FOUNDATIONS ADVANCES

Volume 80 (2024)

Supporting information for article:

Realizations of crystal nets. I. (Generalized) derived graphs

Gregory McColm

Appendix B. An algorithm

The following algorithm constructs a simple graph from a voltage graph of one vertex v and a self-dual edge e (and hence the derived graph has one orbit of each). We again use the notation $(\boldsymbol{A}, \boldsymbol{a})_1 = \boldsymbol{A}$ and $(\boldsymbol{A}, \boldsymbol{a})_2 = \boldsymbol{a}$. For a given placement g, let $(g\omega(v))_1 = \{\boldsymbol{M} : (\boldsymbol{M}, \boldsymbol{b}) \in g\omega(v)\}$ be the corresponding orientation set. For a realization, given a vertex $\rho((v, g\omega)) = \mathbf{v} \in \mathbb{R}^d$ of placement g, say that $(\mathbf{v}, g)$ is a *vertex placement pair*, and similarly, given an edge $[\mathbf{u}, \mathbf{v}]$ of placement ${}^g\gamma(e)$ that $([\mathbf{u}, \mathbf{v}], {}^g\gamma(e))$ is an *edge traversal pair*; when the context is clear, we will refer to either as a "pair." The input is the weight group $\omega(v_0)$ of the vertex v_0 and the chosen voltage $\gamma(e)$ of the edge e.

START

COMMENT *We initialize the algorithm with the reference transversal:*

COMMENT *For the vertex in* Δ*, we enter the weight group:*

$\omega(v_0)$ is the weight group of the vertex

$\gamma(e)$ is the voltage of the edge

$VQ := \{(\mathbf{0}, (\boldsymbol{I}, \boldsymbol{0})\omega(v_0))\}$

COMMENT *And the set of all vertices and their placements encountered thus far:*

$P := \{(\mathbf{0}, (\boldsymbol{I}, \boldsymbol{0})\omega(v_0))\}$

COMMENT *For the edge in* Δ*, we enter two edge traversal pairs as that vertex has one incoming edge of that edge orbit and one outgoing.*

$EP := \{([\mathbf{0}, (\gamma(e))_2], \gamma(e)), ([(\gamma(e))_2, \mathbf{0}], \gamma(e))\}$

COMMENT *And the set of lattice vectors encountered (thus far):*

$L := \varnothing$

COMMENT *In this loop, we pop the top placement pair in VQ, expand it, and put new edges in E (although at least one edge is not new, which is not a problem since the operation is set union), and deal with new vertices* *and lattice vectors (if any) appropriately*

while $VQ \neq \varnothing$ **do**

COMMENT *"Pop" the vertex placement pair at the front of the queue*

$p :=$ the vertex placement pair at the front of VQ

$VQ := VQ$ with p deleted

COMMENT *Expand the vertex of pair* p

$\mathbf{u} := (p)_1$

$g := (p)_2$

COMMENT *Now for the vertex figure about* $\mathbf{u}$*: use* $\omega(v_0)$ *to generate the incident edges of the vertex figure by applying elements of* $\omega(v_0)$ *to the reference edge* ee *and its dual* $\overline{ee}$*.*

$VF_{\text{out}} := \{([\mathbf{u}, {}^{gf}\gamma(e)(\mathbf{u})], {}^{gf}\gamma(e)) : f \in \omega(v_0)\}$

$VF_{\text{in}} := \{([{}^{gf}\gamma(e)^{-1}(\mathbf{u}), \mathbf{u}], {}^{gf}\gamma(e)^{-1}) : f \in \omega(v_0)\}$

$EP := EP \cup VF_{\text{out}} \cup VF_{\text{in}}$

COMMENT *Generate the vertex placement pairs of the neighborhood of* $\mathbf{u}$

NBHD $:= \{({}^{gf}\gamma(e)(\mathbf{u}), gf\gamma(e)\omega(v_0)) : f \in \omega(v_0)\}$

COMMENT *We check the placement of each vertex in the neighborhood of* $\mathbf{u}$ *and add that vertex to the queue if not equivalent to any extant vertex; but if it is equivalent to an extant vector, add the appropriate lattice vector.*

for $(\mathbf{w}, h\omega(v_0)) \in$ NBHD **do**

for $(\mathbf{v}, k\omega(v_0)) \in P$ do

if $h^{-1}k \in \omega(v_0)$ **then**

COMMENT *The orientation is extant, so add to lattice group*

$L := L \cup \{\mathbf{w} - \mathbf{v}\}$

else

COMMENT *The orientation is new, so add vertex to queue*

$VQ := VQ \cup \{(\mathbf{w}, h\omega(v_0))\}$

end if

COMMENT *Now add to the vertex placement pairs*

$P := P \cup \{(\mathbf{w}, h\omega(v_0))\}$

end do

end do

end do

COMMENT *We obtain the edges and vertices from their placements and traversals*

$E := \{(ep)_1 : ep \in EP\}$

$V := \{(p)_1 : p \in P\}$

return V, E, L

HALT

If the original group $\mathcal{G}$ is crystallographic in $\mathbb{R}^d$, every vertex placement set $g\omega(v)$ has an orientation set. As there is a finite upper bound on the orders of d-dimensional point groups, and the number of their cosets, there are finitely many equivalence classes of orientations, and hence if Δ is finite, the algorithm will generate only finitely many vertices and halt.