Alpine: A Global Solver for Mixed-Integer Nonlinear Programs (MINLPs)

Kaarthik Sundar (A-1), Harsha Nagarajan (T-5), Russell Bent (T-5), Hassan Hijazi (T-5);

CNLS visiting students: Mowen Lu (Clemson), Site Wang (Clemson)

Funding: LDRD 20170201ER (CMD), CNLS

LA-UR-18-23774

Problem Statement

Find the global optimum of the following MINLP:

$$\begin{aligned} \min \quad & f(x,y) \\ \text{subject to:} \quad & c_i(x,y) = 0 \quad \forall i \in E \\ & c_i(x,y) \leqslant 0 \quad \forall i \in I \\ & x \in \mathbb{R}^n;\ y \in \{0,1\}^m \end{aligned}$$

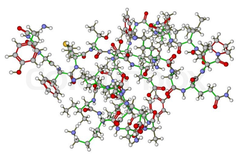
where, the functions f and c_i are continuous and differentiable. The problem is NP-hard and arises in many engineering applications. Furthermore, algorithms to solve the above problem to global optimality have also been identified to be a bottleneck for a variety of *machine learning* problems.

Motivating Applications

Power systems

Gas pipeline networks

Chemical process networks



Molecular structure completion

Philosophy of Alpine

The state-of-the-art technique (spatial branch-and-bound) goes by the philosophy of sub-dividing the MINLP into a large number of "easy-to-solve" sub-problems, exponential number of them, which will in turn be used to find the globally optimal solution to the MINLP.

In contrast, Alpine goes by the philosophy of solving a sequence of small number of increasingly harder Mixed-Integer Linear Programs (MILP), utilizing state-of-the-art MILP solvers like CPLEX and Gurobi.

Solution techniques used in Alpine

Piecewise convex relaxations

Outer-approximation to solve mixed-integer convex problems

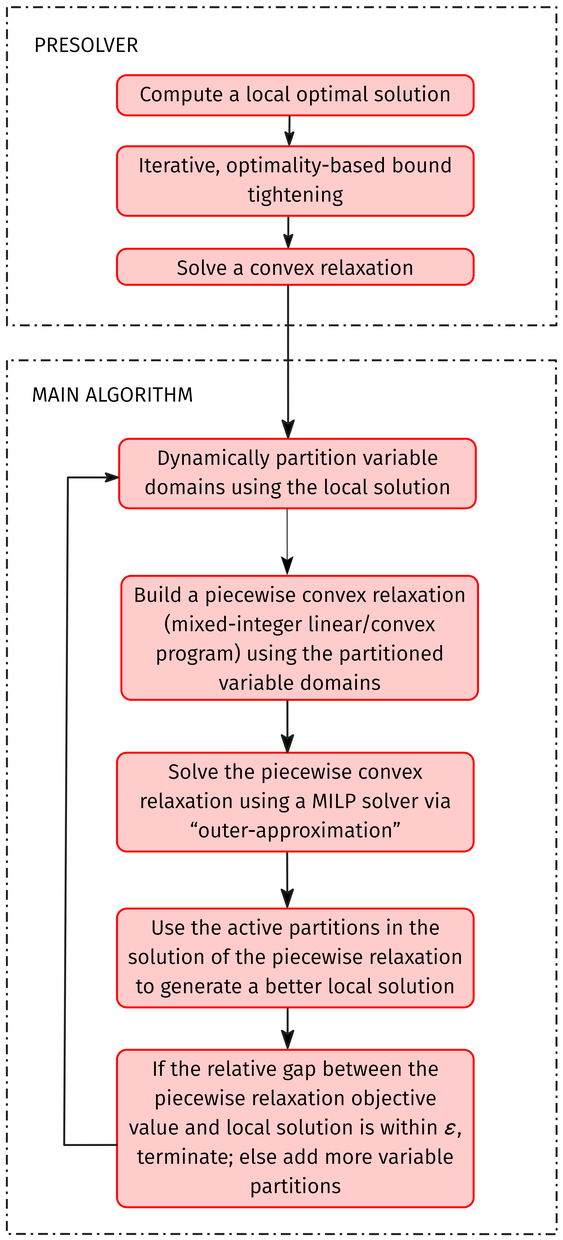
Dynamic partitioning - *new idea in the field of global optimization*

State-of-the-art solvers

Open-source: COUENNE, SCIP

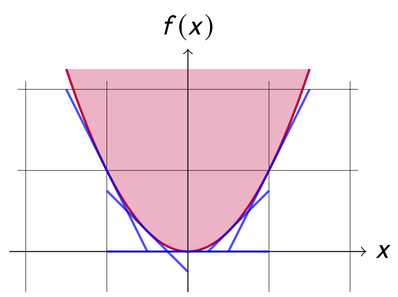
Commercial: BARON, LindoAPI, ANTIGONNE

MILP-based algorithm in Alpine



Outer-approximation

Alpine solves the piecewise convex relaxations via the technique of outer-approximation. For a function defined in the domain $x \in X$



$$f(x) \geqslant f(x^k) + \nabla f(x^k)^\intercal (x - x^k) \ \ \forall x^k \in X$$

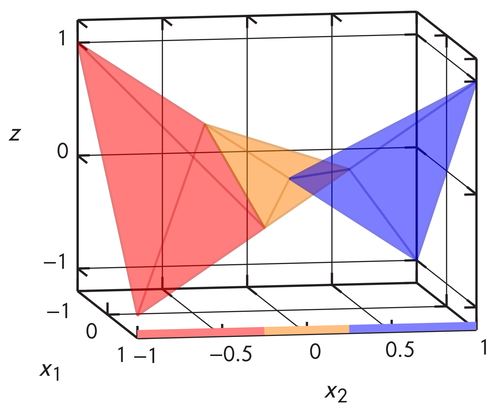
Source code

The source code for Alpine is written using the Julia programming language and can be found at `https://github.com/lanl-ansi/Alpine.jl`

