Optimal Solution for the Number of Kobon Triangles with Eleven Lines

Kyle Wood
9/24/24

https://web.archive.org/web/20171111045109/http://www.tik.ee.ethz.ch/sop/publicationListFiles/cb2007a.pdf

The above paper proves the following:

"If (n mod 3) $\in \{0, 2\}$, then all configurations
that meet the upper bound (1) are perfect configurations. In
these cases $n(n-2) \equiv 0 \bmod 3$ hence $K(n) = n(n-2)/3$."

"But these perfect configurations are only
possible for odd n according to Lemma 3. Therefore for n
mod 3 $\in \{0, 2\}$ and $n \equiv 0 \bmod 2$ the upper bound cannot
be reached. These two conditions can be summarized as n
mod 6 $\in \{0, 2\}$."

For the case of k = 11, the upper bound referenced is 33. If it were shown that there is no perfect configuration that has 33 triangles, then it would be shown that there are no configurations that can meet this upper bound whatsoever, and that the current best arrangement containing 32 triangles is the best possible (along with any alternatives that have 32 triangles).

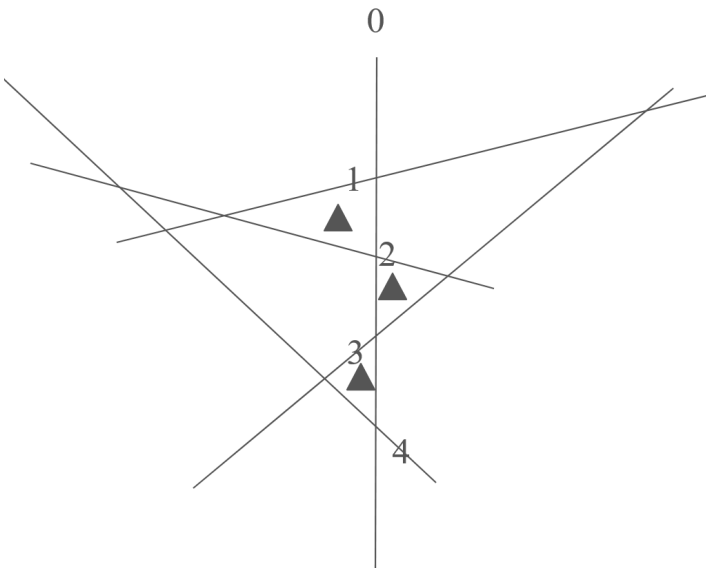

FIGURE 1: a model for k = 5 (not a perfect arrangement, but the base for one)

All lines in a perfect arrangement are not parallel to any other (rule 1), and no point contains 3 lines (rule 2). The number of line segments for a given k, n(n − 2), is 99 for k = 11, and since no segment can be the side for two triangles in such an arrangement, each line will be the base for k - 2 triangles alternating on either side of the line. FIGURE 1 shows an arrangement for k = 5. For the purpose of classifying different arrangements, the first line, line “0”, will be vertical, and the labels for all other lines will be determined by the order from top to bottom in which they intersect line 0. Also, the triangle made with lines 0, 1 and 2 will be on the left side of the line.

Now, for any arrangement that meets rule 1 and 2 which has at least one line with alternating triangles, it may be described by being oriented similarly to FIGURE 1, and then for each line, recording the order from left to right in which it intersects all other lines (line 0 is entered in order from 1 to k - 1, and is the same for all sets as this is how the other lines are defined. Therefore, it is omitted to cut out redundancy). For FIGURE 1, for example, line 1’s entry would be 4, 2, 0, 3, line 2’s would be 4, 1, 0, 3, line 3’s 4, 0, 2, 1, and line 4’s 2, 1, 3, 0. The amount of triangles for a given record can be derived simply without needing to physically “draw” the lines. For instance, lines 1 2 and 4 make a triangle because line 1’s entry has 2 and 4 next to each other, line 2’s has 1 and 4 next to each other, and line 4’s has 1 and 2.

With this configuration, however, there are many records that cannot actually be physically constructed, such as 1: 0, 2, 3, 4 - 2: 0, 1, 3, 4 - 3: 0, 1, 2, 4 - 4: 0, 1, 2, 3

If the goal is to be able to generate all valid sets, the first step is to arrange a list of the intersection points themselves as they appear from left to right. The points with 0 in them can be collectively represented simply by a 0. For example, FIGURE 1 would be described [(2, 4), (1,

2), (3, 4), 0, (2, 3), (1, 3)]. The order of the points which are a part of the triangles whose base contains the 0 line can be ignored; regardless of how they are ordered, they will result in the same record for the individual line records. Therefore, they can also be represented by the 0, shortening the list to [(2, 4), 0, (1, 3)].

In the case where k = 5, all orderings of the values in the set [(2, 4), 0, (1, 3)] result in valid arrangements (ones that can be physically drawn with the given restrictions), but note that this isn't the case for higher values of k.

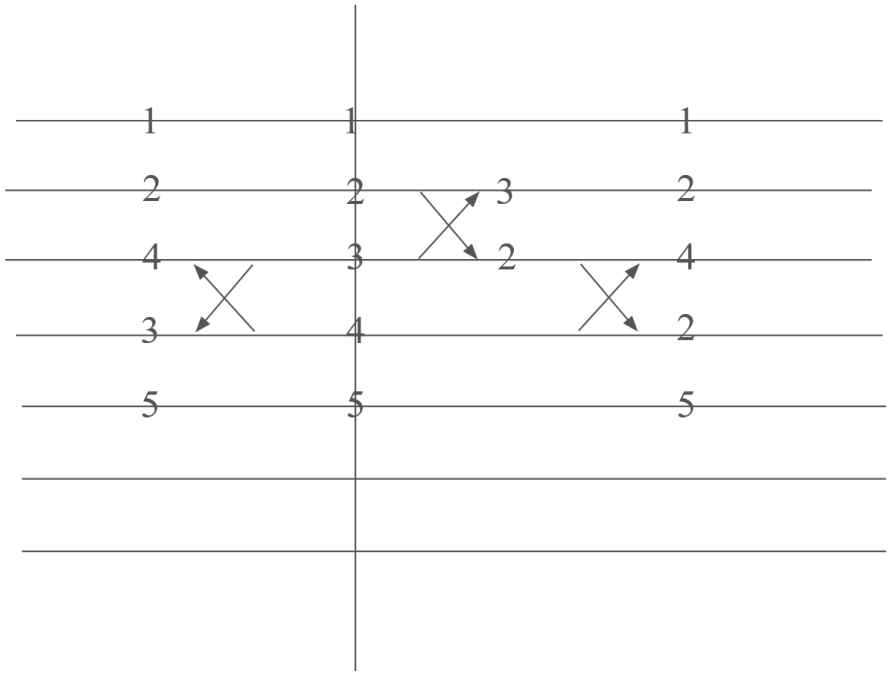

FIGURE 2: model representing proper swaps

To have two lines intersect, they must be vertically next to each other. In FIGURE 2, each line apart from line 0 is represented as a horizontal line, and their labels are "swapped" one at a time. This represents the two lines intersecting, and helps show when it is and isn't possible to have an intersection at a given point in the sequence. Lines 2 and 4 cannot intersect in FIGURE 2 until they are next to each other, which happened when lines 2 and 3 had intersected. Lines 1 and 5 cannot intersect on either side yet. If the order of 2 or more sequential swaps do not contain any lines in common, that can be done at the same time, as the resulting record for individual lines is unaffected.

By generating according to this process, all the resulting sets will have no parallel lines and no point containing 3 lines, as long as every combination of possible points is used by the end.

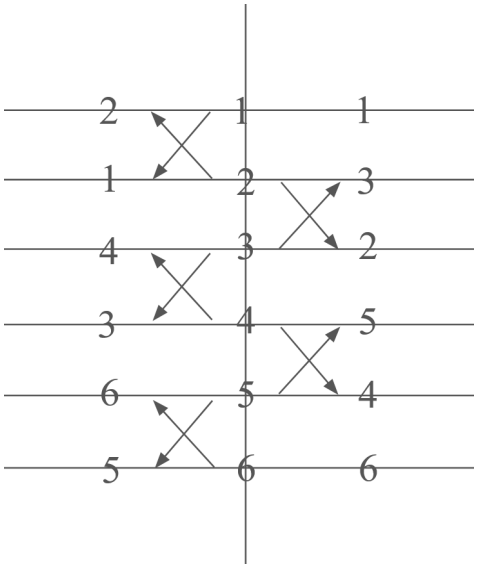

FIGURE 3: the set swaps that are made for any perfect configuration, demonstrated with k = 7

