CSE 507 Project: Hybrid Numerical & SMT Optimization

Ruizhe Shi[a], Evan Wang[a], Jaedon Rich[a] and lipson[a]

[a]*University of Washington, Seattle, U.S.*

Abstract

Optimizing positive rational functions $\mathbb{R}^n \to \mathbb{R}$ with polynomial inequality constraints using interval arithmetic, affine interval representations, Bernstein polynomial form, and branch-and-bound methods. Our source code is available at https://github.com/srzer/cse-507-project

Keywords: interval arithmetic, SMT, Bernstein, affine interval from, dReal

1. Introduction

There are two key elements to introduce: Interval Arithmetic and the dReal SMT solver. Given a collection of variables x_j whose values are each constrained to given intervals $I_j = [a_j, b_j]$, the problem of interval arithmetic is determining the most precise interval to which some arithmetic combination of these variables are constrained. For instance, given x and y , both constrained to the interval $[0, 1]$, we can say that $x + y$ is constrained to the interval $[0, 2]$ and $x - y$ is constrained to the interval $[-1, 1]$. Complications emerge when there are dependencies between our variables; for instance, supposing we have the relation $x = y$, then $x - y$ is actually constrained to the interval $[0, 0]$. We can restate this problem as follows: given a polynomial on a box $f : I_1 \times I_2 \times \cdots \times I_m \to \mathbb{R}$, determine upper and lower bounds on the value of f subject to constraints $g_i(x_1, x_2, ..., x_m) \geq 0$.

There are a number of approaches to interval arithmetic. The goal is to find bounds that are as tight as possible, as efficiently as possible. In general, different approaches have different strengths and weaknesses, and no single method dominates all others across all test cases. Various types of branch-and-prune methods, such as those described below, are used, along with different heuristics for search and subdivision. Some methods are explicitly designed for polynomials, while others can generalize to transcendental functions.[1] Given any set of inequalities involving arithmetic combinations and compositions of polynomials and transcendental functions in several variables (a very broad scope), we are interested in determining whether there are some real assignments of the input variables

*

[1]Some algorithms handle transcendental functions by simply using Taylor approximations.

December 07, 2025

that will simultaneously satisfy every given inequality. Note equalities can be given as a pair of inequalities, and so are also included. An example problem would be: Is there some real x, y satisfying $\sin(x) + y = e^{xy} \wedge \tan(x^2 + y^2) \geq \log(x - y)$? These are difficult problems; in fact, generically they are undecidable. This means we cannot simply feed these problems into an SMT solver using nonlinear real arithmetic with transcendental functions.

However, the problem becomes decidable if we take two actions: First, we want to restrict our search to a compact domain, say a box, by putting bounds on every variable individually. Second, we can introduce an error tolerance instead of looking for an exact solution. That is, we can choose any $\delta > 0$, and then for every constraint $f(x) \geq 0$, we weaken this to $f(x) \geq -\delta$, and an equality $f(x) = 0$ will become $|f(x)| \leq \delta$.

1.0.1. dReal Algorithm Overview

The dReal solver implements this relaxation to obtain a δ -complete algorithm for determining satisfiability: If there is a solution to system of inequalities in the given bounded domain up to a tolerance of δ , then dReal will find it; if there is no solution, dReal returns UNSAT. Notice that if there is no solution up to a tolerance of δ , then there is also no solution to the exact original problem. The complexity class is PSPACE, but in practice dReal performs very well.

dReal uses a branch-and-prune approach: Given the initial box constraint, it will perform interval arithmetic to obtain bounds on the constraint functions to determine whether the constraints are potentially feasible. If not, it returns UNSAT. Otherwise, it subdivides the box into smaller boxes and repeats the procedure on each sub-box: If a sub-box is determined to be infeasible based on the interval arithmetic bounds, it is not explored further, it is 'pruned'. Otherwise, the sub-box is again subdivided into smaller boxes. Interval arithmetic gives imperfect bounds on a function, but subdividing to smaller boxes with tighter interval constraints usually results in more precise bounds, which is the advantage of subdividing. dReal will continue to either prune or subdivide until the box size is so small that the constraint functions vary by less than δ within the small box, at which point the function value is determined on that box within δ precision.

