Counting All Possible Fan Ships of Anthropomorphic Dinosaurs

Draft version – 1 Aug 2024

Preamble

Often, the discussion arises of which of our favourite GVH dinosaurs would we like to see shipped together. Many unique and interesting combinations have been proposed, and so we may ask the question: exactly how many possible ways are there for use to ship them?

The answer to that question depends on exactly what kind of relationships are considered (couple or polycule), and what is being counted (e.g. only counting each possible relationship or looking at all parallel relationships in an AU?).

The results for counting for these different cases are considered below.

Note that this document is a work in progress, and may include errors or oversights. Any comments or suggestions are welcome.

1.1 Counting individual couples

If we have n dinosaurs to consider for constructing a couple, then there are (n-1) remaining dinosaurs for the other partner, giving $n \cdot (n-1)$ combinations. However, this is double-counting the two possible orderings of the couple, so we would need to divide by 2 to get only the unique combinations. Finally, we need to remove the one excluded sibling combination. This gives a final count of

$$\frac{n(n-1)}{2} - 1$$

For the standard 8-dinosaur group, this gives 8(8-1)/2 - 1 = **27** possible couples that can be formed.

Figure 1: Grid layout of possible couple combinations. There are 27 possibilities here, excluding the duplicates between the upper right / lower left triangles. (Image by Wolf Nanaki)

2.1 Counting individual polycules

Next, as seems to be the frequent case for this fandom, we will consider polycule relationships for the dinosaurs. The sibling exclusion criteria necessitates splitting the combinations into multiple sub-scenarios to avoid any issues. Thus, we will consider three cases: either Fang is in the polycule, or Naser is in the polycule, or neither is in the polycule.

For each of the Fang or Naser cases, we have (n-2) other dinos left who could choose to participate in the polycule. Since each one is given an independent yes/no choice of participating, there would be 2^{n-2} possible combinations of choices. However, we need to exclude the one combination where none of the other dinos chose to participate (can't have a polycule with just Fang or Naser with nobody else), so we're left with $2^{n-2}-1$ combinations for each of the Fang or Naser cases.

For the third sub-case (neither Fang nor Naser participating), we have again have (n-2) participants available, giving 2^{n-2} combinations of choices. Again, we have to exclude the one combination where nobody is participating. We also need to exclude the (n-2) combinations where only one of the (n-2) dinos chose to participate. This gives us $2^{n-2}-1-(n-2)$ combinations for this sub-case.

Summing up the combinations for each of those three sub-cases, we have a total count of

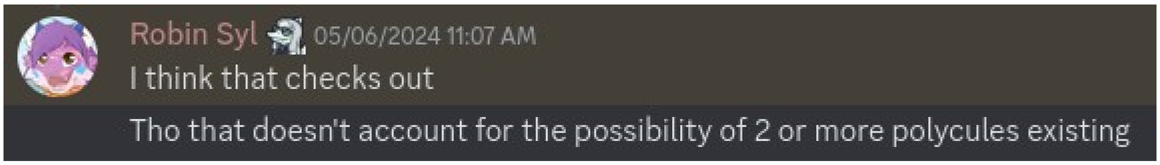
$$(2^{n-2}-1) + (2^{n-2}-1) + (2^{n-2}-1-(n-2)) = 3\cdot 2^{n-2}-n-1$$

For the standard 8-dinosaur group, this gives us a total of $3\cdot 2^{8-2}-8-1$ = **183** possible polycules that could be formed.

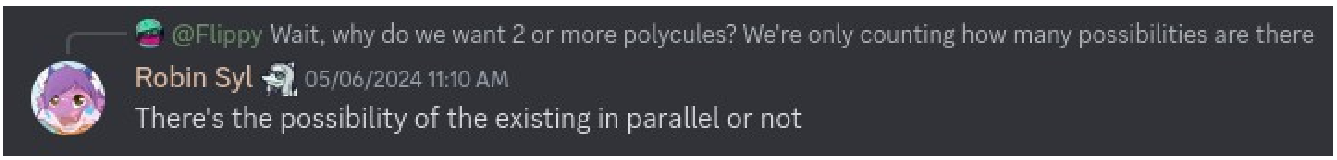
Table 1: Example polycules for each of the three sub-cases (with Fang, with Naser, without Fang or Naser). Other dinos have a yes/no binary choice of whether to participate.

