Let $\mathcal{B}$ be a collection of regions with added pieces. That is, each $B \in \mathcal{B}$ is a region B_R of size k to be tiled, together with a set B_P of n pieces from the 209 (of course $n \geq k$).

For $B \in \mathcal{B}$, let $A(B)$ be the collection of all subsets of B_P of size k, and let the solution set $S(B) \subset A(B)$ be the collection of choices of k pieces which will tile the region. Let's pretend that each $S \in S(B)$ can only tile the region in one way (although it may be that due to a small symmetric subregion in the solution there are 'duplicates').

We will assume that each piece has a 'goodness' p_i such that the probability of being able to tile a region of size k is proportional to the product of these numbers. Throughout, $l_i = \log(p_i)$.

Let F be a set of boundary features and for $j \in F$, $c_j(R)$ be the value of the j^{th} boundary feature of the region R. For example, we used $c_0(R) =$ the number of corners (changes in direction) in the boundary of R, $c_1(R) =$ total length of bad edge in R in units of altitudes, and $c_2(R) =$ the total length of good edge in R in units of half-base of equilateral triangle. ('bad edge' is our name for edge which consists of altitudes of the equilateral triangles, and 'good edge' our name for the other sort). We shall assume there are numbers m_j such that the probability of being able to tile a region R is proportional to

$$\exp\left(\sum_{j\in F} m_j c_j(R)\right).$$

Actually the above assumptions are not really assumptions since you can always declare the probability of being able to tile something is such-and-such. The above 'assumptions' are really just defining the information on which our 'distinguisher' is allowed to depend.

Using the above definitions the probability of being able to tile a region R with a set of pieces X is according to our model

$$P(R,X) = \exp\left(\sum_{i\in X} l_i + \sum_{j\in F} m_j c_j(R)\right). \tag{1}$$

We need to choose l_i and m_j to make $P(R,X)$ high when X can tile R and low when it can't.

The likelihood of the model given the data is

$$\prod_{B\in\mathcal{B}}\left[\prod_{S\in S(B)} P(B_R,S) \prod_{X\in A(B)\setminus S(B)} (1-P(B_R,X))\right].$$

We're going to make a few modifications (described below) to this formula, and after that we'll choose l_i and m_j to maximise the (new) likelihood. [By the way $S =$ the negative logarithm of the likelihood has another interpretation as 'conditional entropy', or residual uncertainty. Imagine trying to explain (or compress) the raw binary sequence of $0,1$ formed by taking all pairs (R,X) (in some random order) and writing 0 if the pieces X cannot tile the region R and writing 1 if they can. If there are N digits altogether in this $0,1$ sequence and proportion p of them are 1 and you compressed it without knowing anything about tiling you would probably reduce it to length $S_0 = -N(p\log(p) + (1-p)\log(1-p))$. This is what you would get for S if you didn't allow $P(R,X)$ to depend on anything, in which case you would choose $P(R,X)$ to be equal to the constant p. At the other extreme we could allow our probabilities to depend on everything and we would get $P(R,X) = 1$ when the set of pieces X can tile R and $P(R,X) = 0$ when it can't. Then the residual uncertainty S would be 0. Of course the problem with such a $P(R,X)$ is that it presumably can only be evaluated by an actual search. We're trying to find the minimum S given we're allowed to do some computation of the form (1). We'll end up with S between 0 and S_0.]

Now we'll describe the modifications to the above formula, but first let's talk instead in terms of the more convenient log likelihood, L:

$$L = \sum_{B\in\mathcal{B}}\left[\sum_{S\in S(B)} \log(P(B_R,S)) + \sum_{X\in A(B)\setminus S(B)} \log(1-P(B_R,X))\right]. \tag{2}$$

What sort of size are the numbers? We mainly used $k = 24$, and n is likely to be at least 60. Say $n = 60$. The typical number of solutions will vary according to how good the pieces and regions are, but let's aim (very roughly) at a few hundred solutions per $B \in \mathcal{B}$. There are $\binom{60}{24} \approx 4 \times 10^{16}$ elements of $A(B)$ (choices of 24 pieces) so $P(R, X)$ will be very small, of the order 10^{-14}, and so we can replace $\log(1 - P(B_R, X))$ with $-P(B_R, X)$ and $A(B) \backslash S(B)$ with $A(B)$ in (2). (Actually this is not really an approximation — we're just changing to a model where the number of solutions to (R, X) has Poisson distribution with parameter $P(R, X)$, which happens to be a very similar model because $P(R, X)$ is so small.)

