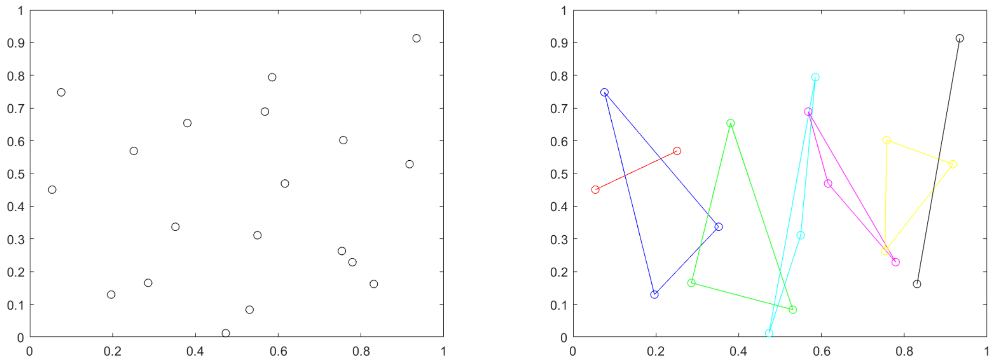

FIGURE 6. A partition of the same 19 points in Figure 2 to give P_7

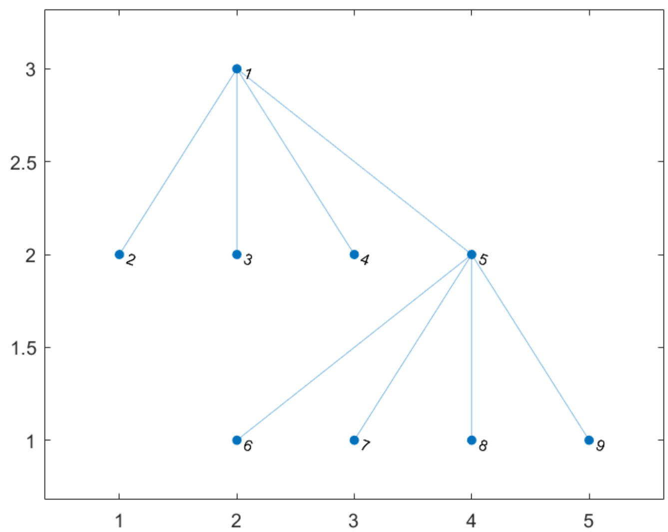

FIGURE 7. A star path graph on 9 vertices. One path that may act as a central path is $(1, 5)$

is the last vertex on the path and is adjacent to k vertices, then the last $3k + 3$ points are in the sets corresponding to v or the vertices adjacent to it and no points prior to that correspond to the set v. If the length is 1, we have just the star graph for which the result holds.

Assume the result holds if the path is of length m. We consider star path graphs which have paths of length $m + 1$. Let G be such a graph with n vertices. We consider the endpoint of the path v_{m+1} and the vertex prior v_m. If we consider the subgraph of G consisting of the path $v_1, \ldots, v_m$ and all vertices adjacent to it except v_{m+1}, this is a star path graph with a path of length m. Let p denote the number of vertices of this graph. By inductive hypothesis, we can represent this graph using $3p-2$ points.

Now given $3n-2$ points, construct the subgraph on the $3p-2$ points with the smallest x coordinates. We will have partition $\{A_1, \ldots, A_p\}$ where we take A_1 to be the set corresponding to v_m. Then take a point $x \in A_1$ and the next largest 3 points to have a Radon partition X, Y with $x \in X$. Our new partition will be $\{A_1 \cup X, \ldots, A_p, Y\}$. Y will correspond to the vertex v_{m+1} and will not intersect any of the other sets due to having larger x coordinates. In addition $A_1 \cup X$ will not intersect any new sets by how we have arranged the points due to the inductive hypothesis. Now as in the proof of Proposition 4, we can add new sets by considering 3 points in iteration for each of the other vertices adjacent to v_{m+1}. Since there were $n-p$ vertices and we used 3 points for each, in total we used $3p-2+3(n-p) = 3n-2$ points. This is the desired number. $\square$

3.4. Enumeration of trees

FIGURE 8. S_1, The only graph on one vertex

3.4.1. S_1. For completion's sake, we start with this obvious result.

PROPOSITION 8. $N_T(S_1) = 1$.

PROOF. For any number of points, there is only one class to put them in which gives the graph on one vertex. Thus, the minimal amount of points is 1. $\square$

FIGURE 9. S_2, the only connected graph on two vertices

3.4.2. S_2.

PROPOSITION 9. $N_T(S_2) = 4$.

