FACETS OF A BALANCED MINIMUM EVOLUTION NETWORK POLYTOPE

A Thesis

Presented to

The Graduate Faculty of the University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Cassandra Durell

May, 2019

FACETS OF A BALANCED MINIMUM EVOLUTION NETWORK POLYTOPE

Cassandra Durell

Thesis

Approved:	Accepted:
____________________	____________________
Advisor Dr. Stefan Forcey	Dean of the College Linda Subich
____________________	____________________
Faculty Reader Dr. J.P. Cossey	Dean of the Graduate School Chand Midha
____________________	____________________
Faculty Reader Dr. Malena Español	Date
____________________	
Department Chair Dr. Kevin Kreider	

ABSTRACT

The balanced minimum evolution (BME) polytope is a structure representative of a problem in biology, in particular in the study of phylogenetic trees. In this scope, the polytope is used to answer the question of how a set of species are related to one another. In this paper we explore generalized instances of the BME polytope for networks. For one of these generalized BME polytopes we focus on the discovery of new facets and their corresponding equations, while for the other we give the facets of the polytope and discuss the relationship that they have to another well known polytope outside of the field of biology. Furthermore, we also provide the dimension reducing equalities that were discovered which hold for every BME polytope and then prove their existence.

ACKNOWLEDGEMETS

I would like to thank Dr. Forcey for allowing me the opportunity to work with him on this project and for everything that I have learned during this process from him that is too much to enumerate. I am grateful for not just his work as an instructor, but as an advisor and mentor during this process. I would also like to thank Dr. Español for reading my thesis and for her role in helping find research that I enjoy as much as I have. Finally, I would like to thank Dr. Cossey for taking the time to read my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In genetics, determining the relationships between species can be done in a multitude of ways, from comparing characteristics, such as whether or not both species have a hinged jaw, to sampling DNA and making statistical comparisons. For our purposes we consider the latter, because it is more precise than the characteristic comparisons.

We choose this since sharing a common feature does not guarantee a relationship between the species and gives limited insight into how closely related they may be. Unlike a characteristic that can be shared among animals that are not related, we have that DNA is unique, like a fingerprint. This means that we may take samples from different species and compare them to determine the dissimilarities through processes such as DNA barcoding. (Forcey) Once data is collected we look to utilize a tree structure to represent our data, placing species close to one another who are more closely related.

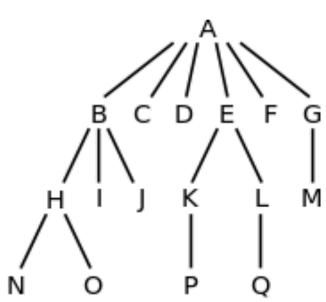


Figure 1.1:
The above is an example of a rooted tree where species B, C, D, E, F, and G are descendants of A and so forth.

In the rooted tree from biology we have that extant species are descendants of those that are closer to the root. This tree, is well-defined under ideal circumstances; however, once we introduce large quantities of species, noise in the data collection, or allow hybridization of species, ambiguities will be introduced that may not be easily overcome and thus the tree representation may be much more difficult to determine. This is why we consider methods in graph theory to find the correct tree representation; it allows us to overcome the ambiguities present. In graph theory, the tree structure used for sorting species is called the Balanced Minimum Evolution tree or BME. More generally, we use the notation BME(n, k) where n represents the number of species being considered and k is the number of bridges in the graph. Here a *bridge* refers to any edge that if it were removed it disconnects the graph. For example, the line in the graph in figure 3.1 that separates the graph into two pieces is a bridge and creates what is called a split. A *split* is a partition of the leaves of the graph into two distinct parts.

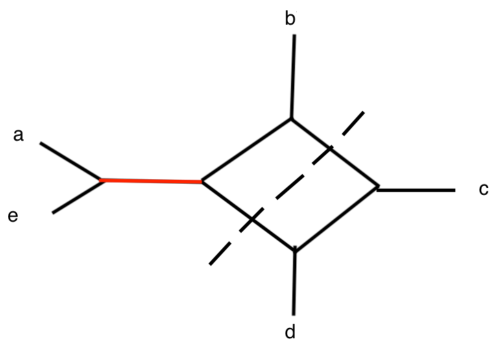


Figure 3.1:
The above network is an example of a vertex of BME(5,1). Here the bridge is represented by the red line and the split is dashed. We see that the network is divided into two parts by the bridge and by the split.

In order to approach this problem in the scope of graph theory we look first to what is called *linear programming*. Linear programming is a process that allows us to take a particular problem and apply an algorithm which will help us to arrive at an answer to the problem. It is important to note that linear programming implies the finding of a solution but not always the optimal solution. In regards to our problem, an algorithm may arrive at a network that shows how the species may be related but it is not guaranteed that the network that the programming arrives upon will be the most correct representation given the collected data. (Forcey) Determining the correctness of the programming happens on a case by case basis. We do not currently have a method to consistently find the most correct network for large quantities of species. Thus the ultimate goal is to find a linear programming approach that will find the most correct network even when there are many ambiguities present, there are large numbers of species, and there is the possibility of species being a hybrid of others.

In order to reach this goal we begin by better understanding generalized network BME polytopes where we consider specific quantities of species and bridges and look to determine their facets. We do this because when facets can be characterized it is also then possible to determine a linear equation that describes the facet and that every vertex of the facet will obey. This will bring us closer to determining a linear programming method that will will yield the most correct tree representation for the generalized BME network polytopes that we consider. Next, we would want to determine which types of facets are present in BME network polytopes of any number of species and bridges. By obtaining the equations of the facets that carry through BME of all numbers of species and bridges we will be closer to determining a linear programming method that minimizes over the polytope and thus returns the most correct tree structure that is representative of the relationship between any number of species and with any network structure.

Aside from aiding in linear programming, finding the facets of the polytope will allow us to run the *simplex method* since it only requires that we know the facet inequalities. The simplex method is a process that produces a *relaxation* of the polytope P from the facet inequalities, where a relaxation of a polytope is any larger polytope (that is also finite) that is larger than P. We look to find a relaxation that is not much larger than P and use it as an approximation of P when finding the polytope itself proves to be too difficult.

Since knowing the facets of the polytope and the facet equalities aid in the construction of linear programming and are required for use of the simplex method, this paper focuses on finding the facets and their linear equations for BME(5,1) and BME(6,0) and leaves the remainder of the process discussed above for further research. In the course of this discovery process we also delve into the relationship between BME(6,0) and another well-known polytope, STSP. In doing so we uncover an alternative definition to a facet of the STSP.

CHAPTER 2

POLYTOPES

As mentioned already we start with the generalized BME network polytopes and with studying their characteristics. A polytope is a multisided shape with flat sides which could be as simple as the cube that we are all familiar with or could be more complex, such as a ten dimensional shape. In this context however the polytope also serves as an organizational tool. To see this we discuss how the polytope is created and we begin by covering terminology that will be necessary for proceeding.

To begin a *graph* is a set of vertices and a set of edges that connect vertices. We have that a graph may also be referred to as a *network.* For our application, we are concerned with networks that are *undirected* which are networks such that traveling from leaf *a* along edge *x* to leaf *b* is the same as traveling from *b* to *a* along edge *x.* In particular, a network of this nature that displays *splits* is called a *split network,* where a split is a partition of the network into two distinct sets of vertices. Furthermore, we have that a *circular split network* is any split network such that the leaves can be read around the graph in a circular fashion where the ordering maintains that for a split A|B the leaves in A must remain grouped together as must the leaves in B. In other words, a leaf that is an element of B may not fall between two leaves that are elements of A in the circular ordering.

To create the polytope we start by listing all of the distinct ways that a split network diagram can be generated with *n* species and *k* bridges. Recall that rotation of a cherry does not

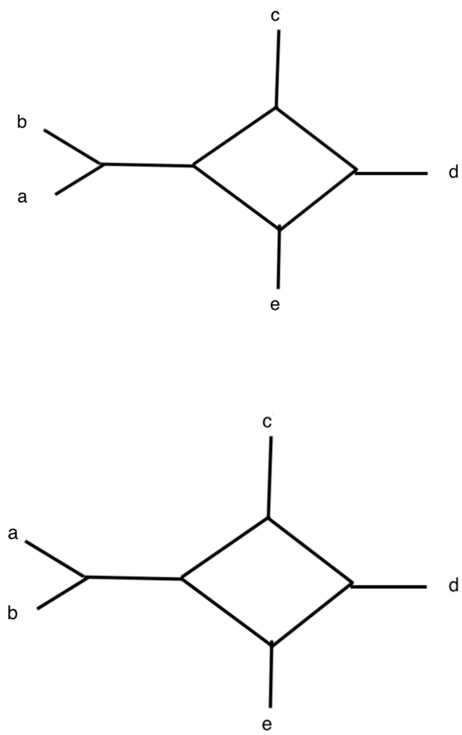


Figure 3.2
Here we have two networks that are the same. The cherry of the network occurs where the bridge splits into two pieces, resembling a pair of cherries on a stem. The rotation of the cherry in the plane allows us to see the second representation of the network which we define to be the same.

produce a distinct network from the original where a *cherry* is a path on three vertices. For example, the portion of the graph to the left of the bridge in figure 3.2 is a cherry on a vertex of BME(5,1).

Once we have compiled all of the potential networks we then must also find each network's corresponding vectors. To find the corresponding vector for a network we consider all of the representations of that network found by rotating the cherries. Again, these representations are of the same network and therefore not unique. For example figure 3.2 shows both representations of the same network which are obtained by rotating the cherry. We then count the ways that different species can occur next to each other, specifically, how many of the different representations of the network place the two species next to each other. (Forcey) For example we consider the network in figure 3.2 where the network can be represented in either way shown. From here we can find that *a* and *b* are next to each other in both networks which garners a two for their value, $x_{ab} = 2$. We can see that following the same reasoning that *a* and *c* are only next to each other in only one of the representations of the network and that *a* and *d* are next to each other never, resulting in a value of 1 and 0 respectively. For consistency we use lexicographical order to represent these figures and thus the vector has the following form

$$x(t) = < x_{ab}, x_{ac}, x_{ad}, x_{ae}, x_{bc}, x_{bd}, x_{be}, x_{cd}, x_{ce}, x_{de} >$$

and hence our network has the following vector representation <2,1,0,1,1,0,1,2,0,2>.

Once we have all of the split network diagrams and their vectors, our next task is create the *polytope*, which is also referred to as a convex hull in a Euclidean space with only finitely may points included. We have that a *convex hull* of a set of points is the smallest convex set that contains all of the points. It may be simpler to think of a convex hull, or polytope, as a set of finitely many points which have been shrink wrapped. There are many examples of convex hulls with which we are all familiar, such as polygons, cubes, and tesseracts. (Forcey)

We can further describe a polytope by discussing its features. We have that any point in the convex hull that cannot be removed without changing the shape of the polytope is called a *vertex*. For example, removing a point that is the corner of a square would result in the convex hull becoming a triangle. And hence, a point that is the corner of a square is then a vertex of a polytope which is a square. Next we have a *face* of the polytope; the set of every point which satisfies exactly a bounding linear inequality of the polytope comprises a face of the polytope. Lastly, we have that a *facet* of the polytope is a face with dimension one less than the dimension of the polytope itself, where the *dimension* of a polytope is the dimension of the smallest Euclidean space needed to contain it. (Forcey) So, for example a cube would have dimension three since that is the smallest space that can contain it and thus the square sides that make up the cube would be the facets of the polytope since they are one dimension less, namely two.

At this point, now that the polytope has been created, if we had an algorithm to minimize over the polytope and obtain the most correct network then we are in a position to apply it. For a small problem we could also apply a *greedy algorithm*, an algorithm that hinges on making choices that at any particular step seem like the most likely to obtain the desired result but that in no way guarantee that this is the most optimal choice overall. For the BME polytope there do exist greedy algorithms, for example: neighbor joining. But, beyond the use of greedy algorithms we do not currently have an effective linear programming method as already mentioned. (Levy) So this is where we must divert and begin to study the polytope itself rather than attempt to begin the process of linear programming.

CHAPTER 3

FACETS OF THE BME(5,1)

3.1 Introduction to BME(5,1)

Our study of the polytope begins with the study of the facets of the polytope. Recall we have that a facet will be a feature of the polytope that has dimension one less than that of the overall polytope. This implies that for a three dimensional polytope, a three dimensional shape that we can visually conceptualize, a facet would be a two dimensional shape. This concept continues for higher dimensional polytopes and in particular the facets of BME(5,1) are four dimensional since BME(5,1) is five dimensional. In the study of facets we look to characterize them by a particular pattern, a pattern for which all of the networks in the facet will adhere. We expect that there will be many types of facets in a polytope that has multiple facets and that each will indeed have a discernible pattern. We also expect that along with each pattern there will be an equality that must logically follow and that the vectors in the facet must obey. Subsequently, it should follow that for any network not in the facet that the equality must not hold and instead will be a strict inequality.

To begin with we start by characterizing the facets of the polytope for specific numbers of species and bridges, in particular five species and one bridge since the facets of this polytope are currently unknown. Networks of this type have the general form seen in figure 3.1 and figure

3.2. In these networks there are five vertices of degree one where labelings may be placed to represent species and the parallelogram represents an ambiguity. An ambiguity can occur due to noise in data that is collected from the DNA, poor data collection, or hybridization of species.

We begin as described in the previous chapter by listing all of the possible networks that can be obtained by utilizing five species and one bridge. When we do so this yields 30 distinct networks and subsequent vectors which we can then input to a program named Polymake. This complete list of networks and the vectors can be found in the appendix of this text. We do expect that the networks possible would total to be 30 since there are 5! ways that one could label the leaves of the network where we then divide by two since the two rotations of the cherry are not unique and again we divide by two since reading the leaves around the network in a cycle clockwise and counter clockwise are also the same network. This results in the following

$$\frac{5!}{(2)(2)} = \frac{120}{4} = 30.$$

Polymake when prompted will then list the number of facets and also the vertices (networks) that happen to be in each facet. It is from this list that we can discern what the pattern will be for each facet as desired. See the appendix for the code that was used for Polymake and the output from the program.

From the list of facets we can easily see that an initial categorization comes from the number of networks in each facet. We see that there are facets that have five, eight, and nine networks in each, and while this does not necessarily mean that all of the facets of the same size will have the same sort of pattern we do know that facet of a certain type/pattern will be required to all be composed of the same number of networks. So, we begin by looking at all of the facets that have five networks in each.

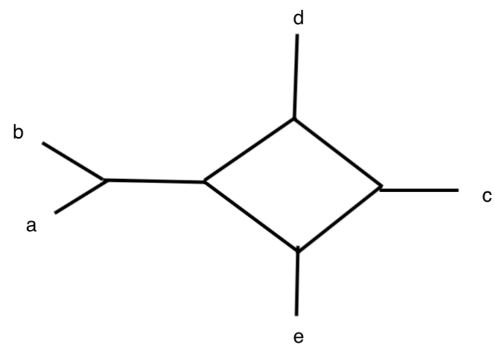
3.2 Cyclic Order Facets

When looking at the twelve facets that have five networks in each we see that all of these adhere to a particular cycle, more precisely a facet of this type will have a certain cycle that every network must display in that facet and it is required that any network not in the facet will not display the cycle. It is important to note that it is not possible for both rotations of the cherry to represent the same cycle so it is only required that one rotation display the cycle be in the facet and, as expected, we require that neither rotation of the cherry display the cycle for a network that is not in the facet. Furthermore, each facet of this type will be associated with a different cycle. For example, we have that the facet comprised of networks 2, 10, 16, 23, and 24 (as given and labeled in the appendix) all display the cycle 1-2-4-3-5-1.

Theorem 1:

We have that there are twelve facets in BME(5,1) such that each has five networks and each one in the facet represents the same cycle, *a-b-c-d-e-a* when reading the leaves around the network in a circle. The vectors of the networks in these facets adhere to the equality

$$x_{ab} + x_{bc} + x_{cd} + x_{de} + x_{ae} = 8.$$



Furthermore, we have that in the case that a network is not in the facet and thus doesn't display the same cycle that the following is true

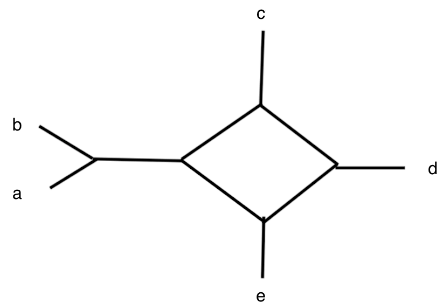
$$x_{ab} + x_{bc} + x_{cd} + x_{de} + x_{ae} < 8.$$

<u>*Proof*</u>:

One can count and verify that there are indeed twelve facets that have five networks that also obey the equality. We can conclude that there are no more than these twelve since there are indeed only twelve ways to form a cycle from five leaves. This is the case since we have that there are 4! ways to create a cycle from five entries and we divide by two since orientation of the cycle is not considered, resulting in

$$\frac{4!}{2} = 12.$$

To verify the equality we arbitrarily label the leaves of the network such that the cycle reads *a-b-c-d-e-a* resulting in:



From the vector $x(t)$ that corresponds to this network we get that $x_{ab} = 2$, $x_{bc} = 1$, $x_{cd} = 2$, $x_{de} = 2$, $x_{ae} = 1$ and so we have that

$$x_{ab} + x_{bc} + x_{cd} + x_{de} + x_{ae} = 8.$$

As an example, consider a network that does not display the same cycle we expect that it would obey the strict inequality. We label it with leaves such that it does not obey the cycle. In particular we have that this network displays the cycle *a-b-d-c-e-a* which results in

$$\begin{aligned} x_{ab} + x_{bc} + x_{cd} + x_{de} + x_{ae} &= 2 + 0 + 2 + 0 + 1 \\ &= 5 \\ &< 8. \end{aligned}$$

We arrive at this since we have that b and c are no longer next to each other which will result in x_{bc} having a value of zero instead of the value one or two that comes from the leaves being next to each other.

In general we consider the maximum value that the equality can yield when the required cycle is not satisfied. In the case that the cycle is satisfied, like in the first network of the proof, we have that the two leaves on the cherry will give a value of two for x_{ab}, as will the value x_{cd} and x_{de}. When we alter this network to not produce the cycle we have that at most two of these can retain their value of two and the third will have a value of zero or one, since allowing all three to remain a value of two would not yield a disruption of the cycle. The remaining values, if not two, must then be a zero or a one. Let's assume, for the sake of finding the maximum value that both of these other values, x_{bd} and x_{ae} are one. Thus we have that $x_{ab} + x_{bc} + x_{cd} + x_{de} + x_{ae}$ can be at most 2+2+1+1+1=7 which is certainly less than eight.