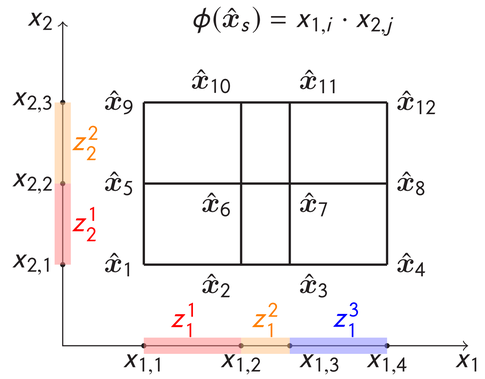
Piecewise convex relaxations - illustrations for a bilinear function

Piecewise convex relaxation of a bilinear function $z = x_1 x_2$, in the domain $[-1, 1]^3$ and with three partitions on the variable x_2, aims to capture the region shown in the 3-D plot (below, to the left)

Piecewise convex envelope for $z = x_1 x_2$



$\phi(\hat{x}_s) = x_{1,i} \cdot x_{2,j}$



The MILP to characterize the piecewise convex relaxation of the biliner function with three and two partitions on x_1 and x_2, respectively, can constructed using the grid analogy (above, to the right). The mathematical formulation, that characterizes all the facets of the MILP using an extreme-point representation, is given by:

$$\begin{bmatrix} x_1 \\ x_2 \\ y \end{bmatrix} = \sum_{s=1}^{12} \lambda_s \begin{bmatrix} \hat{x}_s^1 \\ \hat{x}_s^2 \\ \phi(\hat{x}_s) \end{bmatrix}$$

$$\sum_{s=1}^{12} \lambda_s = 1$$

$$z_1^1 + z_1^2 + z_1^3 = 1$$

$$z_2^1 + z_2^2 = 1$$

$$\lambda_s \geqslant 0 \quad \forall s \in \{1, \ldots, 12\}$$

$$z_1^1, z_1^2, z_1^3, z_2^1, z_2^2 \in \{0, 1\}$$

$$z_1^3 \geqslant \lambda_4 + \lambda_8 + \lambda_{12}$$

$$z_1^3 \leqslant \lambda_3 + \lambda_7 + \lambda_{11} + \lambda_4 + \lambda_8 + \lambda_{12}$$

$$z_1^3 + z_1^2 \geqslant \lambda_3 + \lambda_7 + \lambda_{11} + \lambda_4 + \lambda_8 + \lambda_{12}$$

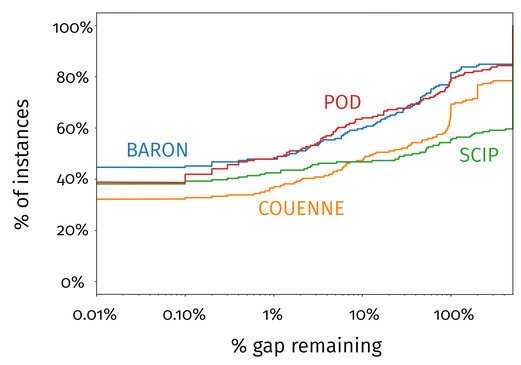
$$z_1^3 + z_1^2 \leqslant \lambda_2 + \lambda_6 + \lambda_{10} + \lambda_3 + \lambda_7 + \lambda_{11} + \lambda_4 + \lambda_8 + \lambda_{12}$$

$$z_2^2 \geqslant \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{12}$$

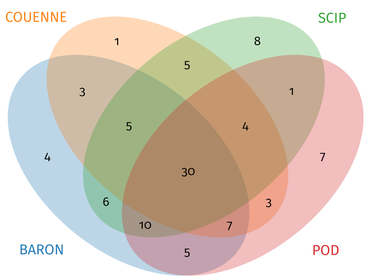
$$z_2^2 \leqslant \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{12}$$

Results

Performance of Alpine is comparable to the state-of-the-art commercial solver, BARON, and better than the state-of-the-art open-source solvers. In the figure below, **higher** the curve, **better** the performance.

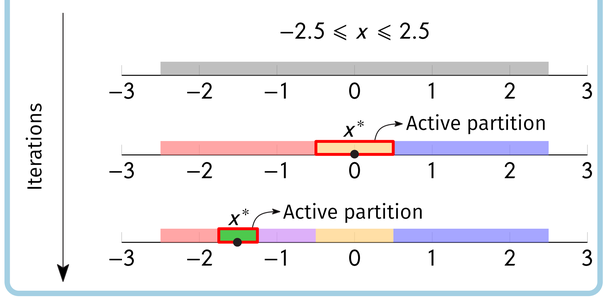


Venn-diagram of instances solved to optimality



Dynamic variable partitioning

The variable partitioning in Alpine is done dynamically, in a non-uniform fashion, guided by the local and the lower bounding solutions



References

[1] H. Nagarajan, M. Lu, S. Wang, R. Bent, and K. Sundar, "An Adaptive, Multivariate Partitioning Algorithm for Tightening Convex Relaxations of Nonlinear Programs," *Journal of Global Optimization*, 2019.

[2] H. Nagarajan, M. Lu, E. Yamangil, and R. Bent, "Tightening McCormick Relaxations for Nonlinear Programs via Dynamic Multivariate Partitioning," in *International Conference on Principles and Practice of Constraint Programming*, 2016.