PROOF. By Radon's Theorem, we know we can construct this graph with 4 points. This is minimal by Proposition 3. $\square$

FIGURE 10. S_3, the only connected graph on three vertices

3.4.3. S_3.

PROPOSITION 10. $N_T(S_3) = 6$.

PROOF. We can appeal to Proposition 5 to determine that the minimal number of points needed is 6. □

We have enumerated all order types and display partitions which induce S_3 for each of them in Table 1.

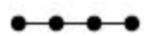

FIGURE 11. P_4, the path on 4 vertices

3.4.4. P_4.

PROPOSITION 11. $N_T(P_4) = 9$.

PROOF. Interestingly enough, after explicit enumeration of all order types on eight points, there exists exactly one point configuration for which it is impossible to generate P_4. This point configuration is displayed in Figure 12. For every other point configuration, we found a partition which induced the path graph on four vertices. From this we assert that $N_T(P_4) \geq 9$.

After an enumeration of all order types on 9 points, we found a partition inducing P_4 for every single order type showing that 9 points is in fact sufficient to construct the path on four vertices. □

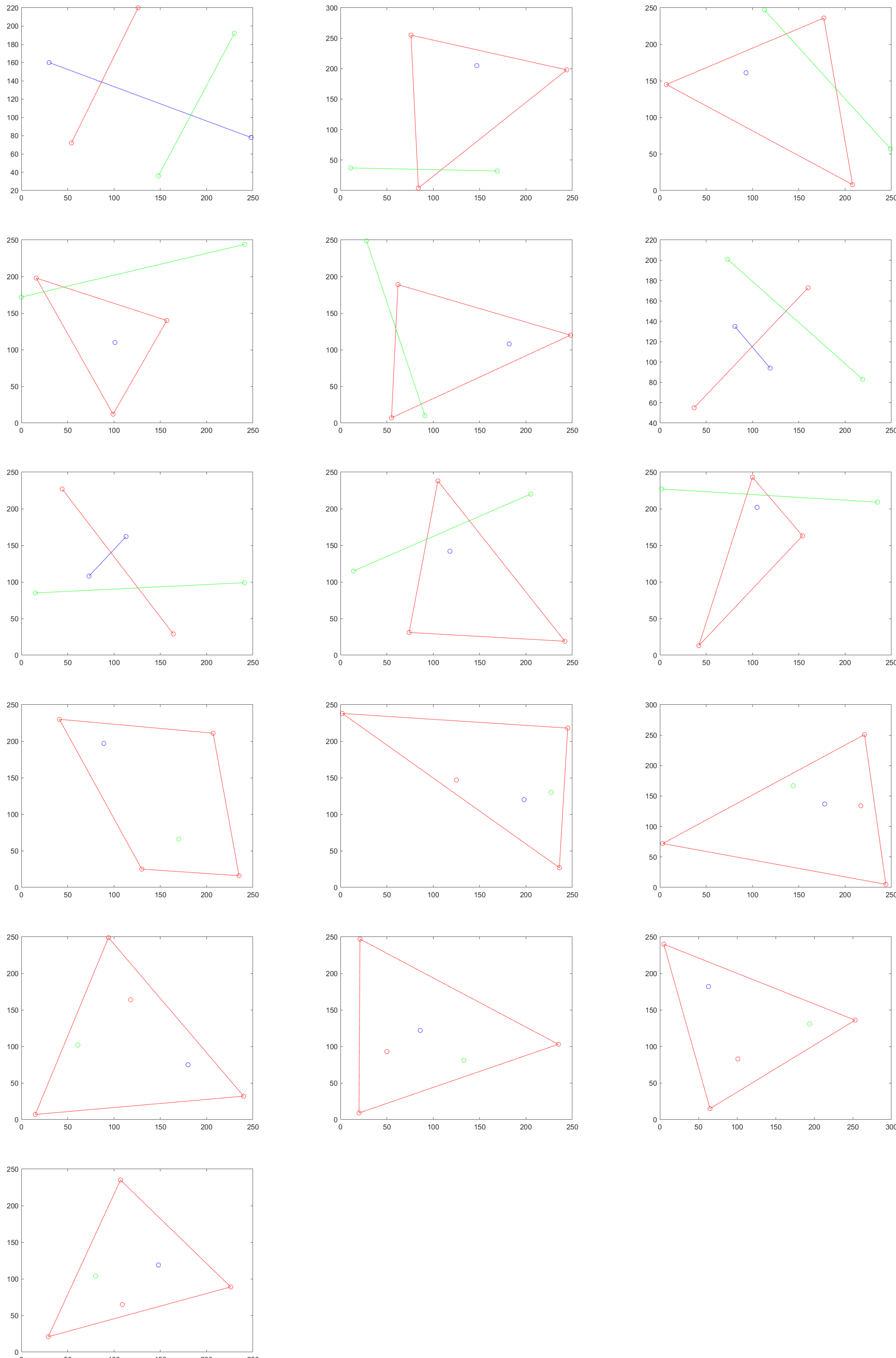

TABLE 1. Partitions inducing S_3 for all order types on 6 points

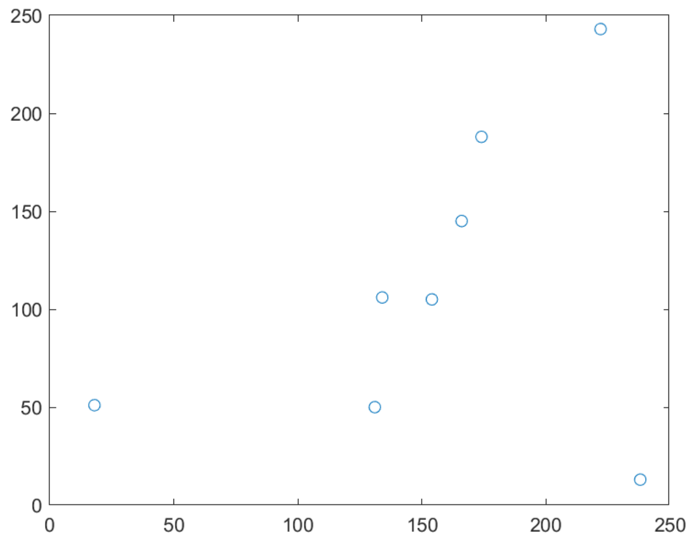

FIGURE 12. A point configuration which cannot be partitioned to induce P_4

TABLE 2. Coordinates of the points for which it is impossible to generate P_4

x	y
222	243
238	13
131	50
154	105
166	145
134	106
174	188
18	51

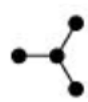

FIGURE 13. S_4, the star on 4 vertices

3.4.5. S_4.

PROPOSITION 12. $N_T(S_4) = 8$.

PROOF. This follows from Proposition 5. In addition, we have enumerated all order types on 8 points and found a partition which would induce S_4. $\square$

FIGURE 14. P_5, the path on 5 vertices

3.4.6. P_5**.**

PROPOSITION 13. $10 \leq N_T(P_5) \leq 13$

PROOF. The lower bound comes from Proposition 3 and the upper bound comes from 6. $\square$

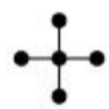

FIGURE 15. S_4, the star on 5 vertices