2.2 Counting polycule ship lists

Now that we've counted all couples and polycule relationships, we're done right?



oh no



Ok... so far we've considered each polycule relationship on its own. However, there could be other dinosaurs left over (not in that relationship) that could form additional relationships within the same head-cannon. A list of ships (or a shipping manifest?). For instance,

How many such distinct lists could be formed? This turns out to be more complicated to calculate, with no simple formula to express it. However, there is a way to compute the answer algorithmically in polynomial (O(n^2)) time.

First, we need to define the method by which we'll construct the ships. Note that we will consider the case with *no* ships to be a valid case, because it simplifies all the math. This *null shipping* case can be excluded after the fact, by subtracting one from the final count at the very end of this algorithm.

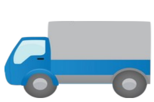
To start, we take Fang and Naser and put them each into their own box on a factory floor.

This will ensure that the sibling exlusion criteria is met by keeping them out of the same polycule. It also has the added benefit of stopping Naser from making puns for a while.

Next, we put the rest of the dinos in a pre-determined order on a moving conveyor belt. When each dino reaches the end of the conveyor belt, they will see some open boxes on the floor. Each box (including Fang’s and Naser’s) contains one or more dinos already in the process of becoming a shipment. The dino on the conveyor belt can choose to either jump into one of the existing boxes, or request a new, empty box to jump into.

Get in the box, Trish.

After all dinos have taken their turns jumping into boxes, we inspect all the contents. Any box containing just a single dinosaur does not make a valid ship, and that dino can jump back out of the box and go about their day. Each of the rest of the boxes constitutes a valid relationship. We seal up the relationship boxes and ship them all out.



Before doing the math on this, we should verify that this process can be used to construct all possible shipping manifests, and tha each combination of choices yields a unique result (otherwise we’d be double-counting ships or missing some combinations).

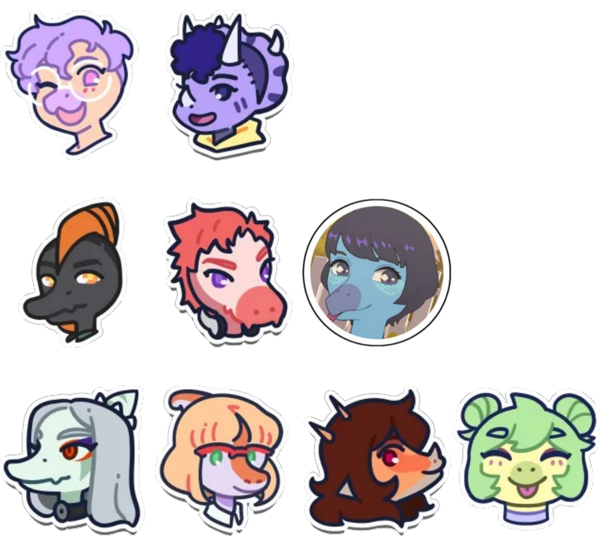
First, we will pre-determine an order for the line of dinos jumping into the boxes:

Alvin sticker by Josephvan

Wait, how did Alvin get there? Anyway, let’s supposed we have a particular ship list to construct, such as the one mentioned earlier:

Teehee 05/06/2024 1:49 PM
Here is my updated agenda folks:
Trisage
Aleeser
Four Femmes
totally canon, and everyone is taken.