∎

Thus we have verified that the equality holds for all networks in the facet and not for networks in other facets. Furthermore, we have successfully shown that all facets of five networks are accounted for by this characterization. When we are using Polymake we also can get an idea of what the Schlegel diagrams look like for the particular facet we are working with. A Schlegel diagram is a way for us to obtain a visual representation of a four dimensional shape by projecting a part of the diagram into the plane. Since our polytope is five dimensional all of its facets will be four dimensional and will require this type of diagram for visual representation. The Schlegel diagram for this particular type of facet of the BME(5,1) polytope all have the basic form of the diagram in figure 3.5, which is a 4-simplex.

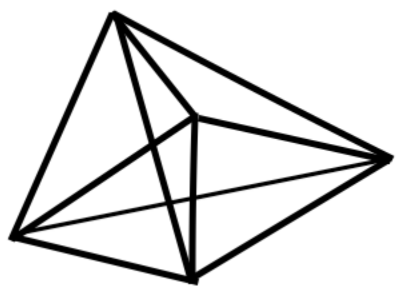


Figure 3.5:
The above are the two basic shapes of the Schlegel diagram for this type of facet.

3.3 Excluded Node Facets

Since all of the facets of five networks are cyclic facets we move on to the facets that have eight networks in each and we find the reoccurring pattern can be described as the following: each network in the facet has two pairs of leaves where the leaves in a pair must always be adjacent, never separated by another leaf. Since there are two pairs we have that naturally there is a node that is excluded from this pairing and may occur in any position except the central single leaf on the cycle which is the right most leaf *e* in figure 3.6, provided, of course, that its does not separate a pair. We require, like in the cyclic facets, that only one rotation of the cherry displays this pattern for a network in the facet and that one that is not in this facet must not display this pattern for either rotation of the cherry. Furthermore, we have that each facet of this type will have different leaves that are chosen to be paired such that each facet is uniquely determined.

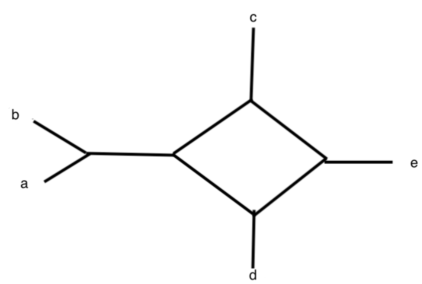
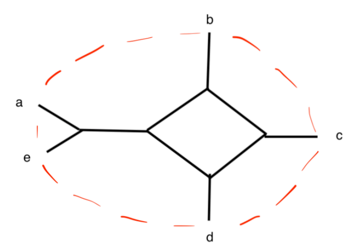


Figure 3.6:
Let *e* be the excluded node. The above shows where the excluded leaf may *not* be placed.

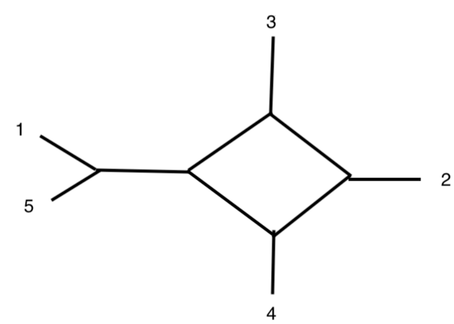
Let the pairs be *a*,*b* and *c,d* and we have that the leaf *e* is excluded. From here it is important to note that the construction of the equality is dependent on the ordering of the pairs in the cycle. In particular one will need to select the leaf that is first of the pair in the cyclic ordering and assign it to be *a* and subsequently the leaf that is its pair will be our *b.* Similarly, in the second pair we select the leaf that is first of the pair in the cyclic order to be *c* and its counterpart in the pair will be *d.* From this specific labeling we obtain the following equality

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 3$$

We arrange the two pairs as adjacent pairs in a cycle of four and read around the network, excluding *e* from this cycle. For each of these networks with pairs *a,b* and *b,c* we get the cycle a-b-c-d-a. We have that the following network is an example of this this cycle



Next we provide an example of how to select the labeling of our nodes. Consider the network



and let it be a part of a facet that has 1,3 and 2,4 as its pairs, leaving 5 to be the excluded node. We have that the first in the cyclic order is 1 and thus we have that the leaf 1 will be *a* in the equality and since the node 3 is in the pair with leaf 1 we have that 3 will be assigned to *b* for the sake of the equality. In the remaining pair we have that 2 is before 4, thus 2 will be assigned to be *c* and 4 will be assigned to *d.* This leaves 5 to be the excluded node *e.* The resulting equality then will be

$$x_{13} + x_{24} - x_{12} - x_{34} = 1+2-0-0 = 3$$

<u>*Theorem 2*</u>:

A facet with 8 networks will have two pairs of leaves in which the leaves in a pair must be next to one another in every network in the facet. For a facet with these such pairs being *a,b* and *c,d* we have that

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 3.$$

If a network is not in the facet it will have that the following inequality holds

$$x_{ab} + x_{cd} - x_{bc} - x_{ad} < 3.$$

<u>*Proof*</u>:

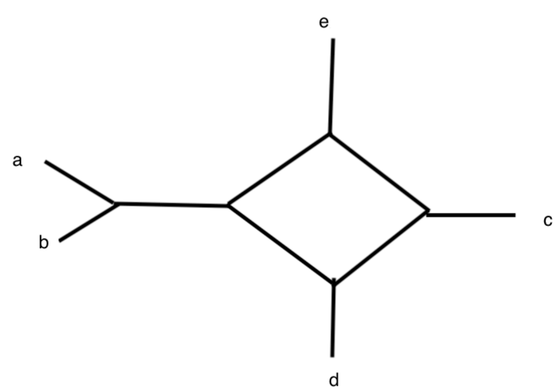
We start by arbitrarily choosing the pairs to be *a,b* and *c,d* which will leave *e* to be the excluded leaf. We have that there are five ways to choose the excluded node and from there we have that there are $\dfrac{2(4-1)!}{2}$ ways to group into pairs. So, we have that there are a total of

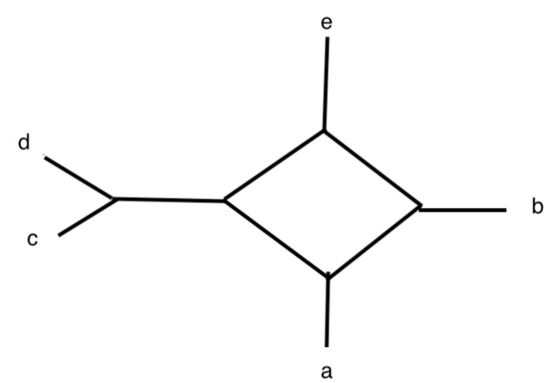
$$\frac{10(4-1)!}{2} = 30$$

ways to create a network that follows the stipulations of being in this type of facet.

We now move on to verifying that the equality holds for networks in the facet. We have that restricting the members of a pair to be next to one another and the restriction on the placement of the node *e* yields two cases to be examined.

Case 1: We place *e* on a side branch, as pictured, and not on a cherry. Doing so will give us two viable labelings for the remaining nodes such that the cycle as read around the network will be *a-b-c-d-a*. These networks are the following:





There are of course two other ways to label the network such that the cycle is maintained but these are excluded from consideration since they do not preserve the pairs' requirement of being adjacent. We then evaluate the left network which yields

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 2 + 2 - 0 - 1 = 3.$$

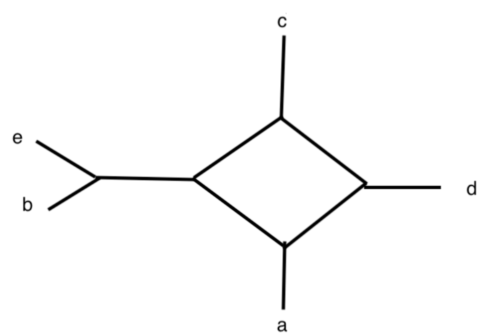
When we look at the right network we have that

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 2 + 2 - 1 - 0 = 3.$$

And thus both possible networks in this case are verified.

We exclude the placement of e onto the other side branch from consideration due to the symmetry of the would be two cases. The result from the leaf e being placed on the other side branch would mirror the case we have just examined.

Case 2: We place e on the cherry (note that the selection of which branch of the cherry will not change the result since the cherry may be rotated) and then label around the network such that the leaves form the cycle *a-b-c-d-a*. This results in four possibilities of network. Since we have that e is on the cherry we have that either the pair *ab* or *cd* will have a node on the cherry as well but not both. Let *ab* be the pair with a node on the cherry.



This will yield $x_{ab} = 1$ since the rotation of the cherry will give that they are next to each other in only one configuration of the network. Likewise, the selection of *ab* having a leaf on the cherry places *cd* with both leaves c and d not on the cherry, resulting in $x_{cd} = 2$.

In this particular configuration we have that the leaves a and c are positioned as far apart on the network as possible and b and d are as well due to the labeling of the nodes needing to obey the cycle a-b-c-d-a. This will give us $x_{ac} = 0$ and $x_{bd} = 0$. Thus the final result will be that

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 1 + 2 - 0 - 0 = 3.$$

The placing of the leaf a on the cherry does not alter the logic applied above and will have the same result. Similarly, we have that selecting c or d to be the node on the cherry will also produce the same consequences and end result. It is left to the reader to verify that these configurations also validate the equality.

It is left to show that the inequality holds for a network that it is not in the facet. For a network to not be in the facet it may be that it fails to have any of the following properties: pairs *a,b* and *c,d* do not maintain that the members of a pair must be adjacent, the excluded leaf *e* is on the restricted branch, or the network does not have a resulting cycle of *a-b-c-d-a*. We break these up into cases.

Case 1: We have that the members of the pairs for the facet are separated. Let the leaves *a* and *b* be the pair that is separated. We have that if a pair is separated such that the members may never be next to each other then we have that x_{ab} must have a value of zero. Thus if we allow x_{cd} to be the largest that it can be and both x_{ac} and x_{bd} to be the smallest that they can be we arrive at

$$x_{ab} + x_{cd} - x_{ac} - x_{ab} = 0 + 2 - 0 - 0 = 2 < 3.$$

Case 2: We let the leaf *e* be on the branch that is not allowed and keep that the members of the pairs are not separated and the cycle is maintained. With *e* on this particular branch if we do not wish to separate our pairs then it must be that one member from each pair is on the cherry. This results in $x_{ab} = 1$ and $x_{cd} = 1$. Let x_{ac} and x_{bd} be the smallest value they can be, thus assigning them to both be 0. We arrive at

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 1 + 1 + 0 + 0 = 2 < 3.$$

Case 3: Let the cycle *a-b-c-d-a* not be achieved by a network. Let the members of pairs not be separated and let *e* be only on an allowed branch. We have that if *e* is on a side branch and not the cherry then every configuration of a network that maintains the members of the pairs being next to each other also will result in the proper cycle. Thus we work with networks that have *e* on the cherry and that follow the restrictions made in our assumption of this case. We have that one of the members of a pair will have a leaf that is on the cherry. Arbitrarily let this be *c*. Thus in order for the pairs to remain together we get that $x_{ab} = 2$ since the leaves *a* and *b* will always be next to each other and $x_{cd} = 1$ since only one orientation of the cherry will place *c* and *d* next to each other. Note that in this configuration it is then required that the nodes *b* and *d* are also always next to each other and that there is an orientation of the cherry in which *a* and *c* are next to each other. The result is

$$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 2 + 1 - 1 - 2 = 0 < 3.$$

And thus we have verified that every way a network could violate a rule of this facet and thus not be a member of the facet will result in the strict inequality from the claim and we are done.

∎

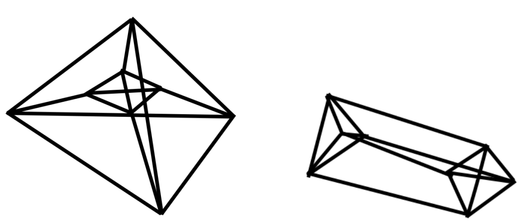


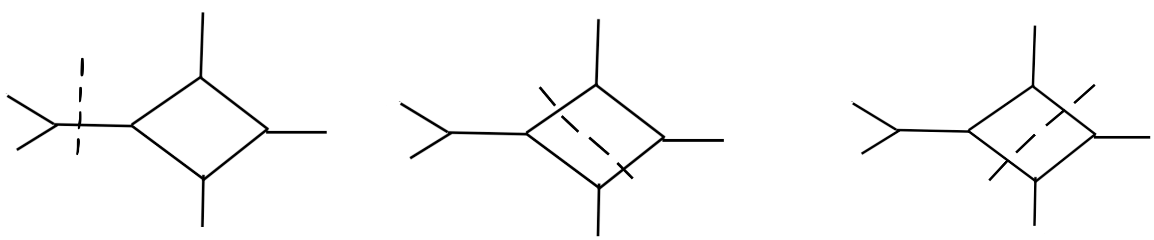
Figure 3.11:
The above are the two basic shapes for the Schlegel diagrams of this facet.

Now we have verified that the equality holds this type of facet and that all thirty of the facets with eight networks must be facets of this type. Like the cyclic facets we can also produce the Schlegel diagrams through polymake.

We have that all of the facets with eight networks are accounted for so we move on to the facets that have nine networks in each.

3.4 Split Facets

For the networks of BME(5,1) there are three ways that we could draw a line and divide the network into two parts or more precisely there are three ways to create a split given this particular network structure.



When examining facets that are composed of nine networks each we find that there are facets that contain networks that display the same split, where here the location of where the split occurs is not important and merely the leaves that are included in each part of the split must always be grouped together for all of the networks in the facet. Hence, we will have that, again, it is only required that one rotation of the cherry display the proper split even though it is possible in this case that both rotations meet the requirements of being in the facet. Furthermore, like all of the aforementioned types of facets, we require that neither of the rotations display the proper split if a network is not in the facet.

Theorem 3:

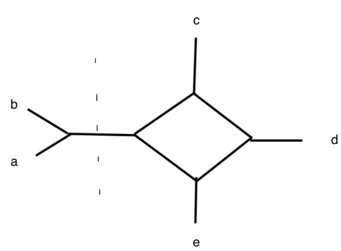
There are precisely ten facets in BME(5,1) that include nine networks with each displaying the same split. The resulting equalities are

$$x_{ab} = 2 \text{ and } x_{cd} + x_{ce} + x_{de} = 4$$

where the split being displayed is $ab|cde$. When a network is not in the vertex it does not display the same split and then we get a strict inequality

$$x_{ab} < 2$$

We have that a split can be displayed in two ways on a network in BME(5,1) and they are as follows:



Proof:

We have counted these such facets and there are indeed ten of this type. We see that there should be no more than these ten since there are ten ways to create a split from five leaves. This is given by $\binom{5}{2}$ which is ten. Hence we should have no more than the ten that have been counted. We will work with the split *ab*|*cde*. Thus we have that $x_{ab} = 2$ follows easily from how the split is defined. In any of the networks above we see that the side of the split with two nodes will always keep the two nodes together in such a way that they may not be apart, not even with the rotation of a cherry, thus resulting in $x_{ab} = 2$. Additionally, we look to verify the equality $x_{cd} + x_{ce} + x_{de} = 4$ and we do so by cases. We have two cases like shown above that will display the split.

Case 1: We have that *a* and *b* form a cherry and arbitrarily label the other nodes. This follows the left most figure in the claim above. We see that we get $x_{cd} = 2$, $x_{de} = 2$, and $x_{ce} = 0$ and thus arrive at $x_{cd} + x_{ce} + x_{de} = 4$. It is easily verified that the other six ways to label nodes *c*, *d*, and *e* will also obey the equality.

Case 2: This case will follow the figure of the right network in the claim above. We have that *a* and *b* do not form a cherry. Again, arbitrarily label the other leaves. We have then that $x_{cd} = 2$, $x_{de} = 1$, and $x_{ce} = 1$. Thus we see that the equality holds and it is easy to verify the other labelings that display this split.

Finally, we look to verify that any network not in the facet, thus not displaying the split, will return $x_{ab} < 2$. We have that if *a* and *b* are not on the same side of the split that they must either not be by each other in any rotation of the cherry and thus $x_{ab} = 0$ or we get that either a or b will be on the cherry but not both either resulting in $x_{ab} = 0$ or $x_{ab} = 1$. Thus we see that the equality does not hold for networks not in the vertex.

∎

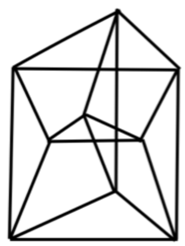


Figure 3.14:
Here is the basic shapes for the Schlegel diagram of this type of facet.

We have now verified that facets of this type obey the given equality and we have shown that there are precisely ten of this type of facet. The associated Schlegel diagram is given in figure 3.14.

Since we have accounted for ten of this facet we have that there are still ten more facets of nine networks that have yet to be accounted for.

3.5 Lower Bound Facets

For the remaining ten facets in the polytope we have that they do all adhere to the same pattern. We see that each facet will have two leaves that may never be next to each other. As expected, it is required that no rotation of the cherry places the two leaves next to each other. This restriction implies that the two leaves may be placed in only two different ways; they may either occur on opposite ends of the network where neither is on the cherry on opposite ends of the network where one leaf is on the cherry. We also require that any network not in the facet have that the leaves are adjacent for at least one rotation of the cherry. We also have that each facet of this type has a different pair of leaves that may never be adjacent.

