Learn to Relax: Integrating Integer Linear Programming with Conflict-Driven Search

Jo Devriendt †*, Ambros Gleixner ‡, Jakob Nordström *°

Acknowledgement: Jan Elffers †*

† Lund University, Sweden

‡ Zuse Institut Berlin, Germany

* University of Copenhagen, Denmark

° KTH Royal Institute of Technology, Sweden

jn@di.ku.dk

Slides by Jo Devriendt

Conflict-driven search for 0-1 ILP

an example-driven intro

$$+x + y - z \geq 0$$

$$-\, y + z - v \geq 0$$

$$-\, z + v - w \geq 0$$

$$-\, x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

Conflict-driven Search Loop

Unit
propagation

Conflict-driven Search Loop

- Given 0-1 ILP program **φ** and current assignment **α**, if a constraint **c** ∈ φ would be falsified by assuming x=0 (resp. x=1), extend α with x=1 (resp. x=0)

Unit propagation

Conflict-driven Search Loop

- Given 0-1 ILP program **φ** and current assignment **α**, if a constraint **c** ∈ φ would be falsified by assuming x=0 (resp. x=1), extend α with x=1 (resp. x=0)
- propagate until fixpoint

Unit propagation

Conflict-driven Search Loop

$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$

- Given 0-1 ILP program **φ** and current assignment **α**, if a constraint **c** ∈ φ would be falsified by assuming x=0 (resp. x=1), extend α with x=1 (resp. x=0)
- propagate until fixpoint

Unit propagation

Conflict-driven Search Loop

$$+x + y - z \geq 0$$

$$- y + z - v \geq 0$$

$$- z + v - w \geq 0$$

$$- x + z + w \geq 1$$

- Given 0-1 ILP program **φ** and current assignment **α**, if a constraint **c** ∈ φ would be falsified by assuming x=0 (resp. x=1), extend α with x=1 (resp. x=0)
- propagate until fixpoint

$$\alpha = \{\}$$

Unit propagation

Conflict-driven Search Loop

- Given 0-1 ILP program **φ** and current assignment **α**, if a constraint **c** ∈ φ would be falsified by assuming x=0 (resp. x=1), extend α with x=1 (resp. x=0)
- propagate until fixpoint

Unit propagation

$$+x + y - z \geq 0$$

$$- y + z - v \geq 0$$

$$- z + v - w \geq 0$$

$$- x + z + w \geq 1$$

$$\alpha = \{\}$$

currently no unit propagation

Conflict-driven Search Loop

$$+x + y - z \geq 0$$
$$- y + z - v \geq 0$$
$$- z + v - w \geq 0$$
$$- x + z + w \geq 1$$

- Conflict: some constraint in φ falsified by α

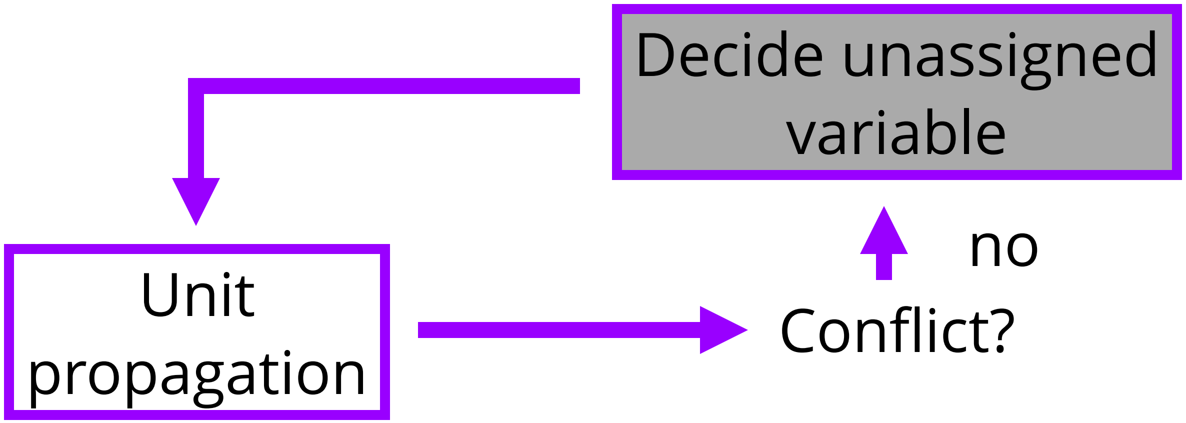
$$\alpha = \{\}$$

Unit propagation → Conflict?

Conflict-driven Search Loop

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

$$\alpha = \{\}$$

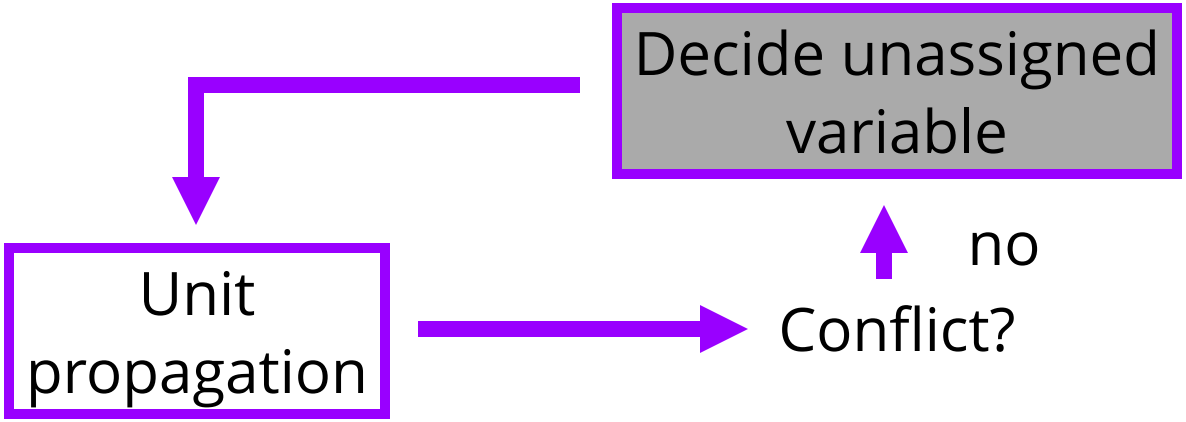


Conflict-driven Search Loop

- Only if unit propagation did not lead to a conflict
  - if no unassigned variable left, return SAT

$$+x + y - z \geq 0$$
$$- y + z - v \geq 0$$
$$- z + v - w \geq 0$$
$$- x + z + w \geq 1$$

$$\alpha = \{\}$$

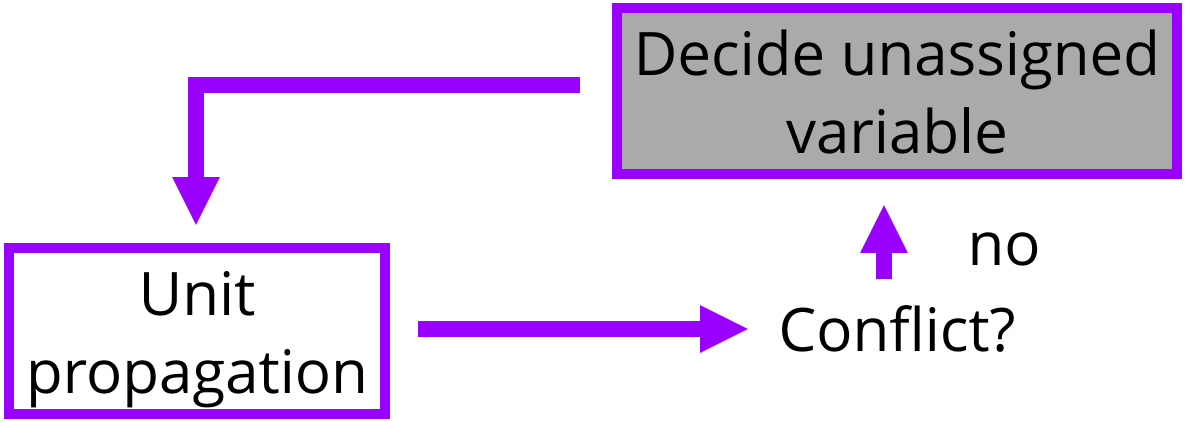


Conflict-driven Search Loop

- Only if unit propagation did not lead to a conflict
  - if no unassigned variable left, return SAT
- Resume unit propagation

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

$$\alpha = \{\}$$

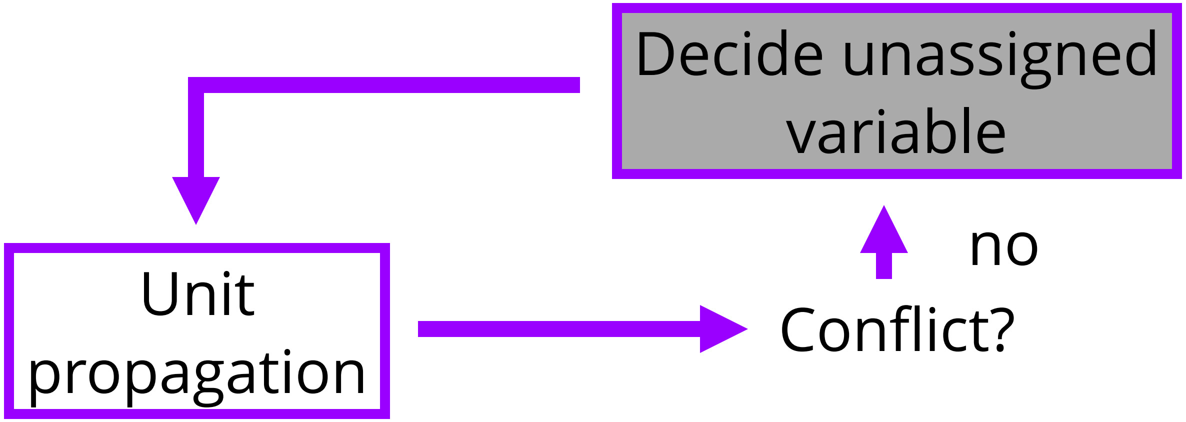


Conflict-driven Search Loop

- Only if unit propagation did not lead to a conflict
  - if no unassigned variable left, return SAT
- Resume unit propagation

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

$$\alpha = \{x = 0\}$$



Conflict-driven Search Loop

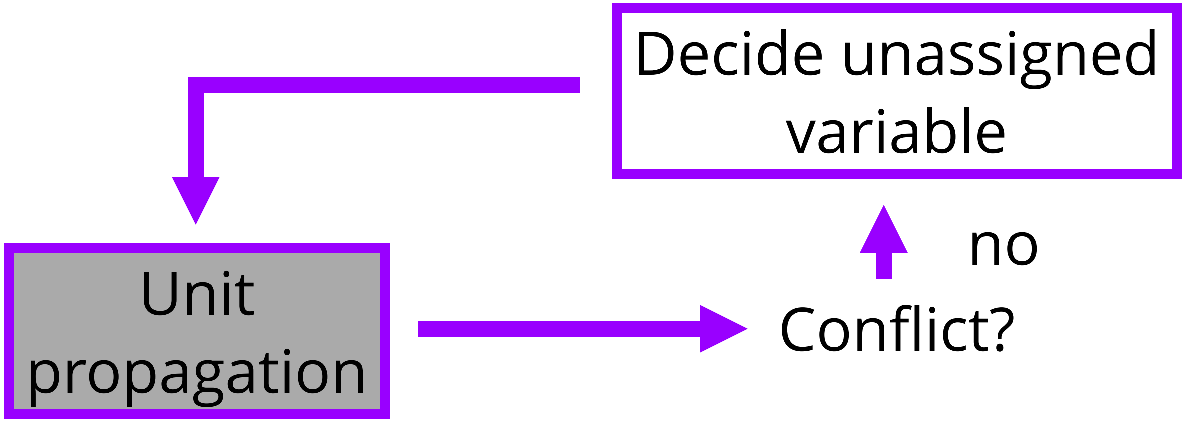
$$+y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$+z + w \geq 1$$

$$\alpha = \{x = 0\}$$



Conflict-driven Search Loop

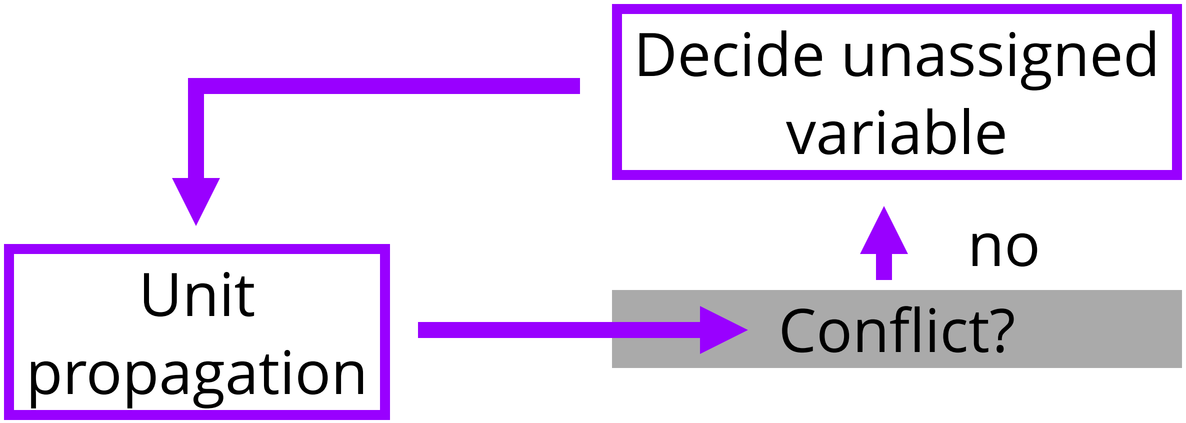
$$+y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$+z + w \geq 1$$

$$\alpha = \{x = 0\}$$



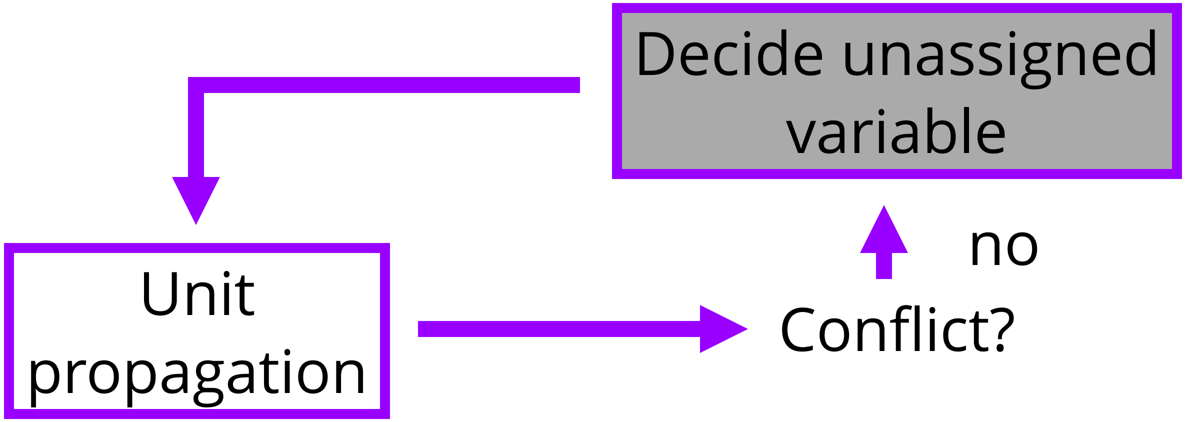
Conflict-driven Search Loop

$$+y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$+z + w \geq 1$$



$$\alpha = \{x = 0, y = 1\}$$

Conflict-driven Search Loop

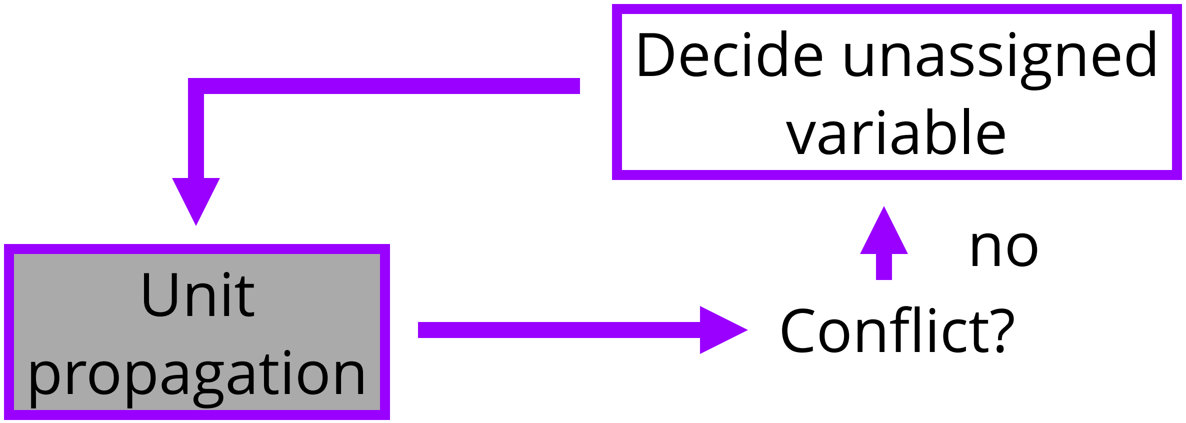
$$1 - z \geq 0$$

$$-1 + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$+z + w \geq 1$$

$$\alpha = \{x = 0, y = 1\}$$



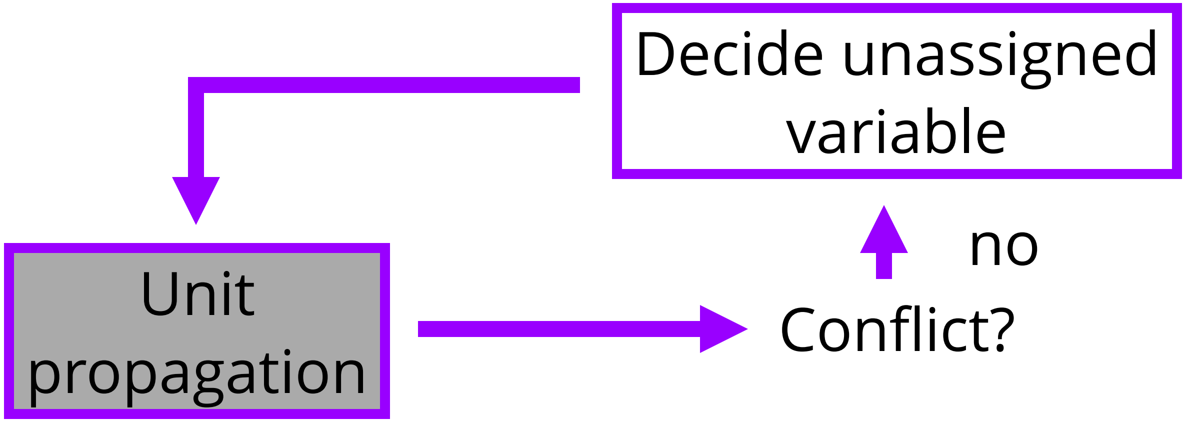
Conflict-driven Search Loop

$$1 - z \geq 0$$

$$-1 + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$+z + w \geq 1$$



$$\alpha = \{x = 0, y = 1\}$$

Conflict-driven Search Loop

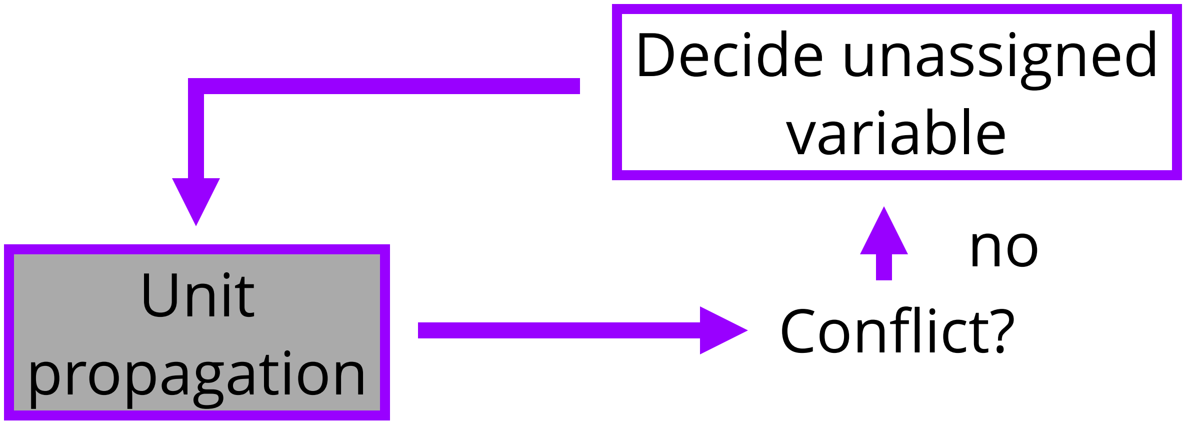
$$1 - z \geq 0$$

$$-1 + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$+z + w \geq 1$$

$$\alpha = \{x = 0, y = 1, z = 1, v = 0\}$$



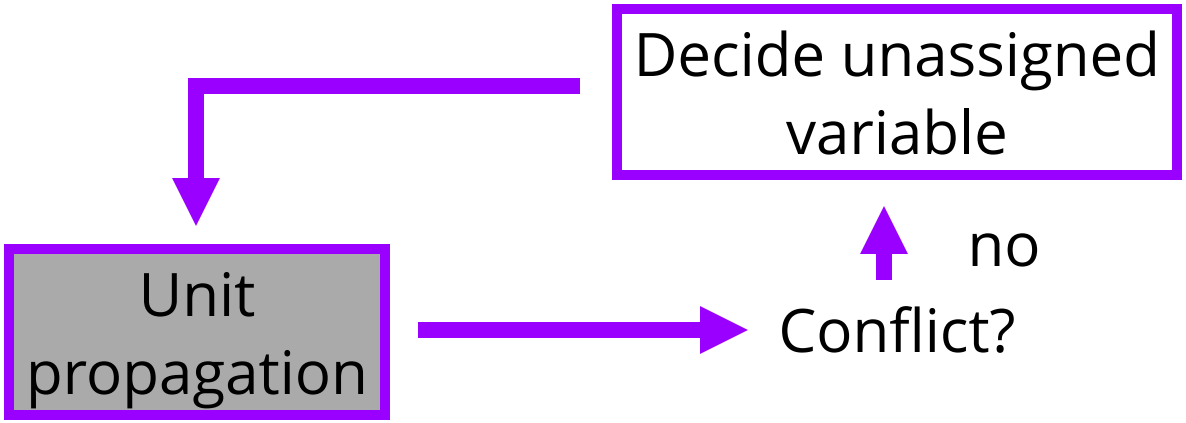
Conflict-driven Search Loop

$$0 \geq 0$$

$$0 \geq 0$$

$$-1 - w \geq 0$$

$$+1 + w \geq 1$$



$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$

Conflict-driven Search Loop

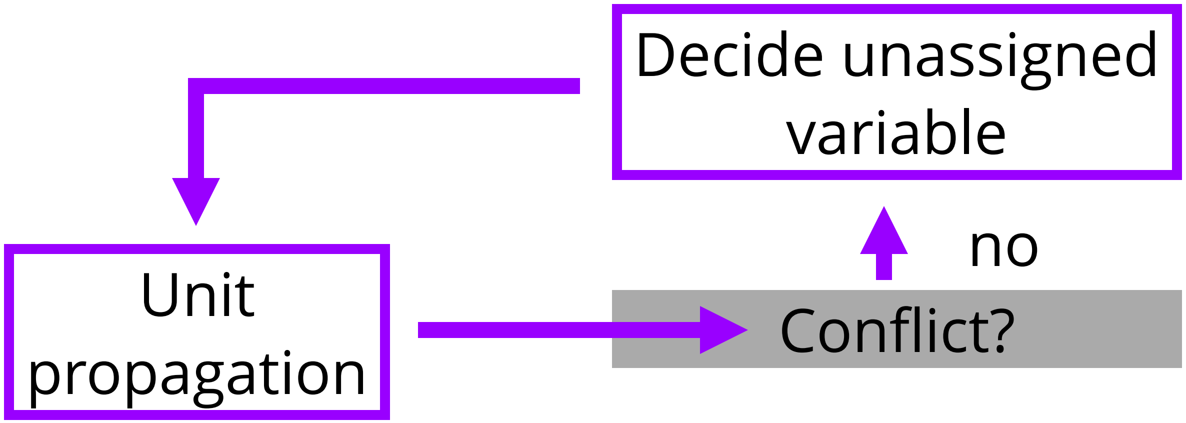
- Conflict: some constraint in φ falsified by α