2. Overview

3. Algorithms

4. Results

Below is Fig. 1.

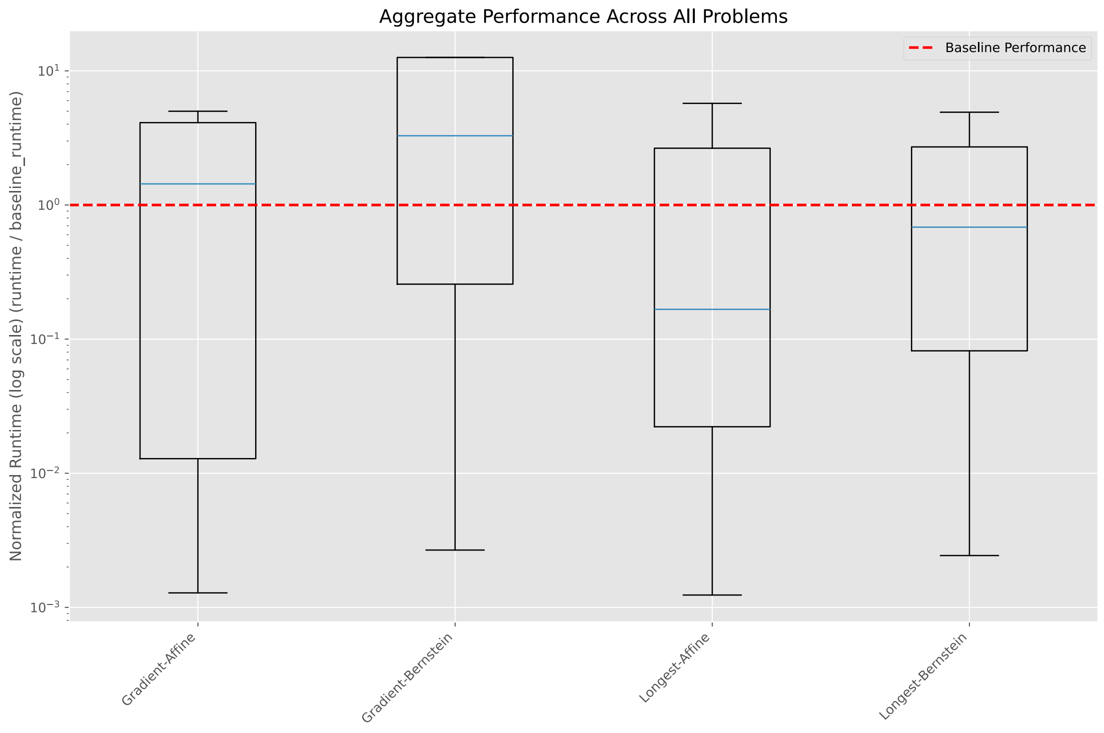


Figure 1: Plot of aggregate runtime performance across testing suite.