Theorem 4:

There are precisely ten facets in BME(5,1) such that each has nine networks and there exists a pair of leaves such that each network in the facet places the two nodes non-adjacently. A consequence of this is that

$$x_{ab} = 0$$

where *a* and *b* are the aforementioned pair. Furthermore, we have that

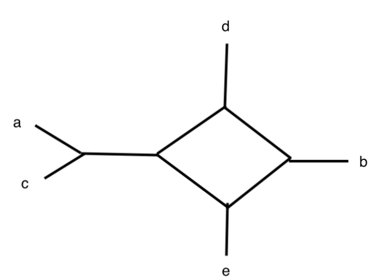
$$x_{cb} + x_{bd} + x_{be} = 4$$

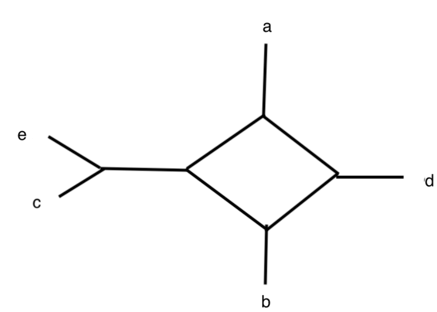
is a necessary consequence. In the case that a network is not included in the facet we should get that

$$x_{ab} > 0$$

$$x_{bc} + x_{bd} + x_{be} < 4$$

The following networks show the two ways that the nodes *a* and *b* can be positioned non-adjacently:





Proof:

It is verifiable that we have found precisely ten such facets in BME(5,1). We can ascertain that these are indeed the only facets of this type given that there are only five ways to choose a pair from our five leaves (recall from the previous proof that selected two leaves from five can be done in $\binom{5}{2}$ ways). Any additional facets that could be found that represent this pattern would subsequently be a repetition of a facet already listed.

Our claim is that for the selected pair a and b we have that $x_{ab} = 0$. We achieve this from how we define x_{ab}. Selecting a and b as our pair places them non-adjacently, hence forcing $x_{ab} = 0$. Furthermore, we look to verify that

$$x_{ab} > 0,$$

when a and b are adjacent. We have that if a and b are adjacent then they must either be next to each other in one of the following ways. We may have that either a or b but not both are on the cherry which would result in $x_{ab} = 1$; a and b are both are on the cherry, resulting in $x_{ab} = 2$; and finally we have the case in which neither a nor b are on the cherry, giving $x_{ab} = 2$. Thus we have that any network not in the facet would have that $x_{ab} > 0$.

∎

Thus we have classified all of the facets and have verified that the equality listed for the classification holds. Furthermore, we have shown that the facets of each type are the only ones of that type, that no more exist that have not been included in the appendix section of this text. And lastly we have included the Schlegel diagrams, so given all of these included features we have a comprehensive look at the facets of this particular BME polytope.

It is left to discuss the polytope itself since we have already talked about the facets which are the features that add the most value. One may assume upon first look that the polytope is ten dimensional if we had not been utilizing Polymake. Polymake gives away the fact that the actual dimension of the polytope is five by giving facets that are of dimension four and when we know that the dimension of the polytope must be greater than the facets by one.

3.6 Dimension Reducing Equalities

Dimension reducing equalities are equalities that hold for all networks in the polytope, unlike facet equalities which hold only for the networks in that particular facet. As mentioned these become important in cases where it is not easy to discern the dimension of the polytope and will become essential for BME of higher numbers of species. However, our study of these equalities began in the case of BME(5,1) so that we might more easily see the format the equalities would have. In doing this we find that five such equalities that hold for all networks in BME(5,1). They are of the form

$$x_{ab} + x_{ac} + x_{ad} + x_{ae} = 4$$

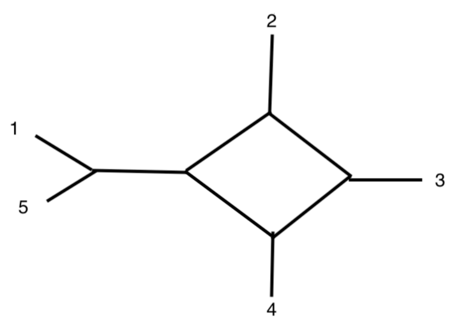
where a is an arbitrarily chosen leaf.

Theorem 5:

Every network in BME(5,1) obeys $x_{ab} + x_{ac} + x_{ad} + x_{ae} = 4$ for a chosen node *a*.

Proof:

We will arbitrarily choose the node *a* and proceed with cases to examine various placements of the chosen node.



Case 1: We place *a* on the cherry and arbitrarily choose the labeling for the other nodes. We have that the other node in the cherry will result in a value of two since the members of the cherry will be next to each other in every orientation. We have that the side branch, labeled 2, to the left will be next to *a* for one rotation of the cherry and similarly the other side branch, labeled 4, will result in a value of one also. Lastly, the furthest branch, labeled 3, will result in a value of zero since this node will never be next to *a* in the cherry. Thus, regardless of the labeling of the nodes *b,c,d,* and *e,* we have that

$$x_{ab} + x_{ac} + x_{ad} + x_{ae} = 2 + 1 + 1 + 0 = 4.$$

Case 2: Place *a* on a side node. Note that the selection of which side branch, either 2 or 4, is irrelevant due to the symmetry of the network. We have that each of the nodes on the cherry will result in a value of one since each will be next to *a* for one orientation of the cherry. We have that the side nodes 2 and 4 will always together yield a value of zero since they are the furthest apart that they can be. Finally, we have that the side node *a* will have a value of two in regards to the furthest node, 3. This results in

$$x_{ab} + x_{ac} + x_{ad} + x_{ae} = 2 + 1 + 1 + 0 = 4.$$

Case 3: Place *a* on the furthest node, labeled with the number 3. This placement of *a* will result in a value of two with each of the side nodes, 2 and 4. And it will result in a value of zero in regards to both of the nodes that are placed in the cherry, hence have

$$x_{ab} + x_{ac} + x_{ad} + x_{ae} = 2 + 2 + 0 + 0 = 4.$$

And we are done.

∎

We include the following table as a way to summarize the results of the previous sections.

Facet Name	Facet Equality	Number of Facets	Number of Vertices in Facet
Cyclic Order Facets	$x_{ab} + x_{bc} + x_{cd} + x_{de} + x_{ae} = 8$	12	5
Excluded Node Facets	$x_{ab} + x_{cd} - x_{ac} - x_{bd} = 3$	30	8
Split Facets	$x_{ab} = 2$ and $x_{cd} + x_{ce} + x_{de} = 4$	10	9
Lower Bound Facets	$x_{ab} = 0$ and $x_{cb} + x_{bd} + x_{be} = 4$	10	9
Dimension Reducing Equalities	$x_{ab} + x_{ac} + x_{ad} + x_{ae} = 4$		

4

DIMENSION REDUCING EQUALITIES

While we are able to find and validate the dimension reducing equalities in regards to the BME(5,1) polytope, like already mentioned, it would be useful to know the general form of the equalities for BME(n, k) and to be able to prove that the equalities hold like we did above. It is our claim that the general form of these equalities is the following

$$x_{ab} + x_{ac} + \ldots + x_{am} = 2^{k+1}$$

for m+1 leaves and for networks with k bridges. We see that in the case of BME(5,1) we expect that the values would total to be $2^{k+1} = 2^2$ and indeed we do have that the values sum to four. Here we provide proof for the general case.

Theorem 6:

For any BME(n, k) we have that for every network in the polytope its vector $x(t)$ must obey equalities of the form

$$x_{ab} + x_{ac} + \ldots + x_{am} = 2^{k+1}$$

where a is an arbitrarily chosen species and $\{b, \ldots, m\}$ is a set of cardinality n-1 which includes all species that are not a.

Proof:

We will prove by induction so we begin with the base case. We let k=0 be the number of bridges present and let n be any number of species. In the case of no bridges we have that there will be only one t-compatible cycle. The *t-compatible cycle* is a cycle that results from reading the leaves around the network in order and then creating a cycle of the same order. From this definition we gather that in cases in which we may rotate portions of a network that we have new a new t-compatible cycle that results from the rotation. For example in BME(5,1) any network will have two t-compatible cycles since there is a cherry that can be rotated at the bridge present and this results in vectors that include the values 0, 1, and 2. So, here we see that having no bridges does imply that there is nothing to rotate and thus there is only one t-compatible cycle for each network. Thus we have that the vector for a network in this type of polytope would only have entries that are 0 and 1. We have then that by construction that any chosen leaf will be next to only two other leaves and being that there is only one cycle this will occur exactly once. This yields

$$x_{ab} + x_{ac} + x_{ad} + \ldots = 1 + 1 + 0 + 0 + \ldots = 2$$

for a chosen leaf a where we arbitrarily choose the leaves next to a to be b and c. And so we see that the equality holds for our base case.

Next we assume that the claim holds for k bridges and any number n species. Thus we may assume that

$$x_{ab} + x_{ac} + \ldots + x_{am} = 2^{k+1}.$$

And we look to show that the claim holds for the case of k+1 bridges, namely that for an arbitrarily chosen leaf a that

$$x_{ab} + x_{ac} + \ldots + x_{am} = 2^{k+1+1} = 2^{k+2}.$$

We look to use our assumptive case in the proof of the quality for k+1 bridges and so in order to do so we must be able to “remove” a bridge from the case of k+1 bridges. So, first we must show that it is possible for a graph to remain externally refined while “deleting” the bridge. Being *externally-refined* requires that any external node of the graph have no more than degree three and while it may be easy to see the effects of removing the bridge in the immediate area, it becomes much more complex for large networks of many species that reach much further than the networks already examined in this text. For this reason we introduce a split in the place of the bridge which will have the same desired effect as removing a bridge and we show then that this is now externally-refined. This should prove easy to do since we can turn to the use of the dual polygonal representation of the network to display the splits of the network. This will show us the changes that are made to the rest of the network when we make this switch. For example the following network and *dual polygonal representation* represent the same relationships between species. We have that each leaf of the network is represented by a side of the polygon and from there we draw lines to represent the splits that are present as in figure 4.1.

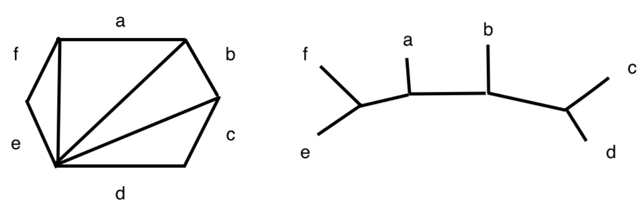


Figure 4.1:
This shows the dual polygonal representation and the corresponding network. Note that they display the same information.

In regards to this polygonal representation we have that a network is *externally-refined* if there are no missing non-crossing diagonals, or in other words, if a diagonal can be draw that would not cross any of the existing diagonals in the polygon then we have that the network is not externally-refined. We opt to use this representation in lieu of drawing the network as we have done previously to show that a split can always be added in place of a bridge and the network remains externally-refined. We choose to do so because we can easily see how other splits will be effected in this representation and the case of large *n* it is not necessarily easy to see this by drawing the network. See figure 4.1 which shows the original graph and the dual polygonal representation and see figure 4.2 for the representation with the addition of the split where the bridge was and then the resultant graph.

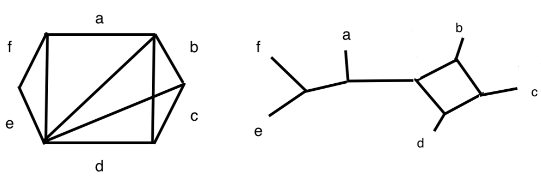


Figure 4.2:
Here the bridge has been eliminated from figure 4.1.

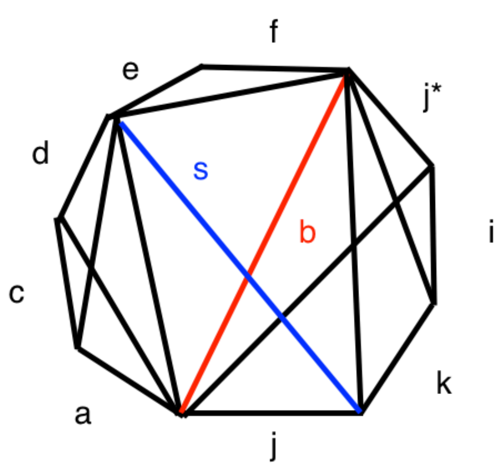


Figure 4.3:
The above shows how a split can be introduced by only crossing the desired bridge and having no effect on the other bridges.

We work first with ten nodes to illustrate the principles that are utilized to show that we may always remove the bridge and replace it with a split. In the figure note that *b* is a bridge since it is a diagonal that is not crossed by any other diagonal. In order to show that can create a split where the bridge is we look to show that it is possible to draw a diagonal that crosses *b* and only crossed diagonals that have already been crossed. In other words, we look to turn just the bridge *b* into a split and leave any other bridges unaffected. We can see that choosing to draw the line *s* is a choice to create a split that does not cross any other bridges.

We have shown for one particular case that we can convert a bridge to a split, so now we look to show that this is possible in the general case. At the worst we see that in any network that all the bridges in the network can at the smallest make a quadrilateral around *b*. Thus we have that it is always possible to draw such a diagonal since bridges at most can triangulate the polygon, thus we can always add a new diagonal which crosses only one bridge.

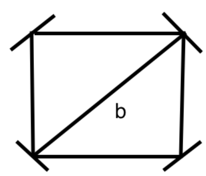


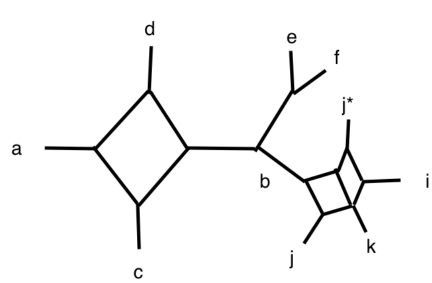
Figure 4.4:
Shows how a quadrilateral can be surrounding a bridge, *b*.

We now proceed with the proof of the equality. We know that we first start by removing a bridge from the case of a network *t* with *n* species and *k*+1 bridges. This process is not dependent upon the removal of any particular bridge so we may select any bridge we wish and we choose that bridge which is the farthest away from the leaf *a* since this will be the easiest to work with. Thus choose bridge *b* to remove for which there are no bridges further from *a* on any path from *a* crossing *b.* After the removal of this bridge to obtain the network *t'* we have that we are now in the case of k bridges for which we assumed that the equality holds so we may assume

$$x_{ab}(t') + x_{ac}(t') + \ldots + x_{am}(t') = 2^{k+1}$$

holds for our new network with the split put in the place of the bridge we chose for removal.

From here we have that there are two potential cases: *a* is not next to a bridge *c* that touches *b* or *a* is adjacent to a bridge *c* that touches *b.*



Case 1:

We have that if *a* is not next to next to a bridge that touches b then we have that there does not exist a rotation of any cherry that will lead to a being next to the leaves that are on the other side of c. Thus we have that

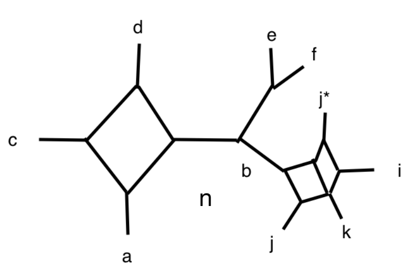
$$x_{ai}(t) = 0 = x_{ai}(t')$$

for any i on the side of c opposite a. Thus we have that if we add the bridge back in that this adds a t-compatible cycle which has the effect of multiplying the current values of the vector by two. Thus we have

$$\begin{aligned} x_{ab}(t) + x_{ac}(t) + \ldots + x_{am}(t) &= 2x_{ab}(t') + 2x_{ac}(t') + \ldots + 2x_{am}(t') \\ &= (2)(2^{k+1}) \\ &= 2^{k+1+1} \\ &= 2^{k+2} \end{aligned}$$

as desired.

Case 2:



We have that a is next to a bridge n which touches b. As in case one we have that there are some values that are far enough away from a that they are zero in both $x(t)$ and $x(t')$ and similarly we have that there are leaves that are on the same side of n as a and for those it is as simple as multiplying their value in the vector by two when we add back in the t-compatible cycle to go from $x(t')$ to $x(t)$. So, we are left interested in what happens to the vector value associated with leaves that are on the opposite side of n from a but that are close enough to have a value of one in the vector for $x(t)$. For example j and $j*$ are two leaves that fit this description. We have that in $x(t)$ that

$$x_{aj}(t) = x_{aj*}(t),$$

or in other words we have that a and j will be next to each other half of the time and the other half of the time a and $j*$ will be adjacent.

Now we have that the removal of b would lead to $x_{aj*} = 0$ as a and $j*$ now will never be next to one another. However, we have that a and j will always be next to each other and there are now half as many t-compatible cycles thus $x_{aj}(t') = x_{aj}(t)$. Hence when going from $x(t')$ to $x(t)$ we yield the result of doubling the value of the vector in regards to all pairs of leaves such as this. And we are done.

∎

CHAPTER 5

FACETS OF BME(6,0)

5.1 Introduction to BME(6,0)

We now look at a new BME polytope using the same methodology that we used for the BME(5,1). We look to accomplish the same goals: characterize the facets and find the associated equalities. However, in addition to these goals we seek to find connections between this polytope and the BME(5,1) polytope for the sake of discovering which types of facets may carry through for different numbers of species and bridges. For example we expect that the split facets will carry through for all number of species and bridges although given without proof and thus is merely accepted as probable. Since the polytope that is constructed here will be the same polytope with the same facets as the Symmetric Traveling Salesman Polytope we will also seek to draw comparisons between the facets of the two. These similarities are intriguing on their own for this specific case but may prove important in terms of linear programming in next stages of research.

The Symmetric Traveling Salesman polytope, or STSP, is a well known polytope that, when minimized over, answers the question, "How can a salesman travel to every city on their agenda precisely once in the most cost effective manner?" In this graph we have that the nodes represent cities and the edges of the graph are the routes between cities. This STSP with *n* cities has the same features as a BME network polytope with *n* species and zero bridges.

So, we proceed in the same manner as for the previous polytope beginning by listing every network possible made from six species and no bridges in the network. However, we are only concerned with the cyclic order of the leaves and thus, although there are numerous ways to draw the graph of BME(6,0), the simplest representation will suffice. This structure can be seen in figure 5.1. As before we then input the corresponding vectors to Polymake to obtain the facets and the list of networks in each facet. The networks and vectors can be found in the appendix as can the code used in Polymake. We see that this process yields a much larger list of facets and many more networks in each facet. From this list we begin by looking for a potential pattern that would hold for a number of facets all consisting of the same number of networks in each.

5.2 Split Facets of BME(6,0)

Before we defined the ways that a split may be present on a network in the BME(5,1) polytope and following the same idea we see that for this new polytope that a split of a network in BME(6,0) may be obtained by drawing the following lines to separate the network in two parts. As expected we see that the presence of split facets carry through to this particular BME polytope. In this instance we have

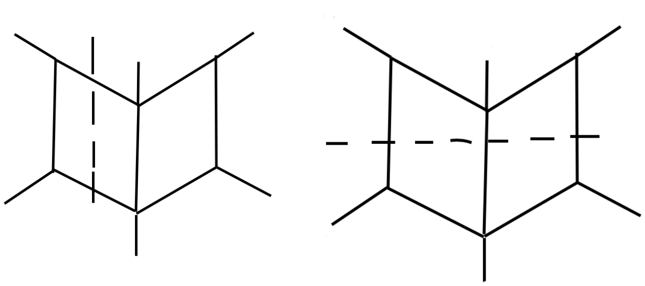


Figure 5.1:
These are the two ways that a split can be draw on BME(6,0)

that there are ten of these facets each consisting of eighteen networks in which the split consists of three leaves in each part of the split and we have that each facet displays a unique split. Like we have done before we look to verify the precise number of facets of this type and that the equality holds.