$$0 \geq 0$$

$$0 \geq 0$$

$$-1 - w \geq 0$$

$$+1 + w \geq 1$$



$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$

Conflict-driven Search Loop

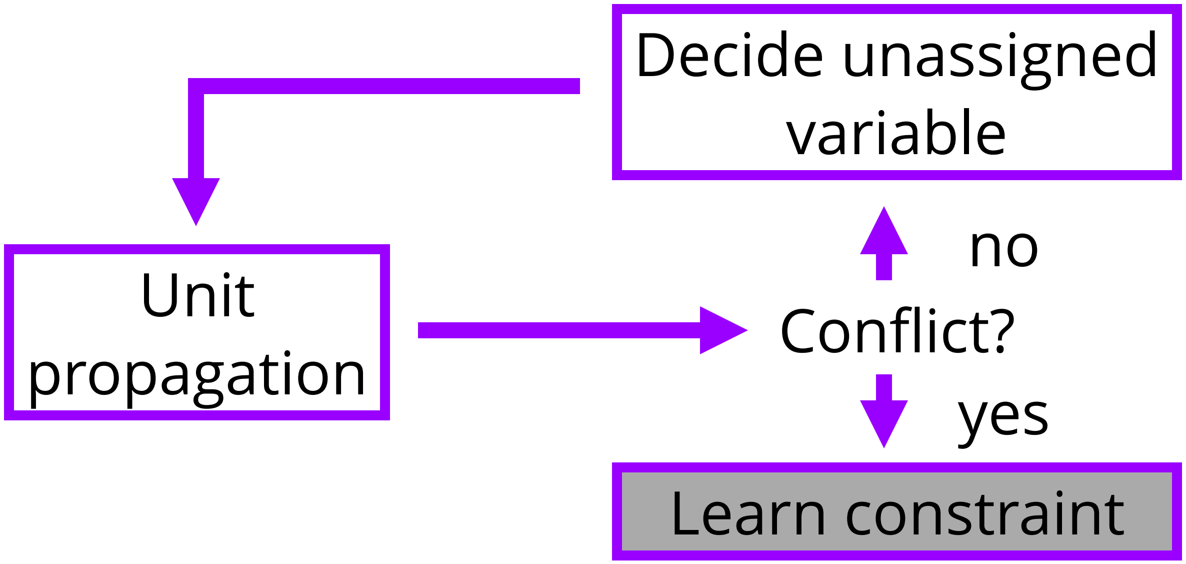
$$0 \geq 0$$

$$0 \geq 0$$

$$-1 - w \geq 0$$

$$+1 + w \geq 1$$

$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$



Conflict-driven Search Loop

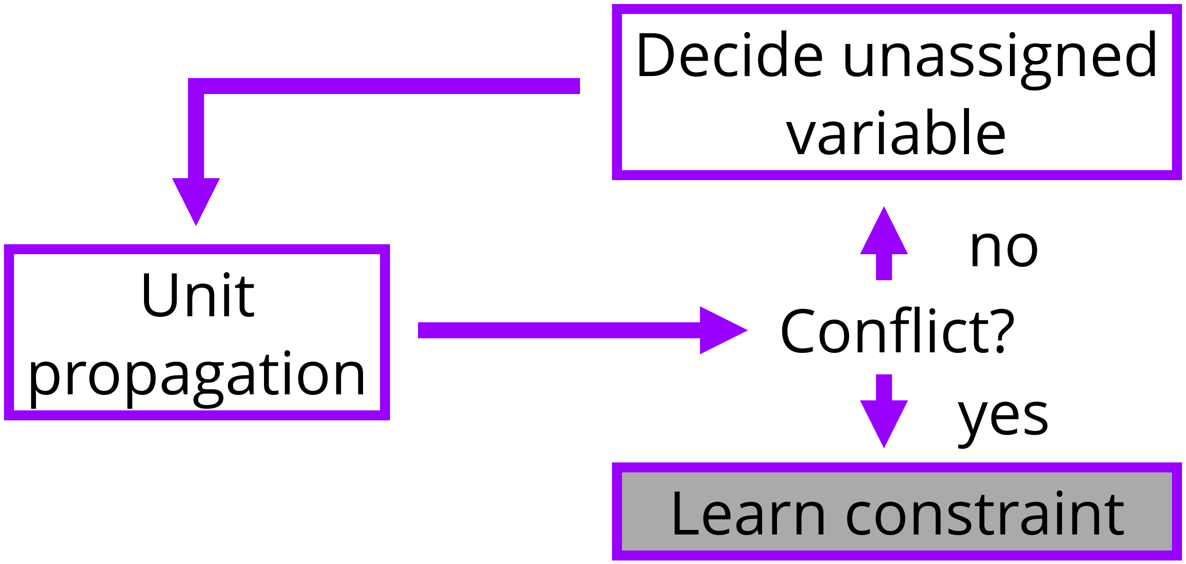
- From falsified constraint and **reasons** leading up to conflict, construct **learned constraint**
  - implied by φ, should prevent same conflict

$$0 \geq 0$$

$$0 \geq 0$$

$$-1 - w \geq 0$$

$$+1 + w \geq 1$$



$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$

Conflict-driven Search Loop

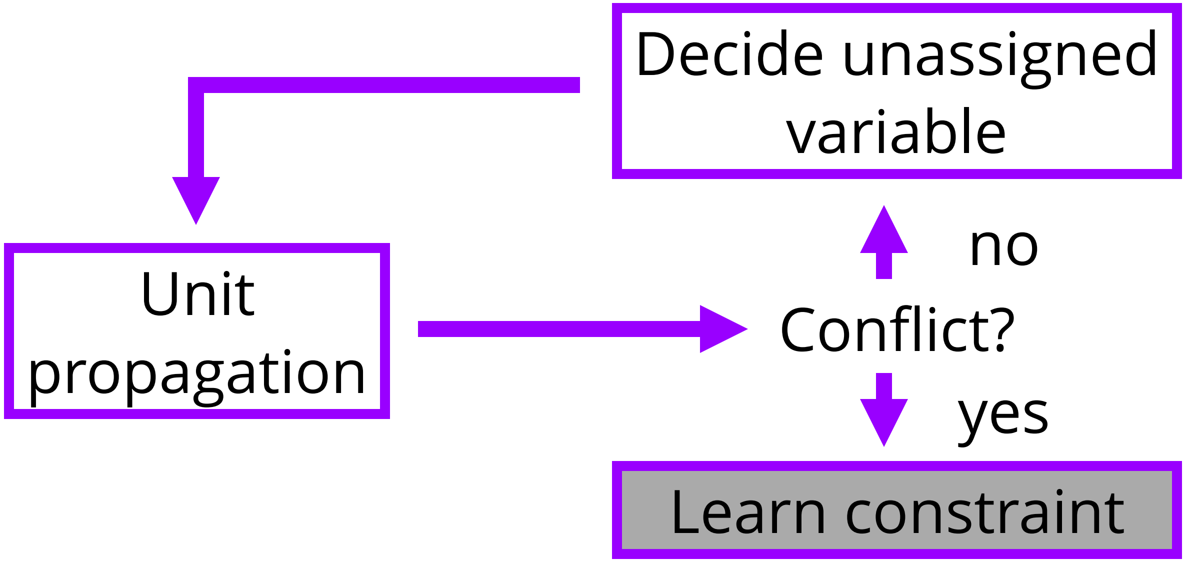
- From falsified constraint and **reasons** leading up to conflict, construct **learned constraint**
  - implied by φ, should prevent same conflict
- Add learned constraint to φ
  - → **learned constraint database**

$$0 \geq 0$$

$$0 \geq 0$$

$$-1 - w \geq 0$$

$$+1 + w \geq 1$$

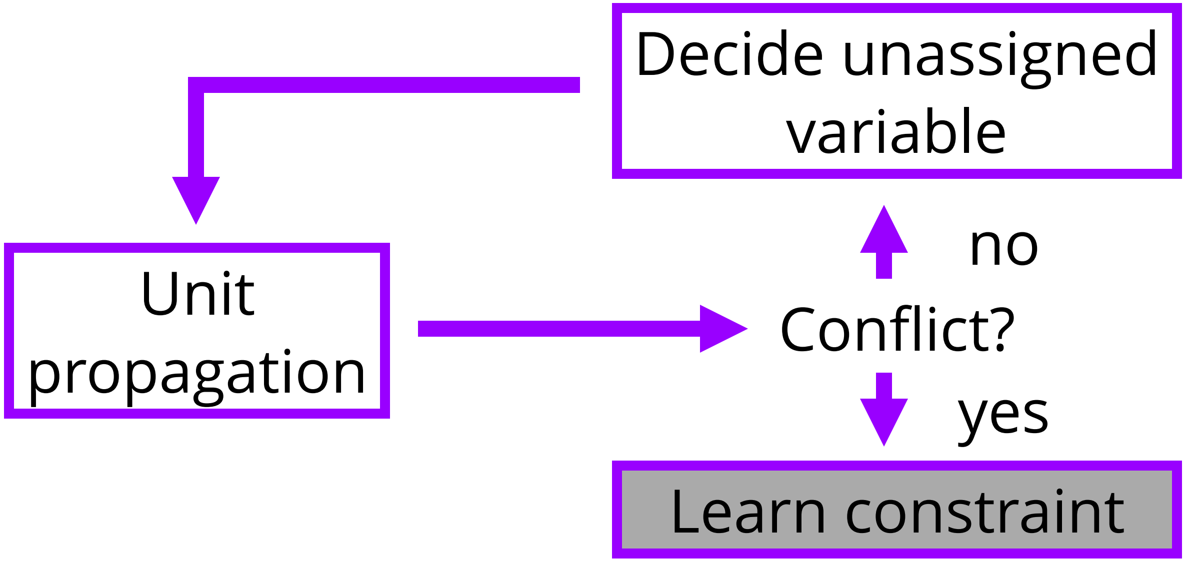


$$\alpha = \{x = 0, y = 1, z = 1, v = 0\}$$

Conflict-driven Search Loop

- From falsified constraint and **reasons** leading up to conflict, construct **learned constraint**
  - implied by φ, should prevent same conflict
- Add learned constraint to φ
  - → **learned constraint database**

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

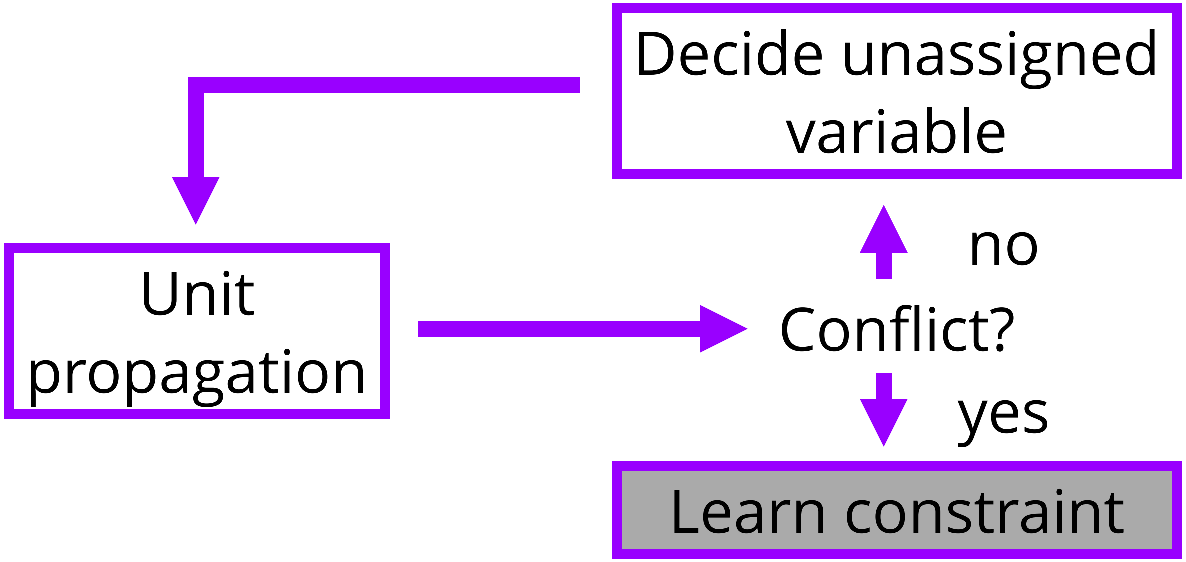


$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$

Conflict-driven Search Loop

- From falsified constraint and **reasons** leading up to conflict, construct **learned constraint**
  - implied by φ, should prevent same conflict
- Add learned constraint to φ
  - → **learned constraint database**

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$



$$\alpha = \{x = 0, y = 1, z = 1, v = 0\}$$

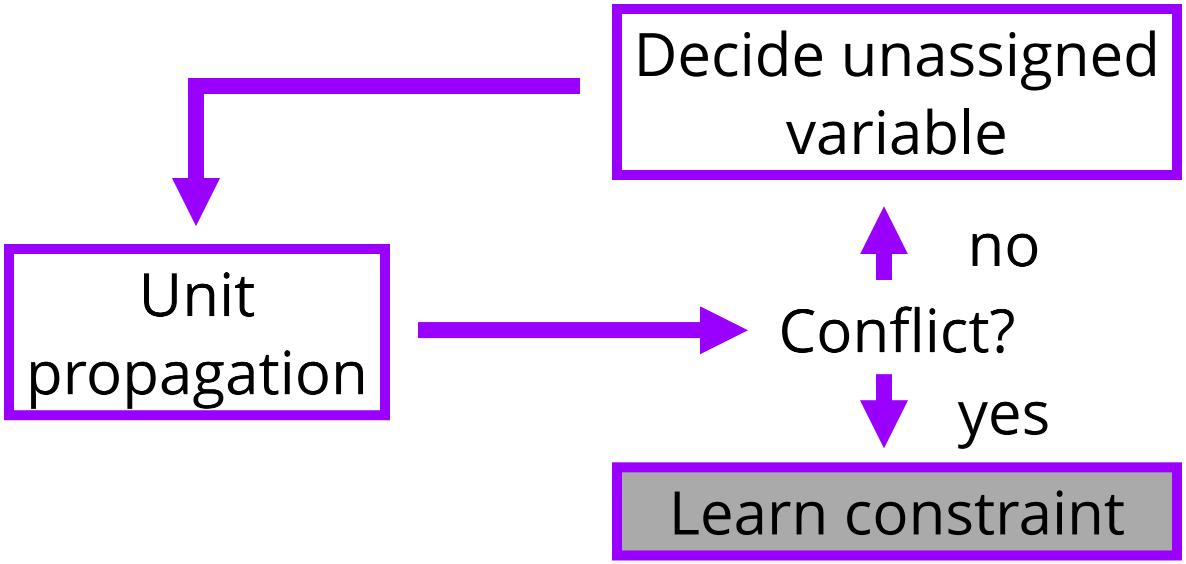
Conflict-driven Search Loop

- From falsified constraint and **reasons** leading up to conflict, construct **learned constraint**
  - implied by φ, should prevent same conflict
- Add learned constraint to φ
  - → **learned constraint database**

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

$$\alpha = \{x = 0, y = 1, z = 1, v = 0\}$$

$$-y - w \geq 0$$



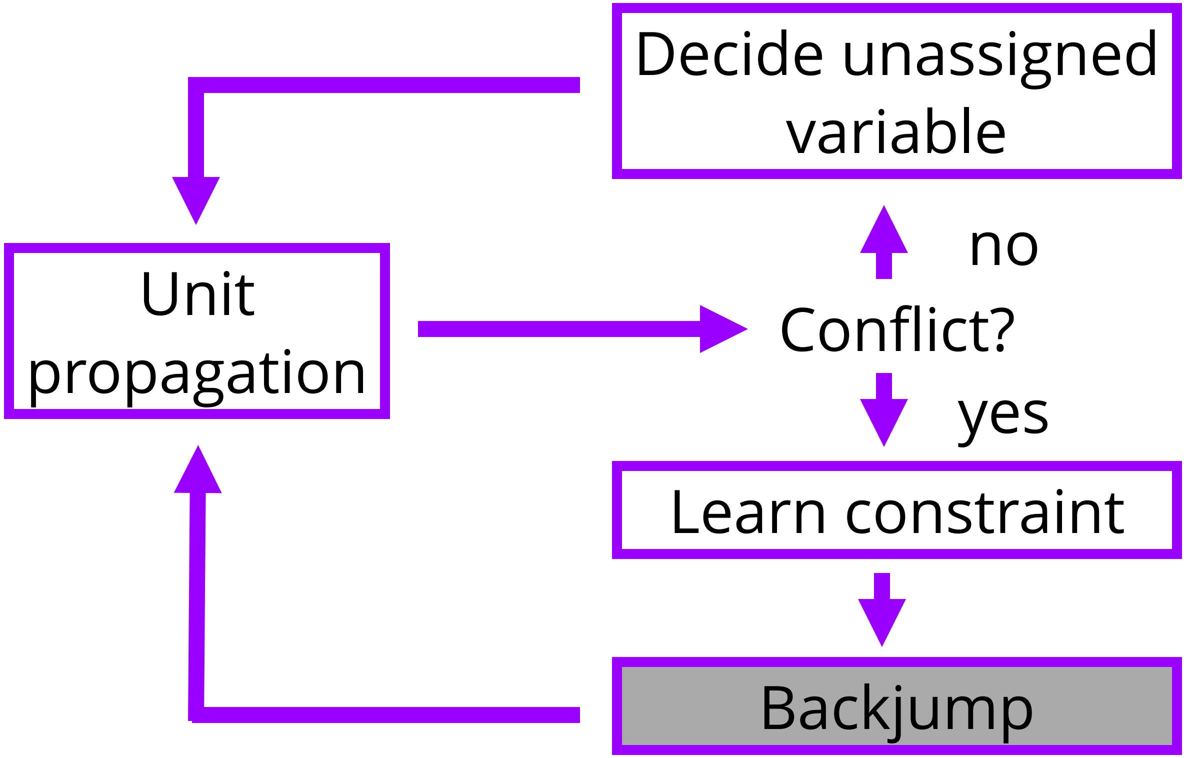
Conflict-driven Search Loop

$$+x + y - z \geq 0$$

$$-\, y + z - v \geq 0$$

$$-\, z + v - w \geq 0$$

$$-\, x + z + w \geq 1$$



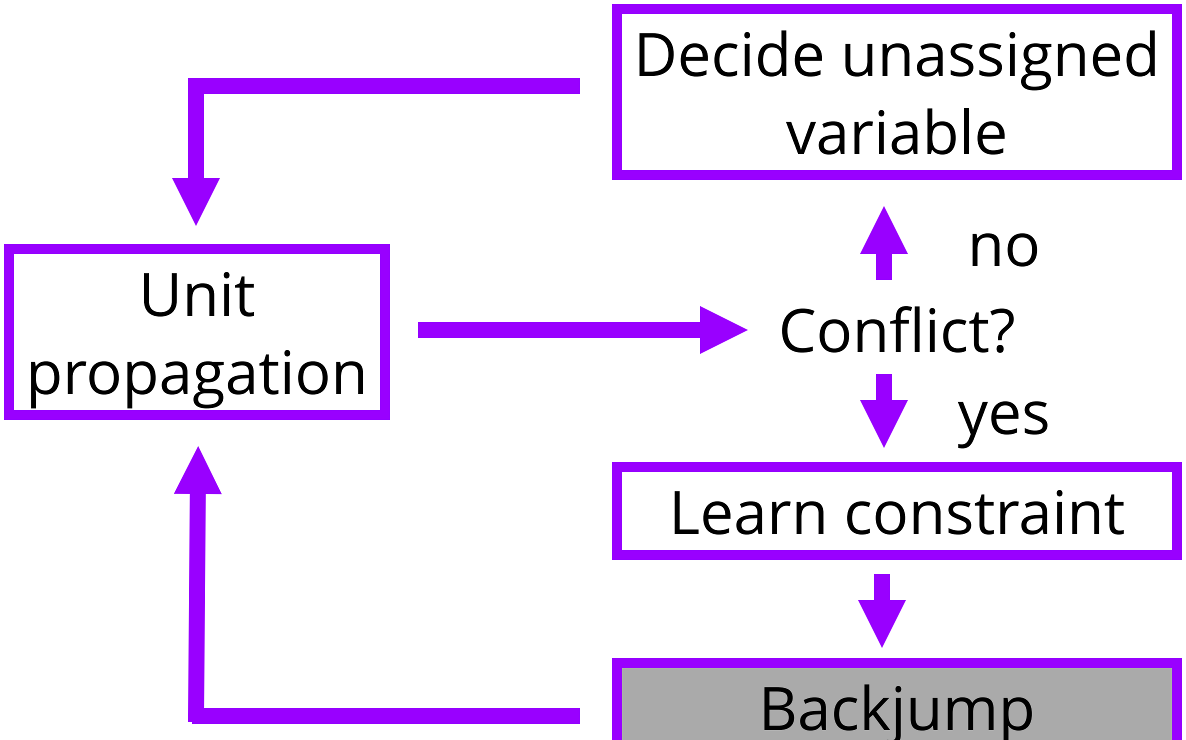
$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$