What we first do is order the participants within each ship, relative to the predetermined total ordering. In this example we would have:



The leading member of each relationship (in this case, Sage, Naser, and Fang) will need to make the choice of starting a new box when they get to the end of the conveyor belt, because nobody ahead of them in line (who already jumped into boxes) would be part of their relationship. The subsequent members of each relationship would then need to choose to jump into the same box as the relationship leader. For example: Naser would need to start a new box, Reed would need to choose the same box as Naser, and Alvin would also need to choose the same box as Naser.

Putting all these choices into the line order, we would have the following:

: Start a new box (Box 1)
: Start a new box (Box 2)
: Same box as Fang (Box 1)
: Same box as Naser (Box 2)
: Same box as Fang (Box 1)
: Start a new box (Box 3)
: Same box as Fang (Box 1)
: Same box as Sage (Box 3)
: Same box as Naser (Box 2)

Note that we were not left with any arbitrary choices to make for the dinosaurs; Their choices were entirely dictated by which polycule they were a member of (or, if they were not in a relationship, they would need to choose to start a new box on their own). Fang and Naser are forced to start out in the first two boxes separately, since they are at the beginning of the line and never form part of the same relationship.

This process outlined above should also be general enough that it could be applied to any list of ships – just follow the recipe, and we will get a list of boxes to put the dinosaurs in. So, if we can find a way to count the combinations of all possible choices for this relationship factory, then this would correspond to the total number of possible parallel polycule combinations.

So now for the math. Let's suppose we are in the middle of the relationship creation process, and have some number of shipping boxes already started (b), and some number of dinosaurs remaining on the conveyor belt (r). Ignoring the choices that were already made to get us where we currently are, how many possible combinations of choices do we have *left*, that the dinosaurs still on the conveyor belt can still make?

For lack of imagination, let's call this partial counting function $f(b,r)$.

For the degenerate case where there are no dinosaurs remaining on the conveyor belt, there are no further choices to make, so we have a single combination (whatever's already in the boxes), so $f(b,0) = 1$.

For the cases where $r > 0$, we can find a recurrence relation to f with smaller values of r, as follows: The next dinosaur on the conveyor belt can choose to jump into one of the b existing boxes, after which there are (r-1) remaining dinosaurs left to make $f(b,r-1)$ possible remaining choices. Or, the dinosaur could choose to start their own new box, after which there would be (r-1) remaining dinosaurs, now with (b+1) boxes available, giving $f(b+1,r-1)$ possible remaining choices. Summing up the combinations from both cases (jumped in one of the existing b boxes, or starting new box), we have the recurrence equation:

$$f(b,r) = b \cdot f(b,r-1) + f(b+1,r-1)$$

We can now start filling out a table of values for f. We can start with the degenerate values $f(b,0) = 1$:

Pre-existing boxes *b*

Remaining dinosaurs *r*	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1									
2									
3									
4									
5									
6									
7									
8									

We can they apply the recurrence relation $f(b,r) = b \cdot f(b,r\text{-}1) + f(b+1,r\text{-}1)$ to fill out the next row:

Pre-existing boxes *b*

Remaining dinosaurs *r*	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	1	2	3	4	5	6	7	8	
2									
3									
4									
5									
6									
7									
8									

.. and the next row:

Pre-existing boxes *b* (columns); Remaining dinosaurs *r* (rows)

	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	1	2	3	4	5	6	7	8	
2	2	5	10	17	26	37	50		
3									
4									
5									
6									
7									
8									

Ex: 37 = 5·6+7

... and so on, until we have enough values for our dinosaur group size:

Pre-existing boxes *b* (columns); Remaining dinosaurs *r* (rows)