For this facet it is important to make the distinction that since we arbitrarily labeled the leaves of the networks that the positions on this type of network are not unique. Thus we have that any rotation of the leaves that maintain the same cycle produces the same network. A consequence of this is that while a network may not look like it displays the proper split, if we simply rotate the leaves around the network we modify the network so that it displays the split in a feasible way.

Theorem 7:

There are precisely ten facets of the BME(6,0) polytope that display a particular split where the split consists of three leaves in each part. Let the split for the facet be $abc|def$ then we have that any network in the facet follows the equality

$$x_{ab} + x_{ac} + x_{bc} = 2$$

and any network that is not in the facet will obey

$$x_{ab} + x_{ac} + x_{bc} < 2.$$

Proof:

We have that there are $\binom{6}{3}$ ways to select which leaves will be in which part of the split however we have that selecting the leaves *a,b,* and *c* is the same as selecting the remaining leaves *d,e,* and *f.* Thus we then must divide that quantity by two, resulting in ten total ways that we can create a split of this type and no more and so we have verified that the ten facets we have found are truly the only facets of this type in this polytope. It is left to verify the equality and the inequality.

Let the split displayed by the networks in the facet be *abc|def*. We have two cases for which the split may occur.

<u>Case 1</u>:

Let *a* be between *b* and *c.* Thus we have that x_{ab} has a value of one and x_{ac} does as well. Subsequently we have that *b* and *c* are not adjacent since *a* falls between them so the value of x_{bc} must be zero. Thus the resultant equation is

$$x_{ab} + x_{ac} + x_{bc} = 1+1+0 = 2.$$

<u>Case 2</u>:

Let *a* not be between *b* and *c.* Thus we have that either x_{ab} or x_{ac} must be one but not both. Additionally, since *b* and *c* must be adjacent in this scenario we have that

$$x_{ab} + x_{ac} + x_{bc} = 2.$$

Thus we have verified the equality for all possible cases.

It is left to show that for a network not in the facet that the inequality holds. If a network is not in the facet then we may assume that the split is not displayed, thus at least one leaf of the three is not adjacent to the others of the leaves *a,b,* and *c.* Thus we have that the values associated with that leaf must be zero, leading to a decrease in the value in the equality. So, we have that this implies that for networks not in the facet that the inequality holds.

∎

Since we have verified that there are ten of these facets in the polytope we have accounted for all of the facets with eighteen networks. If we follow the established procession for describing facets that has been established by the work done on BME(5,1) then our next step would be to discuss the Schlegel diagram however for this particular polytope we have that the facets are not fourth dimensional or lower so there is not a visual representation that can be included for these facets. Thus, moving forward they will be excluded from the discussion of the facets in this polytope.

5.3 Adjacent Node Facets

For this polytope we have already found a characterization that required the use of splits however, an astute reader may have noticed that only one potential split was utilized in the description of this facet from the list of potential splits given previously. This next type of facet may be described in two different ways the first being with the use of a split. We see that the facets that involve 24 networks each all display a unique split of the form $ab|cdef$. Equivalently, we can describe this facet as having two leaves that are adjacent for every network in the facet with each facet having a distinct pair that are adjacent. This circumvents the need for rotating the leaves around the network until they are in a position to properly display the split like in the previous facet. Being that the adjacency of leaves requires no additional complications we opt to proceed forward with this description.

Theorem 8:

There are fifteen facets that have a pair of leaves that are adjacent in every network of the facet. Let a and b be the two leaves that are kept adjacent. Then we have that for every network in the facet the following equality holds

$$x_{ab} = 1 .$$

And for every network that is not in the facet we have that $x_{ab} < 1$.

Proof:

We have that if we need to choose two leaves to be kept together, then for six leaves we have that there are $\binom{6}{2}$ ways to do this which results in fifteen ways to make the selection. Thus we have that there are fifteen facets of this type which accounts for all of the facets made up of 24 networks.

For the equality corresponding to the facet we have that any network must maintain the adjacency of a and b thus we have that x_{ab} must have a value of one for every network since they must always occur next to each other. Furthermore, we have that for any network not in the facet, the two may not occur next to each other, thus x_{ab} must be zero which is indeed less than one and we are done.

∎

5.4 Lower Bound Facets

We have that, like the previous polytope, there are facets for which the networks in the facet have a particular pair of leaves that are never adjacent. We have that each facet has a distinct pair that it keeps separated. This can be seen in the facets that have 36 networks in each and since we are selecting a pair from six potential leaves, we should see that there are fifteen ways to do this and thus fifteen facets that are of this type. For proof of this quantity see the proof given for the adjacent node facets. It is left then to show that the equality and inequality hold for networks in and not in the facet respectively.

Theorem 9:

We have that there are fifteen facets having a pair of leaves that may never occur next to one another in any network in the facet. This is given by the equality

$$x_{ab} = 0$$

where *a* and *b* are the leaves that may never be adjacent. Subsequently we have that if a network is not in the facet then

$$x_{ab} > 0.$$

Proof:

We have that for each network in the facet that the leaves *a* and *b* must never be next to one another. Thus this will always result in a value of

$$x_{ab} = 0.$$

We have conversely that if a network is not in the facet then the leaves *a* and *b* must occur next to one another on the given network thus resulting in

$$x_{ab} = 1$$

which is indeed greater than zero. And we are done.

∎

5.5 Pairs Facets

We now only have left the facets which have nine networks in each. Each facet pairs the six leaves and in six of the nine networks in the facet the leaves in a pair are adjacent. In the other three networks there is one pair per network that is placed opposite one another on the network. Note that in each of these three networks only one pair is placed opposite and the other two remain adjacent. In the three networks that have one of the pairs placed opposite it is important to note which other leaves are placed opposite since those will be used later to determine the equality that defines the facets.

We have that there are indeed 60 of these such facets; this is the case since there are $\binom{6}{3} = 20$ ways to choose the pairs that will be adjacent and then there are three ways to choose

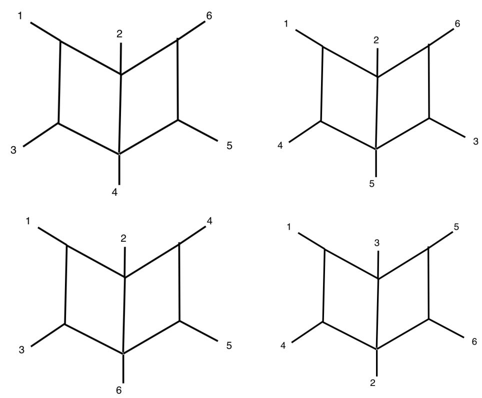


Figure 5.2:
The above are networks from a pairs facet. We have that the primary pairs are *13, 26,* and *45.* The first network shows an example of when all primary pairs are kept adjacent. The other three are examples of when the primary pairs are placed opposite. Thus the secondary pairs are *25, 46, 34, 15, 16,* and *23.*

which leaves are placed opposite of each other in the three networks with a pair placed opposite in each. This results in 20(3) = 60 possible ways to define a facet of this type.

For the equality that defines the facet, let the pairs be as follows: a and b, c and d, and e and f. Let the following leaves be opposite one another in the three non-adjacent pair networks: c and f, d and e, a and f, b and e, a and d, b and c. Then the equality for the facet is

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 4$$

Note, that the leaves that are adjacent pairs are excluded from the sum and any pair of leaves that are opposite each other in the three non-adjacent networks are doubled in the sum.

<u>*Theorem 10*</u>:

There are facets in BME(6,0) such that each has six networks in which the pairs of leaves are always adjacent and three networks where one pair has leaves opposite of each other in the network. In the three non-adjacent pair networks the leaves that are opposite have their vector element doubled and the three pairs are excluded from the sum. The equality is given by

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 4.$$

We also have that for a network that is not in the facet that the following holds

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} > 4.$$

Proof:

We deal with the six adjacent pair networks and the three non-adjacent pair networks separately.

Case 1:

Consider the three networks that have the pairs non-adjacent. Without loss of generality we choose the network with the pair *a* and *b* placed opposite one another. We have that the other two pairs must remain adjacent and there are four unique ways that the two pairs can be filled in each having the same effect, so we arbitrarily choose to place the leaves such that they make the cycle *a-c-d-b-f-e-a* when read around the network. Thus we have that

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 1 + 0 + 1 + 0 + 0 + 1 + 0 + 1 + 0 + 0 + 0 + 0$$

$$= 4.$$

We have that the other ways we could have placed the two pairs would end in a similar result. And thus we are done.

Case 2:

Consider the six networks that have the pairs placed adjacent. Arbitrarily label the the network to read around in the cycle *a-c-d-e-f-b-a.* Thus we have that the resultant equality is

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 1 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 2(1) + 0$$

$$=4.$$

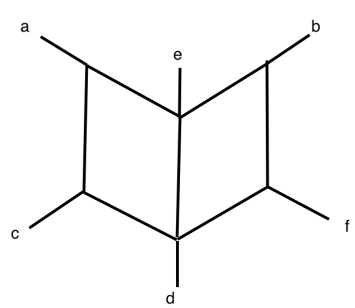
The other cases will follow similarly.

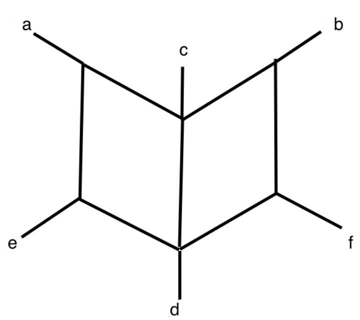
It is left to show that for any network which is not in the facet that the inequality holds. For ease we refer to the leaves that are pairs with the pairs adjacent or placed opposite of one another as the primary pairs. We refer to the pairs of leaves that are opposite in the three networks with the adjacent pairs placed opposite one another as secondary pairs. We have that there are two ways for a network to not be in the facet, the first being a network which has one of the prescribed primary pairs be not adjacent and also not placed opposite one another. The second is a network which has one of the primary pairs placed opposite but one of the secondary pairs is not placed opposite. We handle these cases separately.

Let the primary pairs be *ab, cd,* and *ef* and let *af, ad, bc, cf, be,* and *de* be the secondary pairs. Thus we have that the equality should read

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} > 4$$

as desired.





Case 1: We begin with the case with a primary pair *ab* not adjacent and also not placed opposite. Thus we have that *a* and *b* must have precisely one leaf between them. We have that this leaves two sub cases: the case with one of the primary pairs placed adjacent and the other placed neither adjacent nor opposite, the other case being one of the primary pairs placed opposite and the other placed neither opposite nor adjacent.

Case i: One of the primary pairs is adjacent and the other two are not adjacent and not opposite. In this case we have that the equality must be as follows

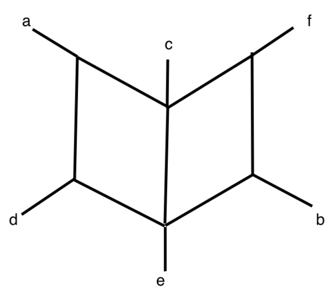
$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 1 + 0 + 1 + 0 + 0 + 0 + 2 + 1 + 0 + 0 + 0 + 1$$

$$= 6 > 4.$$

Case ii: One of primary pairs is opposite and the other two are neither adjacent nor opposite. In this case we have that the equality must be as follows

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 1 + 0 + 1 + 0 + 2 + 0 + 0 + 1 + 0 + 0 + 2 + 1$$

$$= 8 > 4.$$



Case 2: We have that a primary pair is placed opposite. Let that primary pair be *ab.* Thus we have then that the required secondary pairs that must be opposite each other in this case must be *cd* and *de* as the other secondary pairs require *a* or *b* to be an element of a pair. Since this case requires that a secondary pair not be placed opposite then we have that the two secondary pairs must both be placed adjacent.

Thus the resultant equation must be

$$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 0 + 2 + 0 + 0 + 0 + 0 + 2 + 1 + 0 + 2 + 2 + 0$$

$$= 9 > 4.$$

And we are done.

$$\blacksquare$$

At this point covered all of the types of facets for the BME(6,0) polytope so the table here gives the facets and their properties. It is left to the next section to compare these facets to the facets of the Symmetric Traveling Salesman Polytope with six cities considered.

We again include the following table as a way to summarize our results.

Facet Name	Facet Equality	Number of Facets	Number of Vertices in Facet
Split Facets	$x_{ab} + x_{ac} + x_{bc} = 2$	10	18
Adjacent Node Facets	$x_{ab} = 1$	15	24
Lower Bound Facets	$x_{ab} = 0$	15	36
Pairs Facets	$x_{ac} + 2x_{ad} + x_{ae} + 2x_{af} + 2x_{bc} + x_{bd} + 2x_{be} + x_{bf} + x_{ce} + 2x_{cf} + 2x_{de} + x_{df} = 4$	60	9

CHAPTER 6

COMPARING BME WITH STSP

The BME polytope can in some cases be compared to the Symmetric Traveling Salesman Polytope, specifically when there are no bridges present, since they have the same construction. An example of this is the BME(6,0) polytope for which we have just discussed the facets. We have for both that there are 100 facets and 60 vertices; these are known for the STSP and can be found in the appendix for BME(6,0). In the following we discuss the three types of facets for each and how they are related.

In both polytopes we have 15 lower bound facets which are described in the same manner and have the same properties and thus require no further comparison so we begin with the *subtour elimination facets* of the STSP. These facets are described as being all of the ways that you can create a subtour on our six cities where a *subtour* is a tour that displays a split, or any degenerate tour that is formed between intermediate notes and not connected to the origin. (Grotschel) Based on this description we expect that these are comparable to the facets in BME(6,0) which are defined by how they display a split. There are 25 of these subtour elimination facets and thus we expect that these are comparable to both of the 10 split facets of BME(6.0) and the 15 adjacent node facets.

In the STSP we have that any grouping of adjacent nodes will constitute a split, thus we end up with all 25 ways to create a split. However, in BME(6,0) we have that leaves being adjacent is not enough to say the two sets of leaves define a split. We have that a split may only be drawn in the ways outlined in figure 22. Thus, we have that there are only ten ways to do this.

However, consider the adjacent node facets of BME(6,0), these give us the ways that leaves may be adjacent and since we are only concerned with the cyclic order of the leaves in the network then we may rotate the leaves around the network until the adjacent nodes are together on one side of a split. Thus the adjacent node facets are essentially the same as the split facets and it is then reasonable to compare these two types of BME(6,0) facets to the subtour elimination facets of the STSP. We then have that there are 25 facets in BME(6,0) that correspond to the subtour elimination facets in STSP as expected.

Lastly, we consider the last 60 facets of STSP which are *comb facets*. We have that a comb is comprised of a vertex set H (the handle of the comb) and vertex sets $T_1, \ldots, T_p$ (the teeth of the comb) where $p \geq 3$ and p is odd. We require that the sets of teeth are disjoint and also that the sets $H \cap T_j$ and $T_j \backslash H$ are nonempty for all j. (Grotschel) We offer the pairs facets of BME(6,0) as a substitute to this idea since comparatively speaking they are more simply defined than the comb facets of STSP.

CHAPTER 7

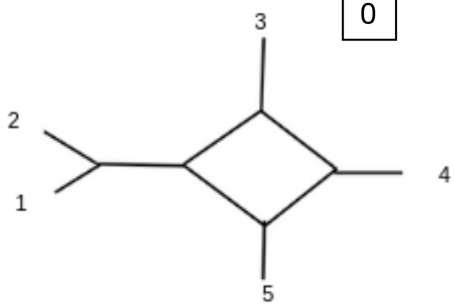
CONCLUSION

In taking steps toward creating a linear programming method to minimize over the polytope we have found and characterized the facets of BME(5,1), accompanying each of these facet descriptions with the corresponding linear equality. We have also found that some of these facets carry through to BME(6,0) and we expect that they are also a part of the BME network polytope for *n* species and *k* bridges. These are important steps in understanding the faces and facets of the BME network polytope which grow quickly in number as the number of vertices also grows. This understanding will advance us further in the search for a linear programming method and in finding a relaxation of the polytope, both of these allowing us to answer questions in the field of biology.

APPENDIX

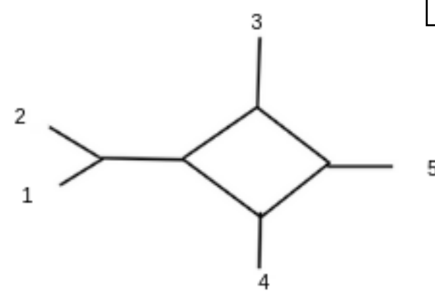
The following are the networks that comprise the BME(5,1) polytope. Each is accompanied by its vector and each vector is later put into polymake to create the polytope and find the list of facets. The boxed number that accompanies each network is the number that Polymake uses to identify the network in its determination of networks in facets.