$$-y - w \geq 0$$

Conflict-driven Search Loop

- Backtrack based on learned constraint

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$



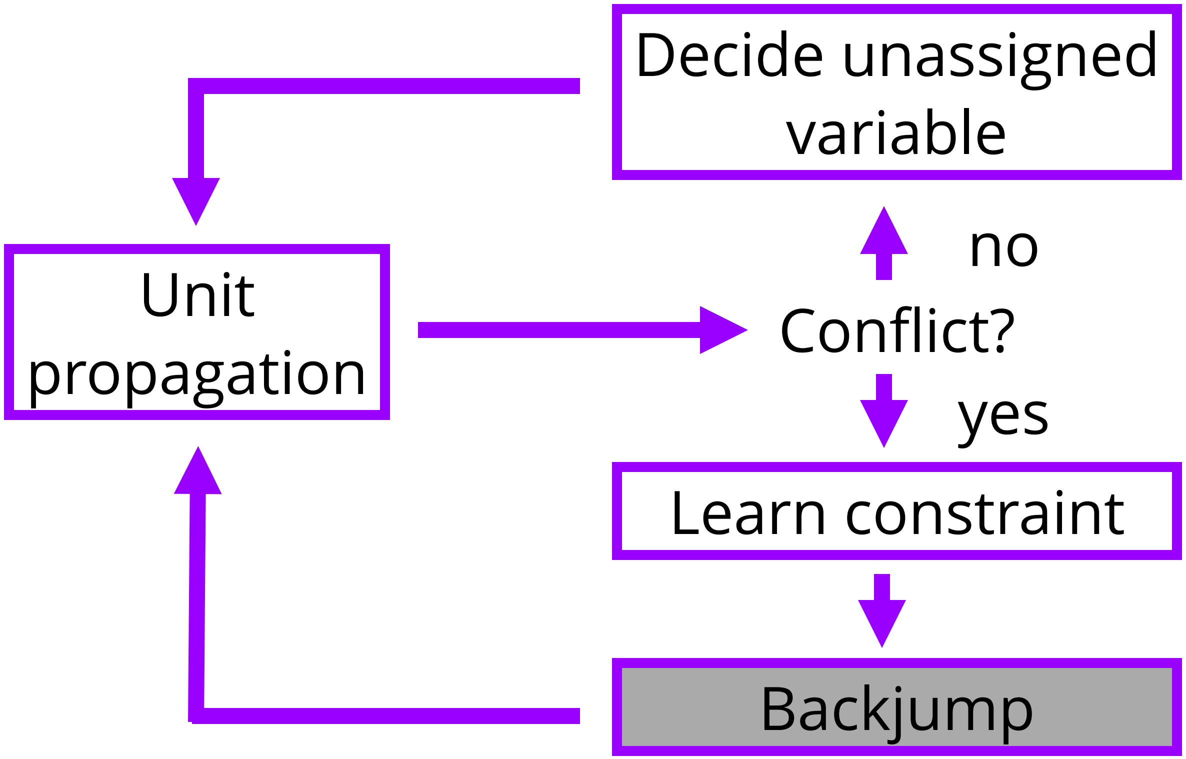
$$\alpha = \{x = 0, y = 1,$$
$$z = 1, v = 0\}$$

$$-y - w \geq 0$$

Conflict-driven Search Loop

- Backtrack based on learned constraint
- Resume unit propagation

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$



$$\alpha = \{x = 0, y = 1, z = 1, v = 0\}$$

$$-y - w \geq 0$$

Conflict-driven Search Loop

- Backtrack based on learned constraint
- Resume unit propagation

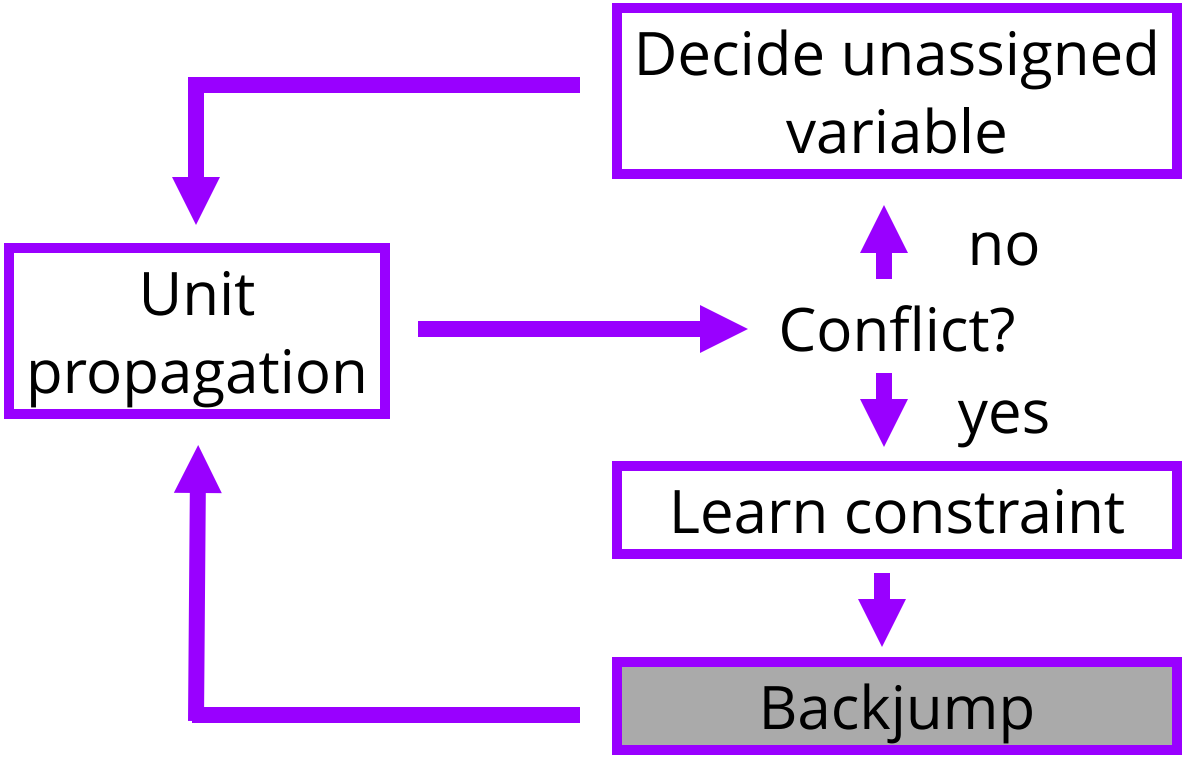
$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$

$$\alpha = \{\}$$



$$-y - w \geq 0$$

Conflict-driven Search Loop

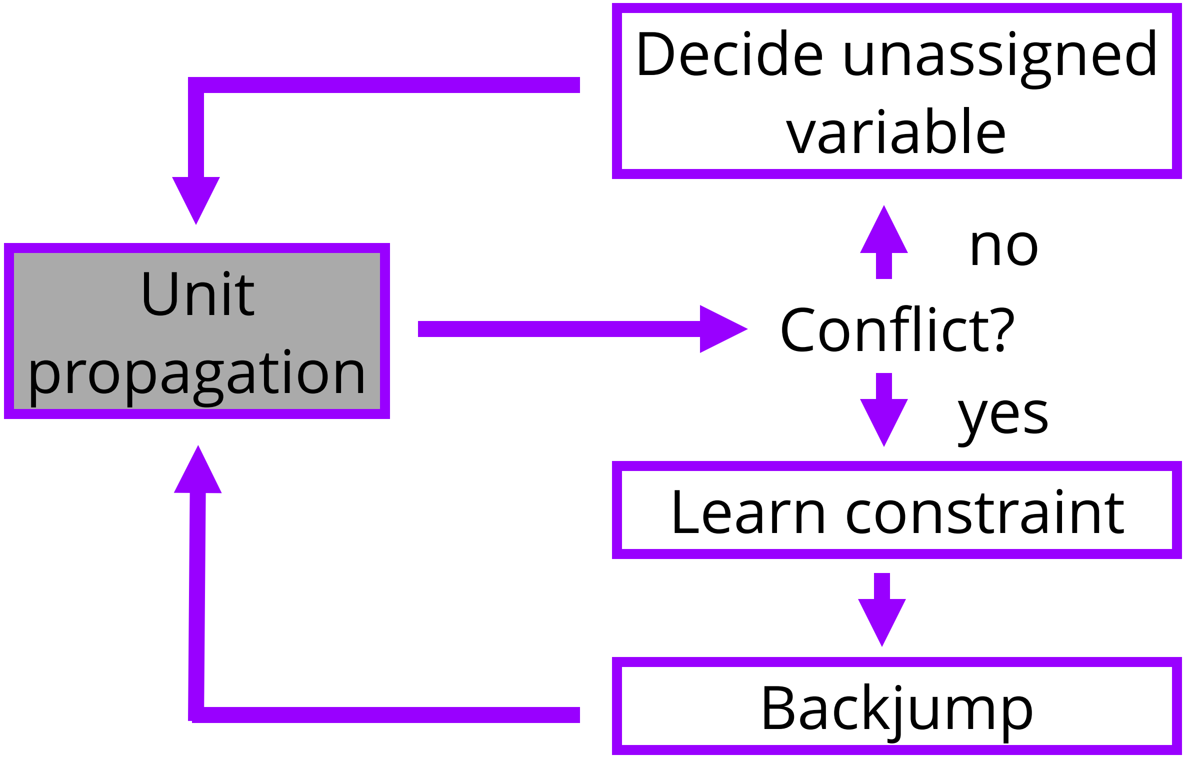
$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$

