IMPACT 22 chair challenge

Polyhedral projection in $\mathbb{Q}$

Problem: formulation, significance

Let D be a polyhedron in $\mathbb{Q}^n$.

Find the set P in $\mathbb{Q}^r$, $r < n$: $\{x \mid \exists\ (x, y) \in D\}$

Canonical projection (along axes)

More general image to lower-dimensional space can be decomposed into full-rank affine transformation followed by canonical projection

- “Sufficient” problem to tackle rational image problem

Most combinatorial useful polyhedral operation in $\mathbb{Q}^n$

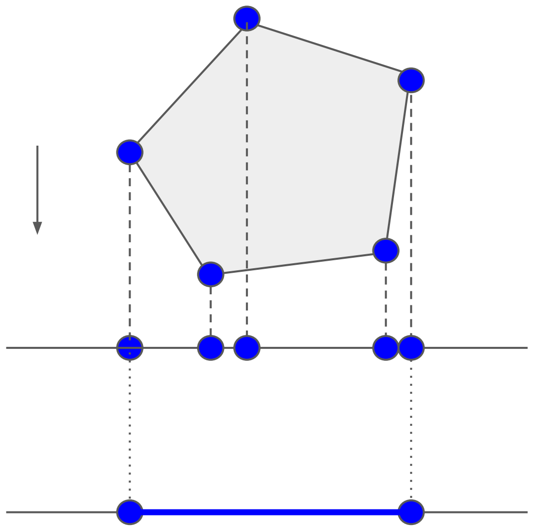
- Others exist (e.g. convex hull) but aren’t necessary in polyhedral compilation

Algorithm 1: vertex-based projection

1- Express polyhedron as a convex combination of vertices and a non-negative combination of rays (Minkowski form)

2- Project the vertices and rays

3- Simplify (remove redundant vertices and rays)



Algo 1 exposes first problem

Combinatoriality of polyhedra: m constraints can combine into $C_n(m)$ vertices

Converse is true also, but loop nest representations tend to be light on constraints, heavy on vertices

- One of ISL postulates

In practice: we cannot afford to compute all the vertices of D, P or any intermediate in the process

- Only compute some ? Not unreasonable.

Next Algo works on constraints

Algo 2: Pairwise inequality elimination (Fourier-Motzkin)

Uses gauss-style elimination, by zeroing out one coefficient using a pair of opposite-coef-sign inequalities:

$$(1) : \mathbf{a_1}x + b_1y + c_1 \geq 0$$
$$(2) : \mathbf{a_2}x - b_2y + c_2 \geq 0$$

$\Rightarrow$ b2 x (1) + b1 x (2)

$$b_2\mathbf{a_1}x + \cancel{b_2b_1y} + b_2c_1 + b_1\mathbf{a_2}x - \cancel{b_1b_2y} + b_1c_2 \geq 0$$

$$\mathbf{a'}x + c'_1 \geq 0$$

Choose an order of the (n-r) dimensions to project out

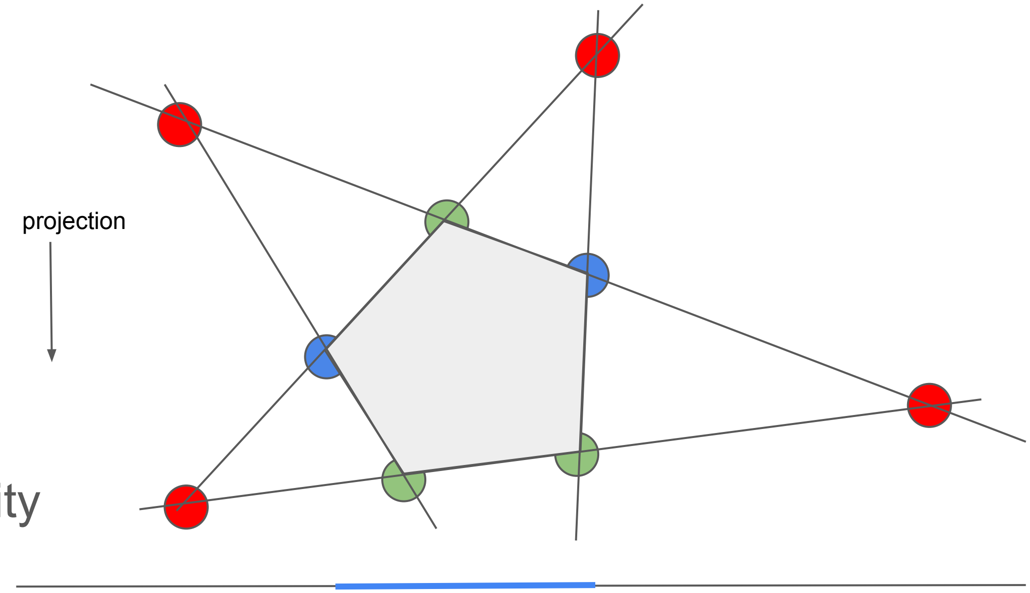
Eliminate them one by one.

Remove redundant constraints

Algo 2 exposes second problem

While coefficient signs somewhat limit the constraints that get combined, many pairwise combinations are not faces of the projected polyhedron

- Redundant constraints
- Useful constraints (faces of P')
- Avoided eliminations



Leads to a 2^{2^n} worst-case complexity

Optimizations

Exploited:

Remove redundant faces after eliminating each dimension (LeVerge)

Avoid some redundancy by keeping track of how projected inequalities were formed (Imbert)

- Commutativity of intersection
- Degenerate faces

Unexploited (?):

Commutativity of dimension choice (among the n-r)

- All sequences of dimensions lead to the same result

Algo 3 - Parametric linear programming

Formulate the problem as: domain in x such that there exists a value of y in D(x,y).

- There exists a point y iff there exists *some* minimum point along some linear objective function f(y)
- Compute the parametric minimum of f(y). It will have the form:

$$\min = v_1 \text{ if } A_1(x) \geq 0$$

$$= v_2 \text{ if } A_2(x) \geq 0$$

… etc

Projection is the domain in x for which there exists a minimum: $P(x) = U_k\{ A_k(x) \geq 0 \}$

- We know that the projection is convex
- We can choose f to make the computation simpler, w/ fewer *k*s
- Complexity ?

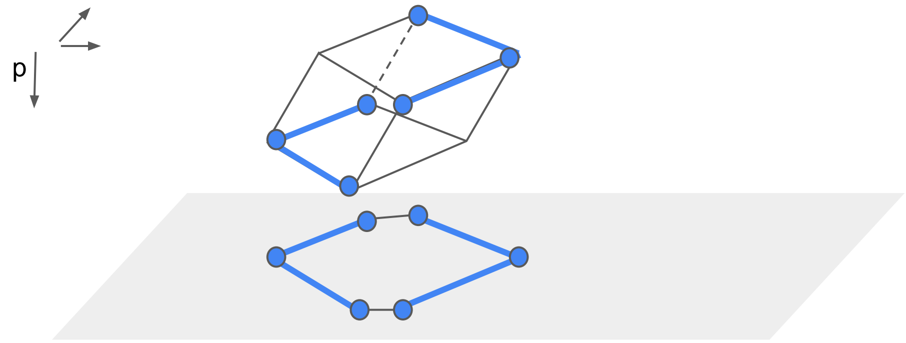
Other interesting conjecture

Let p(x, y) the projection function, i.e., P(x) = image(D, p)

Let p'(x, y) the dual of the p , represented by Ker(p)

Let G the set of function spanned by the vectors of p': G = span(p')

Conjecture: The extremal face of D along any function g ∈ G maps to a face of P.



Challenge

Level 1 - Find a new algorithm that isn't just a reformulation of Algos 1, 2 or 3

Level 2 - The found algorithm is faster than ISL, FMLib, FPL (if vectorized)

Present it at IMPACT 2024!

Benchmark will be posted by the chairs