To generate exclusively perfect sets, that is, sets that follow the previous rules used to generate this set along with each segment being the side for exactly 1 triangle, one simple rule can be added. When making a swap, take note of the new polygon which was enclosed. Looking at FIGURE 3, which has the starting moves made for k = 7, the available moves on the right side are (1, 3), (2, 5), and (4, 6). Let's say the next move taken is (2, 5). This encloses a pentagram, with corners (2, 5), (2, 3), (0, 3), (0, 4), and (4, 5). Any time a move is taken which encloses a polygon other than a triangle, every side of the polygon must be the base for a different triangle. In this case, there is already a triangle for lines 0, 2, 3, lines 0, 3, 4, and lines 0, 4, 5, but the swaps (3, 5) and (2, 4) are now required to resolve the pentagon.

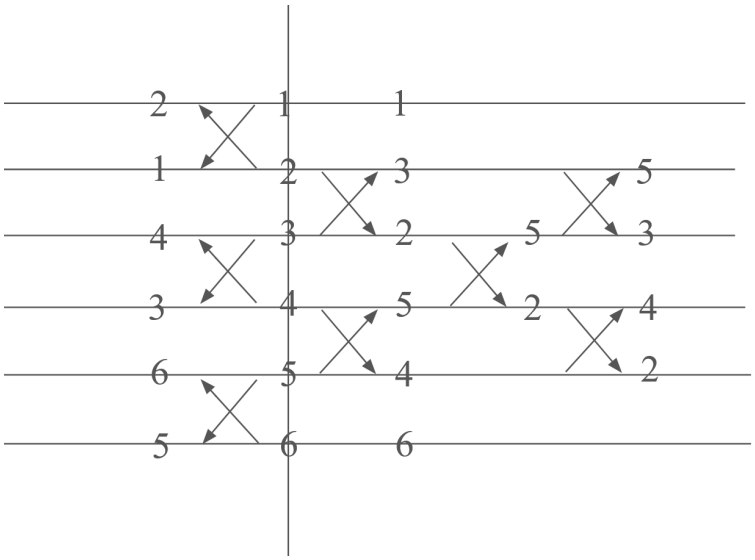

FIGURE 4: model with a primary swap and the two required secondary swaps

In the case here, as shown by FIGURE 4, swaps (3, 5) and (2, 4) are required to resolve this, creating triangles with lines 2, 3, 5 and lines 2, 4, 5. If (3, 5) or (2, 4) had already been used, or if either secondary swap created a polygon other than a triangle, then the primary swap, (2, 5), would not be allowed.

The lines at the top and bottom can never be used for primary swaps, and can only be swapped when required for a secondary swap, because a primary swap would require a line be above and below it to resolve the move. On the right side, they also cannot be used in a secondary swap, because neither would form a triangle.

The code provided generates every valid set of swaps for each side individually, taking into account only the swaps for the other side that are part of the 0 line and are therefore concrete. It creates a record for each, and assigns a code unique to the set of swaps on the given side.

If a perfect configuration exists for a given value k, it will have one of every possible swap, meaning that one side will have a perfect complement to the other, where each swap between the two of them will have been performed exactly once. A dictionary is created where the keys are the code for each of the entries in the record of right side arrangements. Then, each entry in the record of left side arrangements is checked against the corresponding right side arrangement to see if they are disjointed and if the length of their combined set of swaps is equal to max_length, which is ((k - 2)(k - 3)) / 2, the number of swaps excluding swaps with 0 and swaps in the form (n, n + 1).

For k = 9, a match is found (2 matches, but they are mirror image), resulting in the arrangement:

[(1, 5), (3, 7), (1, 7), (1, 3), (4, 6), (1, 6), (1, 4), (2, 8), (1, 8), (3, 6), (4, 8), (3, 8), (5, 7), (6, 8), (5, 8), 0, (2, 5), (3, 5), (2, 4), (2, 7), (4, 7), (2, 6)]

This describes and can be used to construct the solution for k = 9.

However, for k = 11, there are no matches between all possible sets constructed via the outlined method. There is no perfect construction for k = 11, and therefore it is not possible to construct an arrangement with 33 triangles. 32 is the optimal solution.

"G. Clément and J. Bader. Tighter Upper Bound for the Number of Kobon Triangles. Draft Version, 2007" (PDF). Archived from the original (PDF) on 2017-11-11. Retrieved 2008-03-03.