$$\alpha = \{\}$$



$$-y - w \geq 0$$

Conflict-driven Search Loop

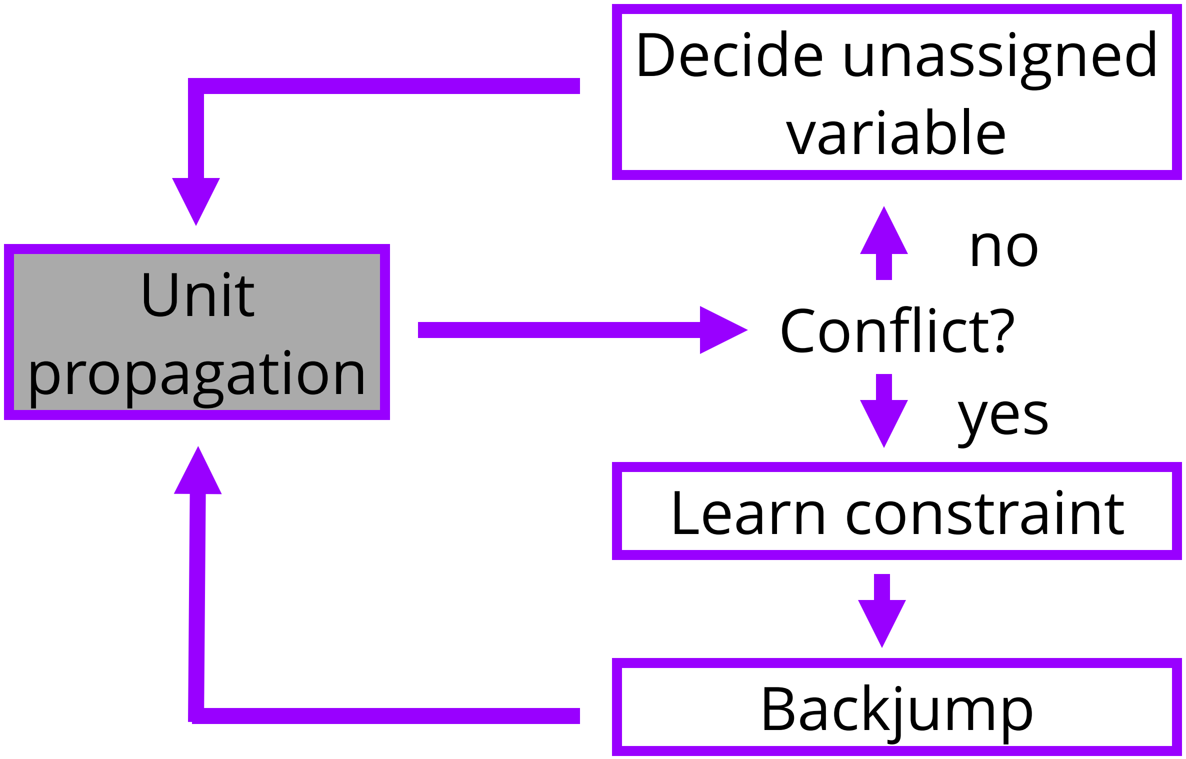
$$+x + y - z \geq 0$$

$$-\,y + z - v \geq 0$$

$$-\,z + v - w \geq 0$$

$$-\,x + z + w \geq 1$$

$$\alpha = \{\}$$



$$-y - w \geq 0$$

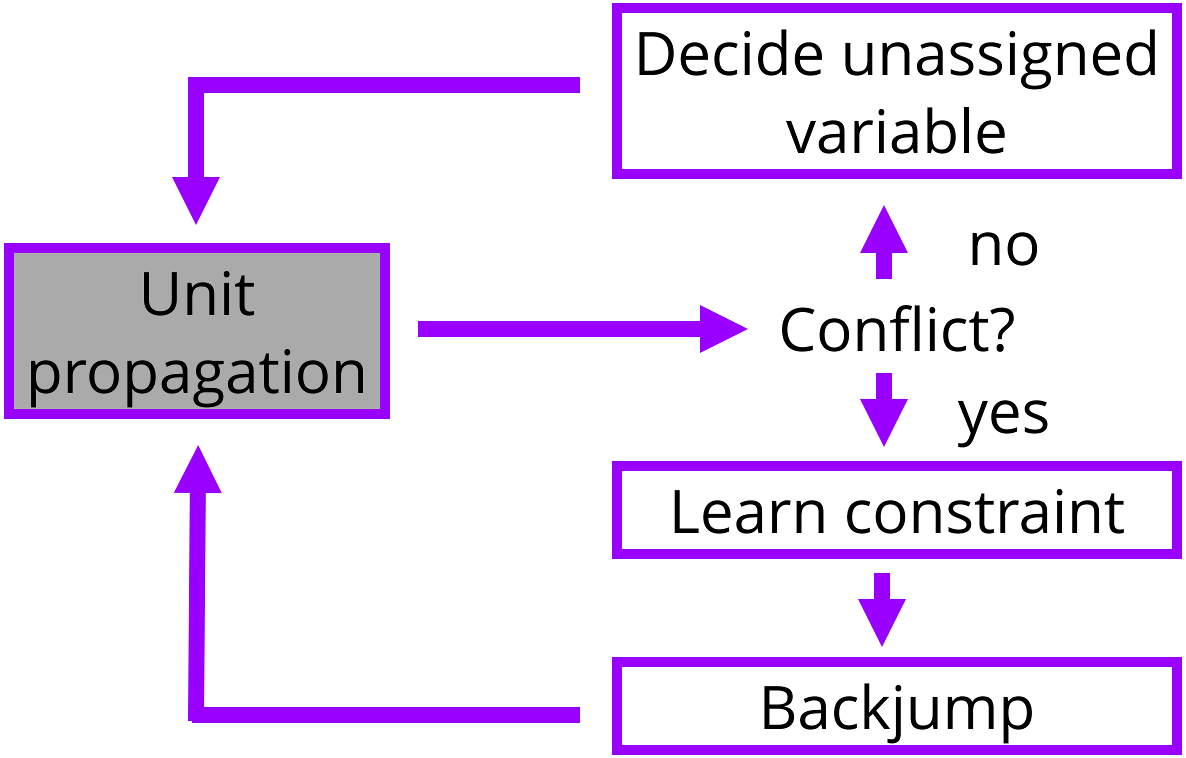
Conflict-driven Search Loop

$$+x - z \geq 0$$

$$+z - v \geq 0$$

$$-z + v \geq 0$$

$$-x + z \geq 1$$



$$\alpha = \{y = 0, w = 0\}$$

$$-y - w \geq 0$$

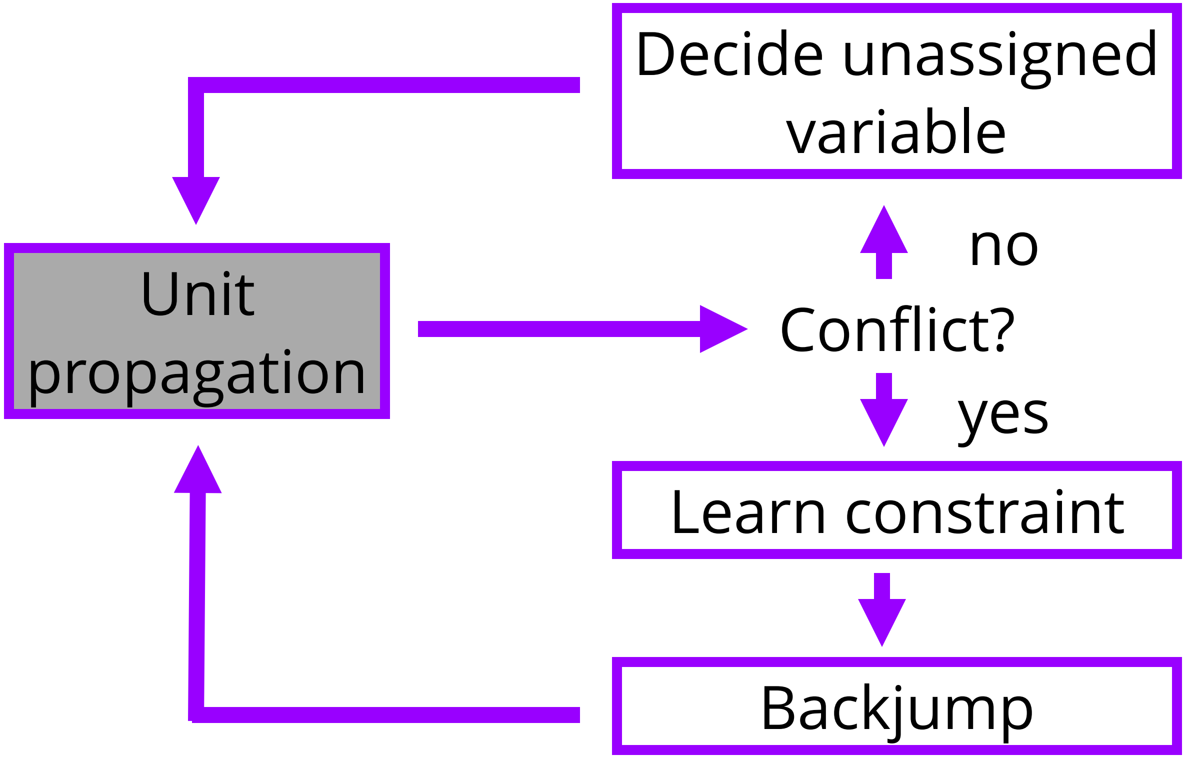
Conflict-driven Search Loop

$$+x - z \geq 0$$

$$+z - v \geq 0$$

$$-z + v \geq 0$$

$$-x + z \geq 1$$



$$\alpha = \{y = 0, w = 0\}$$

$$-y - w \geq 0$$

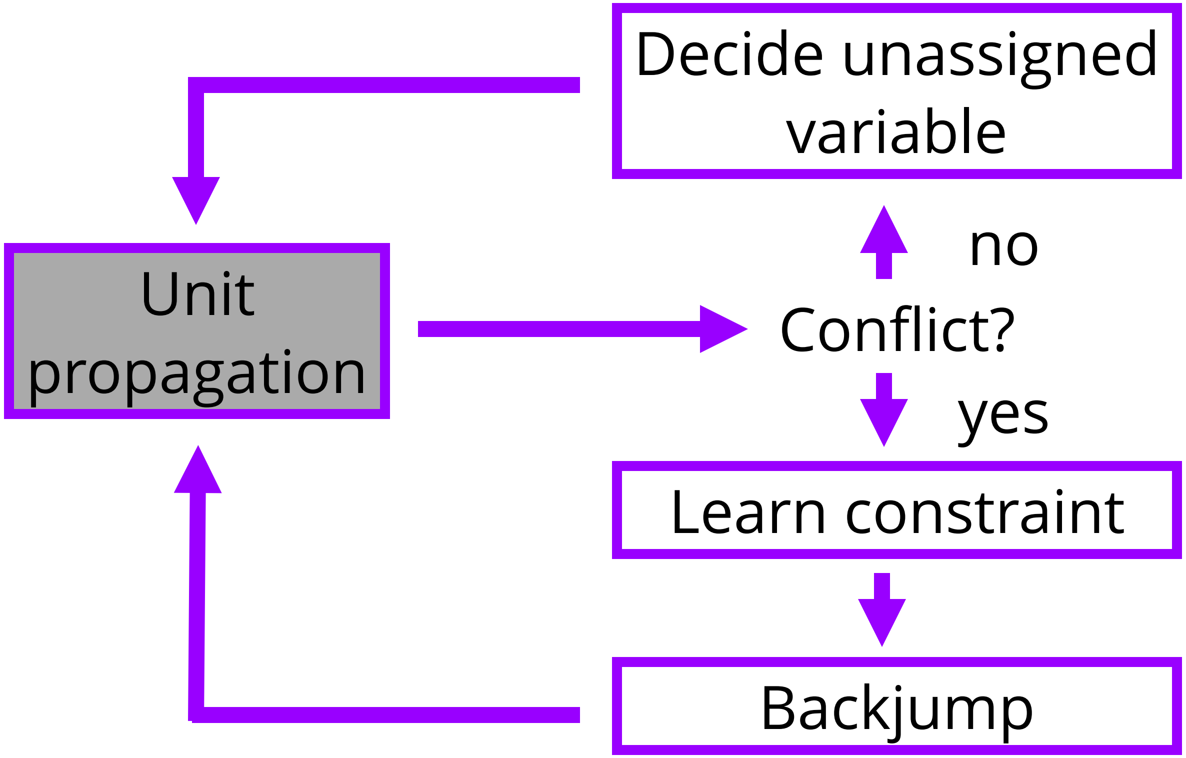
Conflict-driven Search Loop

$$-1 \geq 0$$

$$1 - v \geq 0$$

$$-1 + v \geq 0$$

$$1 \geq 1$$



$$\alpha = \{y = 0, w = 0, x = 0, z = 1\}$$

$$-y - w \geq 0$$

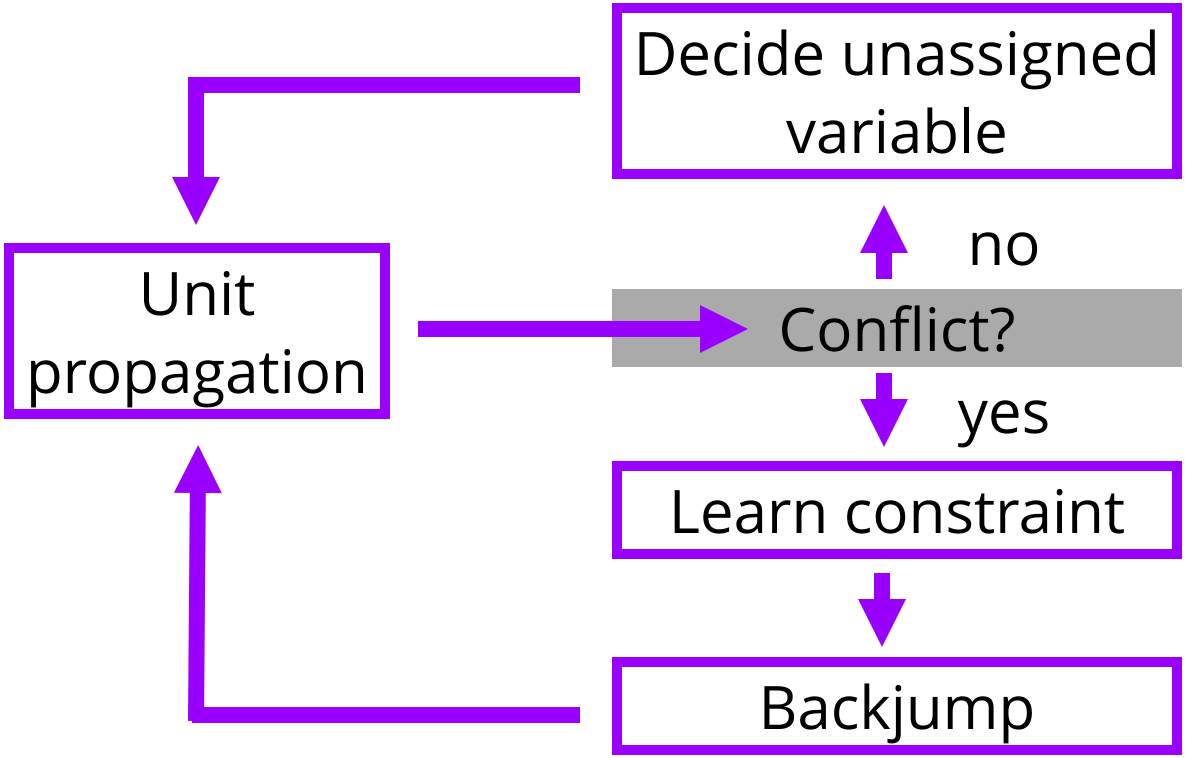
Conflict-driven Search Loop

$$-1 \geq 0$$

$$1 - v \geq 0$$

$$-1 + v \geq 0$$

$$1 \geq 1$$



$$\alpha = \{y = 0, w = 0, x = 0, z = 1\}$$

$$-y - w \geq 0$$

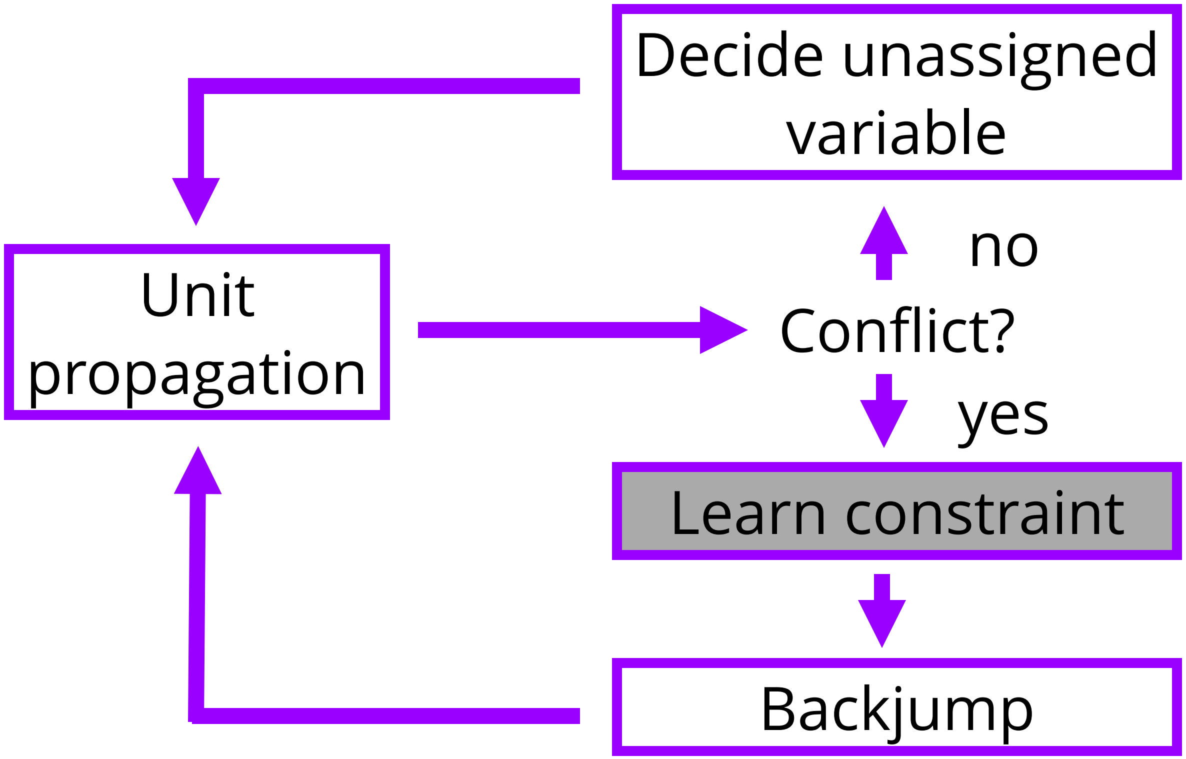
Conflict-driven Search Loop

$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$



$$\alpha = \{y = 0, w = 0, x = 0, z = 1\}$$

$$-y - w \geq 0$$

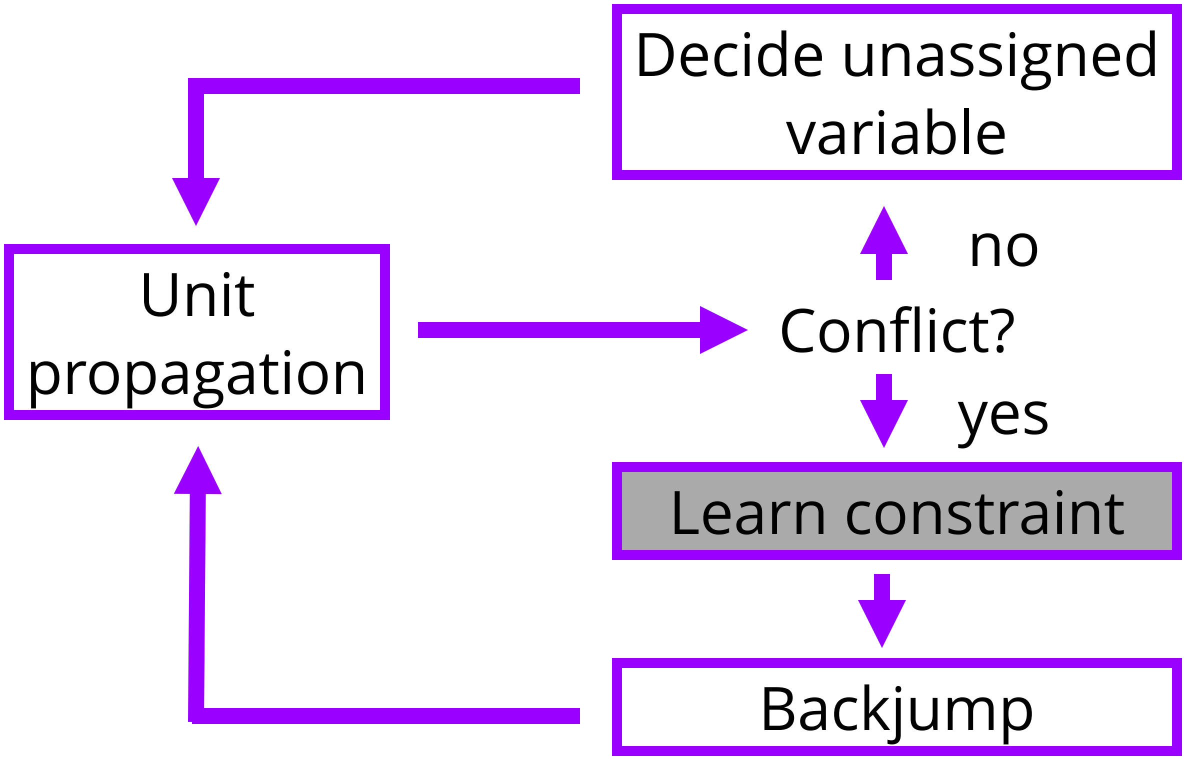
Conflict-driven Search Loop

$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$



$$\alpha = \{y = 0, w = 0, x = 0, z = 1\}$$

$$-y - w \geq 0$$

Conflict-driven Search Loop

$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

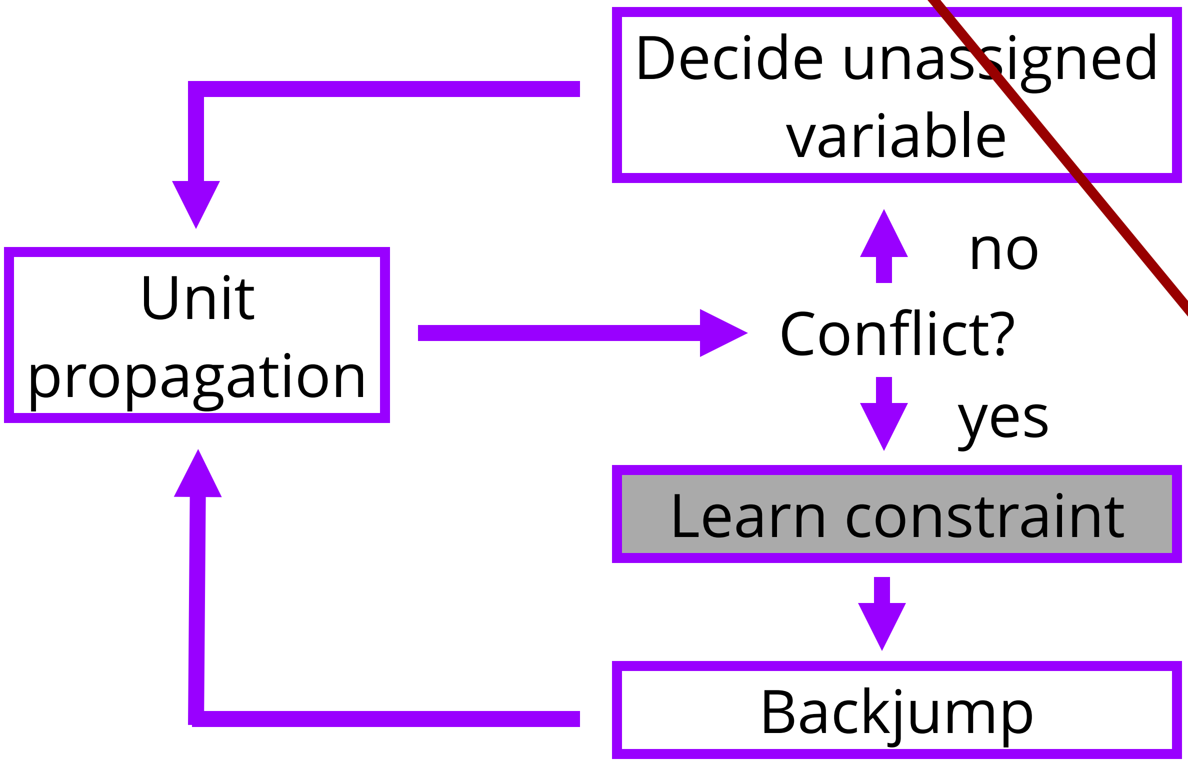
$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$

