Mixed-integer formulations for piecewise linear functions: Modern approaches
Joey Huchette and Juan Pablo Vielma
M.I.T.

PIECEWISE LINEAR FUNCTIONS

- Piecewise linear (PWL) function:
  \[ x \in P \implies f(x) = a^T x + b_i \quad \forall i \in [d] \]
  (Assume continuity for this work)
- Typically \( \{P_i\}_{i=1}^d \) are “simple” polyhedra (intervals, triangles, etc.)
- Applications: Econ., operations, engineering
- Use them to approximate nonlinear functions
- Contributions: host of new, fast formulations

FORMULATING PWL FUNCTIONS

- Want to embed piecewise linear functions in optimization problems
- If \( f \) is convex, exists canonical transformation to linear optimization problem
- If \( f \) is non-convex...
  - Modify simplex algorithm (e.g. Gurobi?)
  - Use tailored algorithm to handle PWL structure directly (e.g. SOS2 branching)
  - MIP formulations!
- Over a dozen MIP formulations for univariate PWL functions! Which do we use?

UNIVARIATE PWL FUNCTIONS

- \( f : D \rightarrow \mathbb{R}, D = [l, u] \subset \mathbb{R} \)
- Each domain piece \( P_i \) is subinterval of \( D \)
- Existing approaches:
  - Apply standard MIP formulations for disjunctive constraints (NC, GC)
  - Incremental (Inc) with good branching
  - Logarithmic independent branching formulation (Log) of Vielma and Nemhauser

BIVARIATE PWL FUNCTIONS

- \( f : D \rightarrow \mathbb{R}, D = [P, u] \times [P, v] \subset \mathbb{R}^2 \)
- Grid \( D \) along \( x \) and \( y \), triangulate subregions
- Each domain piece \( P_i \) is triangle of \( D \)
- Existing approaches highly dependent on structure of triangulation

ZIG-ZAG FORMULATION

- Take \( (a^k)_{k=1}^n \leq \{0, 1\}^r \) (r = log_2(d)) as Gray code: adjacent vectors differ in exactly one bit
- Log: graph of \( f \) embedded using Gray code
- Define \( a^k \) as number of times \( v^k \) changes value
- Integer zig-zag (ZZI) formulation: graph of \( f \) embedded using \( (a^k)_{k=1}^n \subset \mathbb{Z}^r \):
  \[
  \sum_{j=1}^n a^j \lambda_j \leq z_k \quad \forall k \in [r]
  \]
  \[
  \sum_{j=k+1}^n a^j \lambda_j \geq z_k \quad \forall k \in [r]
  \]
- \( \{a^k\}_{k=1}^n \) is a linear transformation of \( \{0, 1\}^r \), giving zig-zag (ZZ) formulation:
  \[
  \sum_{j=1}^n a^j \lambda_j \leq z_k + \sum_{k=1}^r 2^{r-k} a^k \quad \forall k \in [r]
  \]
  \[
  \sum_{j=k+1}^n a^j \lambda_j \geq z_k + \sum_{k=1}^r 2^{r-k} a^k \quad \forall k \in [r]
  \]
- \( (\lambda, z) \in \Delta^d \times \mathbb{Z}^r \)

ZIG-ZAG BRANCHING

- ZZ1: Branching \( z_1 \leq 0 \) (Left) and \( z_1 \geq 1 \) (Right)
  - ZZ1 uses general integer control variables to emulate good incremental branching of Inc
  - ZZ1 branching is more balanced (i.e. volume)
  - Smaller portion of domain attaining worst approximation for concave \( f \)

BINARIAN COMPUTATIONS

- Transportation problem, concave nondecreasing objective function with \( N \) pieces
- CPLEX 12.7.0, using JuMP package for PWL formulation (ask me for a sticker!)

INDEPENDENT BRANCHING (IB)

- Rewrite disjunctive constraint \( \bigcup_{i=1}^d P^i \) as \( \bigcap_{i=1}^d \left( Q(f^i) \cup Q(F^i) \right) \)
  - \( Q(S) = \{ \lambda \in \Delta^d : \lambda(x_{\leq d}, \leq 0) \} \), “face of simplex”
  - Have complete combinatorial characterization in terms of representability, size
  - Construct biclique cover for conflict graph \( \mathcal{E} = \{(u, v) \in \Delta^d : \{u, v\} \text{ do not share triangle}\} \)
  - Log is an IB formulation! (if power of two)
  - Construction idea for grid triangulation:
    1. Apply “aggregated” SOS2 along \( x \) and \( y \)
    2. Enforce “triangle selection” with a constant number of additional levels
    3. Size: \( \log_2(\# \text{ of triangles}) + \text{constant} \)

OPTIMAL IB FORMULATIONS

- Combinatorial characterization \( \implies \) notion of optimality (smallest # of levels)
- Can find smallest MIP formulation with a MIP!
- Vanilla MIP not scalable, but works well for small problems (need specialized algorithm)
- \( 4 \times 4 \) arbitrary grid triangulation: