An Algorithm-Independent Measure of Progress for Linear Constraint Propagation

Boro Sofranac [1 2] Ambros Gleixner [1 3] Sebastian Pokutta [1 2]

{sofranac, gleixner, pokutta}@zib.de

[1]Zuse Institute Berlin

[2]Berlin Institute of Technology

[3]HTW Berlin

October 14, 2021

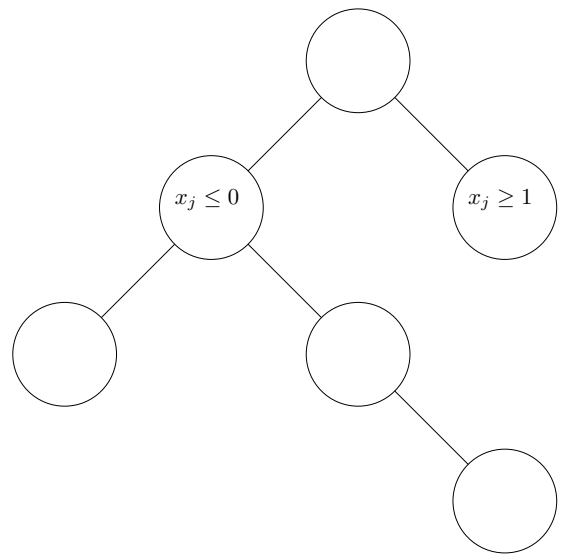
Introduction

Mixed Integer Linear Programming (MIP)

MIP is defined as:

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s. t.} \quad & Ax \leq b \\
& l_j \leq x_j \leq u_j \qquad \forall j \in \mathcal{N} \\
& x_j \in \mathbb{Z} \qquad\quad \forall j \in \mathcal{I}
\end{aligned}
\tag{1}
$$

1. Theory: MIPs are $\mathcal{NP}$-hard
2. Practice: surprisingly fast solvers exist
3. Branch-and-bound algorithm most common



Branch-and-bound search tree

Bounds Tightening in MIP

$$-1 \leq x_1 + x_2 + x_3 \leq 1$$
$$-9 \leq x_1 \leq 9$$
$$-1 \leq x_2 \leq 1$$
$$-1 \leq x_3 \leq 1$$

Bounds Tightening in MIP

$$-1 \leq x_1 + x_2 + x_3 \leq 1$$
$$-9 \leq x_1 \leq 9$$
$$-1 \leq x_2 \leq 1$$
$$-1 \leq x_3 \leq 1$$
$$u_1^{\text{new}} : x_1 \leq 1 - x_2 - x_3 \leq 3$$

Bounds Tightening in MIP

$$-1 \leq x_1 + x_2 + x_3 \leq 1$$
$$-9 \leq x_1 \leq 9$$
$$-1 \leq x_2 \leq 1$$
$$-1 \leq x_3 \leq 1$$
$$u_1^{\text{new}} : x_1 \leq 1 - x_2 - x_3 \leq 3$$
$$\ell_1^{\text{new}} : x_1 \geq -1 - x_2 - x_3 \geq -3$$

Bounds Tightening in MIP

Generalization:

$$\ell_j^{\text{new}} = \frac{\underline{\beta} - \overline{\alpha}_j}{a_j} \leq x_j \leq \frac{\overline{\beta} - \underline{\alpha}_j}{a_j} = u_j^{\text{new}},\ a_j < 0$$

$$\ell_j^{\text{new}} = \frac{\overline{\beta} - \underline{\alpha}_j}{a_j} \leq x_j \leq \frac{\underline{\beta} - \overline{\alpha}_j}{a_j} = u_j^{\text{new}},\ a_j > 0$$

$$\lceil \ell_j^{\text{new}} \rceil \leq x_j \leq \lfloor u_j^{\text{new}} \rfloor \text{ if } x_j \in \mathbb{Z}$$

Iterative Bounds Tightening Algorithm (IBTA)

1. Unique greatest fixed point
2. Bounds consistency at fixed point
3. Tolerance-based termination in practice

Motivations

Premature stalling of IBTAs

- Improvements are not guaranteed to be monotonically non-increasing!

Performance comparison of different IBTAs

- Different IBTAs can traverse different ”paths” towards the fixed point.
- When stopped early, which one performed better?

Designing stopping criteria:

- Different ”paths” towards the fixed point $\rightarrow$ average improvements might also be different!
- Hence, different IBTAs might need different stopping criteria. How to decide when to stop?

