The Meßthaler-Wulff Project

Julia Meßthaler

Blazingly fast code for finding all crystals (subsets of a graph) that can be constructed using only transformations that locally minimize surface energy.

The Problem

Let G be some graph and $C \subset G$ a crystal. Furthermore let $\eta : G \to \wp(G)$ denote the neighbors of a given node. Then we can define the surface energy of the crystal C as

$$\xi_C \coloneqq \sum_{n \in C} l_n^1$$

where l_n^1 denotes the forwards loneliness of the node n and is itself defined as

$$l_n^1 \coloneqq \#\{n_0 \in \eta(n) \mid n_0 \notin C\}$$

The idea now is to find optimal ξ_C by doing locally minimizing transformations. We call a node n optimal in forwards mode if for the specific C there is no node n_0 such that $l_{n_0}^1 < l_n^1$. The same definition can be applied to backwards mode, for this however we use $l_n^0 \coloneqq \#\{n_0 \in \eta(n) \mid n_0 \in C\}$.

A locally optimal addition is now simply a node with optimal l_n^1 and a locally optimal removal is a node with optimal l_n^0.

Our goal in this project is to explore what crystals we can construct by only using such locally optimal transformations.

The Additive Simulation

This class encapsulates a current state representing a crystal and methods to find out what locally optimal transformations can be applied or to apply said transformations. It is optimized to be able to support $O(1)$ operations. A simplified definition of an additive simulation instance is $S_A = (\xi_C, B_0, B_1)$ where ξ_C is the energy of the current crystal and B_i are the boundaries, defined as follows:

$$B_i = \{n \in C \mid l_n^{1-i} > 0\}$$

The boundaries are represented by `PriorityStack` instances and support the following operations:

- Getting the loneliness for a node
- Setting the loneliness for a node
- Unsetting the loneliness for a node, effectively removing it from the boundary
- Getting the nodes that have minimal loneliness

In its essence `PriorityStack` is an implementation of a priority queue optimized for this specific use-case.

S_A now basically only has to support one operation: Moving a node from one boundary to the other.

Let n be the affected node and m the mode[1]. If $m = 1$ the energy becomes

$$\begin{aligned} \xi'_C &= \xi_C + l_n^1 - l_n^0 \\ &= \xi_C + l_n^1 - (\#\eta(n) - l_n^1) \\ &= \xi_C + 2l_n^1 - \#\eta(n) \end{aligned}$$

Since backwards and forwards are inverse for $m = 0$ the energy must be

$$\begin{aligned} \xi'_C &= \xi_C - l_n^1 + l_n^0 \\ &= \xi_C + l_n^0 - (\#\eta(n) - l_n^0) \\ &= \xi_C + 2l_n^0 - \#\eta(n) \end{aligned}$$

[1] This is 0 for backwards and 1 for forwards