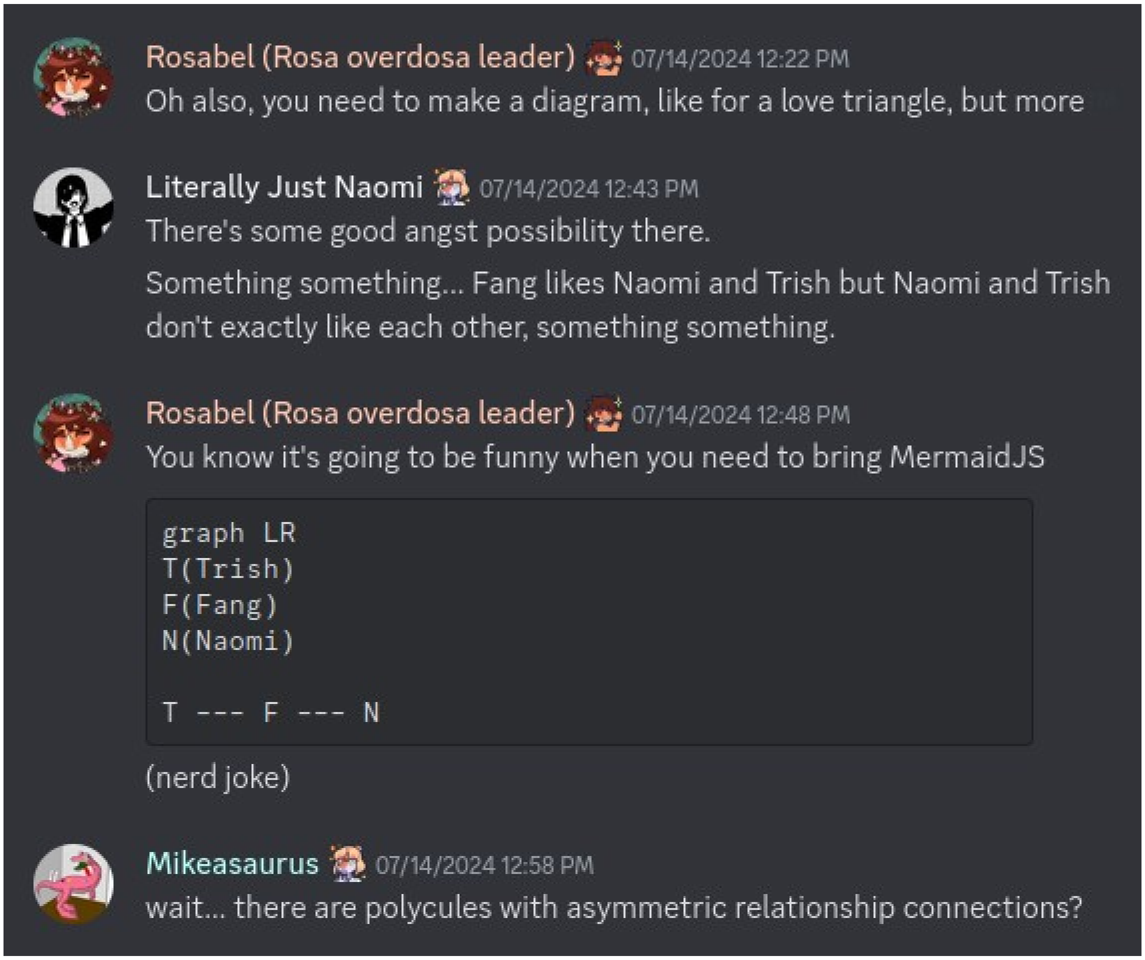
	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	1	2	3	4	5	6	7	8	
2	2	5	10	17	26	37	50		
3	5	15	37	77	141	235			
4	15	52	151	372	799				
5	52	203	674	1915					
6	203	877	3263						
7	877	4140							
8	4140								

Ok, that was a bit of work, but we're almost done. Now, what is our final count? Well, in our case we have to pre-allocate Fang and Naser into the first two boxes, so we start with 2 existing boxes. We have 6 remaining dinosaurs, so we just need to get the number of possible choices from the table (at r=6, b=2). This gives an answer of 3263 choices, including the *null* ship. Excluding that boring possibility, we arrive at the grand total of **3262** possible sets of parallel polycule relationship combinations for our standard 8-dinosaur group.

And now, we are definitely done!

...right?