3.4.7. S_5**.**

PROPOSITION 14. $N_T(S_5) = 10$.

PROOF. This follows from Proposition 5. $\square$

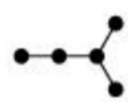

FIGURE 16. The Y graph on 5 vertices

3.4.8. Y**.**

PROPOSITION 15. $10 \leq N_T(Y) \leq 13$

PROOF. We note that this is a star path graph and so applying Proposition 7 we have an upper bound of 13. The lower bound comes from Proposition 3. $\square$

CHAPTER 4

Future directions

In this chapter, we list different ways upon which this study can be expanded on.

First, there are plenty ways in which the algorithm for enumerating the order types can be expanded upon. Methods of improving upon our algorithm for enumeration include:

- Applying algorithms specific to certain classes of graphs. As of right now, it is primarily a brute force algorithm with a few heuristics applied to reduce the size of the search space. There certainly exist more methods specific to larger graphs which may be utilized to reduce the search space in order to examine larger graphs.
- Parallelization of the enumeration of order types. The algorithm is naively parallelizable, being a brute force enumeration. Applying this algorithm onto a cluster would significantly decrease the time of execution.

There also are problems of complexity which emerge regarding computing the Tverberg number.

- Is the problem of computing the Tverberg Number of a given graph in NP?
- Is determining if a point configuration can induce a certain graph in NP?

Our study has also been restricted to two dimensions. The problem of computing Tverberg numbers naturally generalizes to multiple dimensions. We have these questions regarding higher dimensions:

- How is the Tverberg number of a graph related to the number of dimensions on which the point set exists in? Tverberg's Theorem gives results for complete graphs, but the question remains for trees as to how the number is related to the dimension of the space.
- Is there a way to enumerate point configurations in 3 or more dimensions to determine the Tverberg Number?