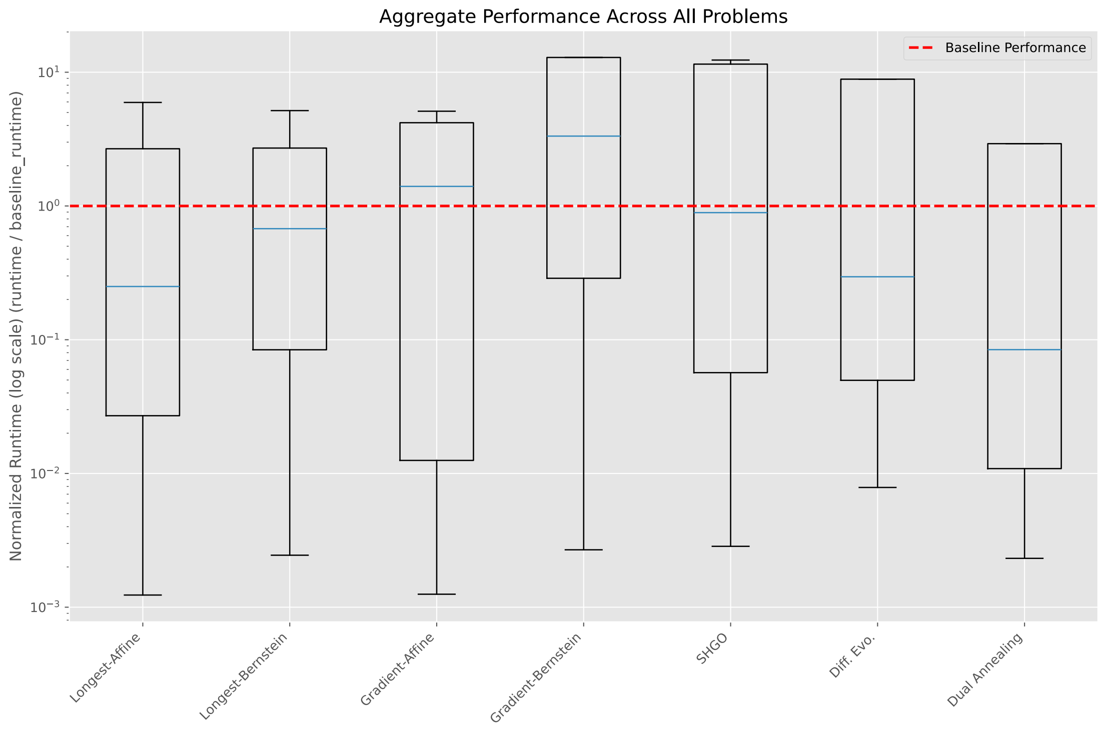


Figure 2: Plot of aggregate runtime performance across testing suite with standard numeric optimization methods.