$$-y - w \geq 0$$

$$0 \geq 1$$

$$\alpha = \{y = 0, w = 0, x = 0, z = 1\}$$



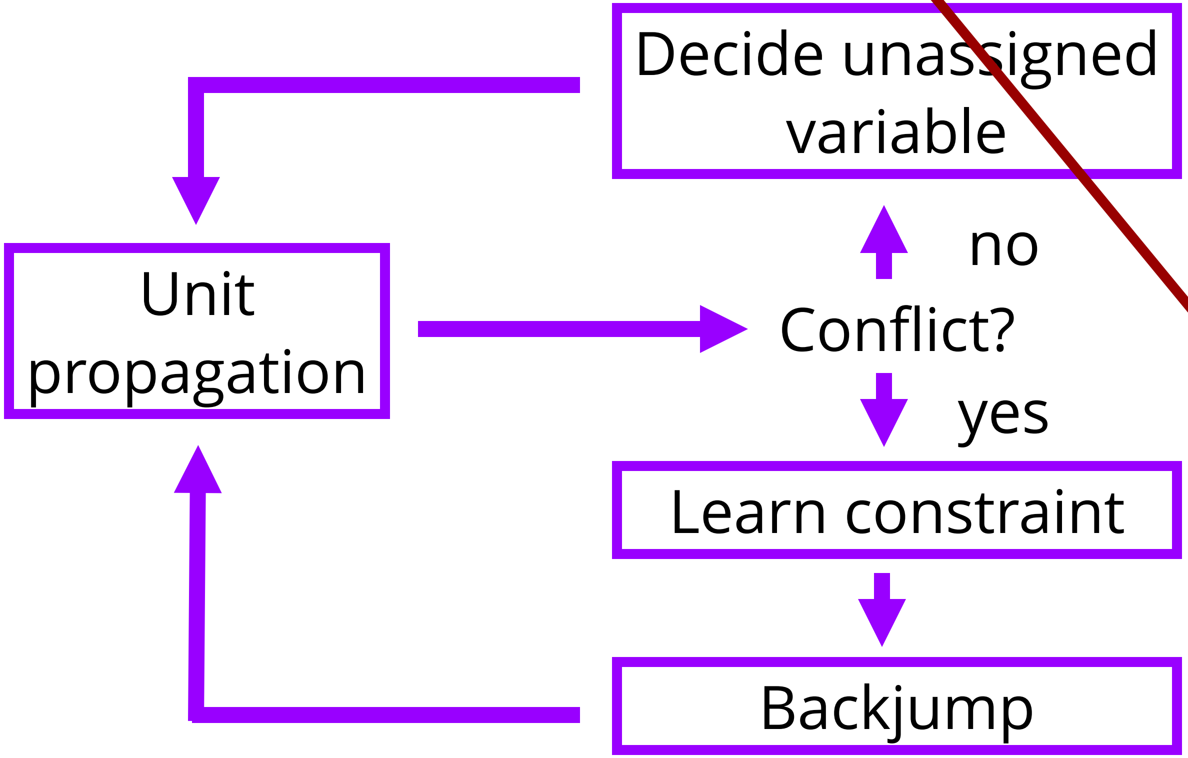
Conflict-driven Search Loop

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

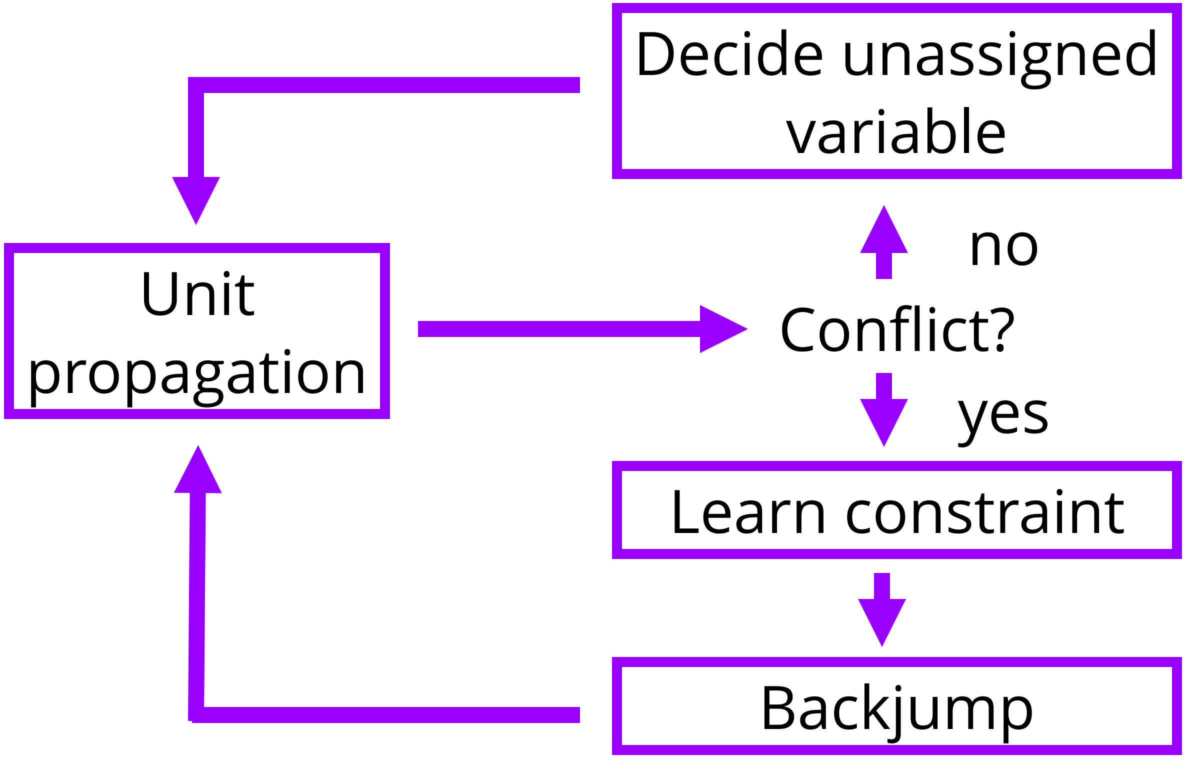
UNSAT! $0 \geq 1$

$$\alpha = \{y = 0, w = 0, x = 0, z = 1\}$$

$$-y - w \geq 0$$

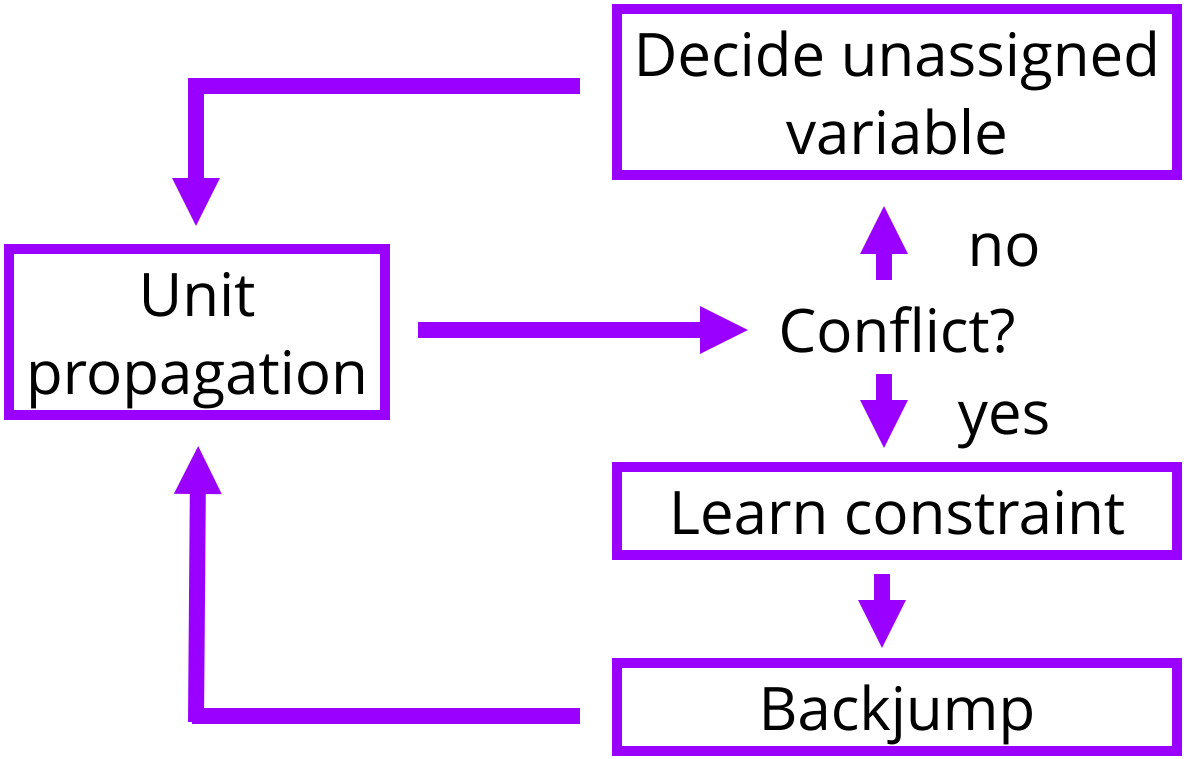


Conflict-driven Search Loop



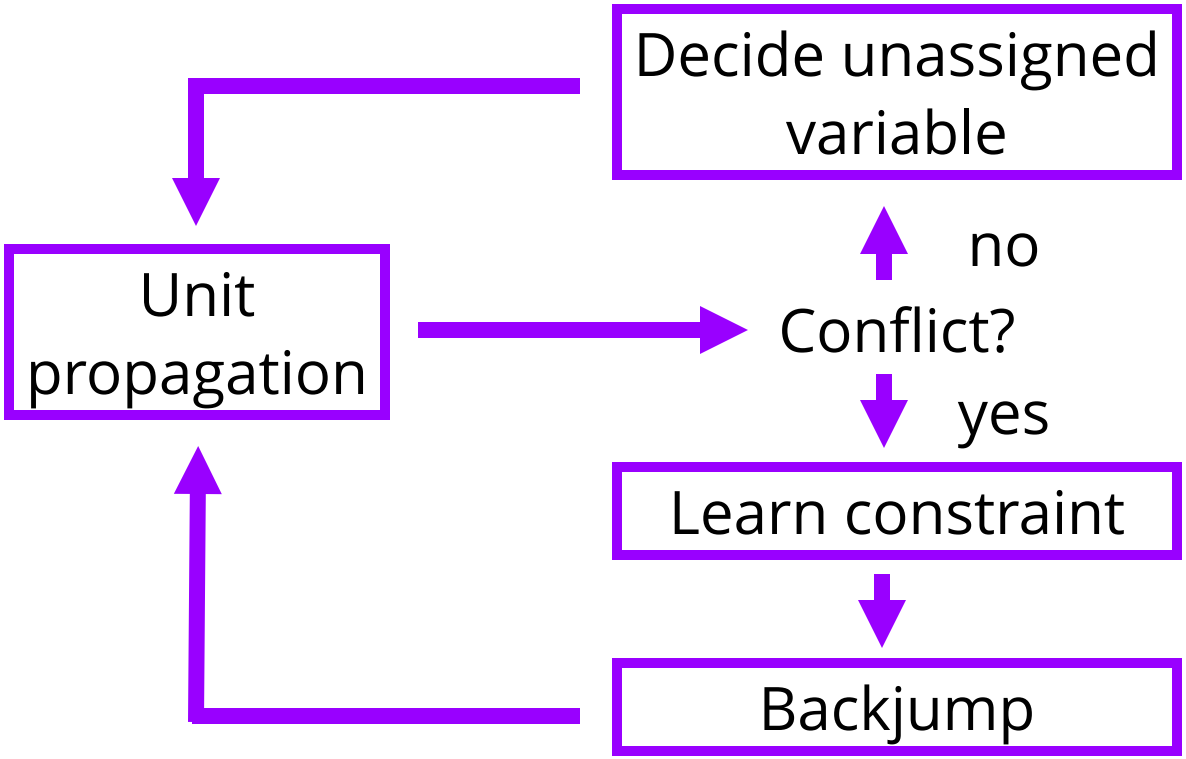
Conflict-driven Search Loop

- Learning constraints pushes search forward



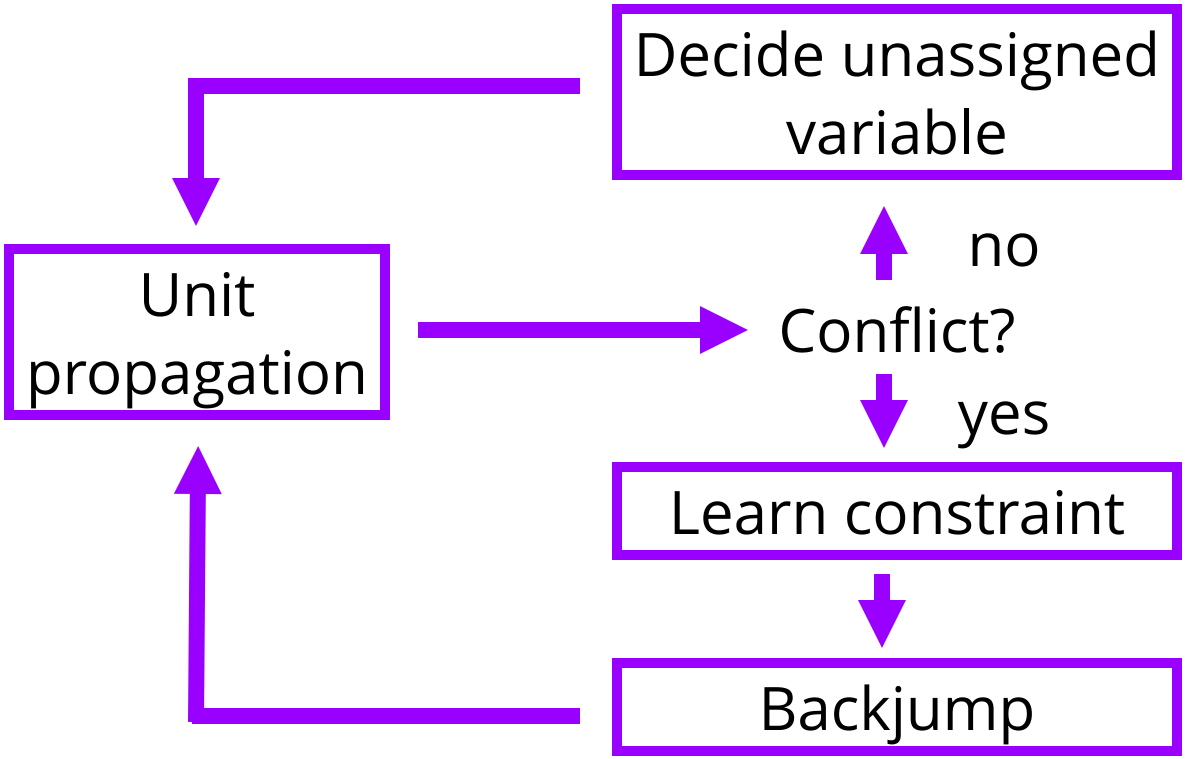
Conflict-driven Search Loop

- Learning constraints pushes search forward
- Thousands of conflicts per second



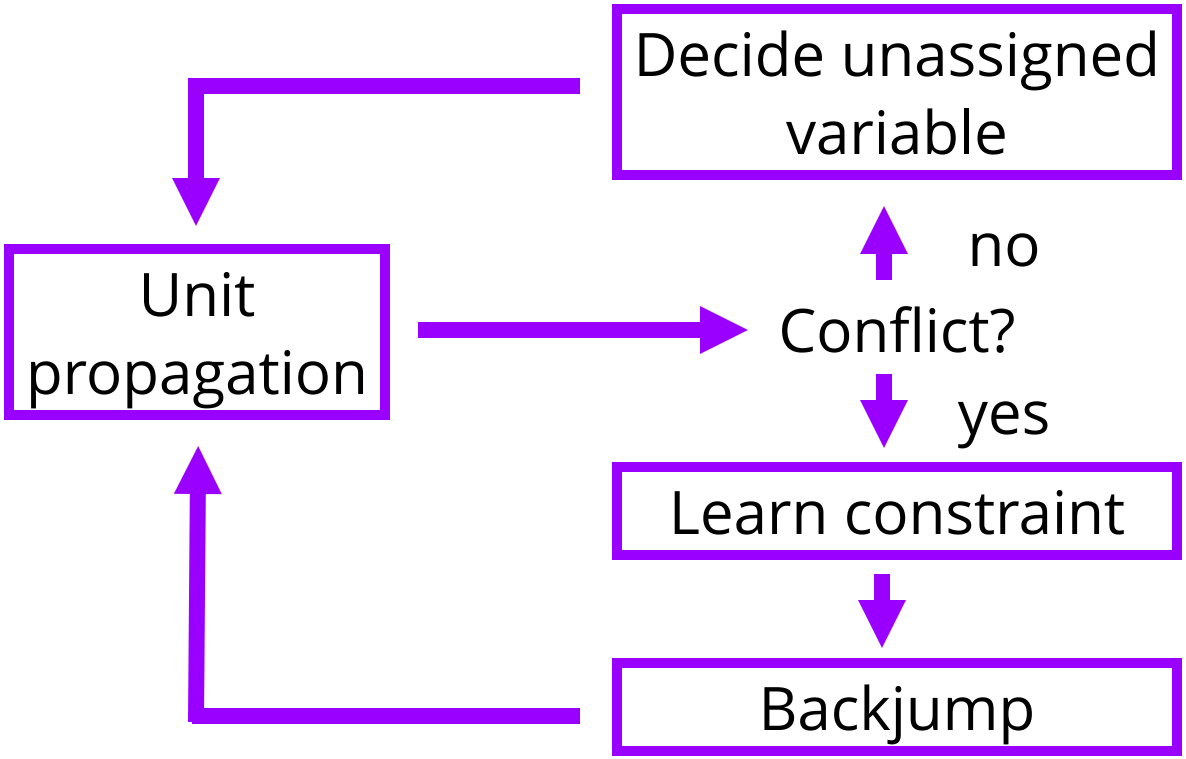
Conflict-driven Search Loop

- Learning constraints pushes search forward
- Thousands of conflicts per second
- Highly optimized unit propagation



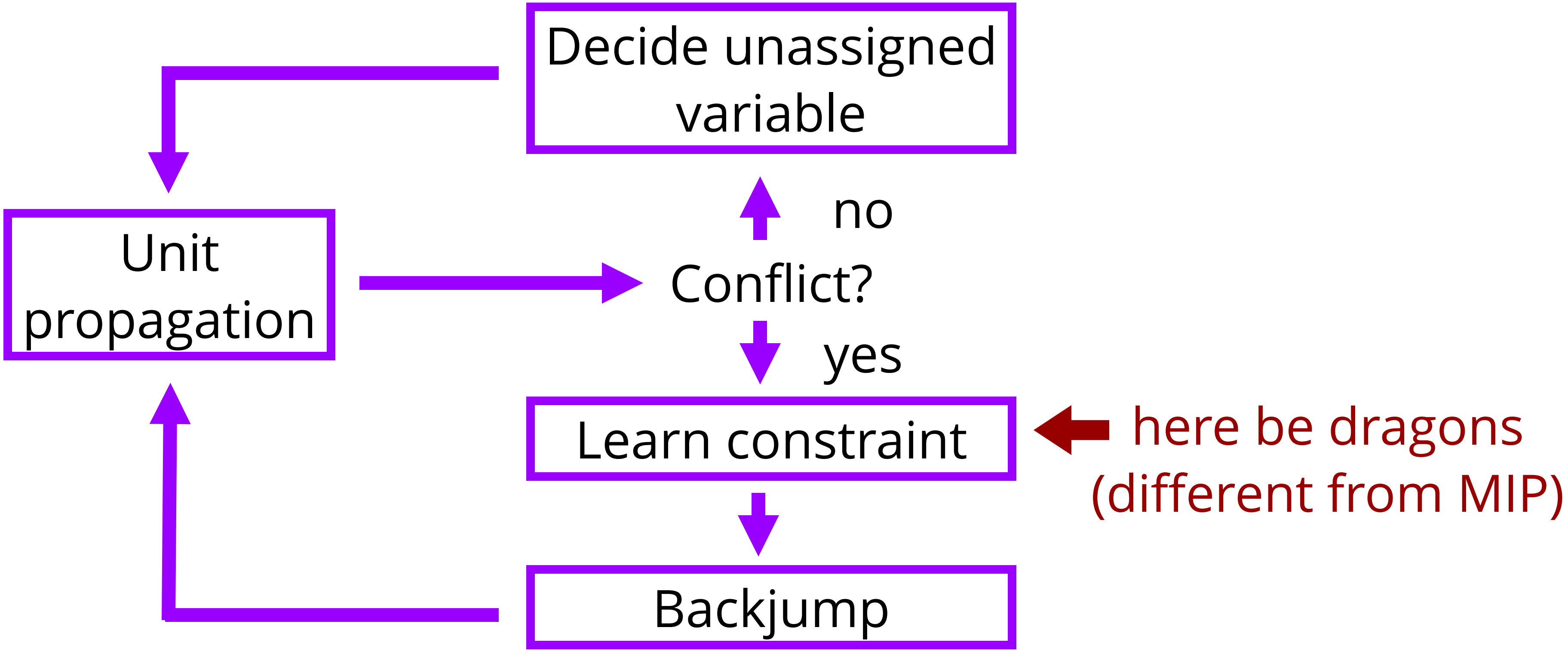
Conflict-driven Search Loop

- Learning constraints pushes search forward
- Thousands of conflicts per second
- Highly optimized unit propagation
- First proposed for Boolean satisfiability (**SAT**) [MS96,BS97,MMZZM01]



Conflict-driven Search Loop

- Learning constraints pushes search forward
- Thousands of conflicts per second
- Highly optimized unit propagation
- First proposed for Boolean satisfiability (**SAT**) [MS96,BS97,MMZZM01]
- Generalized to pseudo-Boolean (**PB**) solving [CK05,SS06,LP10,EN18]
  - many variations possible



Another look at our example...

$$+x + y - z \geq 0$$

$$- y + z - v \geq 0$$

$$- z + v - w \geq 0$$

$$- x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

Another look at our example...

$$+x + y - z \geq 0$$

$$-y + z - v \geq 0$$

$$-z + v - w \geq 0$$

$$-x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

- It's rationally infeasible!

Another look at our example...

$$+x + y - z \geq 0$$

$$-\ y + z - v \geq 0$$

$$-\ z + v - w \geq 0$$

$$-\ x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

- It's rationally infeasible!
- Could be solved without search

Another look at our example...

$$+x + y - z \geq 0$$

$$- y + z - v \geq 0$$

$$- z + v - w \geq 0$$

$$- x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

- It's rationally infeasible!
- Could be solved without search
- In theory: rationally infeasible programs are easy for conflict-driven PB search

Another look at our example...

$$+x + y - z \geq 0$$
$$- y + z - v \geq 0$$
$$- z + v - w \geq 0$$
$$- x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

- It's rationally infeasible!
- Could be solved without search
- In theory: rationally infeasible programs are easy for conflict-driven PB search
- In practice: PB solvers **timeout** on certain rationally infeasible programs [EGNV18]
  - unit propagation is **local**
  - **wrong constraints** are learned

Another look at our example...

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

$$x, y, z, v, w \mapsto \{0, 1\}$$

- It's rationally infeasible!
- Could be solved without search
- In theory: rationally infeasible programs are easy for conflict-driven PB search
- In practice: PB solvers **timeout** on certain rationally infeasible programs [EGNV18]
  - unit propagation is **local**
  - **wrong constraints** are learned

How about integrating an LP solver?

Another look at our example...

$$+x + y - z \geq 0$$
$$-y + z - v \geq 0$$
$$-z + v - w \geq 0$$
$$-x + z + w \geq 1$$

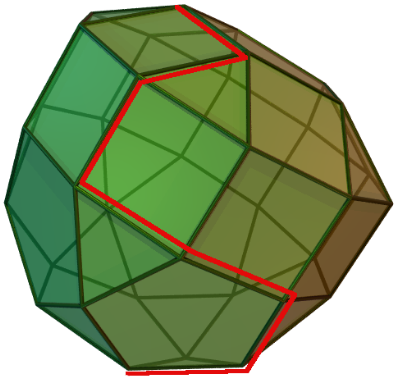
$$x, y, z, v, w \mapsto \{0, 1\}$$

- It's rationally infeasible!
- Could be solved without search
- In theory: rationally infeasible programs are easy for conflict-driven PB search
- In practice: PB solvers **timeout** on certain rationally infeasible programs [EGNV18]
  - unit propagation is **local**
  - **wrong constraints** are learned

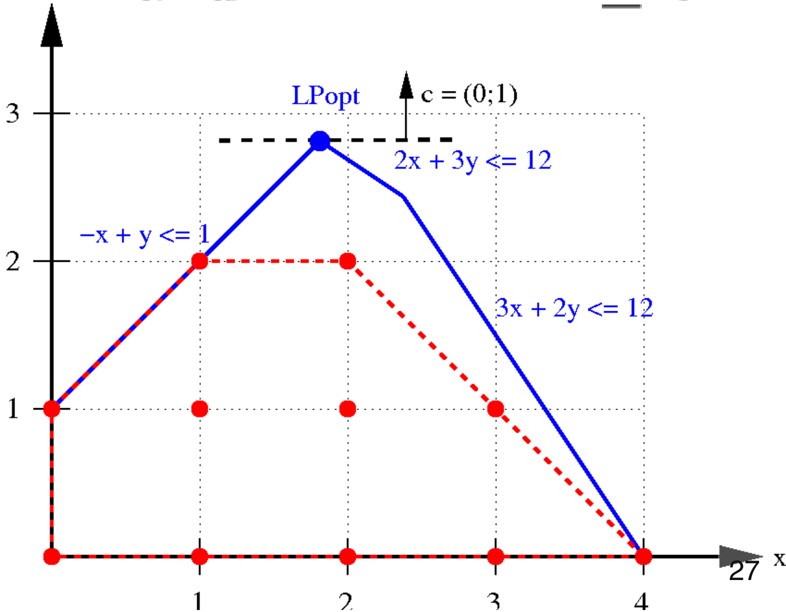
How about integrating an LP solver?

our work

Linear Programming (LP) solver

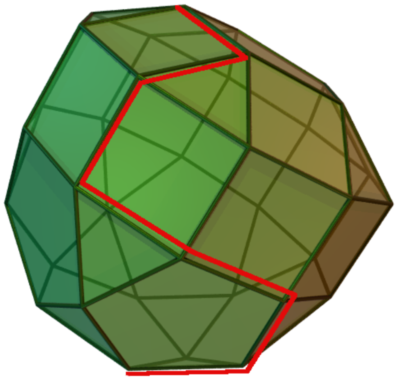


$$\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array}$$

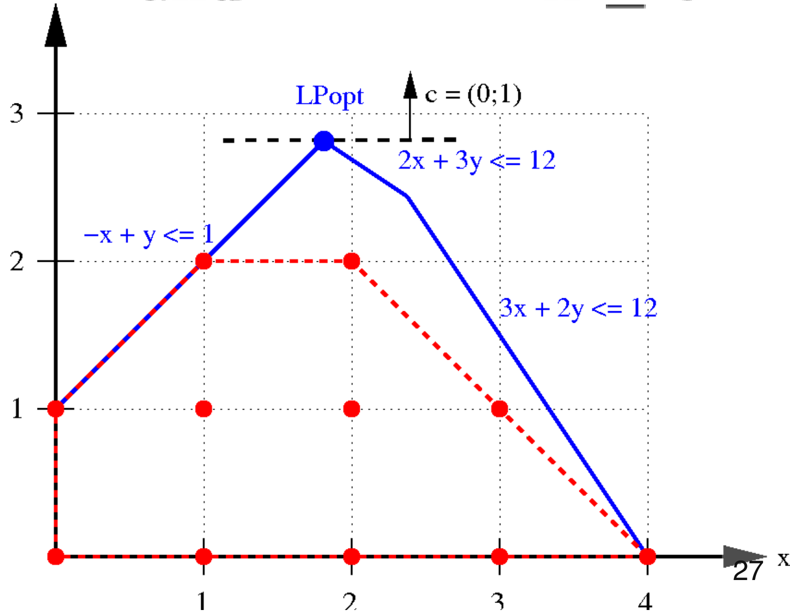


Linear Programming (LP) solver

- Input:
  - LP relaxation of φ
  - variable bounds α
  - objective function

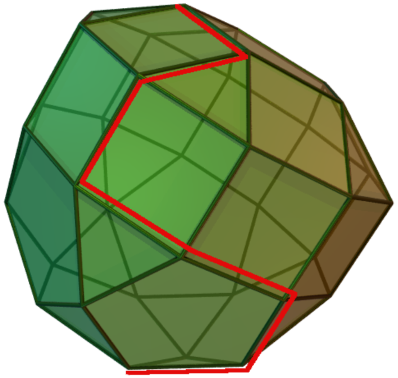


$$
\begin{array}{ll}
\text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\
\text{subject to} & A\mathbf{x} \leq \mathbf{b} \\
\text{and} & \mathbf{x} \geq \mathbf{0}
\end{array}
$$

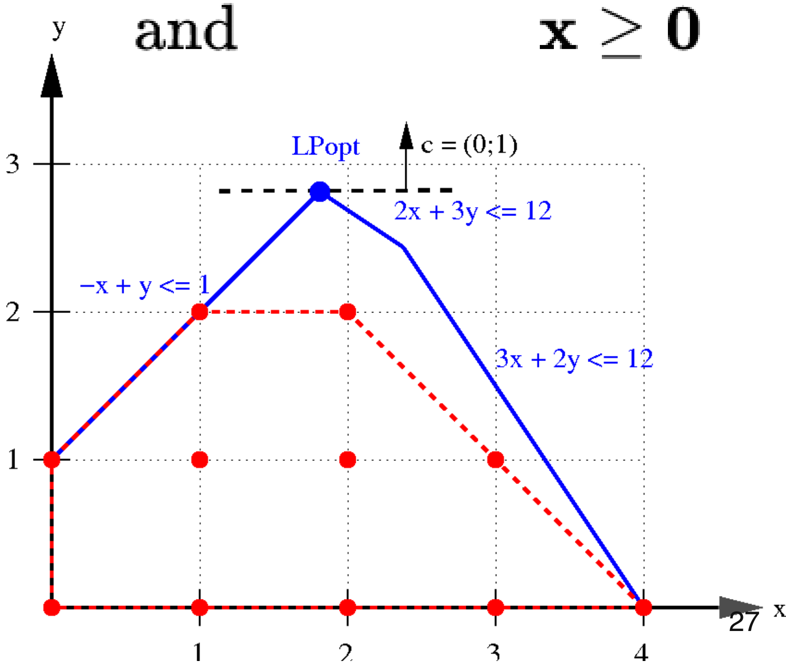


Linear Programming (LP) solver

- Input:
  - LP relaxation of φ
  - variable bounds α
  - objective function
- Output: either
  - optimal rational solution
  - **Farkas multipliers**
    - define a positive linear combination of constraints in φ, falsified by α

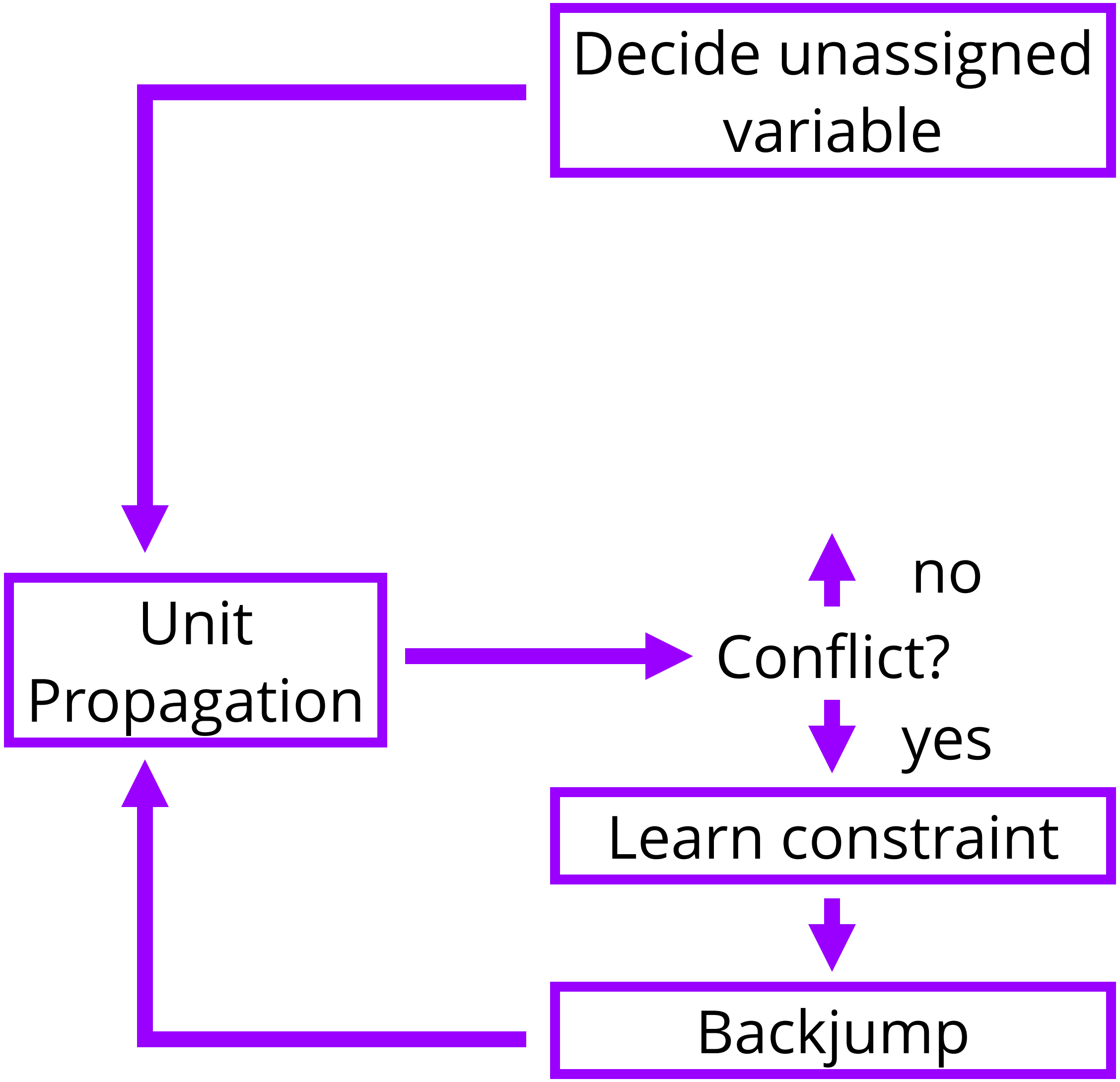


$$\begin{array}{ll} \text{minimize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array}$$