0



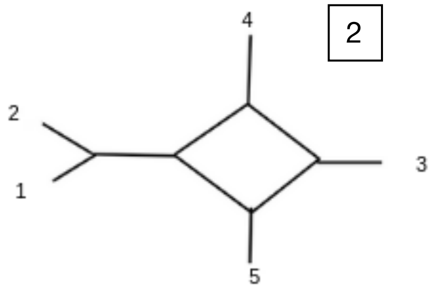
<2,1,0,1,1,0,1,2,0,2>

1



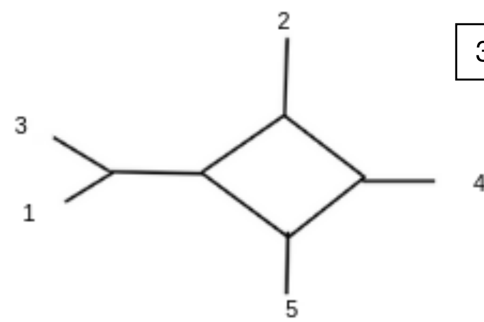
<2,1,1,0,1,1,0,0,2,2>

2



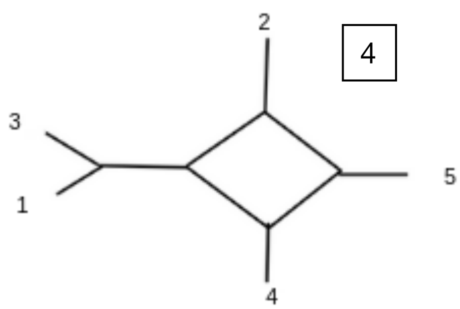
<2,0,1,1,0,1,1,2,2,0>

3



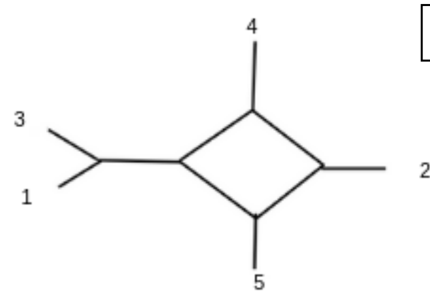
<1,2,0,1,1,2,0,0,1,2>

4



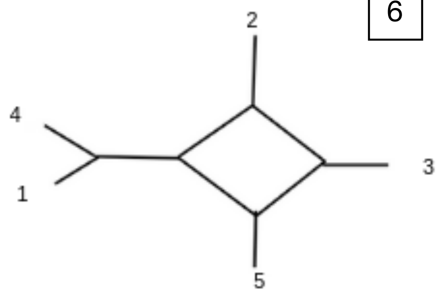
<1,2,1,0,1,0,2,1,0,2>

5



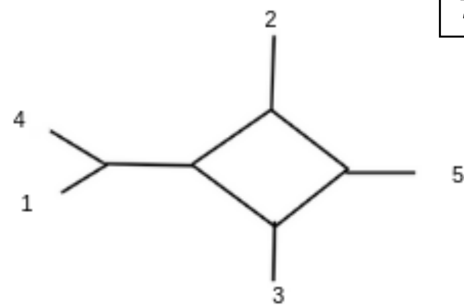
<0,2,1,1,0,2,2,1,1,0>

6



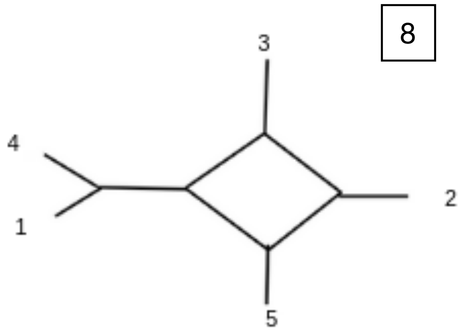
<1,0,2,1,2,1,0,0,2,1>

7



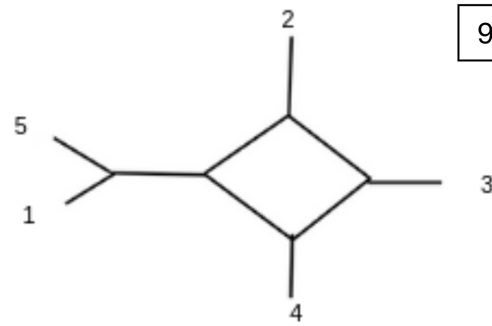
<1,1,2,0,0,1,2,1,2,0>

8



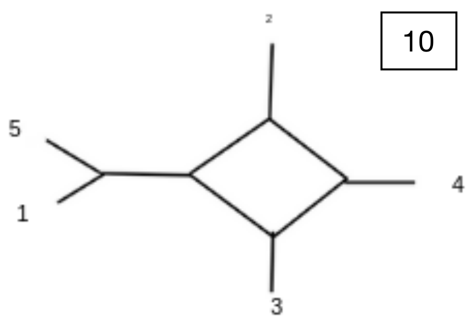
<0,1,2,1,2,0,2,1,0,1>

9



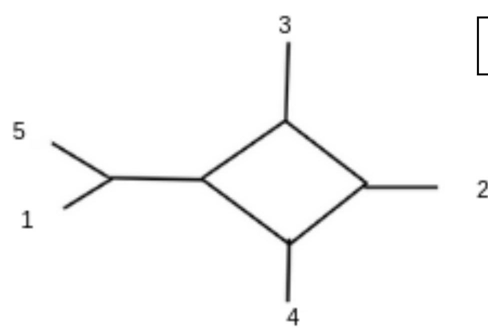
<1,0,1,2,2,0,1,2,0,1>

10



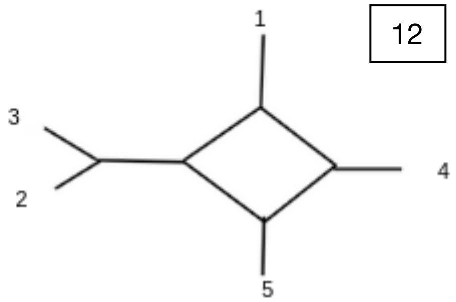
<1,1,0,2,0,2,1,2,1,0>

11



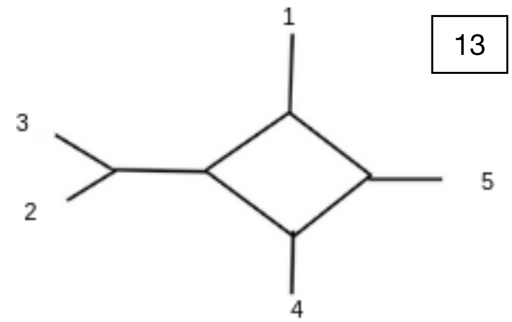
<0,1,1,2,2,2,0,0,1,1,>

12



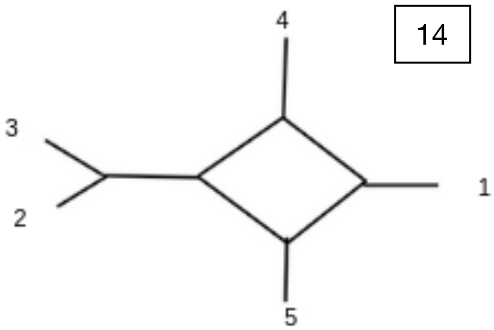
<1,1,2,0,2,0,1,0,1,2>

13



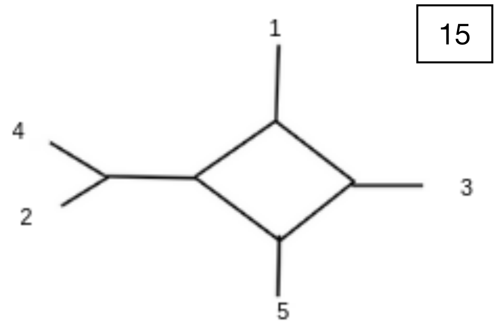
<1,1,0,2,2,1,0,1,0,2>

14



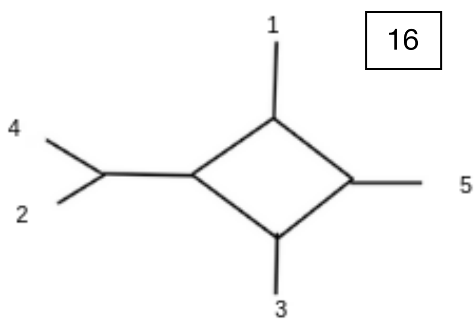
<0,0,2,2,2,1,1,1,1,0>

15



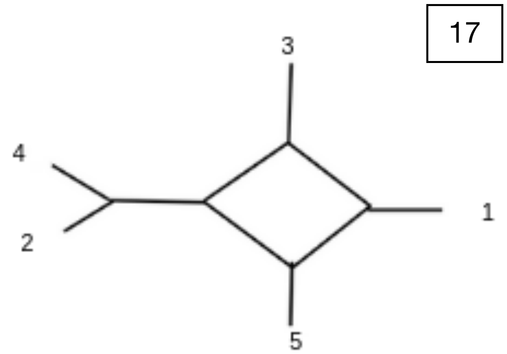
<1,2,1,0,0,2,1,0,2,1>

16



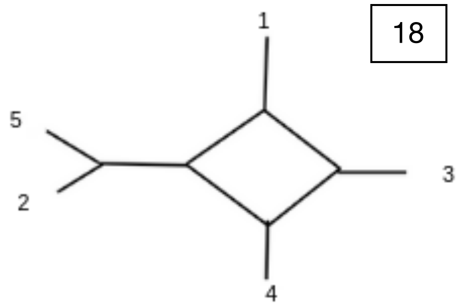
<1,0,1,2,1,2,0,1,2,0>

17



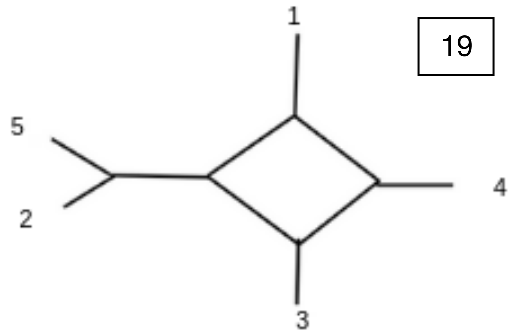
<0,2,0,2,1,2,1,1,0,1>

18



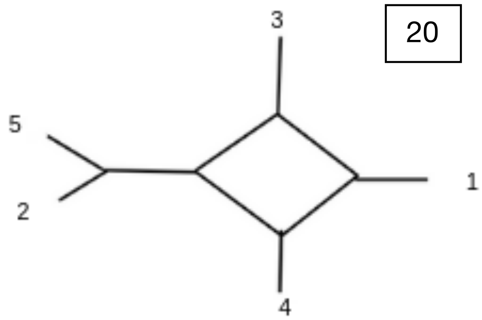
<1,2,0,1,0,1,2,2,0,1>

19



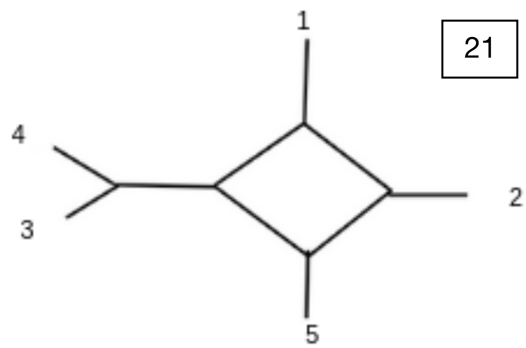
<1,0,2,1,1,0,2,2,1,0>

20



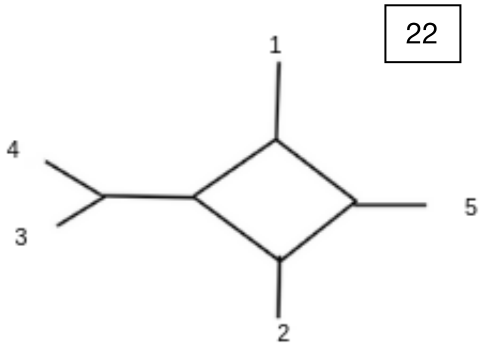
<0,2,2,0,1,1,2,0,1,1>

21



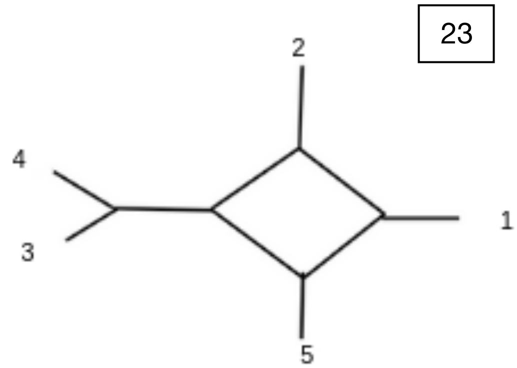
<2,1,1,0,0,0,2,2,1,1>

22



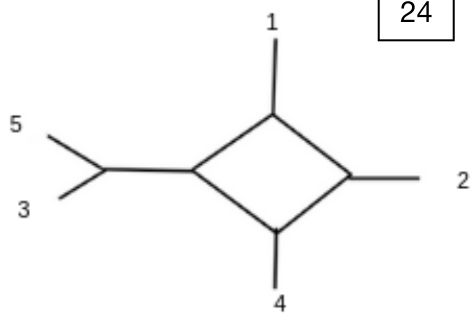
<0,1,1,2,1,1,2,2,0,0>

23



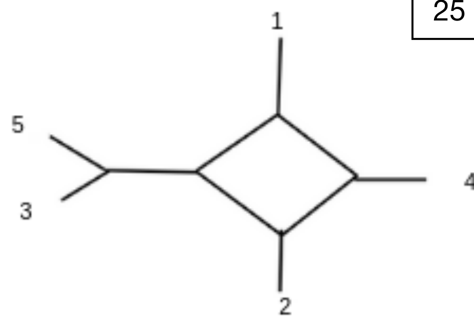
<2,0,0,2,1,1,0,2,1,1>

24



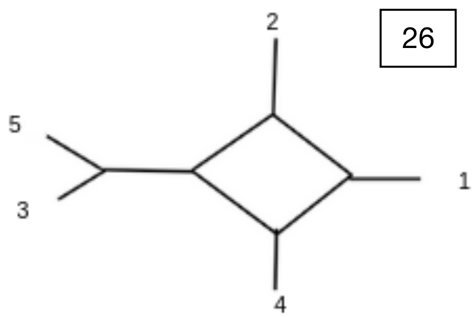
<2,1,0,1,0,2,0,1,2,1>

25



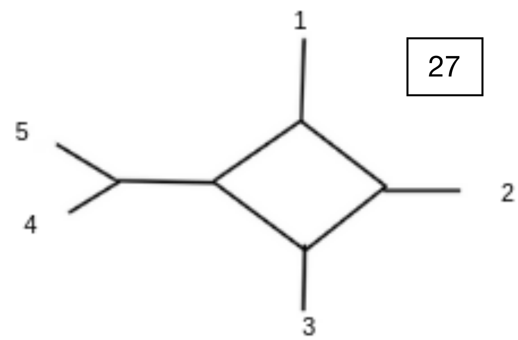
<0,1,2,1,1,2,1,0,2,0>

26



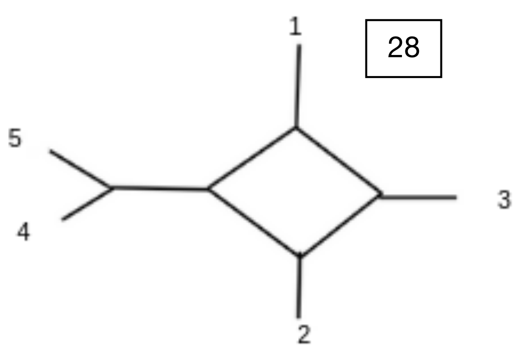
<2,0,2,0,1,0,1,1,2,1>

27



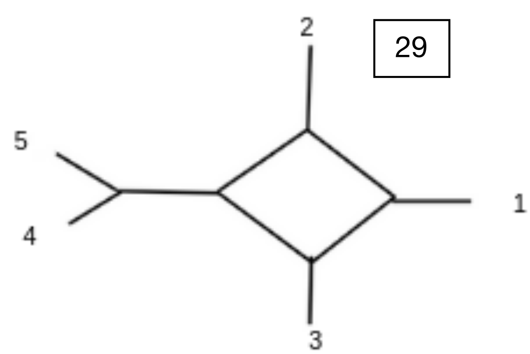
<2,0,1,1,2,0,0,1,1,2>

28



<0,2,1,1,2,1,1,0,0,2>

29



<2,2,0,0,0,1,1,1,1,2>

For the program polymake it is expected that the networks are given by their vector for input. This is not done in any particular order as input but it is important to note that polymake does assign a number to the networks in the polytope to establish an order. It assigns a value of zero to the first vector/network and proceeds through the natural numbers in order through to the last network which is assigned to 29.

The first section of coding in the following section is the input used to create the matrix of data as described in the previous paragraph. The following line then takes that information and assigns the matrix to be the points of the polytope and from there we can begin to request information about the polytope. Next we have the f-vector command which gives us information about the features of the polytope. We have that the f-vector below tells us that there are 30 vertices, 120 edges, 390 faces that are not facets, and that there are 62 facets.

The following line of code then outputs the Schlegel diagram and from the next line we get a list of the facets. In that list of facets we also are given which networks are in each facet, denoted by the number that Polymake assigns to the network. It is from this list that we work to find the characteristics of the facets.