There also exist natural questions bounding the number of partitions which will induce a certain graph. For example from the survey [**3**], the conjecture is stated that every set $(r-1)(d+1)+1$ points in $\mathbb{R}^d$ has at least $(r-1)!^d$ Tverberg partitions. Could a lower bound on the number of partitions of star graphs and path graphs exist relating the number of points to the number of partitions?

These and other questions would constitute valuable research to gaining a deeper understanding of the nature of the partitioning a given set and the interaction between how the convex hulls of the partition.

Bibliography

1. Oswin Aichholzer, *Enumerating order types for small point sets with applications*, http://www.ist.tugraz.at/staff/aichholzer/research/rp/triangulations/ordertypes/, 2006.
2. Oswin Aichholzer, Franz Aurenhammer, and Hannes Krasser, *Enumerating order types for small point sets with applications*, Order **19** (2002), no. 3, 265–281.
3. Imre Brny and Pablo Sobern, *Tverberg's theorem is 50 years old: a survey*, (2017).
4. Paul Erdös and George Szekeres, *A combinatorial problem in geometry*, Classic Papers in Combinatorics, Springer, 2009, pp. 49–56.
5. Leo Moser and Max Wyman, *Stirling numbers of the second kind*, Duke Mathematical Journal **25** (1958), no. 1, 29–43.
6. Martin Tancer, *d-representability of simplicial complexes of fixed dimension*, arXiv preprint arXiv:1107.1170 (2011).
7. Helge Tverberg, *A generalization of radon's theorem*, Journal of the London Mathematical Society **1** (1966), no. 1, 123–128.
8. Günter M Ziegler, *3n colored points in a plane*, Notices of the AMS **58** (2011), no. 4, 550–557.

APPENDIX A

MATLAB Code

In this section, we present the MATLAB code which was used to enumerate all partitions for all order types on a given number of points. All MATLAB scripts and functions can be made available on request.

```
function intersect = hull_intersect(P1, P2)
% HULL_INTERSECT
% Determines if two convex hulls intersect
% Args:
%     P1: N X 2 matrix listing points in counter-clockwise order
%     P2: M X 2 matrix listing points in counter-clockwise order
% Returns:
%     intersect: Boolean indicating if the sets intersect

n1 = size(P1, 1);
n2 = size(P2, 1);

% We swap if |P1| < |P2|
if n1 < n2
    tmp = P1;
    P1 = P2;
    P2 = tmp;
    n1 = size(P1, 1);
    n2 = size(P2, 1);
end

if n1 == 1 && n2 == 1
    intersect = all(P1 == P2);
elseif n1 == 2 && n2 == 1
    intersect = (norm(P1(1,:) - P2) + norm(P1(2,:) - P2) == norm(P1(1,:) - P1(2,:)));
elseif n1 > 2 && n2 == 1
    % If one is contained in the other
    intersect = inpolygon(P2(1,1), P2(1,2), P1(:,1), P1(:,2));
elseif n1 == 2 && n2 == 2
    % If lines intersect
    intersect = ~isempty(poly2poly(P1',P2'));
elseif n1 > 2 && n2 == 2
    edge_int = isintersect(P1', P2');
    intersect = edge_int;
end

end
```

FIGURE 1. Routine for determining if the convex hulls of two point sets intersect

```
function [int_graph] = intersection_graph(points,groups)
% INTERSECTION_GRAPH
% Computes the intersection graph of a point set given a partition
% Args:
%   points: n X 2 matrix of the points
%   groups: n X 1 matrix of the class each point is in
% Returns:
%   int_graph: The intersection graph

point_groups = splitapply(@(x){x}, points, findgroups(groups));
[n, ~] = size(point_groups);
hulls = cell(n, 1);
for i=1:n
    group = point_groups{i};
    if size(group, 1) < 3
        hulls{i} = group;
    else
        hulls{i} = group(convhull(group),:);
    end
end

adj_matrix = zeros(n);

% Construct the Adjacency Matrix
for i=1:n
    for j=1:i-1
        hull1 = hulls{i};
        hull2 = hulls{j};
        intersect = hull_intersect(hull1, hull2);
        if intersect
            % Using symmetry of the matrix
            adj_matrix(i, j) = 1;
            adj_matrix(j, i) = 1;
        end
    end
end
int_graph = graph(adj_matrix);
end
```

FIGURE 2. Routine for determining the intersection graph of a point set given a partition

```
function [bool, rep] = has_graph(points, g, partitions)
% HAS_GRAPH
% Checks if a point set can be partitioned to induce a given graph.
% It does this by enumeration of partitions, stops if a partition is found.
% Args:
%     points: N X 2 matrix of points
%     g: The graph to induce
%     partitions: An M X N matrix listing all partitions of the points
% Returns:
%     bool: true if partition is found
%     rep: The partition of the points which was found

bool = false;
rep = false;

num_partitions = size(partitions,1);
for i=1:num_partitions
    int_graph = intersection_graph(points, partitions(i,:)');
    if isomorphism(int_graph, g)
        bool = true;
        rep = partitions(i,:);
        break
    end
end

end
```