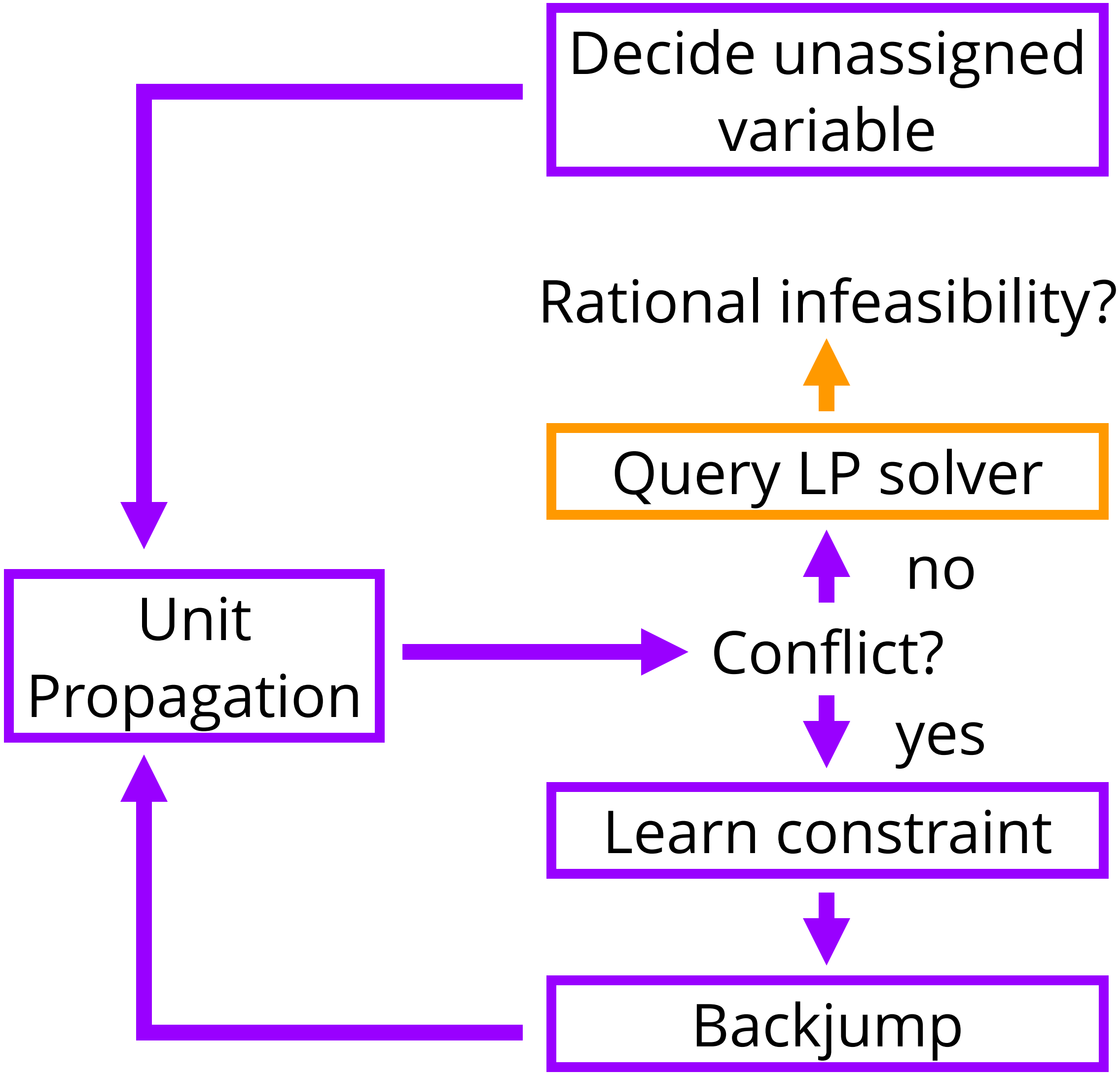
Conflict-driven search loop

with LP solver call



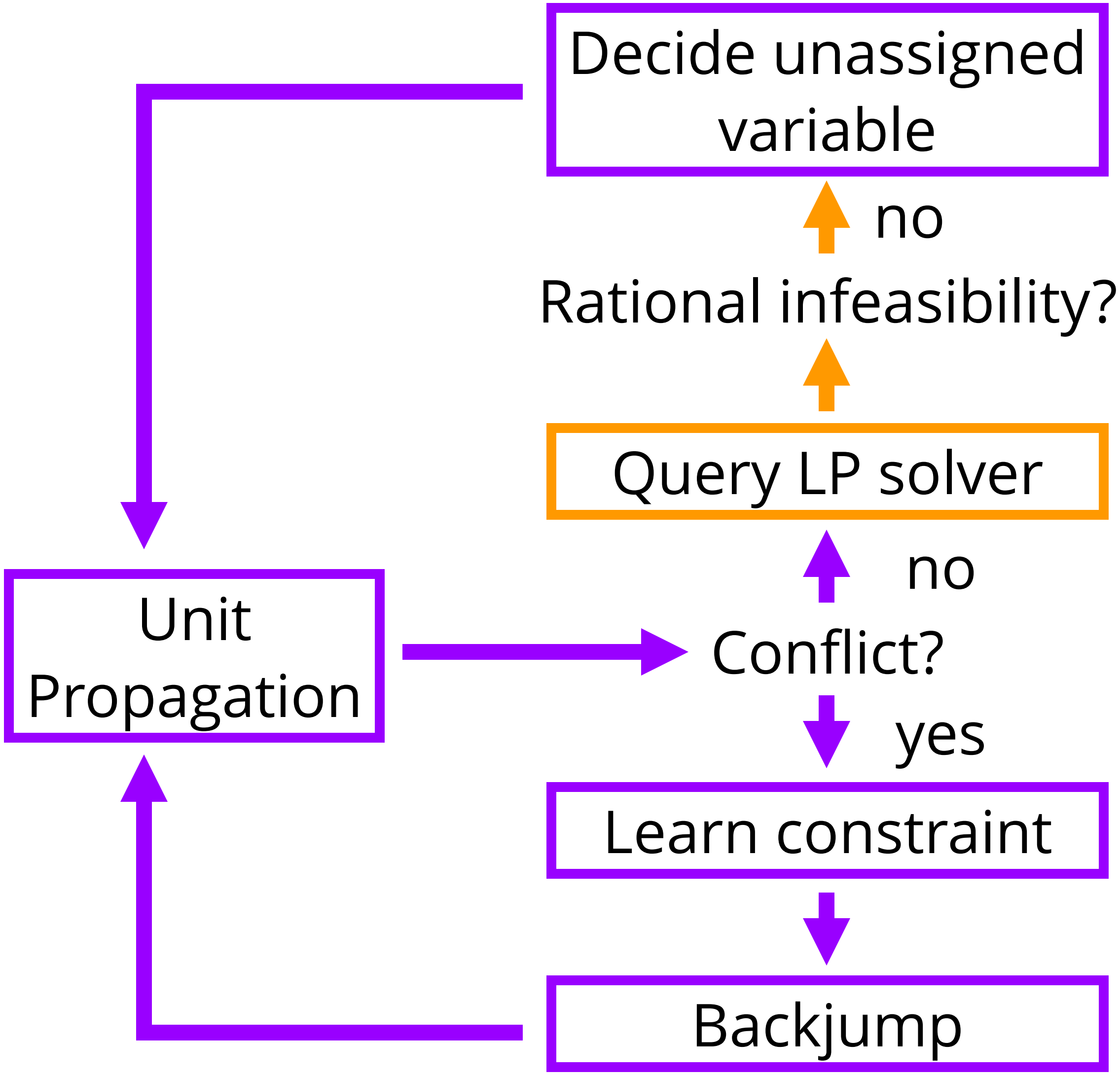
Conflict-driven search loop

with LP solver call



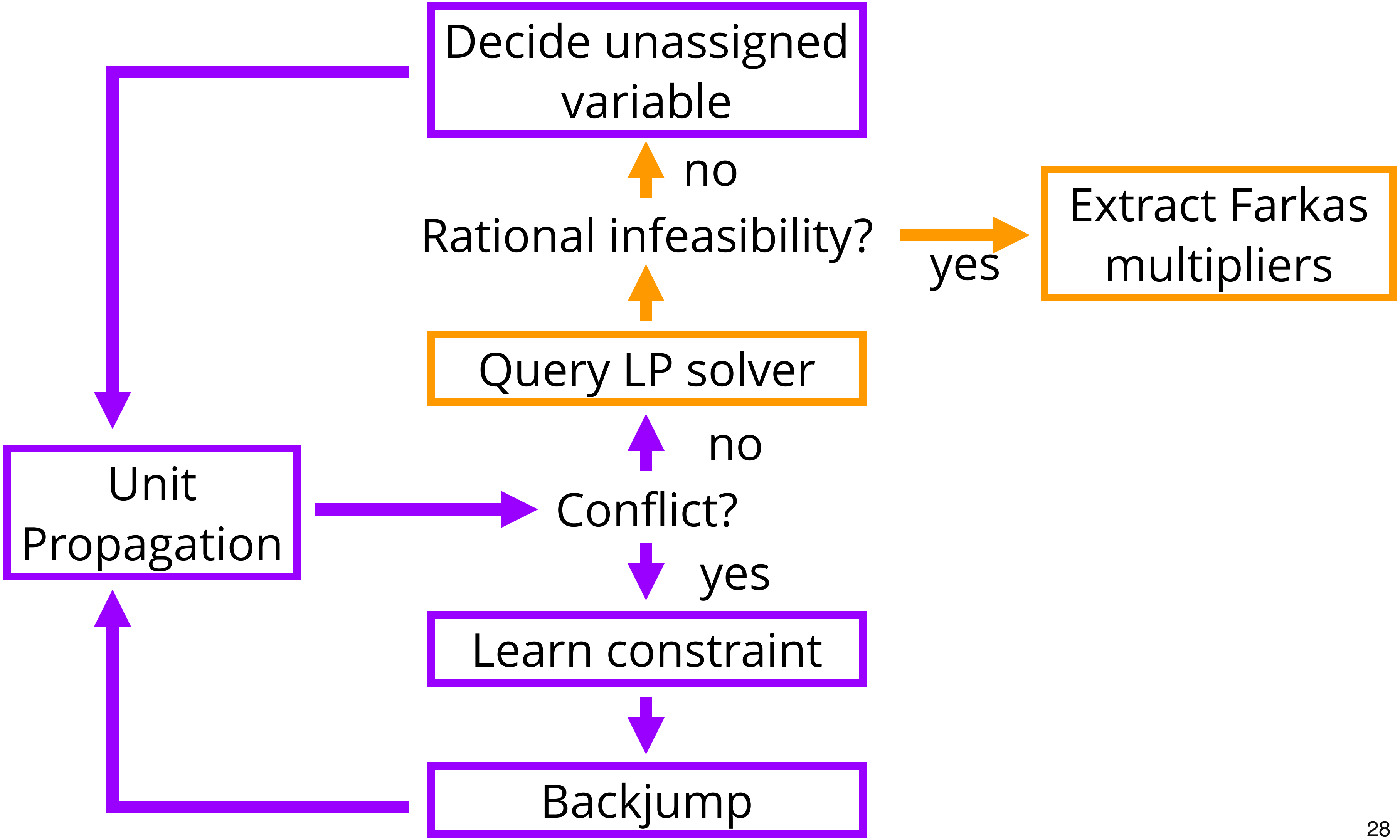
Conflict-driven search loop

with LP solver call



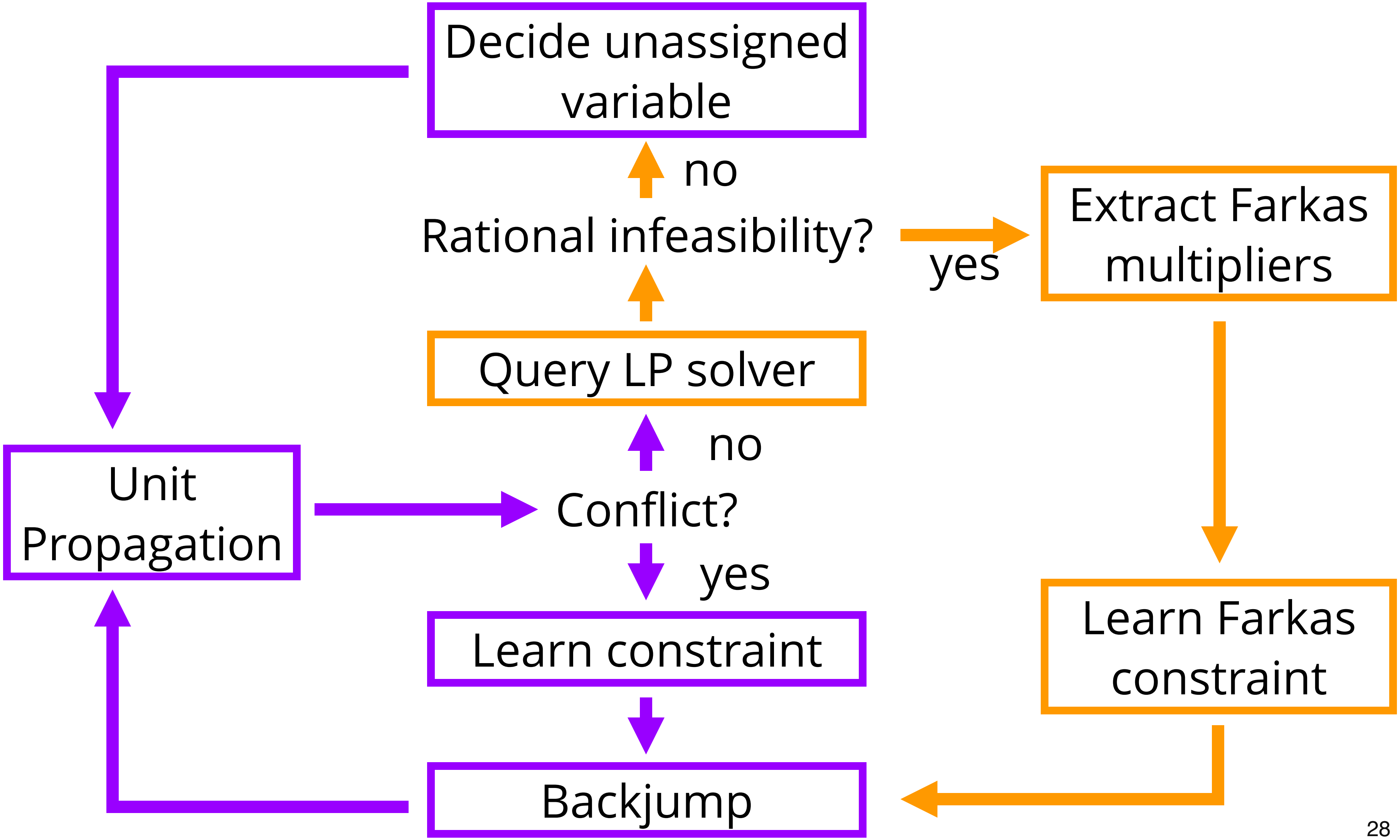
Conflict-driven search loop

with LP solver call



Conflict-driven search loop

with LP solver call



Rational infeasibility example

$$
\begin{aligned}
+a + x + y - z &\geq 0 \\
+b - y + z - v &\geq 0 \\
-z + v - w &\geq 0 \\
-x + z + w &\geq 1 \\
+a - b &\geq 0
\end{aligned}
$$

Rational infeasibility example

$$+a + x + y - z \geq 0$$

$$+ b - y + z - v \geq 0$$

$$- z + v - w \geq 0$$

$$- x + z + w \geq 1$$

$$+ a - b \geq 0$$

$$\alpha = \{a = 0, b = 0\}$$

Rational infeasibility example

Farkas multipliers

$$+a + x + y - z \geq 0 \quad \times 1$$

$$+ b - y + z - v \geq 0 \quad \times 1$$

$$- z + v - w \geq 0 \quad \times 1$$

$$- x + z + w \geq 1 \quad \times 1$$

$$+ a - b \geq 0 \quad \times 0$$

$$\alpha = \{a = 0, b = 0\}$$

Rational infeasibility example

Farkas multipliers

$$+a + x + y - z \geq 0 \quad \times 1$$

$$+b - y + z - v \geq 0 \quad \times 1$$

$$-z + v - w \geq 0 \quad \times 1 \qquad +a + b \geq 1$$

$$-x + z + w \geq 1 \quad \times 1$$

$$+a - b \geq 0 \quad \times 0$$

$$\alpha = \{a = 0, b = 0\}$$

Two technical hurdles

Two technical hurdles

- LP solvers are **slow** compared to conflict-driven search loop
  - limit calls to LP solver
  - limit LP solver running time
  - deterministic measure: **balance #conflicts** in conflict-driven solver to **#pivots** in LP solver

Two technical hurdles

- LP solvers are **slow** compared to conflict-driven search loop
  - limit calls to LP solver
  - limit LP solver running time
  - deterministic measure: **balance #conflicts** in conflict-driven solver to **#pivots** in LP solver
- LP solver uses inexact floating point arithmetic
  - learned constraint must be implied by φ
  - **recalculate** Farkas constraint with exact arithmetic
  - **verify** Farkas constraint is still conflicting
  - **post-process** Farkas constraint to eliminate noise

Further ideas

- Every once in a while, run LP solver to completion at root

Further ideas

- Every once in a while, run LP solver to completion at root
  - use (optimal) rational solution as **value heuristic**

Further ideas

- Every once in a while, run LP solver to completion at root
  - ☺ use (optimal) rational solution as **value heuristic**
  - ☺ generate **Chvátal-Gomory (CG) cuts**
    - add to both LP solver and learned constraint set

Further ideas

- Every once in a while, run LP solver to completion at root
  - use (optimal) rational solution as **value heuristic**
  - generate **Chvátal-Gomory (CG) cuts**
    - add to both LP solver and learned constraint set
- Generate "**deep**" Chvátal-Gomory cuts from internal search nodes
  - valid at root node, so safe to add as learned constraint

Further ideas

- Every once in a while, run LP solver to completion at root
  - use (optimal) rational solution as **value heuristic**
  - generate **Chvátal-Gomory (CG) cuts**
    - add to both LP solver and learned constraint set
- Generate "**deep**" Chvátal-Gomory cuts from internal search nodes
  - valid at root node, so safe to add as learned constraint
- Add **learned constraints as cuts** to the LP solver

Working implementation

- PB solver **RoundingSat** [EN18]
  - Strong ILP constraint learning
  - Performed well in past PB competitions
- LP solver **SoPlex** [ZIB]
  - SCIP's native LP solver
  - State-of-the-art open source

Design choices

- #pivots/#conflicts ≤ 1
- CG cut **parallelism check**
- for decision instances, **minimize sum of variables** in SoPlex
- for pure CNFs, **deactivate** LP techniques
- **128 bit precision** to calculate CG cuts and Farkas constraints

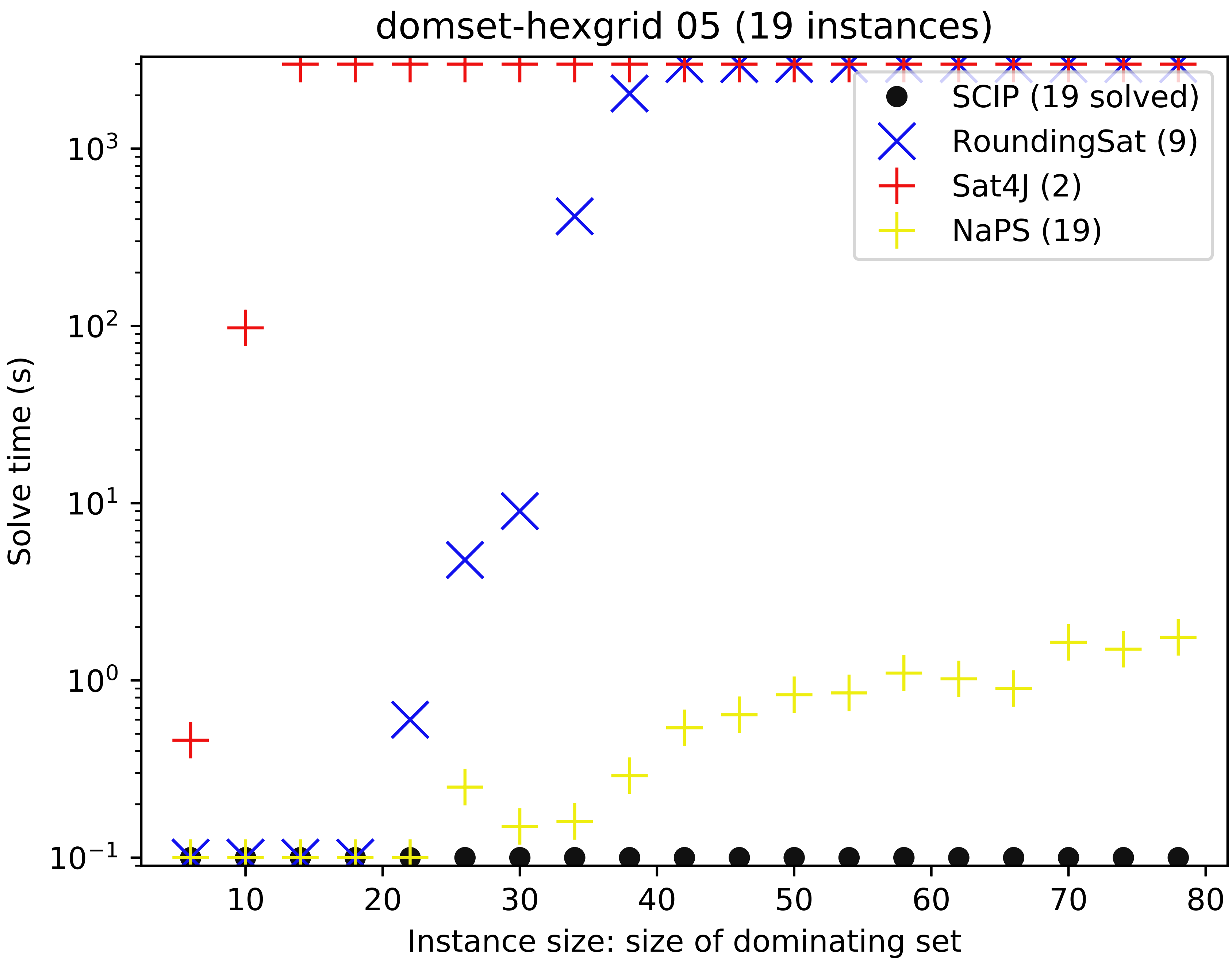
Experiments!