Handling Initially Infinite Bounds

Reducing Infinite Bounds

- Some bounds start as infinities.
- The possibility of reducing these to finite value does not depend on the actual finite values in the system.
- The first finite values of the initially infinite bounds depend on the order of propagation.

Observation

- There are k iterations at the beginning with at least one infinite to finite reduction, and none thereafter!

$\Rightarrow$ Reduction of infinities is a fundamentally different process from finite improvements that follow.
$\Rightarrow$ Measure these processes independently.

Reducing Finite Bounds

- All bounds are finite now.
- Two propagation algorithms might not start at the same value towards the unique fixed point! $\Rightarrow$ still can not compare them fairly

Definition (weakest variable bounds)

Given an optimization problem of the form (1) with starting variable bounds ℓ^s and u^s, we call $\overline{\ell}_j$ *weakest lower bound* of variable j if

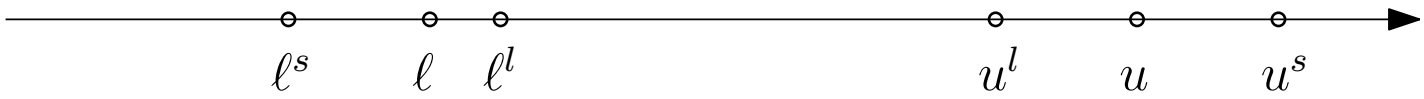
- $\overline{\ell}_j = -\infty$ and no IBTA can produce a finite lower bound $\ell_j \in \mathbb{R}$, or
- $\overline{\ell}_j \in \mathbb{R}$ and no IBTA can produce a finite lower bound $\ell_j \in \mathbb{R}$ with $\ell_j < \overline{\ell}_j$.

We call $\overline{u}_j$ *weakest upper bound* of variable j if

- $\overline{u}_j = \infty$ and no IBTA can produce a finite upper bound $u_j \in \mathbb{R}$, or
- $\overline{u}_j \in \mathbb{R}$ and no IBTA can produce a finite upper bound $u_j \in \mathbb{R}$ with $u_j > \overline{u}_j$.

The Measuring Function

The Measuring Function



Schematic representation of the starting (index s), current (no index) and limit bounds (index l) for a given variable on the real line. In this example, $\ell^s = \overline{\ell} \in \mathbb{R}$ and $u^s = \overline{u} \in \mathbb{R}$.

Infinite reductions progress $\mathcal{P}^{\text{inf}}$

$$n^{\text{total}} = |\{j = 1, \ldots, n : \ell_j^s = -\infty, \ell_j^l \in \mathbb{R}\}| + |\{j = 1, \ldots, n : u_j^s = \infty, u_j^l \in \mathbb{R}\}|,$$

$$n^{\text{current}} = |\{j = 1, \ldots, n : \ell_j^s = -\infty, \ell_j \in \mathbb{R}\}| + |\{j = 1, \ldots, n : u_j^s = \infty, u_j \in \mathbb{R}\}|,$$

$$\mathcal{P}^{\text{inf}} = \frac{n^{\text{current}}}{n^{\text{total}}}, n^{\text{total}} \neq 0.$$

The Measuring Function

Schematic representation of the starting (index s), current (no index) and limit bounds (index l) for a given variable on the real line. In this example, $\ell^s = \overline{\ell} \in \mathbb{R}$ and $u^s = \overline{u} \in \mathbb{R}$.

Finite reductions progress $\mathcal{P}^{\text{fin}}$

$$\mathcal{P}_{\ell_j} = \begin{cases} \frac{\ell_j - \overline{\ell}_j}{\ell_j^l - \overline{\ell}_j} & \text{if } \ell_j > \overline{\ell}_j \text{ and } \overline{\ell}_j \neq \ell_j^l \\ 0 & \text{otherwise} \end{cases}, \quad \mathcal{P}_{u_j} = \begin{cases} \frac{\overline{u}_j - u_j}{\overline{u}_j - u_j^l} & \text{if } u_j < \overline{u}_j \text{ and } \overline{u}_j \neq u_j^l \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{P}^{\text{fin}} = \|\mathcal{P}_\ell\|_1 + \|\mathcal{P}_u\|_1 = \sum_j (\mathcal{P}_{\ell_j} + \mathcal{P}_{u_j}).$$

Applications of the Progress Measure

Premature Stalling Effect

- Improvements are not guaranteed to be monotonically non-increasing.

$\Rightarrow$ stopping criteria might stop the algorithm prematurely!