Code: BME(5,1)

```
$points=new Matrix([[1,2,1,0,1,1,0,1,2,0,2],[1,2,1,1,0,1,1,0,0,2,2],[1,2,0,1,1,0,1,1,2,2,0],
[1,1,2,0,1,1,2,0,0,1,2],[1,1,2,1,0,1,0,2,1,0,2],[1,0,2,1,1,0,2,2,1,1,0],[1,1,0,2,1,2,1,0,0,2,1],
[1,1,1,2,0,0,1,2,1,2,0],[1,0,1,2,1,2,0,2,1,0,1],[1,1,0,1,2,2,0,1,2,0,1],[1,1,1,0,2,0,2,1,2,1,0],
[1,0,1,1,2,2,2,0,0,1,1],[1,1,1,2,0,2,0,1,0,1,2],[1,1,1,0,2,2,1,0,1,0,2],[1,0,0,2,2,2,1,1,1,1,0],
[1,1,2,1,0,0,2,1,0,2,1],[1,1,0,1,2,1,2,0,1,2,0],[1,0,2,0,2,1,2,1,1,0,1],[1,1,2,0,1,0,1,2,2,0,1],
[1,1,0,2,1,1,0,2,2,1,0],[1,0,2,2,0,1,1,2,0,1,1],[1,2,1,1,0,0,0,2,2,1,1],[1,0,1,1,2,1,1,2,2,0,0],
[1,2,0,0,2,1,1,0,2,1,1],[1,2,1,0,1,0,2,0,1,2,1],[1,0,1,2,1,1,2,1,0,2,0],[1,2,0,2,0,1,0,1,1,2,1],
[1,2,0,1,1,2,0,0,1,1,2],[1,0,2,1,1,2,1,1,0,0,2],[1,2,2,0,0,0,1,1,1,1,2]]);
```

$p=new Polytope(POINTS=>$points);

print $p->F_VECTOR;
30 120 210 180 62

$p->SCHLEGEL

print $p->VERTICES_IN_FACETS;

<u>Output from Polymake:</u>

{1 2 7 15 21 24 26 29}

{0 1 2 21 23 24 26 27 29}

{1 4 7 12 15 20 21 26 29}

{1 3 4 12 15 20 28 29}

{0 1 3 4 12 13 27 28 29}

{0 1 3 13 23 24 27 29}

{0 3 4 13 17 18 28 29}

{0 2 10 18 21 23 24 29}

{3 5 10 15 17 18 24 29}

{2 5 7 10 15 18 21 24 29}

{3 4 5 15 17 18 20 28 29}

{4 5 7 15 18 20 21 29}

{0 3 10 13 17 18 23 24 29}

{0 1 4 12 21 26 27 29}

{1 3 6 11 15 16 24 25}

{2 6 9 14 16 19 23 26 27}

{2 10 16 23 24}

{8 9 14 19 22}

{2 7 19 21 26}

{0 4 8 9 18 19 21 22}

{2 5 7 10 15 16 24 25}

{6 8 9 12 14 19 26 27}

{0 2 9 10 18 19 21 22 23}

{0 2 9 19 21 23 26 27}

{5 10 17 18 22}

{1 3 6 11 12 15 20 25 28}

{0 4 8 9 12 19 21 26 27}

{4 5 7 8 18 19 20 21 22}

{6 7 8 12 14 19 20 25 26}

{3 5 11 15 17 20 25 28}

{1 2 6 7 15 16 24 25 26}

{5 7 15 20 25}

{6 11 14 16 25}

{5 7 8 14 19 20 22 25}

{2 5 7 10 14 16 19 22 25}

{2 9 10 14 16 19 22 23}

{6 8 11 12 14 20 25 28}

{3 5 10 11 15 16 17 24 25}

{0 9 10 13 17 18 22 23}

{3 11 13 17 28}

{5 10 11 14 16 17 22 25}

{2 6 7 14 16 19 25 26}

{6 8 9 11 12 13 14 27 28}

{3 10 11 13 16 17 23 24}

{4 8 12 20 28}

{2 5 7 10 18 19 21 22}

{9 10 11 13 14 16 17 22 23}

{6 9 11 13 14 16 23 27}

{1 6 7 12 15 20 25 26}

{5 8 11 14 17 20 22 25 28}

{0 4 8 9 13 17 18 22 28}

{0 4 8 9 12 13 27 28}

{1 3 6 11 13 16 23 24 27}

{1 6 12 26 27}

{8 9 11 13 14 17 22 28}

{4 5 8 17 18 20 22 28}

{1 2 6 16 23 24 26 27}

{0 9 13 23 27}

{4 7 8 12 19 20 21 26}

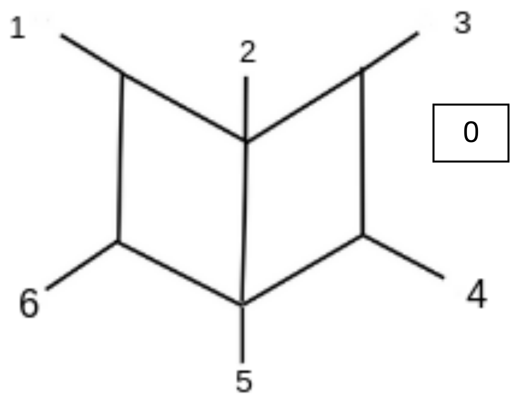
{1 3 6 11 12 13 27 28}

{0 4 18 21 29}

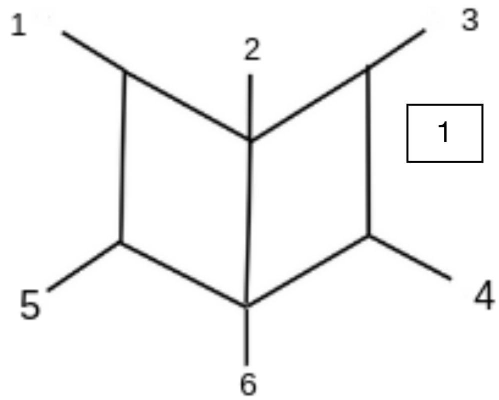
{1 3 15 24 29}

Here we have that the facets are color coded based upon the number of networks in each, with five networks in the facet being the blue highlighted, yellow being the facets of eight networks and pink being those that have nine networks in each.

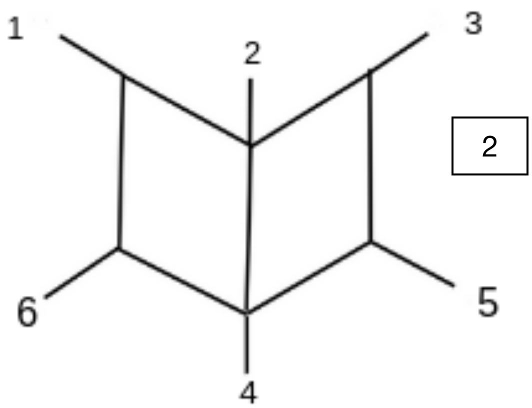
The following are the networks that make up the BME(6,0) polytope. They too are given along with their vector which is used later for the polymake input.