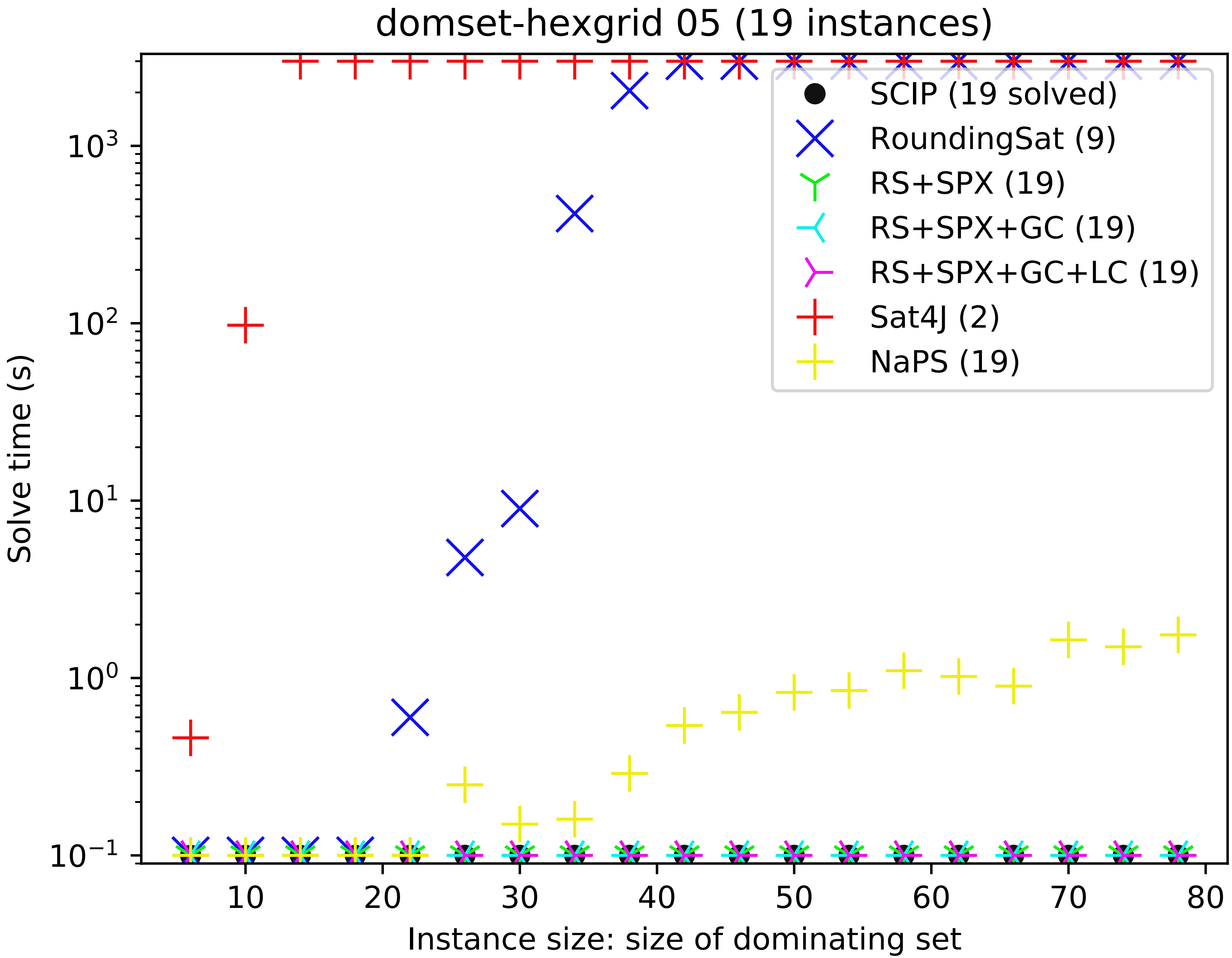
Compare state-of-the-art

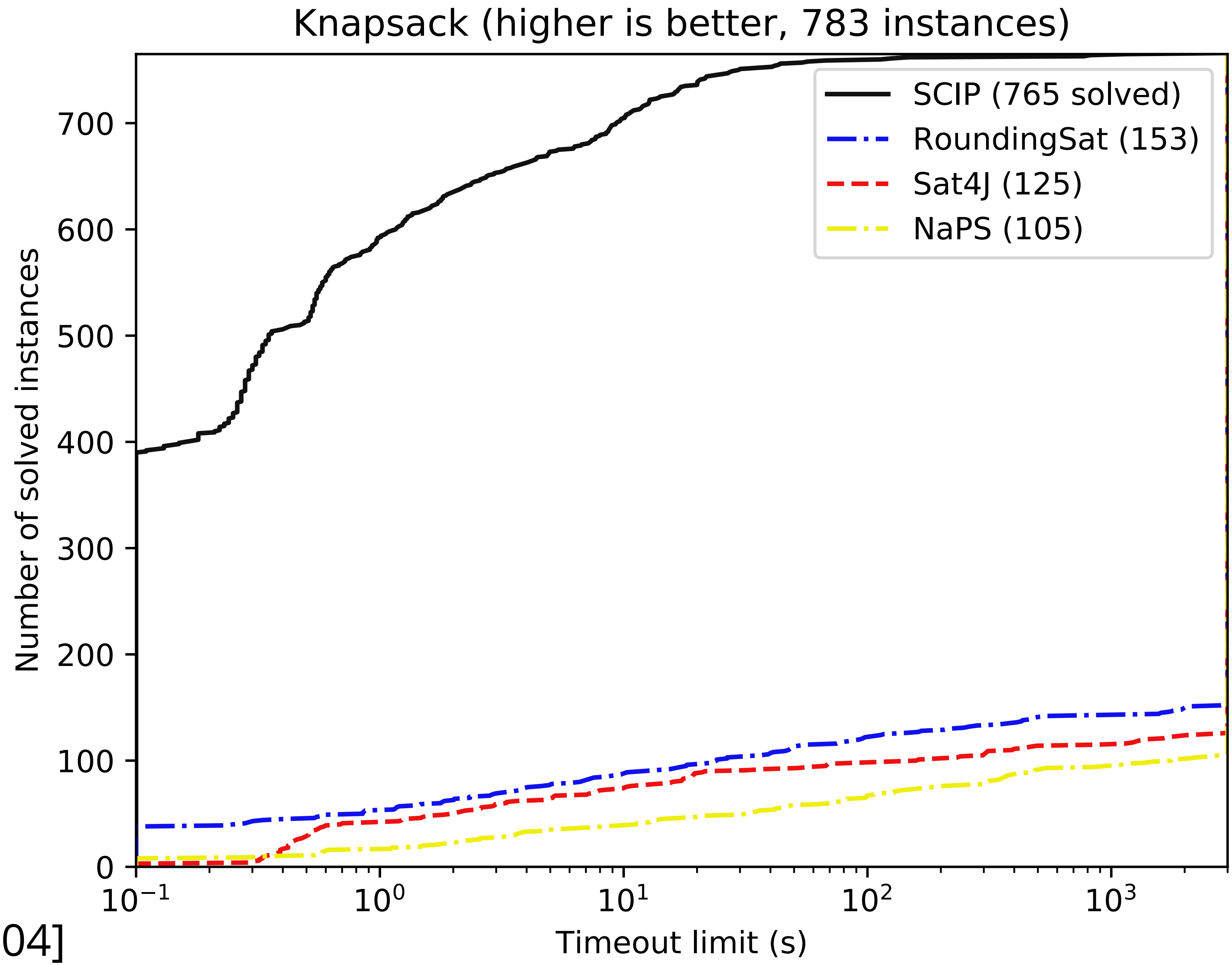
- RoundingSat
- Sat4J
- NaPS
- SCIP
- Gurobi
- CPLEX

to implementations

- RS+SPX
- RS+SPX+GC
- RS+SPX+GC+LC

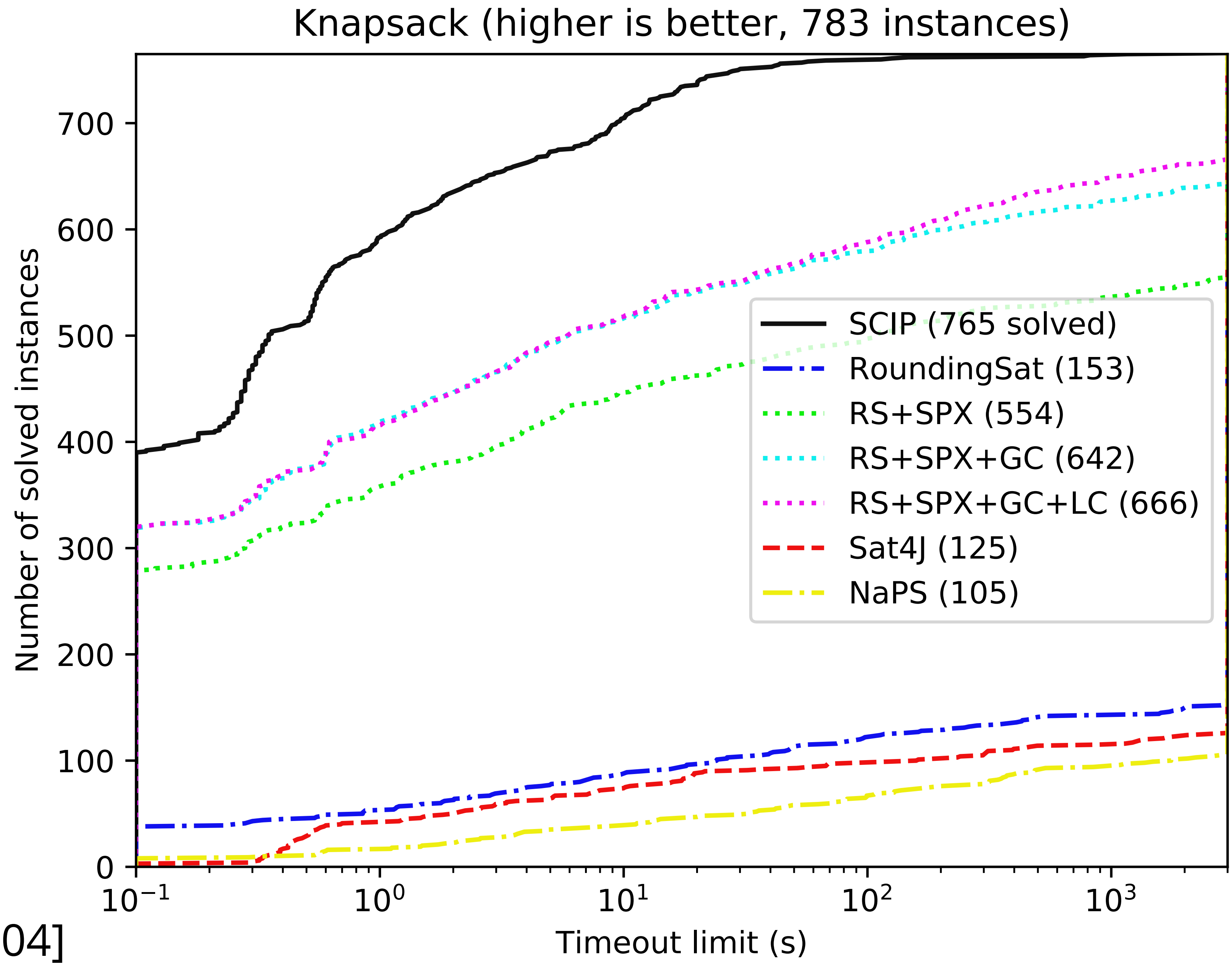

domset-hexgrid 05 (19 instances)
SCIP (19 solved)
RoundingSat (9)
Sat4J (2)
NaPS (19)
Solve time (s)
10^3
10^2
10^1
10^0
10^{-1}
10
20
30
40
50
60
70
80
Instance size: size of dominating set


domset-hexgrid 05 (19 instances)
SCIP (19 solved)
RoundingSat (9)
RS+SPX (19)
RS+SPX+GC (19)
RS+SPX+GC+LC (19)
Sat4J (2)
NaPS (19)
Solve time (s)
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10
20
30
40
50
60
70
80
Instance size: size of dominating set

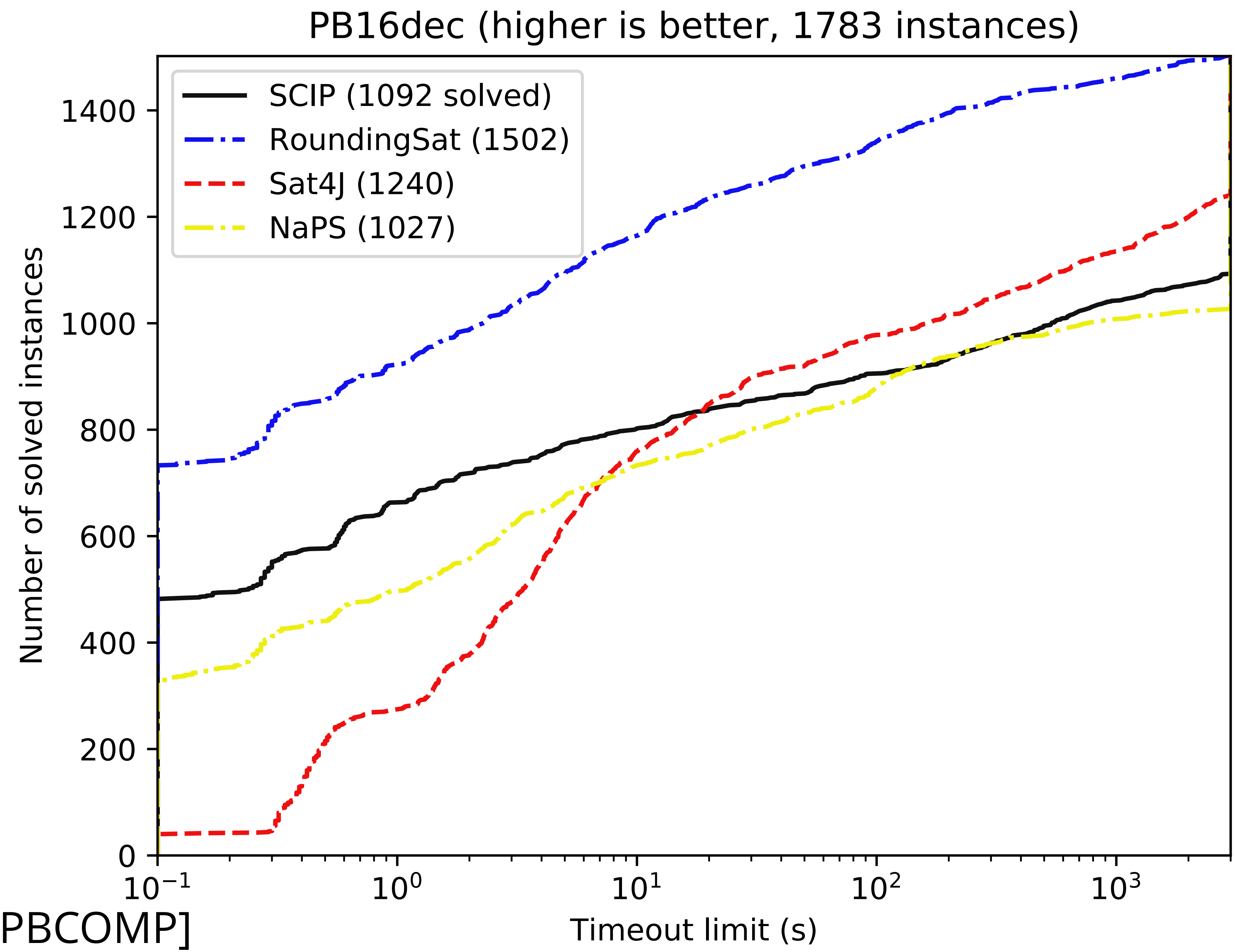


Knapsack (higher is better, 783 instances)
SCIP (765 solved)
RoundingSat (153)
Sat4J (125)
NaPS (105)
Number of solved instances
0
100
200
300
400
500
600
700
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Timeout limit (s)

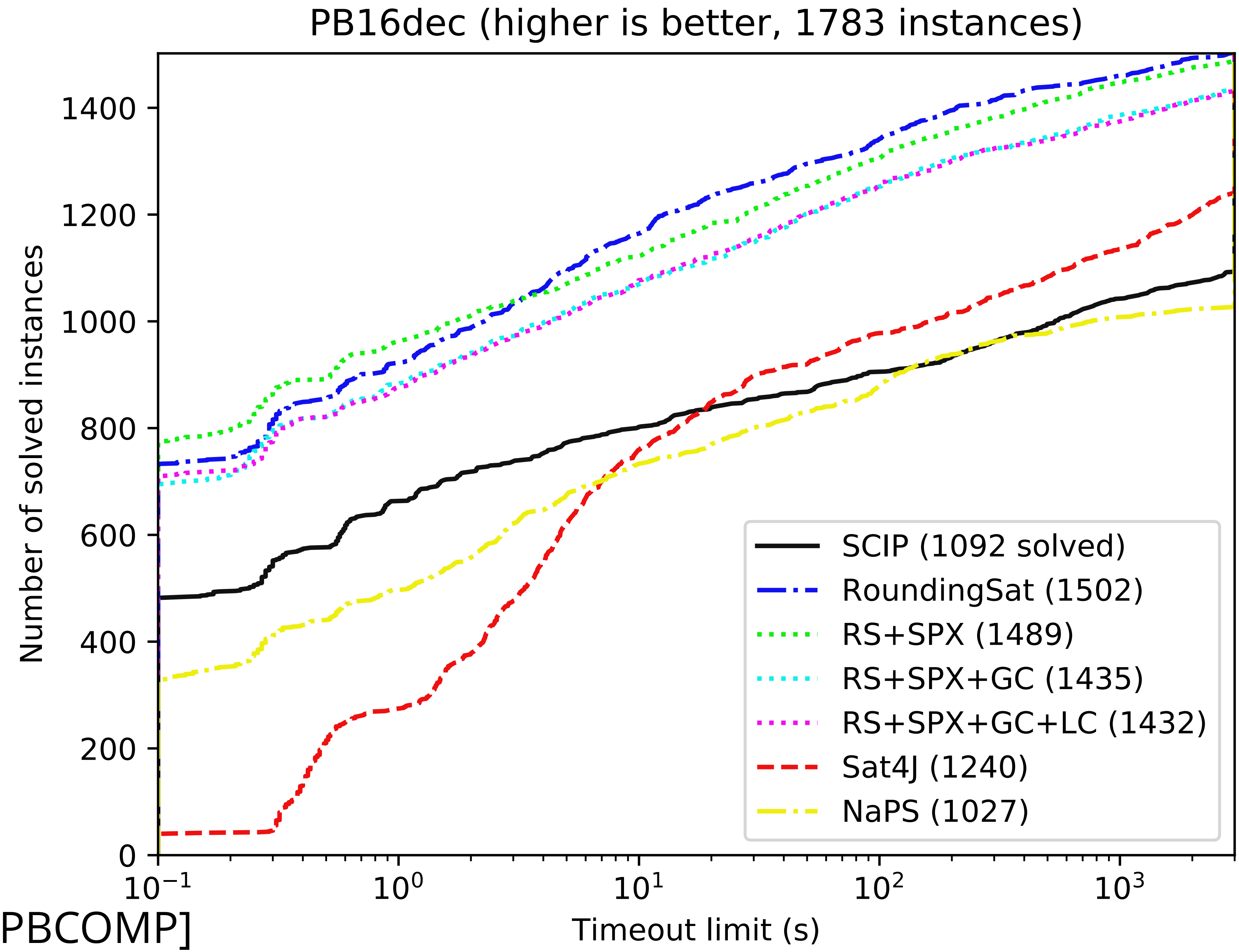
[P04]


Knapsack (higher is better, 783 instances)
Number of solved instances
Timeout limit (s)
0
100
200
300
400
500
600
700
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
SCIP (765 solved)
RoundingSat (153)
RS+SPX (554)
RS+SPX+GC (642)
RS+SPX+GC+LC (666)
Sat4J (125)
NaPS (105)