3.1 Counting generalized polycules



Rosabel (Rosa overdosa leader) 07/14/2024 12:22 PM
Oh also, you need to make a diagram, like for a love triangle, but more

Literally Just Naomi 07/14/2024 12:43 PM
There's some good angst possibility there.
Something something... Fang likes Naomi and Trish but Naomi and Trish don't exactly like each other, something something.

Rosabel (Rosa overdosa leader) 07/14/2024 12:48 PM
You know it's going to be funny when you need to bring MermaidJS

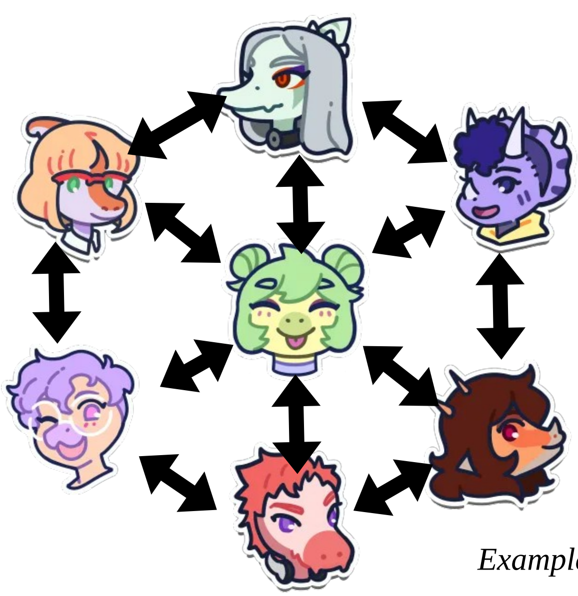
```
graph LR
T(Trish)
F(Fang)
N(Naomi)

T --- F --- N
```

(nerd joke)

Mikeasaurus 07/14/2024 12:58 PM
wait... there are polycules with asymmetric relationship connections?

Whoops... I didn't take into account the complexity of polycule relationships. I assumed that all people in the polycule would have a direct connection to every other person, but that's not necessarily the case. This opens up many possible combinations of connections.



Example: "The wheel of fortune"

How to even begin counting all these possibilities? We'll define two functions to help us along the way. Let's assume we have *n* characters available.

First, let's define *p(n)* to be the number of ways to build a polycule with exactly those *n* characters with nobody left out (and assuming no restrictions on who can participate... maybe lock Naser away in the basement for now).

Also, let's define *u(n)* to be the number of ways to connect those *n* characters in general, without requiring that they form a coherent polycule (the *u* stands for allowing *un*connected). There could be multiple independent relationships in there, and people left out of relationships entirely).

u(n) is actually easy to derive. There are $\binom{n}{2}$ possible interconnections for a graph of size *n*, and we're allowing each interconnection to have the choice of being there or not. This gives a total number of combinations of

$$u(n) = 2^{\binom{n}{2}}$$

There's another way to count *u(n)* as well. Let's pick one of the *n* characters as a reference character, let's say Fang. We group all the relationship possibilities into categories based on how many people Fang is in a relationship with. In our random mess of connections in *u(n)*, Fang could find themselves alone and unconnected, or in a relationship with 1, 2, ..., *(n-1)* other people.

How many of the *u(n)* combinations result in Fang being alone (connected to zero other people)? Well, putting Fang off to the side, there are *(n-1)* other characters who can be arbitrarily connected in *u(n-1)* different ways, so for case "0" we have a count of *u(n-1)* possibilities.

How many of the *u(n)* combinations result in Fang being connected to exactly one other person? There are *(n-1)* possible people Fang could be in that relationship with, and then the other *(n-2)* people have *u(n-2)* possible ways to be arbitrarily connected amongst themselves. So for case "1" we have a count of
(n-1)·u(n-2) possibilites.