Now the formula for L looks like

$$L = \sum_{B \in \mathcal{B}} \left[\left(\sum_{S \in S(B)} \log P(B_R, S) \right) - \sum_{X \in A(B)} P(B_R, X) \right].$$

Let $n_B = |S(B)|$ = the number of solutions in B. The trouble with using the above L is that n_B will vary a lot. Some B might have $n_B = 0$, others $n_B = 5$ and a few B might have $n_B = 100000$, because it is hard to accurately control how many solutions you get when you add a lot of pieces. This L will be dominated by the terms arising from B for which n_B is particularly large. It will turn out that we may as well not have bothered exhausting 99.9% of the regions, because the p_i and m_j will end up being geared almost entirely towards minimising the terms in L arising from B for which n_B is very large. So most of our data will be ignored and the model will spend all of its effort trying to 'explain' the B for which n_B is very large, and the values it gets for p_i and m_j will be pretty useless. When we chose lots of different B, we were trying to make a uniform sample from the (very large) space of all possible B. In effect with the above L we are weighting each sample point according to n_B, but this is bad because the results associated with a given B are not independent. We decided to smooth this problem out by using a fiddle factor, w_B, to weight each B. w_B is defined as

$$w_B = \frac{1}{n_B} \left(1 - \frac{3}{\sqrt{n_B + 9}} \right).$$

This definition is obviously a bit arbitrary, but it probably doesn't matter too much. $w_B n_B$ is chosen to be between 0 and 1 and increasing in n_B. Our final version of L is then

$$L = \sum_{B \in \mathcal{B}, n_B > 0} w_B \left[\left(\sum_{S \in S(B)} \log P(B_R, S) \right) - \sum_{X \in A(B)} P(B_R, X) \right].$$

This can be rewritten equivalently as

$$L = \sum_i a_i l_i + \sum_j b_j m_j - \sum_{B \in \mathcal{B}, n_B > 0} w_B \exp \left(\sum_j m_j c_j(B_R) \right) A_k(B_P), \tag{3}$$

where

$$A_k(X) = \sum_{\substack{Y \subset X \\ |Y| = k}} \prod_{i \in Y} p_i,$$

and the constants a_i, b_j are defined by

$$a_i = \sum_{B \in \mathcal{B}, n_B > 0} \left[w_B \sum_{\substack{S \in S(B) \\ i \in S}} 1 \right]$$

and

$$b_j = \sum_{B \in \mathcal{B}, n_B > 0} \left[w_B \sum_{S \in S(B)} c_j(B_R) \right].$$

The nice thing about L is that it can be evaluated quickly. The only problematic term might be $A_k(B_P)$, but in fact we can do the necessary sum over all possible piece subsets even though there might be, e.g., $\binom{60}{24} \approx 4 \times 10^{16}$ of them. If we really had to do 10^{16} work to evaluate L, then everything above would be useless because we would never know how to choose p_i and m_j to maximise L. So we weren't really free to choose any model we liked — we had to keep in mind the need to actually be able to calculate the likelihood. To evaluate $A_k(X)$ we list the elements of X in order and make use of the fact that for $i \in X$,

$$A_k(X) = p_i A_{k-1}(X \backslash \{i\}) + A_k(X \backslash \{i\}).$$

So if $X = \{x_1, \ldots, x_n\}$ we can build up to $A_k(X)$ in n stages, where at the r^{th} stage we keep track of $A_i(\{x_1, \ldots, x_r\})$ for all relevant i.

Now it is pretty clean optimising (3) over l_i and m_j because it is smooth and concave (so a local maximum is a global maximum and there are no places to get stuck), and it is easy to evaluate first and second derivatives. For example if $i \in X$,

$$\frac{\partial}{\partial l_i} A_k(X) = p_i A_{k-1}(X \backslash \{i\}).$$

By the way, the main problem with broadening this model to include interaction terms between pieces is that L will no longer be possible to evaluate, because we'll be faced instead of $A_k(X)$ with sums like this

$$\sum_{\substack{Y \subset X \\ |Y|=k}} \prod_{i,j \in Y} b_{i,j},$$

which I think are impossible in general. We did experiment with a restricted set of interaction terms, e.g. the 100 most important pairs of pieces, but this did not obviously make a great improvement, and it slowed everything down so we put it aside, intending perhaps to return to it when we had done other things that seemed more important! There are other (non maximum likelihood) ways of evaluating the effectiveness of model parameters, but we weren't sure how good they would be, and we didn't pursue them. So our final model never included piece–piece interaction terms, although we do have some fairly good data about the 100 or so most important interactions. piece–boundary interactions, on the other hand, would be easy to introduce, but we did not get around to trying this.