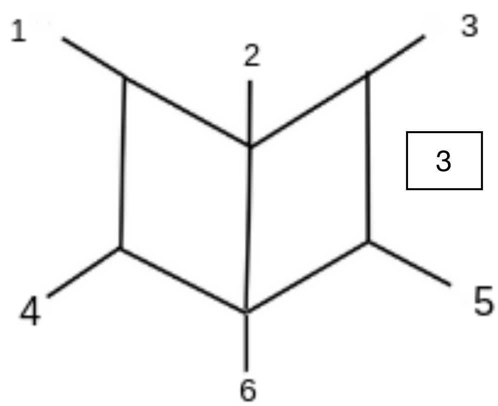
<1,0,0,0,1,1,0,0,0,1,0,0,1,0,1>



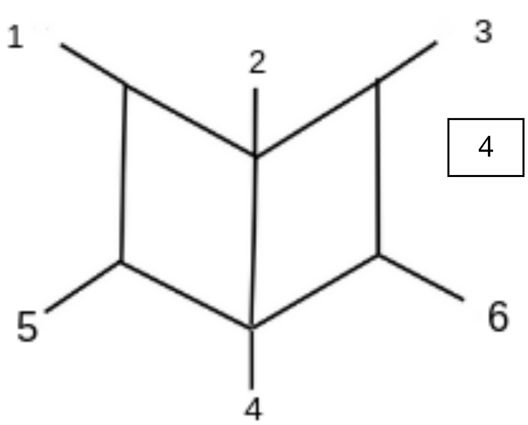
<1,0,0,1,0,1,0,0,0,1,0,0,0,1,1>



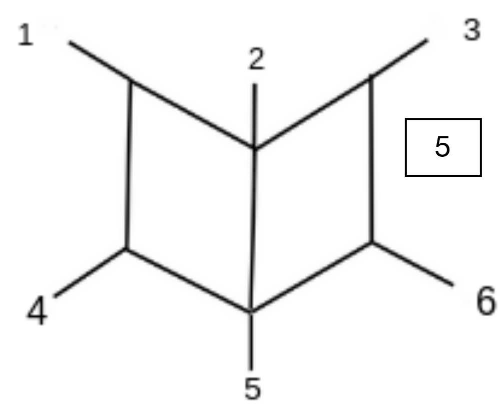
<1,0,0,0,1,1,0,00,0,1,0,1,1,0>



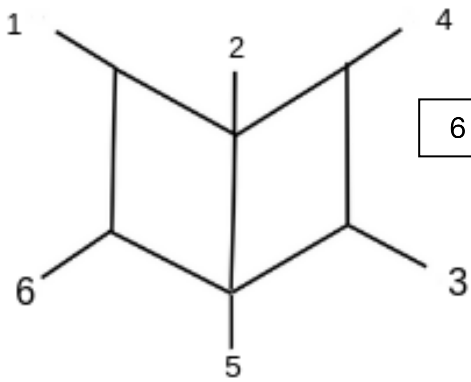
<1,0,1,0,0,1,0,0,0,0,1,0,0,1,1>



<1,0,0,1,0,1,0,0,0,0,0,1,1,1,0>

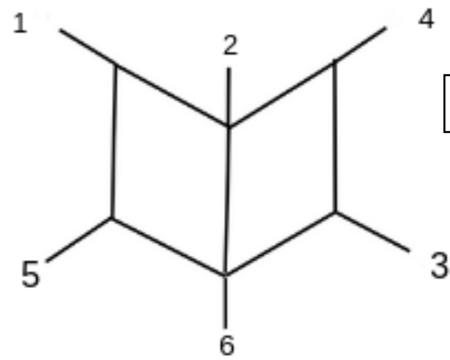


<1,0,1,0,0,1,0,0,0,0,0,1,1,0,1>



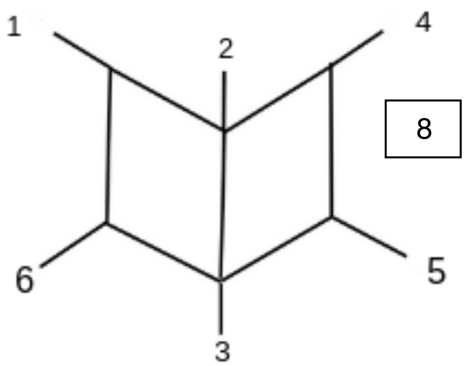
6

<1,0,0,0,1,0,1,0,0,1,1,0,0,0,1>



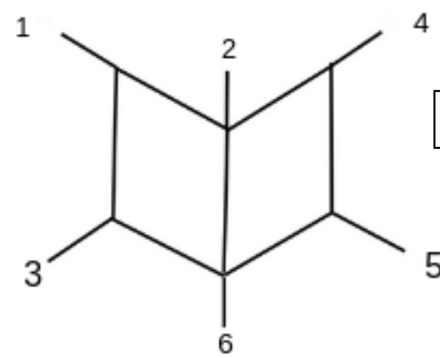
7

<1,0,0,1,0,0,1,0,0,1,0,1,0,0,1>



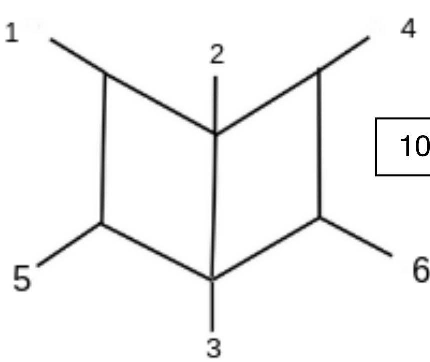
8

<1,0,0,0,1,0,1,0,0,0,1,1,1,0,0>



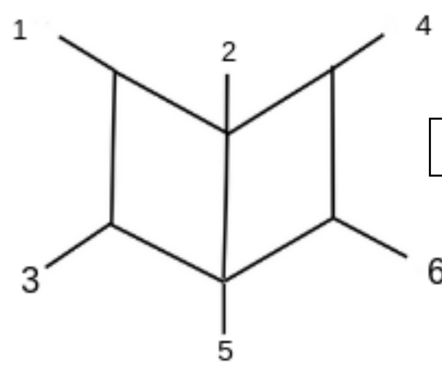
9

<1,1,0,0,0,0,1,0,0,0,0,1,1,0,1>



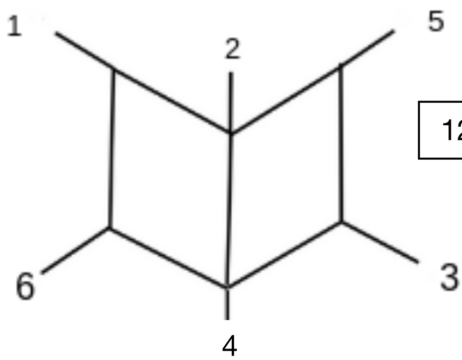
10

<1,0,0,1,0,0,1,0,0,0,1,1,0,1,0>



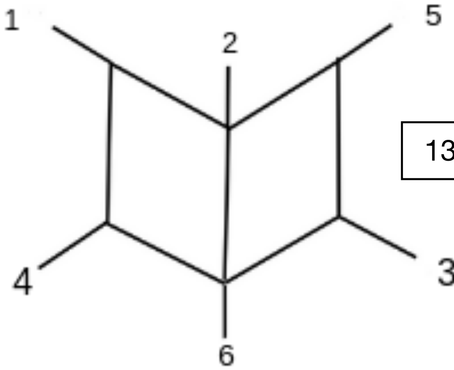
11

<1,1,0,0,0,0,1,0,0,0,1,0,0,1,1>



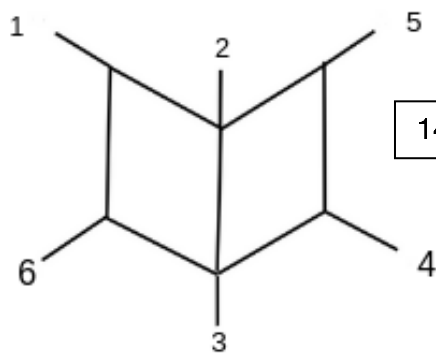
12

<1,0,0,0,1,0,0,1,0,1,1,0,0,1,0>



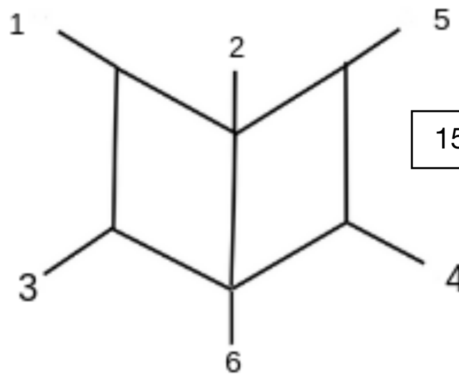
13

<1,0,1,0,0,0,0,1,0,0,1,1,0,1,0>



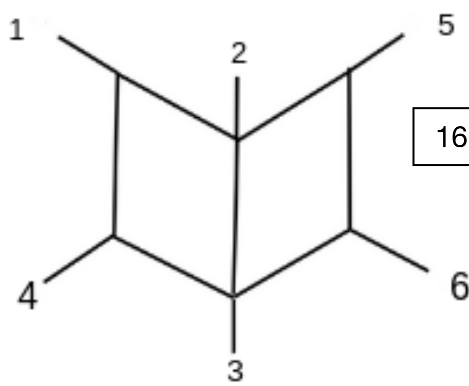
14

<1,0,0,0,1,0,0,1,0,1,0,1,1,0,0>



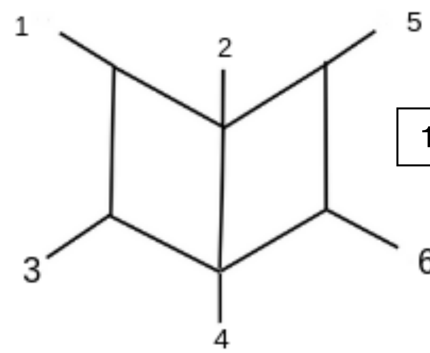
15

<1,1,0,0,0,0,0,1,0,0,0,1,1,1,0>



16

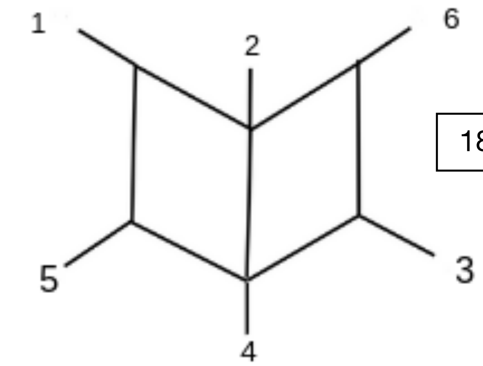
<1,0,1,0,0,0,0,1,0,1,0,1,0,0,1>



17

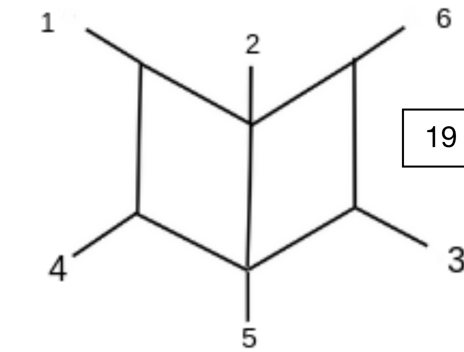
<1,1,0,0,0,0,0,1,0,1,0,0,0,1,1>

18



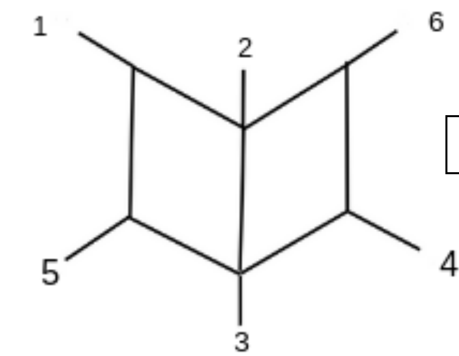
<1,0,0,1,0,0,0,0,1,1,0,1,1,0,0>

19



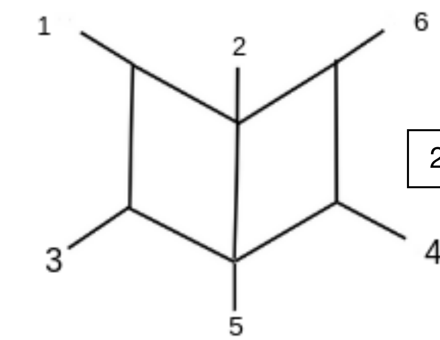
<1,0,1,0,0,0,0,0,1,0,1,1,1,0,0>

20



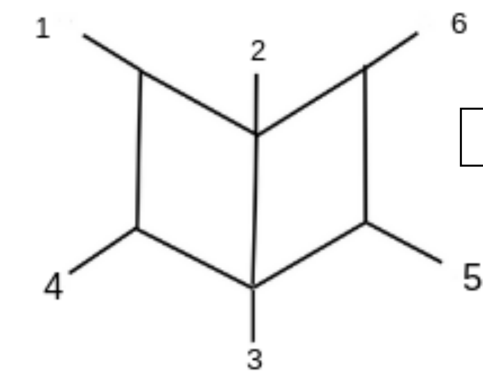
<1,0,0,1,0,0,0,0,1,1,1,0,0,1,0>

21



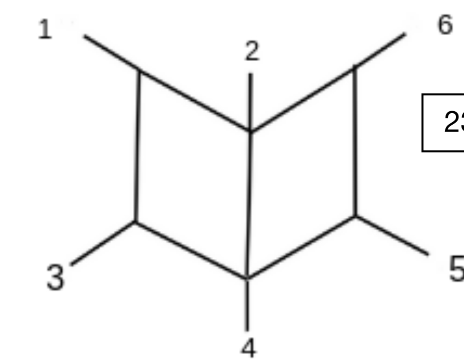
<1,1,0,0,0,0,0,0,1,0,1,0,1,1,0>

22



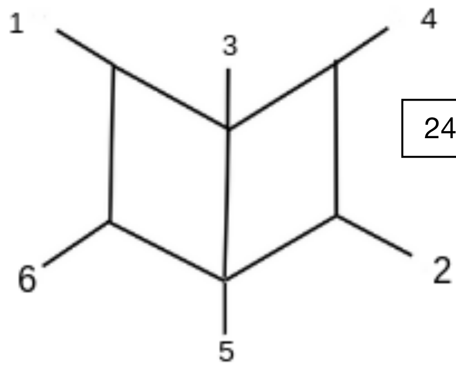
<1,0,1,0,0,0,0,0,1,1,1,0,0,0,1>

23



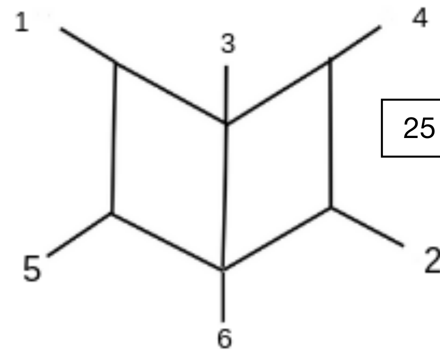
<1,1,0,0,0,0,0,0,1,1,0,0,1,0,1>

24



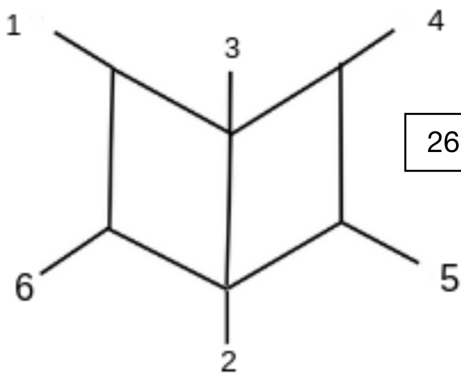
<0,1,0,0,1,0,1,1,0,1,0,0,0,0,1>

25



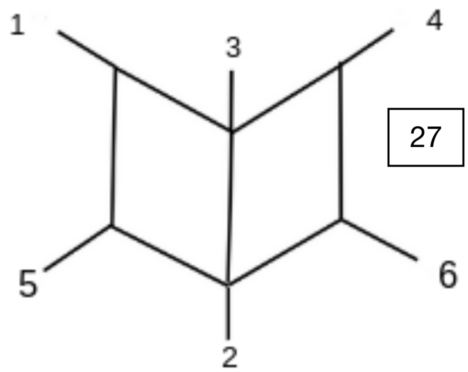
<0,1,0,1,0,0,1,0,1,1,0,0,0,0,1>

26



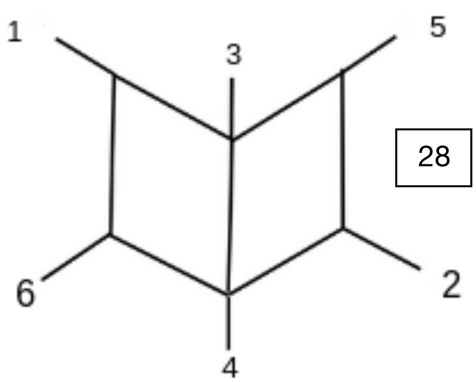
<0,1,0,0,1,0,0,1,1,1,0,0,1,0,0>

27



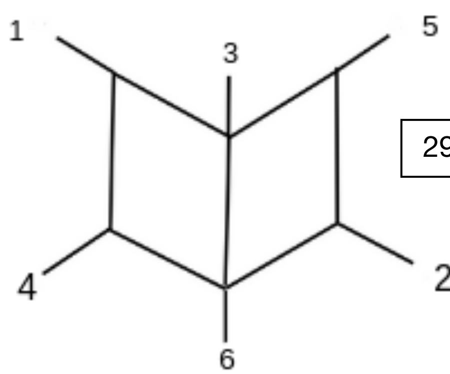
<0,1,0,1,0,0,0,1,1,1,0,0,0,1,0>

28

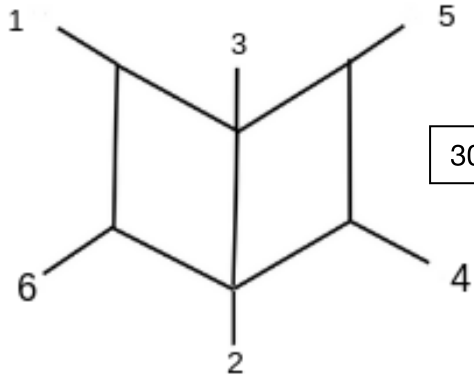


<0,1,0,0,1,0,1,1,0,0,1,0,0,1,0>

29

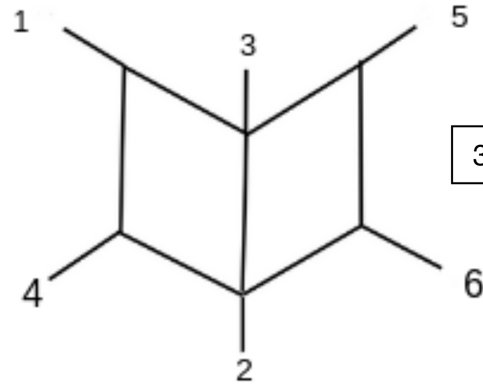


<0,1,1,0,0,0,0,1,1,0,1,0,0,1,0>



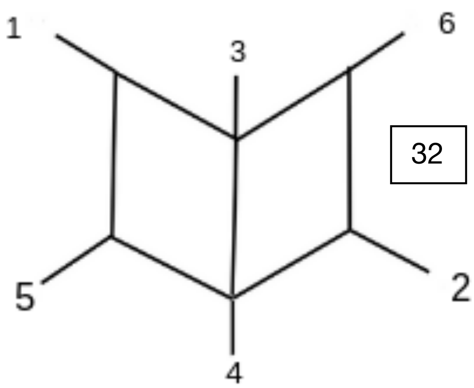
30

<0,1,0,0,1,0,1,0,1,0,1,0,1,0,0>



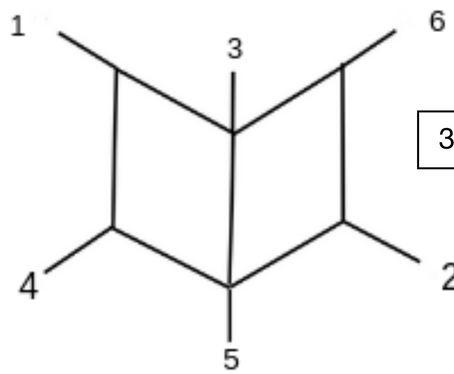
31

<0,1,1,0,0,0,1,0,1,0,1,0,0,0,1>



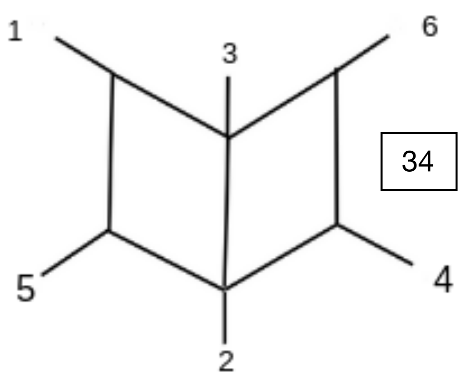
32

<0,1,0,1,0,0,1,0,1,0,0,1,1,0,0>



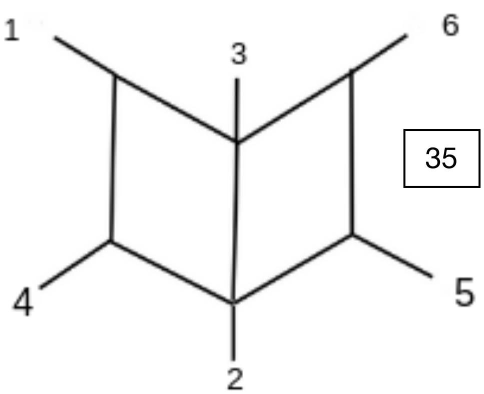
33

<0,1,1,0,0,0,0,1,1,0,0,1,1,0,0>



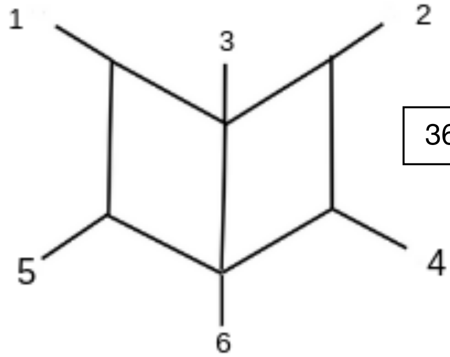
34

<0,1,0,1,0,0,1,1,0,0,0,1,0,1,0>



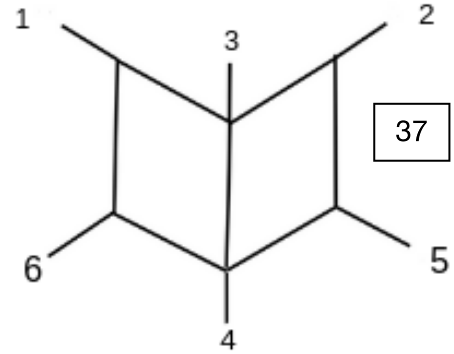
35

<0,1,1,0,0,0,1,1,0,0,0,1,0,0,1>



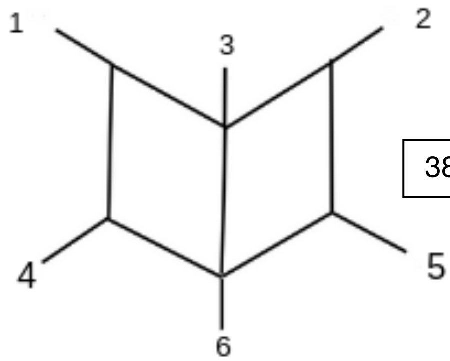
36

<0,1,0,1,0,1,1,0,0,0,0,0,0,1,1>



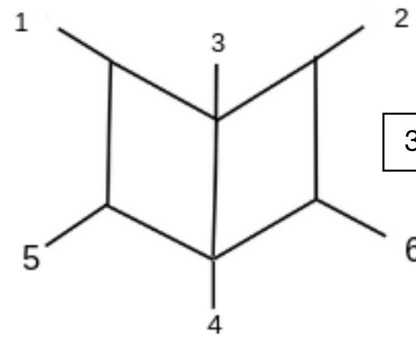
37

<0,1,0,0,1,1,0,1,0,0,0,0,1,1,0>



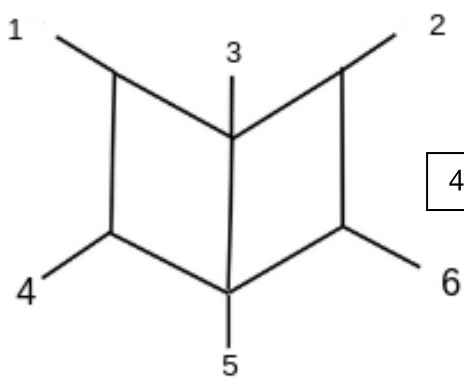
38

<0,1,1,0,0,1,0,1,0,0,0,0,0,1,1>



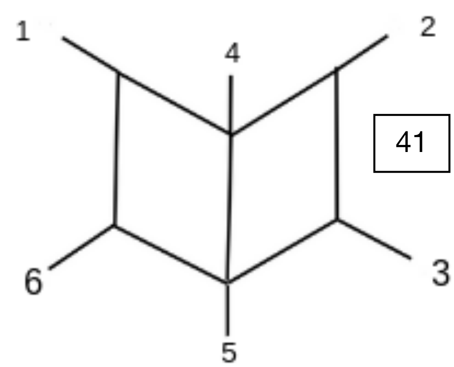
39

<0,1,0,1,0,1,0,0,1,0,0,0,1,1,0>



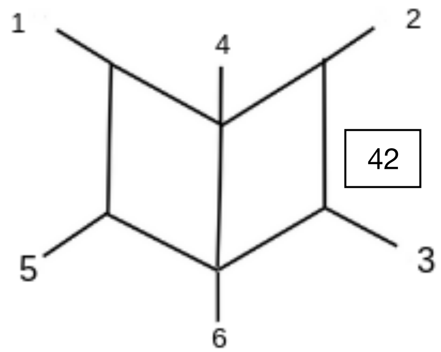
40

<0,1,1,0,0,1,0,0,1,0,0,0,1,0,1>



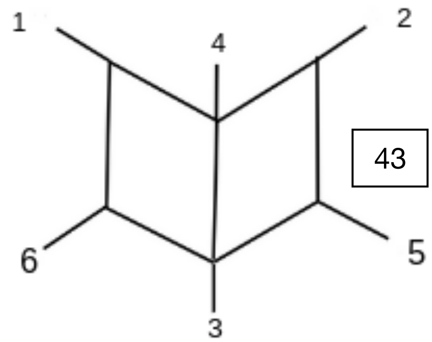
41

<0,0,1,0,1,1,1,0,0,0,1,0,0,0,1>



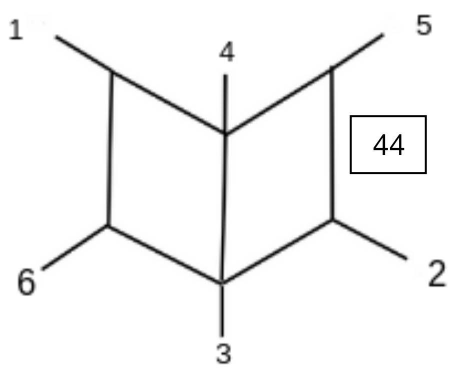
42

<0,0,1,1,0,1,1,0,0,0,0,1,0,0,1>



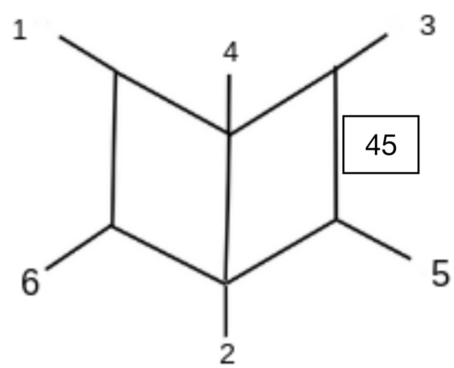
43

<0,0,1,0,1,0,1,1,0,0,1,1,0,0,0>



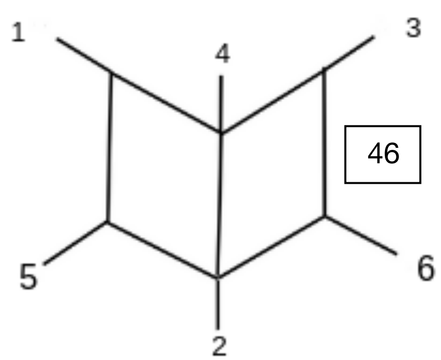
44

<0,0,1,0,1,1,0,1,0,0,0,1,1,0,0>



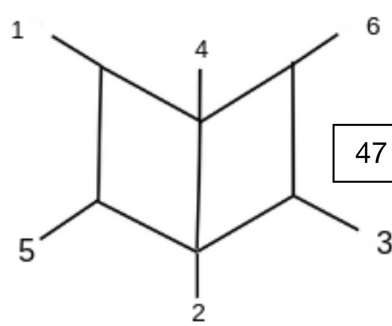
45

<0,0,1,0,1,0,0,1,1,1,1,0,0,0,0>



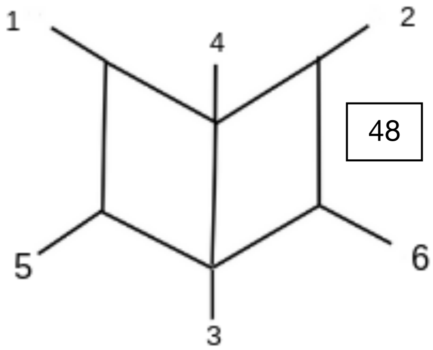
46

<0,0,1,1,0,0,0,1,1,1,0,1,0,0,0>



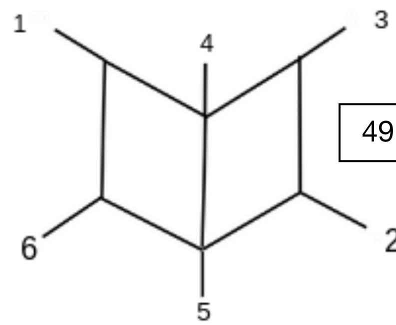
47

<0,0,1,1,0,1,0,1,0,0,0,1,0,1,0>



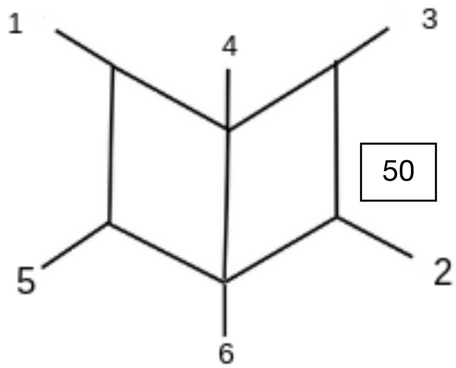
48

<10,0,1,1,0,0,1,0,1,0,1,1,0,0,0>



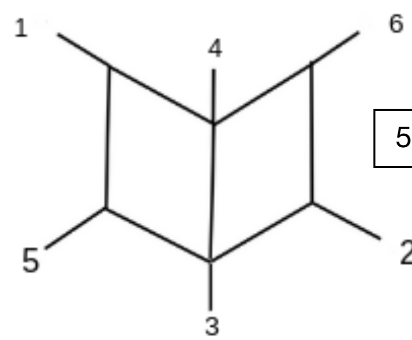
49

<0,0,1,0,1,1,0,1,0,1,0,0,0,0,1>



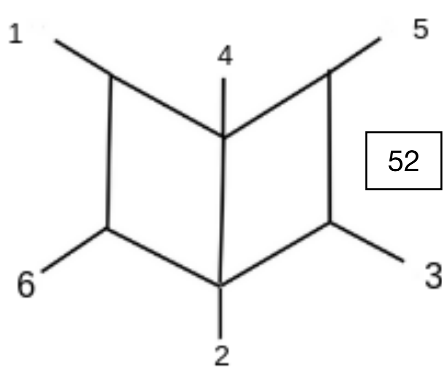
50

<0,0,1,1,0,1,0,0,1,1,0,0,0,0,1>



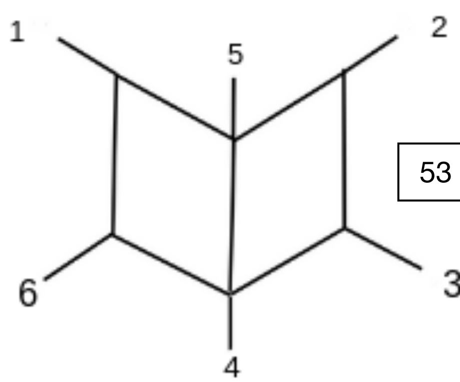
51

<0,0,1,1,0,1,0,0,1,0,1,0,0,1,0>



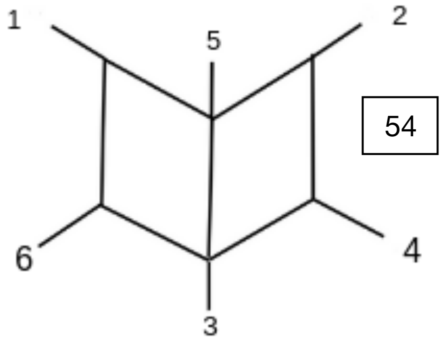
52

<0,0,1,0,1,1,0,0,1,0,1,0,1,0,(



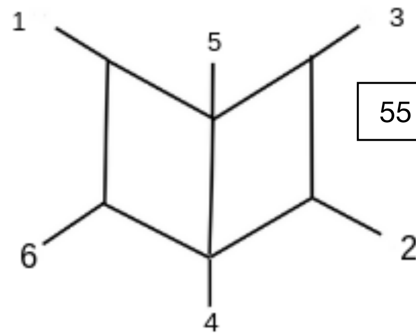
53

<0,0,0,1,1,1,0,1,0,1,0,0,0,1,0>



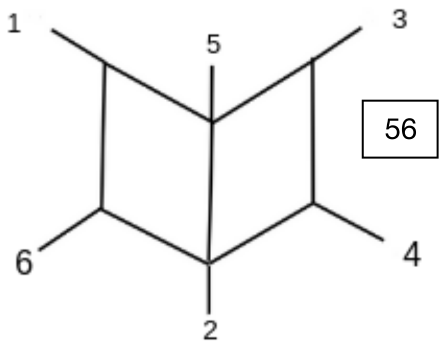
54

<0,0,0,1,1,0,1,1,0,1,0,1,0,0,0>



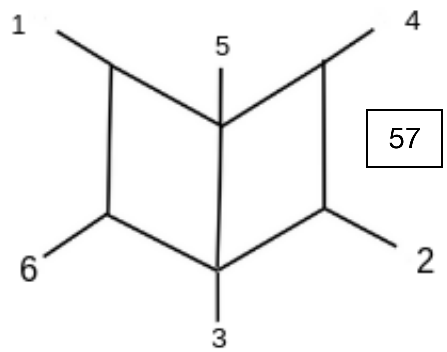
55

<0,0,0,1,1,1,1,0,0,0,1,0,0,1,0>



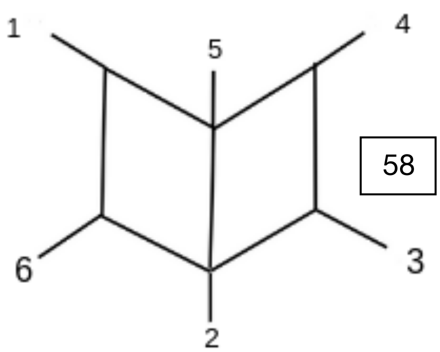
56

<0,0,0,1,1,0,1,0,1,1,1,0,0,0,0>



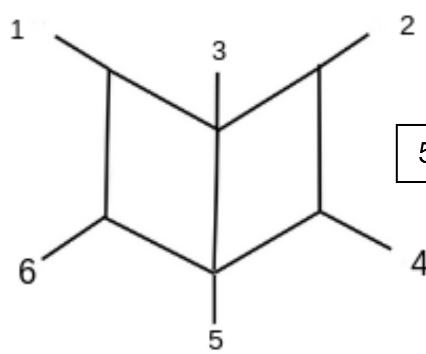
57

<0,0,0,1,1,1,1,0,0,0,0,1,1,0,0>



58

<0,0,0,1,1,1,0,0,1,1,0,0,1,0,0>



59

<0,1,0,0,1,1,1,0,0,0,0,0,1,0,1>

The following is the code that is input into Polymake for BME(6,0) to obtain the same information as what was provided for BME(5,1).

```
$points=new Matrix([[1,0,0,0,1,1,0,0,0,1,0,0,1,0,1],[1,0,0,1,0,1,0,0,0,1,0,0,0,1,1],
[1,0,0,0,1,1,0,0,0,0,1,0,1,1,0],[1,0,1,0,0,1,0,0,0,0,1,0,0,1,1],[1,0,0,1,0,1,0,0,0,0,0,1,1,1,0],
[1,0,1,0,0,1,0,0,0,0,0,1,1,0,1],[1,0,0,0,1,0,1,0,0,1,1,0,0,0,1],[1,0,0,1,0,0,1,0,0,1,0,1,0,0,1],
[1,0,0,0,1,0,1,0,0,0,1,1,1,0,0],[1,1,0,0,0,0,1,0,0,0,0,1,1,0,1],[1,0,0,1,0,0,1,0,0,0,1,1,0,1,0],
[1,1,0,0,0,0,1,0,0,0,1,0,0,1,1],[1,0,0,0,1,0,0,1,0,1,1,0,0,1,0],[1,0,1,0,0,0,0,1,0,0,1,1,0,1,0],
[1,0,0,0,1,0,0,1,0,1,0,1,1,0,0],[1,1,0,0,0,0,0,1,0,0,0,1,1,1,0],[1,0,1,0,0,0,0,1,0,1,0,1,0,0,1],
[1,1,0,0,0,0,0,1,0,1,0,0,0,1,1],[1,0,0,1,0,0,0,0,1,1,0,1,1,0,0],[1,0,1,0,0,0,0,0,1,0,1,1,1,0,0],
[1,0,0,1,0,0,0,0,1,1,1,0,0,1,0],[1,1,0,0,0,0,0,0,1,0,1,0,1,1,0],[1,0,1,0,0,0,0,0,1,1,1,0,0,0,1],
[1,1,0,0,0,0,0,0,1,1,0,0,1,0,1],[0,1,0,0,1,0,1,1,0,1,0,0,0,0,1],[0,1,0,1,0,0,1,0,1,1,0,0,0,0,1],
[0,1,0,0,1,0,0,1,1,1,0,0,1,0,0],[0,1,0,1,0,0,0,1,1,1,0,0,0,1,0],[0,1,0,0,1,0,1,1,0,0,1,0,0,1,0],
[0,1,1,0,0,0,0,1,1,0,1,0,0,1,0],[0,1,0,0,1,0,1,0,1,0,1,0,1,0,0],[0,1,1,0,0,0,1,0,1,0,1,0,0,0,1],
[0,1,0,1,0,0,1,0,1,0,0,1,1,0,0],[0,1,1,0,0,0,0,1,1,0,0,1,1,0,0],[0,1,0,1,0,0,1,1,0,0,0,1,0,1,0],
[0,1,1,0,0,0,1,1,0,0,0,1,0,0,1],[0,1,0,1,0,1,1,0,0,0,0,0,0,1,1],[0,1,0,0,1,1,0,1,0,0,0,0,1,1,0],
[0,1,1,0,0,1,0,1,0,0,0,0,0,1,1],[0,1,0,1,0,1,0,0,1,0,0,0,1,1,0],[0,1,1,0,0,1,0,0,1,0,0,0,1,0,1],
[0,0,1,0,1,1,1,0,0,0,1,0,0,0,1],[0,0,1,1,0,1,1,0,0,0,0,1,0,0,1],[0,0,1,0,1,0,1,1,0,0,1,1,0,0,0],
[0,0,1,0,1,1,0,1,0,0,0,1,1,0,0],[0,0,1,0,1,0,0,1,1,1,1,0,0,0,0],[0,0,1,1,0,0,0,1,1,1,0,1,0,0,0],
[0,0,1,1,0,1,0,1,0,0,0,1,0,1,0],[0,0,1,1,0,0,1,0,1,0,1,1,0,0,0],[0,0,1,0,1,1,0,1,0,1,0,0,0,0,1],