Definition (Premature Stalling)

Let $\mathcal{P} : \mathbb{R}_+ \to [0, 100]$ be a progress function of finite domain reductions. Then, the propagation algorithm is said to *prematurely stall* with coefficients $p, q \in \mathbb{R}_{\infty \geq 0}$ at round $r \in \{2, \ldots, k\}$ if:

1. no infinite reductions in round r,
2. $\mathcal{P}'(t(r)) < p$, and
3. there exists $x \in [t(r), 100]$ such that $\mathcal{P}''(x) > q$.

Premature Stalling Effect

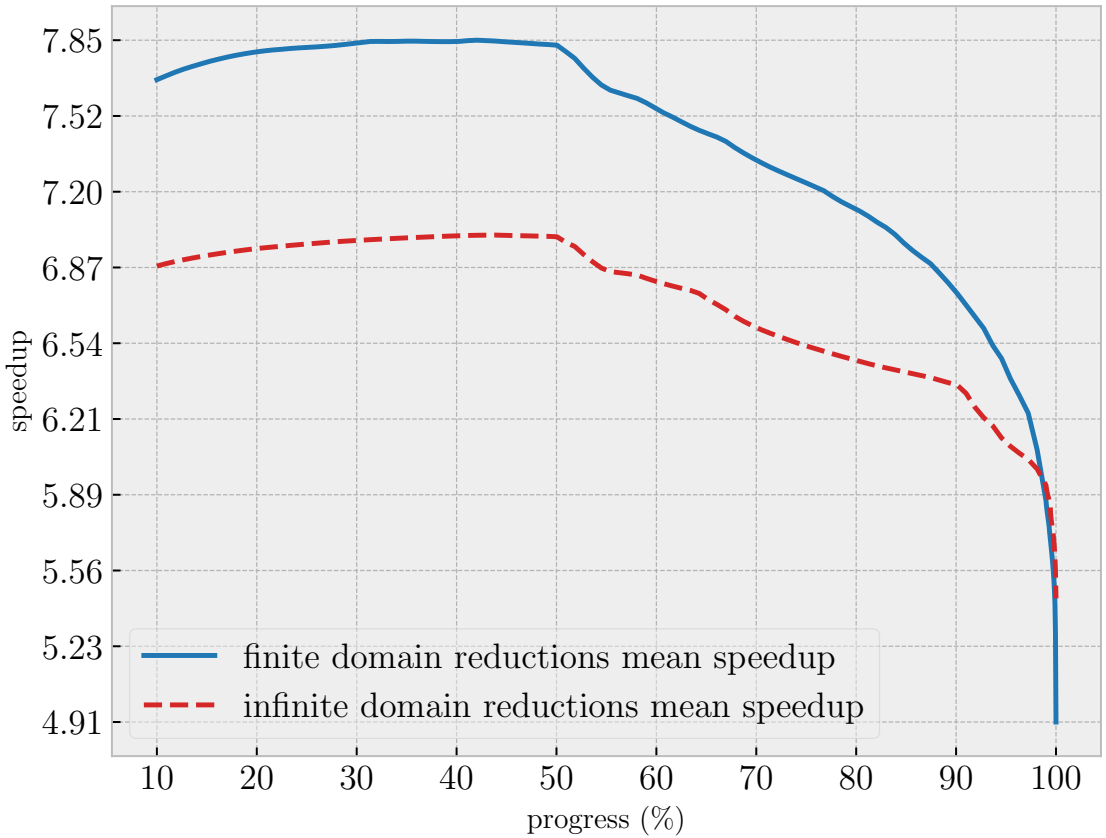
- Test set: MIPLIB 2017 (432 instances which may potentially stall)
- Two different IBTAs: *seq_prop* (sequential) and *gpu_prop* (gpu-parallel)

Table: Number of premature stalls in the test set for different values of parameters p and q

		# stalls	
p	**q**	*seq_prop*	*gpu_prop*
∞	0.0	48	44
0.1	0.0	14	18
0.1	0.2	1	0
0.1	0.5	0	0
0.5	0.5	1	0
0.5	2.0	0	0

Sequential VS GPU Propagation in Practice

- Test set: MIPLIB 2017 (540 instances with progress measurements)



Speedup of the finite and the infinite domain reductions of *gpu_prop* over *seq_prop* for different percentages of progress made.

Concusions & Outlook

Conclusions

- Tolerance-based stopping criteria heuristics is reasonable.
- GPU-based algorithm *gpu_prop* performs even better in real-life than originally reported.
- Future applications of the progress measure possible, e.g. design new stopping criteria etc.

Outlook

- No conceptual barriers for going beyond the linear case.
- Strengthening the definition of the weakest bounds.