More generally, how many of the *u(n)* combinations result in Fang being connected to exactly *m* other people ($0 \le m < n$)? There are $\binom{n-1}{m}$ ways to choose *m* of those people from the *(n-1)* total available. Then there are *p(m+1)* possible ways of arranging that polycule of Fang plus *m* others. The remaining *(n-m-1)* other people can be arbitrarily connected in *u(n-1-m)* different ways. So for case "*m*" we have $\binom{n-1}{m} \cdot p(m+1) \cdot u(n-1-m)$ possibilities.

Summing up all these cases, we can calculate the total number of arbitrary interconnections *u(n)* as:

$$u(n) = \sum_{m=0}^{n-1} \binom{n-1}{m} \cdot p(m+1) \cdot u(n-1-m)$$

where we define the degenerate case *p(1)* = 1 to make the math work for case “0”.

We already know that *u(n)* = $2^{\binom{n}{2}}$, so plugging that in we have:

$$2^{\binom{n}{2}} = \sum_{m=0}^{n-1} \binom{n-1}{m} \cdot p(m+1) \cdot 2^{\binom{n-1-m}{2}}$$

Rearranging that, we can get the recursive formula:

$$p(n) = 2^{\binom{n}{2}} - \sum_{m=0}^{n-2} \binom{n-1}{m} \cdot p(m+1) \cdot 2^{\binom{n-1-m}{2}}$$

Starting with *p(1) = 1*, we can use that formula to find higher values, e.g.

$$p(2) = 2^{\binom{2}{2}} - \binom{2-1}{0} \cdot p(1) \cdot 2^{\binom{2-1-0}{2}}$$

$$= 2^1 - \binom{1}{0} \cdot p(1) \cdot 2^{\binom{0}{}}$$

$$= 2 - 1 \cdot 1 \cdot 1$$

$$= 1$$

$$p(3) = 2^{\binom{3}{2}} - \binom{3-1}{0} \cdot p(1) \cdot 2^{\binom{3-1-0}{2}} - \binom{3-1}{1} \cdot p(2) \cdot 2^{\binom{3-1-1}{2}}$$

$$= 2^3 - \binom{2}{0} \cdot p(1) \cdot 2^1 - \binom{2}{1} \cdot p(2) \cdot 2^0$$

$$= 8 - 1 \cdot 1 \cdot 2 - 2 \cdot 1 \cdot 1$$

$$= 4$$

Continuing this process, here’s a table of the first 7 values pf *p(n)*:

Number of people in the relationship *n*	Number of ways to form the relationship *p(n)*
1	1
2	1
3	4
4	38
5	728
6	26704
7	1866256

So, for instance, if we keep Naser locked in the basement and put all 7 remaining characters in a single polycule, there are 1866256 ways to do that (with people being either directly connected to each other or indirectly connected via someone else within that polycule). There are other cases we need to count though, too. There could be less than 7 people in the relationship, or Naser could be in the relationship instead of Fang, etc. So there's a bit more counting that needs to be done.

We can follow a similar approach to section 2.1 to break the possibilities into 3 cases: Either Fang is in the relationship, or Naser is in the relationship, or neither Fang nor Naser are in the relationship.

If Fang is in the relationship: assume there are m other people in the relationship as well ($1 \leq m \leq 6$). There are $\binom{6}{m}$ ways to choose those other people, and $p(m+1)$ ways to interconnect the relationship.

This gives the count:

# of combos with Fang: $\sum_{m=1}^{6} \binom{6}{m} p(m+1) = 2038226$

by symmetry, we would also have:

# of combos with Naser: $\sum_{m=1}^{6} \binom{6}{m} p(m+1) = 2038226$

For the third case, where neither Fang nor Naser are in the relationship, assume that m of the remaining people form a relationship ($2 \leq m \leq 6$). There are $\binom{6}{m}$ ways to choose those people, and $p(m)$ ways to interconnect the relationship. This gives the count:

# of combos without Fang or Naser: $\sum_{m=2}^{6} \binom{6}{m} p(m) = 31737$

Summing these all up, we get the grand total of

2038226 + 2038226 + 31737 = 4108189

4108189 possible ways to form a relationship (including generalized polycules).