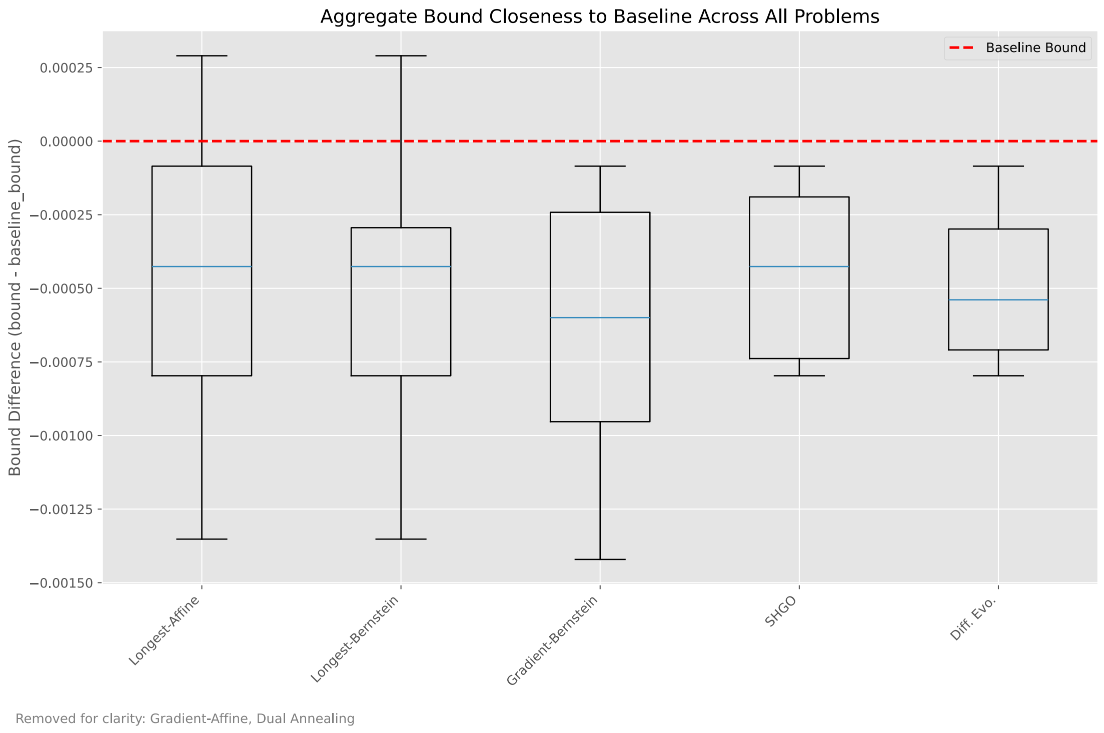


Figure 3: Plot of aggregate bound difference across testing suite. Note that the gradient-split and affine-bound heursitic combination has been ommitted as it tended to produce much larger (and incorrect) bounds on many problems. Dual annealing has been ommitted for a similar reason.

affine bounding heuristic can fail quickly

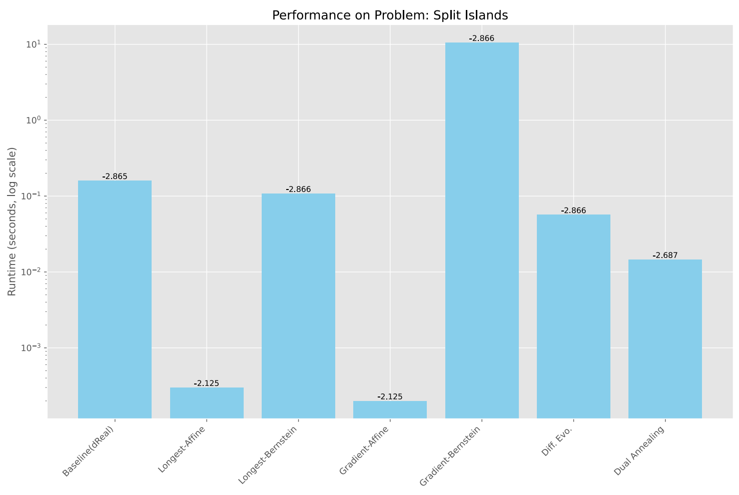


Figure 4: Plot of

sometimes our methods worked well (because of specialization to rational functions?)