Figure 3. Routine for determining if any partition of a point set will induce a given graph

```
function [valid_partitions] = find_valid_partitions(n,k,partitions)
% FIND_VALID_PARTITIONS
% Function for enumerating all valid partitions of n into k parts
% Each partition generated will be unique.
% Args:
%     n: The number to break into k parts
%     k: The number of parts broken into
%     partitions: An m X k matrix listing where each row lists a valid
%         partition into k parts in decreasing order
% Returns:
%     valid_partitions: A matrix of all valid partition

p_count = size(partitions,1);
all_partitions = nproduct(1:k, n);
num_partitions = k^n;

valid_partitions = zeros(size(all_partitions));

counts = zeros(1,k);
l = 1;

for j = 1:num_partitions
    partition = all_partitions(j,:);
    % Count each element
    for i=1:k
        counts(i) = sum(partition==i);
    end
    counts = sort(counts);
    valid = false;
    % If matches one of valid partitions, can add it
    for m=1:p_count
        if all(counts == partitions(m,:))
            valid = true;
            break
        end
    end

    % This is to eliminate partitions which are just relabelling of
    % other partitions.
    for m=1:k-1
        if find(partition == m, 1) > find(partition == m+1, 1)
            valid = false;
            break
        end
    end

    if valid
        valid_partitions(l,:) = partition;
        l = l+1;
    end
end
valid_partitions = valid_partitions(1:l-1,:);
end
```

FIGURE 4. Routine for generating all valid partitions of n points into m subsets

```
function [graphs, counts, reps] = all_intersections(points, k)
% ALL_INTERSECTIONS
% Enumerates all partitions of a point set, and generates all possible
% intersection graphs which can be induced by the point set.
% Args:
%     points: N x 2 matrix of points
%     k: The number of classes in the partition
% Returns:
%     graphs: A cellarray of each graph generated up to isomorphism
%     counts: The counts of each graph
%     reps: A representative partition inducing the graph

n = size(points, 1);
graphs = cell(1);
counts = cell(1);
reps = cell(1);
num_graphs = 0;

partitions = nproduct(1:k, n);
num_partitions = k^n;
for i=1:num_partitions
    % If is a valid partition
    if size(unique(partitions(i,:)), 2) ~= k
        continue
    end
    g = intersection_graph(points, partitions(i,:)');
    found = false;
    % Find a graph isomorphic to g
    for j=1:num_graphs
        if isomorphism(g, graphs{j})
            counts{j} = counts{j} + 1;
            found = true;
            break
        end
    end
    if ~found
        % If not isomorphic to any, create a new cell for this isomorphism
        % class.
        graphs{num_graphs+1} = g;
        counts{num_graphs+1} = 1;
        reps{num_graphs+1} = partitions(i,:);
        num_graphs = num_graphs + 1;
    end
end

end
```

Figure 5. Routine for determining all possible intersection graphs which can be induced by a given point set

```
% This script will enumerate all order types on 8 points and attempt to find a
% partition which will induce the path graph on 4 vertices.
% It will generate plots of partitions for each success
% and plots of the points for each failure.

set(0,'DefaultFigureVisible','off');
% File with all order types
fp = fopen('otypes08.b08');

% Path graph on 4 vertices
g = graph([0 1 0 0; 1 0 1 0; 0 1 0 1; 0 0 1 0]);
num_points = 8;

points = {};
count = 0;

% Generate valid partitions
path_parts = [2 2 2 2];
valid_partitions = find_valid_partitions(8,4,path_parts);

i = 0;
while ~feof(fp)
    xs = fread(fp, num_points*2);
    xs = reshape(xs,2,[])';
    disp(xs);

    % Check if graph is possible to induce
    [bool, reps] = has_graph(xs, g, valid_partitions);

    % If not found
    if ~bool
        f = figure('visible', 'off');
        plot(xs(:,1), xs(:,2), 'o');
        filename = sprintf('path4_2/notfound_%d.png',i);
        saveas(f,filename);
        close(gcf)
    else
        plot_groups(xs, reps');
        filename = sprintf('path4_2/found_%d.png',i);
        saveas(gcf, filename);
        close(gcf);
    end
    i = i+1;
end
```

FIGURE 6. An example script for enumerating all order types to find a point set which cannot induce the path graph on 4 vertices