[0,0,1,1,0,1,0,0,1,1,0,0,0,0,1],[0,0,1,1,0,1,0,0,1,0,1,0,0,1,0],[0,0,1,0,1,1,0,0,1,0,1,0,1,0,0],
[0,0,0,1,1,1,0,1,0,1,0,0,0,1,0],[0,0,0,1,1,0,1,1,0,1,0,1,0,0,0],[0,0,0,1,1,1,1,0,0,0,1,0,0,1,0],
[0,0,0,1,1,0,1,0,1,1,1,0,0,0,0],[0,0,0,1,1,1,1,0,0,0,0,1,1,0,0],[0,0,0,1,1,1,0,0,1,1,0,0,1,0,0],
[0,1,0,0,1,1,1,0,0,0,0,0,1,0,1]]);
```

```
polytope > print $p->F_VECTOR;
60 1230 7245 18210 22642 14820 5265 1050 100
```

```
print $p->VERTICES_IN_FACETS;
```

{1 3 4 5 7 9 10 11 13 15 16 17 18 19 20 21 22 23 25 27 29 31 32 33 34 35 36 38 39 40 42 46 47 48 50 51}

{2 3 4 5 8 9 10 11 13 15 19 21 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 47 48 51 52 55 57 59}

{0 2 3 5 6 8 9 11 12 13 14 15 16 17 19 21 22 23 24 26 28 29 30 31 33 35 37 38 40 41 43 44 45 49 52 59}

{3 5 13 16 19 22 29 31 33 35 38 40 41 42 43 44 45 46 47 48 49 50 51 52}

{9 11 15 17 21 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 59}

{16 17 22 23 24 25 26 27 29 31 33 35 38 40 45 46 49 50}

{0 1 3 5 6 7 9 11 16 17 22 23 24 25 31 35 36 38 40 41 42 49 50 59}

{0 1 4 5 7 9 14 15 16 17 18 23 24 25 26 27 32 33 34 35 36 37 38 39 40 42 44 46 47 49 50 53 54 57 58 59}

{0 1 2 3 6 11 12 17 20 21 22 23 24 25 26 27 28 29 30 31 36 37 38 39 40 41 45 49 50 51 52 53 55 56 58 59}

{0 1 2 3 4 5 36 37 38 39 40 41 42 44 47 49 50 51 52 53 55 57 58 59}

{0 1 2 3 4 5 9 11 15 17 21 23 36 37 38 39 40 59}

{0 1 2 3 4 5 12 13 14 15 16 17 18 19 20 21 22 23 26 27 29 33 37 38 39 40 44 45 46 47 49 50 51 52 53 58}

{1 3 6 7 10 11 12 13 16 17 20 22 24 25 27 28 29 31 34 35 36 38 41 42 43 45 46 47 48 49 50 51 53 54 55 56}

{0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 24 28 34 35 36 37 38 41 42 43 44 47 49 53 54 55 57 59}

{2 3 12 13 28 29 37 38 41 43 44 45 47 49 51 52 53 55}

{12 13 14 15 16 17 24 26 27 28 29 33 34 35 37 38 43 44 45 46 47 49 53 54}

{1 2 3 4 10 11 12 13 15 17 20 21 27 28 29 34 36 37 38 39 47 51 53 55}

{21 26 29 30 33 37 39 40 52}

{0 3 5 22 40 41 49 50 52}

{4 5 18 19 32 33 39 40 42 44 46 47 48 50 51 52 57 58}

{9 19 21 23 30 31 32 33 40}

{0 26 37 40 44 49 52 58 59}

{0 2 4 5 8 9 14 15 18 19 21 23 26 30 32 33 37 39 40 44 52 57 58 59}

{0 1 2 3 4 5 6 7 8 9 10 11 18 19 20 21 22 23 25 30 31 32 36 39 40 41 42 48 50 51 52 55 56 57 58 59}

{0 5 6 7 8 9 14 16 18 19 22 23 24 25 26 30 31 32 33 35 40 41 42 43 44 45 46 48 49 50 52 54 56 57 58 59}

{23 25 26 30 32 39 40 58 59}

{3 5 6 7 8 9 10 11 13 16 19 22 31 35 41 42 43 48}

{8 9 14 15 24 26 28 30 32 33 34 35 37 43 44 54 57 59}

{4 5 7 8 9 10 13 14 15 16 18 19 32 33 34 35 42 43 44 46 47 48 54 57}

{0 2 6 8 12 14 24 26 28 30 37 41 43 44 45 49 52 53 54 55 56 57 58 59}

{7 9 11 17 24 25 34 35 36}

{7 25 32 34 35 42 46 48 54}

{0 12 14 17 24 26 37 49 53}

{6 7 8 9 10 11 24 25 28 30 31 32 34 35 36 41 42 43 48 54 55 56 57 59}

{5 9 14 15 16 18 19 23 33}

{4 8 9 10 15 18 19 21 32}

{2 4 8 10 12 13 14 15 18 19 20 21 26 27 28 29 30 32 33 34 37 39 43 44 45 46 47 48 51 52 53 54 55 56 57 58}

{22 31 41 45 48 50 51 52 56}

{15 18 21 23 26 27 32 33 39}

{18 25 26 27 32 46 54 56 58}

{1 3 4 10 36 42 47 51 55}

{13 29 33 34 35 43 46 47 48}

{3 11 12 13 17 20 21 22 29}

{6 9 11 23 24 25 30 31 59}

{6 24 28 30 31 41 43 45 56}

{12 15 17 21 26 27 28 29 37}

{4 34 36 37 39 47 53 55 57}

{16 35 42 43 44 46 47 49 54}

{20 27 45 46 50 51 53 56 58}

{14 26 44 45 46 49 53 54 58}

{1 2 3 6 10 11 12 20 55}

{10 34 42 43 47 48 54 55 57}

{18 19 20 21 22 23 25 26 27 29 30 31 32 33 39 40 45 46 48 50 51 52 56 58}

{7 18 20 22 25 46 48 50 56}

{19 29 30 31 33 43 45 48 52}

{1 2 4 20 39 51 53 55 58}

{0 1 6 7 12 14 16 17 18 20 22 23 24 25 26 27 45 46 49 50 53 54 56 58}

{0 1 3 6 12 16 17 22 49}

{4 7 9 10 32 34 36 42 57}

{13 14 16 19 33 43 44 45 46}

{0 1 2 3 4 5 6 7 8 10 12 13 14 16 18 19 20 22 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58}

{0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23}

{0 2 4 14 37 44 53 57 58}

{1 18 20 23 25 27 39 50 58}

{2 4 5 8 13 14 15 19 44}

{2 3 4 10 13 19 20 21 51}

{2 30 37 39 52 55 57 58 59}

{0 1 5 7 16 18 22 23 50}

{0 2 5 8 41 44 52 57 59}

{0 6 7 9 14 16 17 23 24}

{6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 43 45 46 48 54 56}

{2 6 8 11 28 30 41 55 59}

{8 30 41 43 48 52 55 56 57}

{9 25 31 32 35 36 40 42 59}

{3 19 21 22 29 31 40 51 52}

{0 1 2 4 6 7 8 9 10 11 12 14 15 17 18 20 21 23 24 25 26 27 28 30 32 34 36 37 39 53 54 55 56 57 58 59}

{1 4 7 10 18 20 25 27 32 34 36 39 42 46 47 48 50 51 53 54 55 56 57 58}

{13 19 20 22 29 45 46 48 51}

{2 3 6 8 10 11 12 13 19 20 21 22 28 29 30 31 41 43 45 48 51 52 55 56}

{7 8 10 18 32 48 54 56 57}

{2 8 10 11 12 13 15 21 28}

{1 4 7 10 12 13 14 15 16 17 18 20 27 34 46 47 53 54}

{1 6 7 11 17 20 22 23 25}

{6 8 10 12 28 43 54 55 56}

{0 2 6 8 12 14 18 19 20 21 22 23 26 30 45 52 56 58}

{0 1 6 7 24 25 36 41 42 49 50 53 54 55 56 57 58 59}

{6 12 14 16 24 43 45 49 54}

{10 11 20 21 25 27 28 29 30 31 32 34 36 39 48 51 55 56}

{12 24 26 27 28 45 53 54 56}

{0 4 5 7 8 9 14 18 57}

{1 27 36 38 39 47 50 51 53}

{17 24 27 28 34 36 37 38 53}

{11 13 15 17 28 29 34 35 38}

{1 3 5 16 38 42 47 49 50}

{15 27 29 33 34 37 38 39 47}

{11 24 28 31 35 36 38 41 59}

{5 13 15 16 33 35 38 44 47}

{5 35 38 40 41 42 44 49 59}

{3 31 36 38 40 41 42 50 51}

{24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59}

We have that the yellow are all of the facets that have 36 networks in each and there are fifteen of these. There are also fifteen pink and these are the facets that have 24 networks in each. The ten green facets are those that have eighteen networks in each and lastly the sixty blue facets are those that have nine networks in each.

BIBLIOGRAPHY

Forcey, Stefan, et al. “Facets of the Balanced Minimal Evolution Polytope.” 4 Aug. 2016.

Forcey, Stefan, et al. “The Minimum Evolution Problem in Phylogenetics: Polytopes, Linear Programming, and Interpretation.” 29 Jan. 2018.

Forcey, Stefan, et al. “Split-Facets for Balanced Minimal Evolution Polytopes and the Permutoassociahedron.” 15 Mar. 2017.

Grotschel, Martin. “On the Symmetric Travelling Salesman Problem: Inequalities.” *Mathematical Programming: North Holland Publishing Company.* 8 Nov. 1977.

Grotshel, M., and W. R. Pulleyblank. “Clique Tree Inequalities and the Symmetric Travelling Salesman Problem.” *Mathematics of Operations Research* , vol. 11, 4 Nov. 1986.

Levy, Dan and Pachter, Lior. “The Neighbor-Net Algorithm.” 6 Feb. 2018.