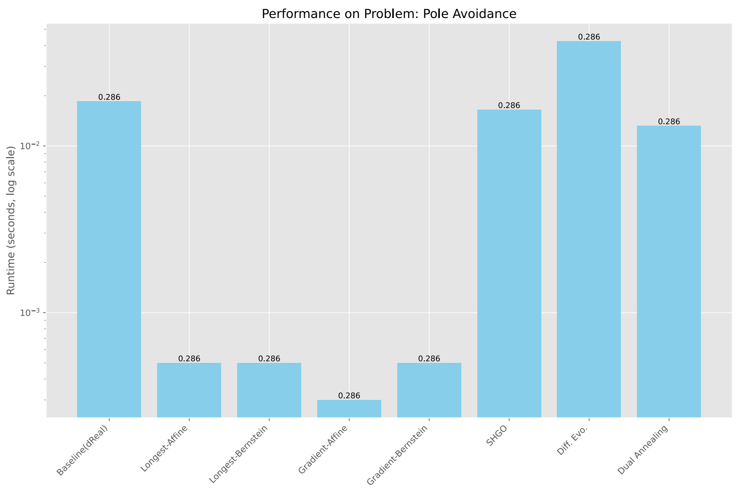
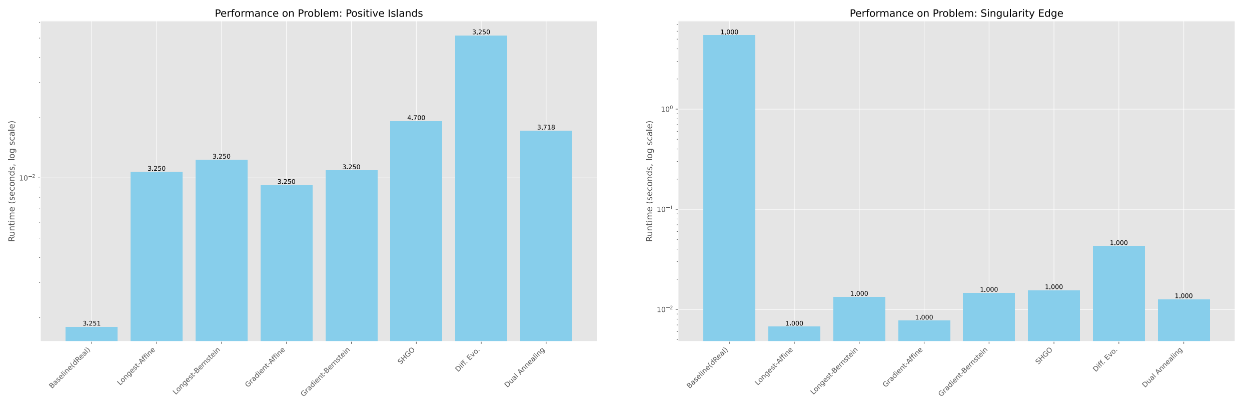


Figure 5: Plot of

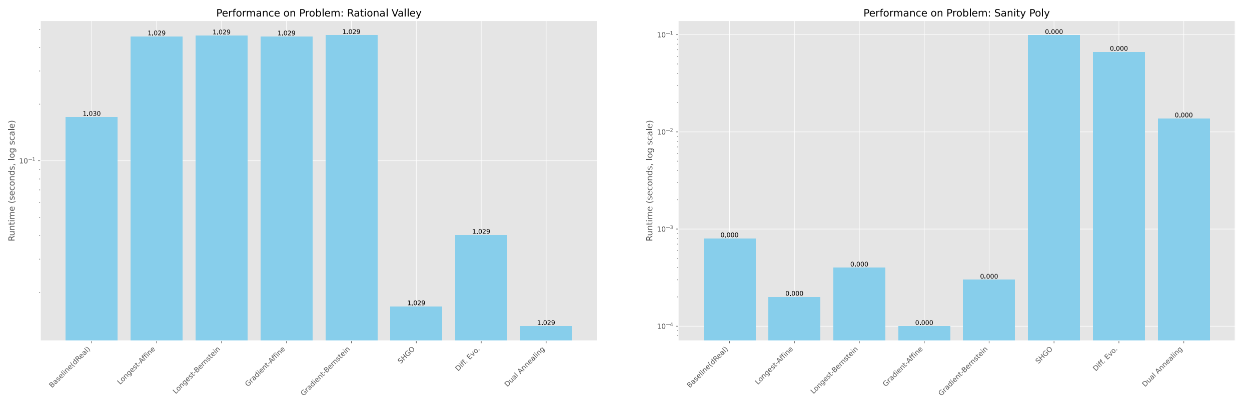
dReal is sometimes an outlier



(a) Positive islands runtime performance. (b) Singularity edge runtime performance

Figure 6: (a) On certain problems, dReal defeats all other methods with marginal error. (b) On other problems, our method is able to match standard numerical optimization performance.

Numerical methods sometimes outperform solver-aided ones



(a) Rational valley runtime performance. (b) Sanity poly runtime performance

Figure 7: (a) On certain problems, dReal defeats all other methods with marginal error. (b) On other problems, our method is able to match standard numerical optimization performance.

4.1. Picking Test Suite

5. Teamwork

6. Course Topics

References