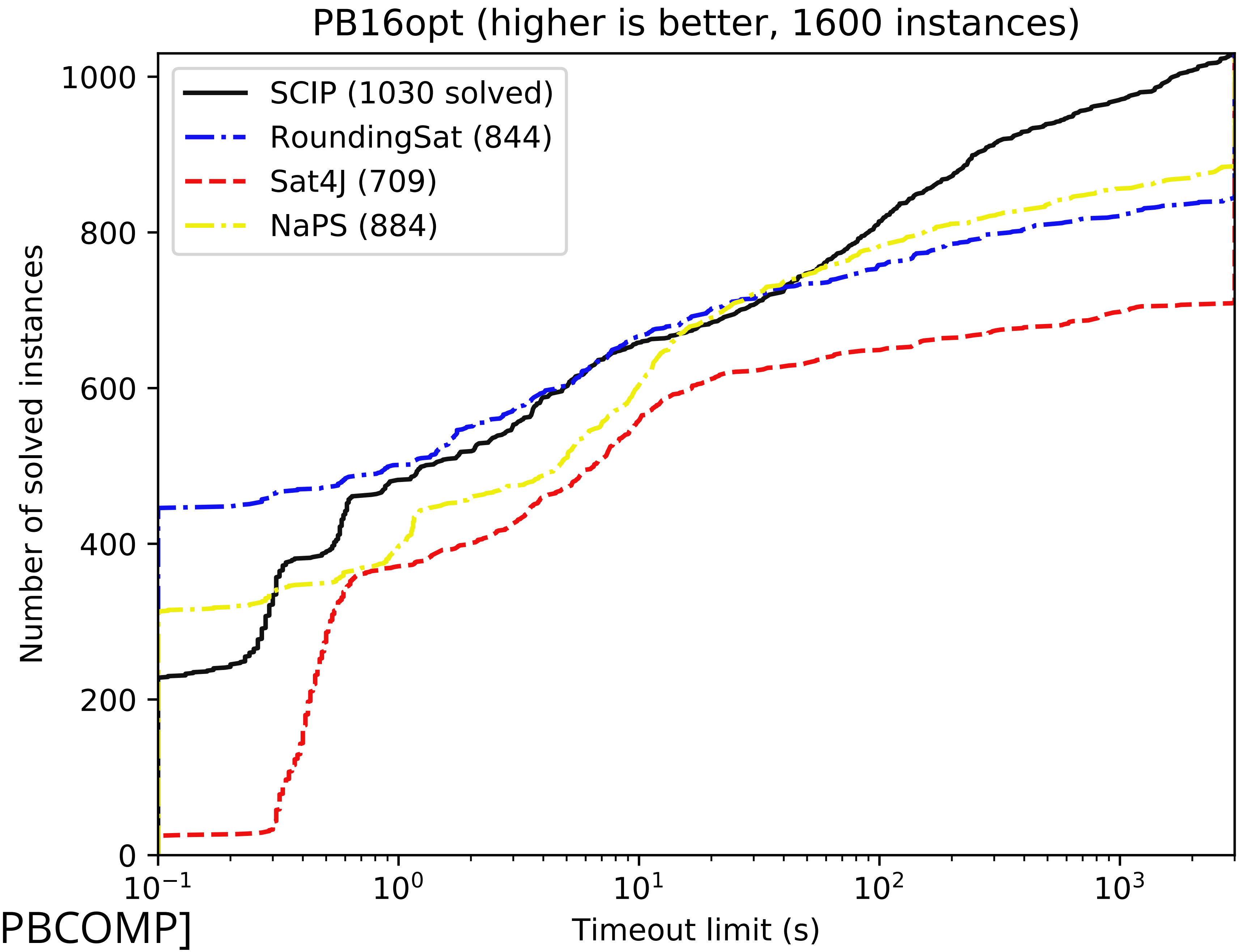
[P04]


PB16dec (higher is better, 1783 instances)
SCIP (1092 solved)
RoundingSat (1502)
Sat4J (1240)
NaPS (1027)
Number of solved instances
0
200
400
600
800
1000
1200
1400
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Timeout limit (s)

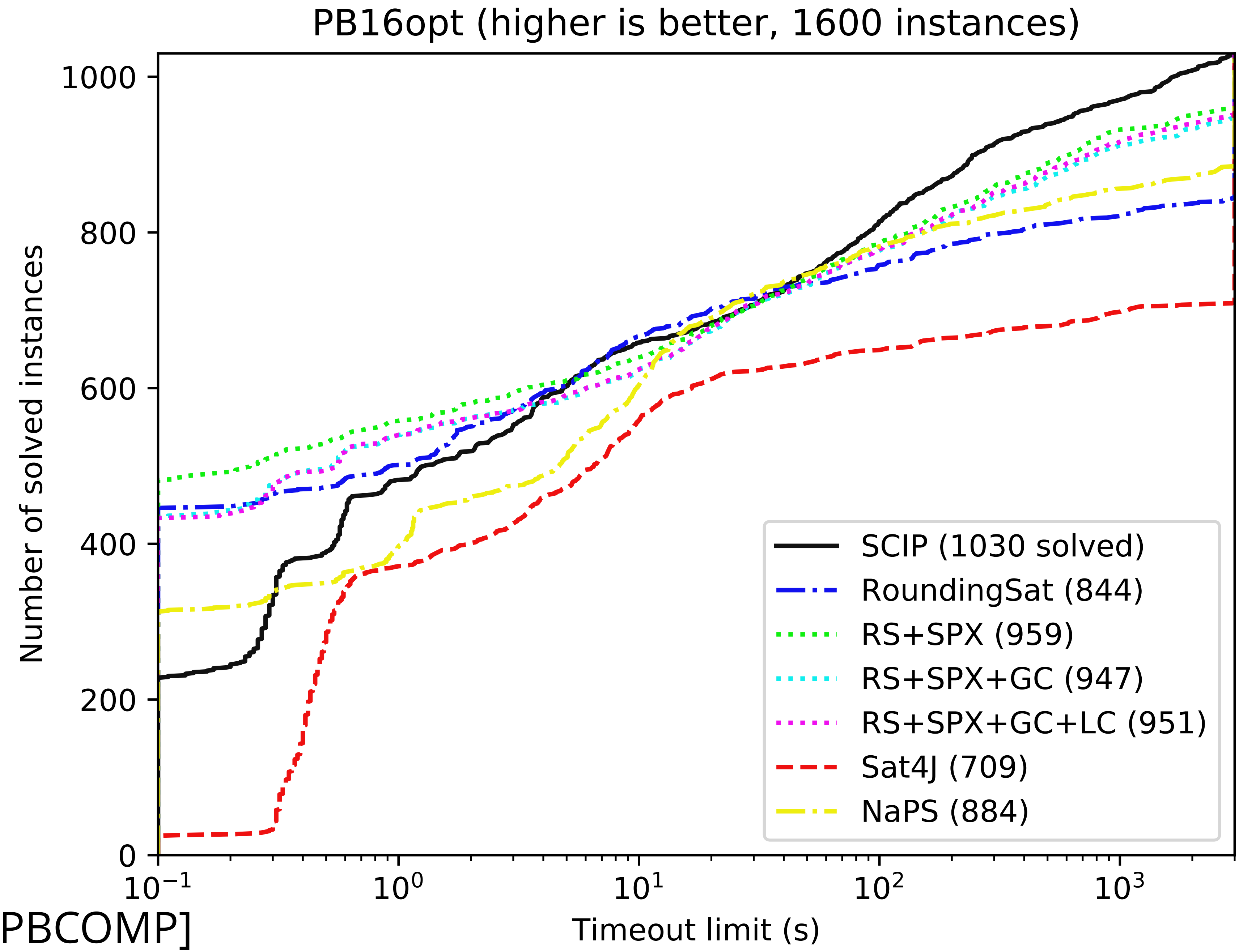
[PBCOMP]


PB16dec (higher is better, 1783 instances)
Number of solved instances
Timeout limit (s)
0
200
400
600
800
1000
1200
1400
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
SCIP (1092 solved)
RoundingSat (1502)
RS+SPX (1489)
RS+SPX+GC (1435)
RS+SPX+GC+LC (1432)
Sat4J (1240)
NaPS (1027)

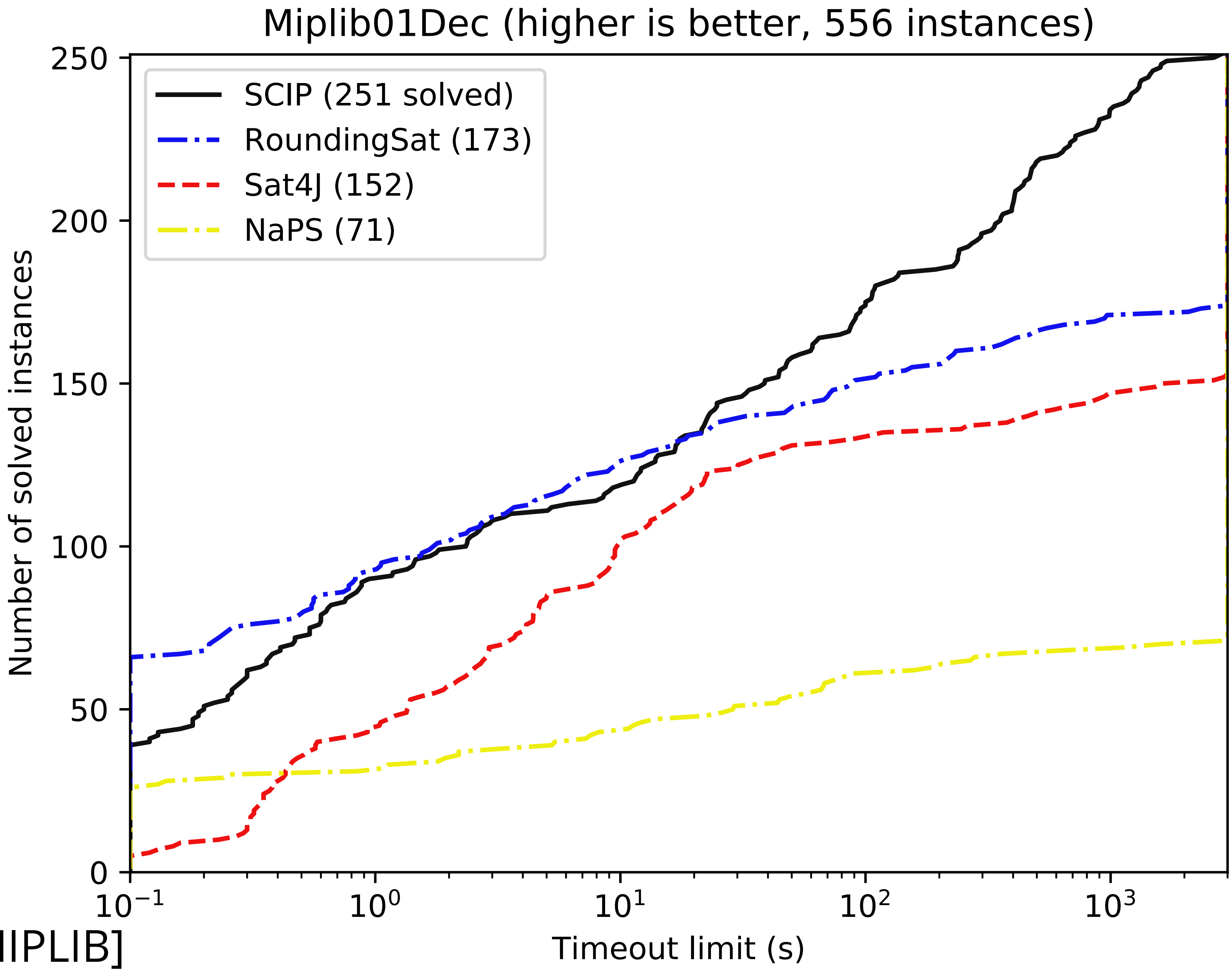
[PBCOMP]


PB16opt (higher is better, 1600 instances)
SCIP (1030 solved)
RoundingSat (844)
Sat4J (709)
NaPS (884)
Number of solved instances
0
200
400
600
800
1000
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Timeout limit (s)

[PBCOMP]

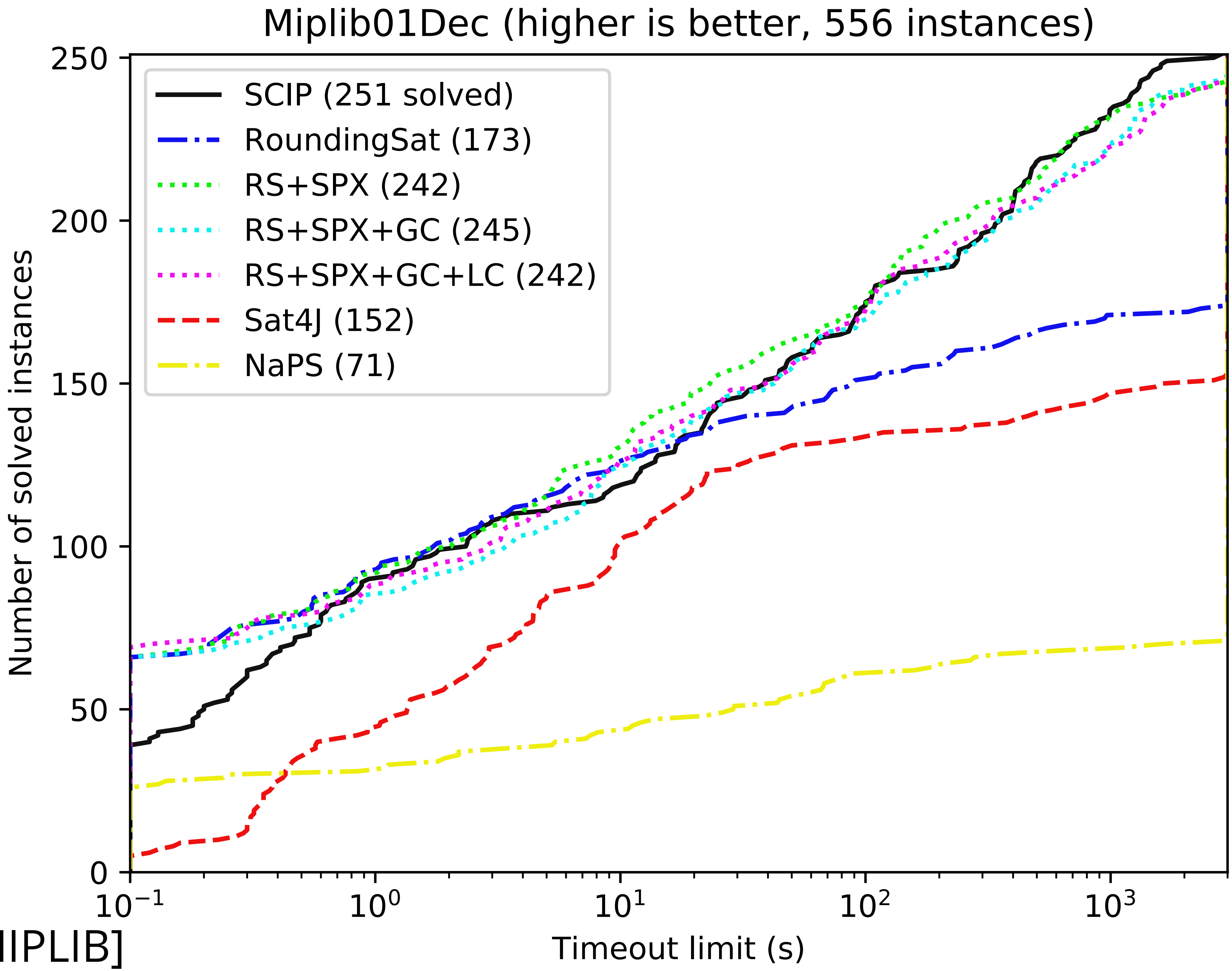

PB16opt (higher is better, 1600 instances)
Number of solved instances
Timeout limit (s)
0
200
400
600
800
1000
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
SCIP (1030 solved)
RoundingSat (844)
RS+SPX (959)
RS+SPX+GC (947)
RS+SPX+GC+LC (951)
Sat4J (709)
NaPS (884)

[PBCOMP]



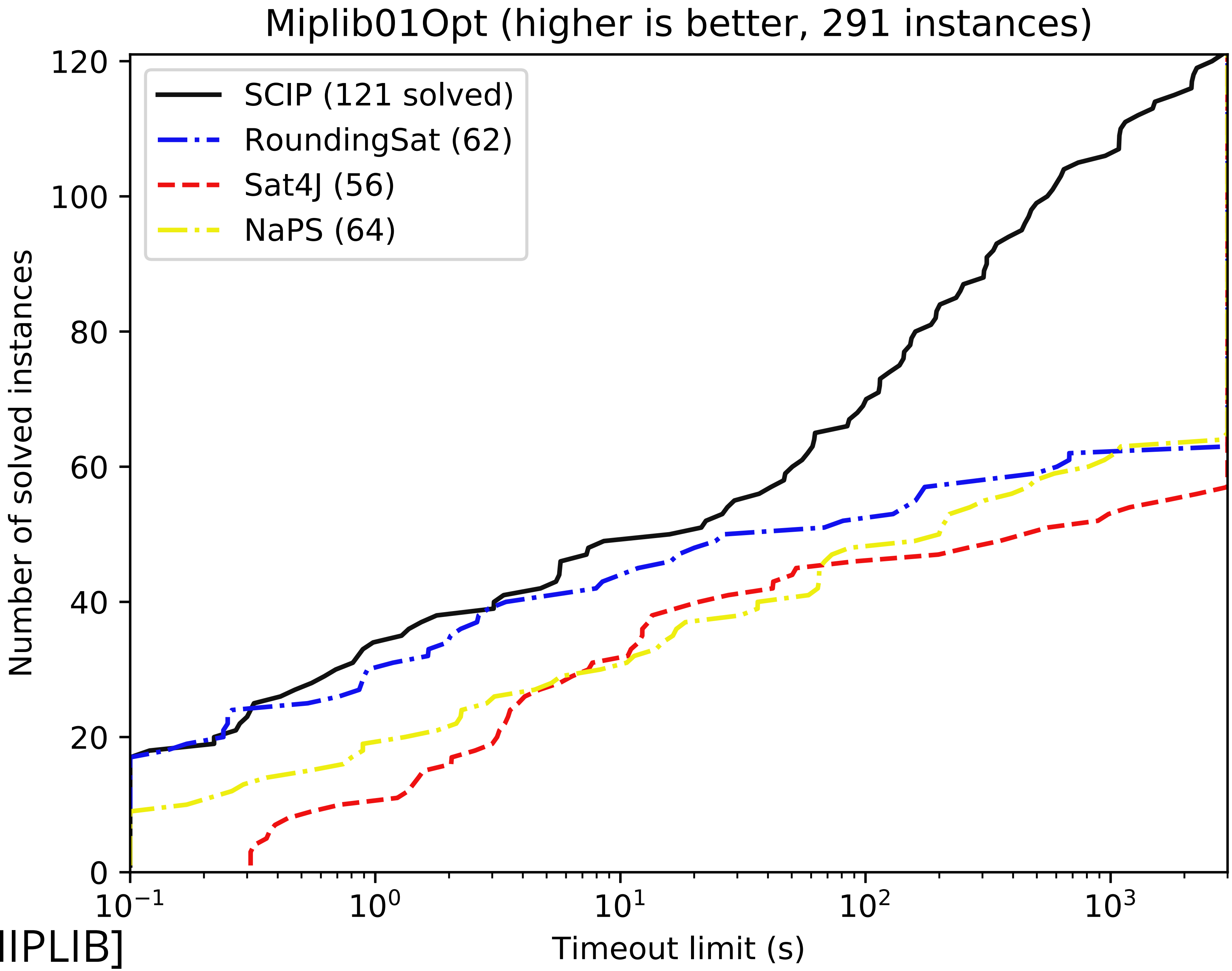
Miplib01Dec (higher is better, 556 instances)
SCIP (251 solved)
RoundingSat (173)
Sat4J (152)
NaPS (71)
Number of solved instances
Timeout limit (s)
0
50
100
150
200
250
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}

[MIPLIB]



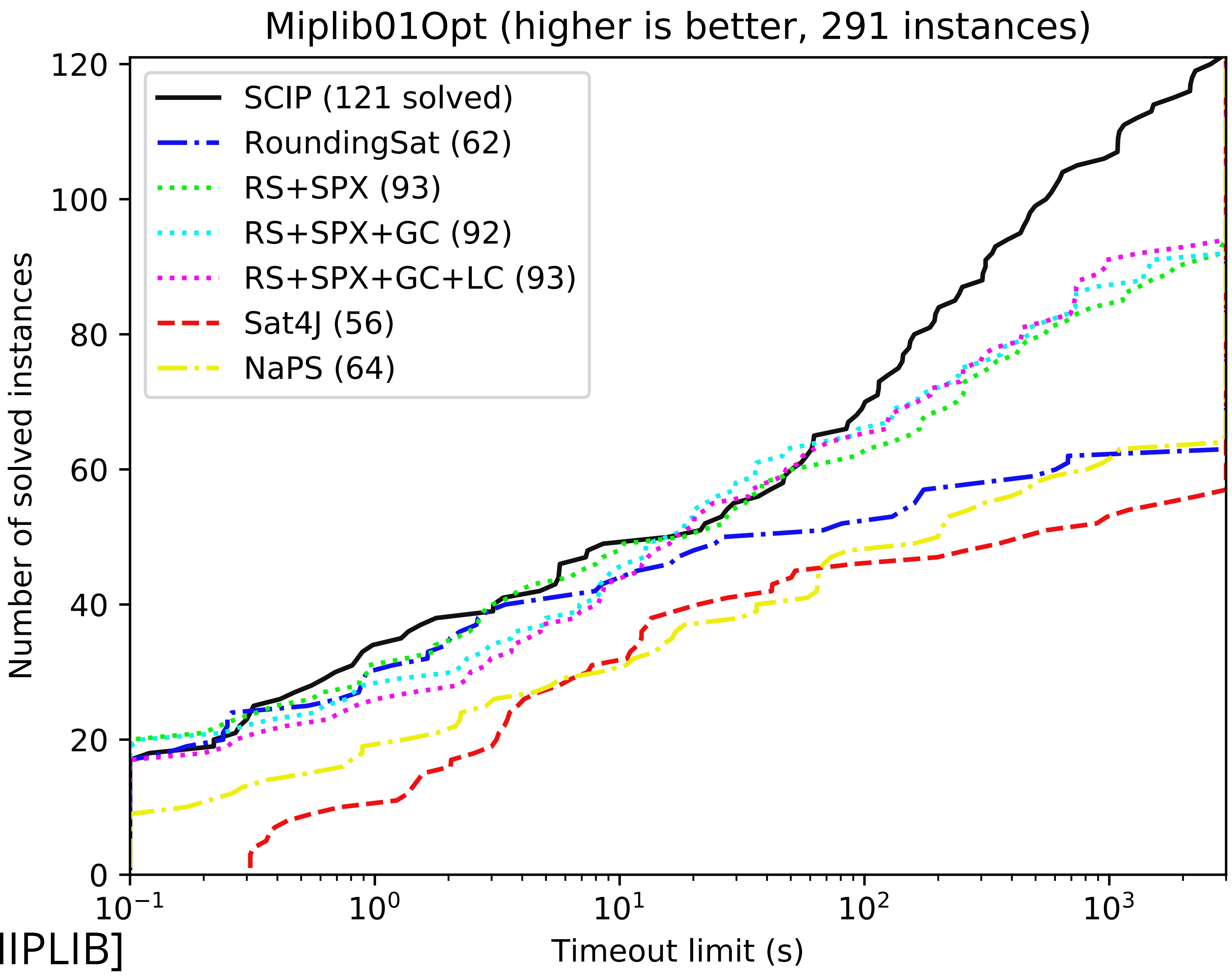
Miplib01Dec (higher is better, 556 instances)
SCIP (251 solved)
RoundingSat (173)
RS+SPX (242)
RS+SPX+GC (245)
RS+SPX+GC+LC (242)
Sat4J (152)
NaPS (71)
Number of solved instances
Timeout limit (s)
0
50
100
150
200
250
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}

[MIPLIB]

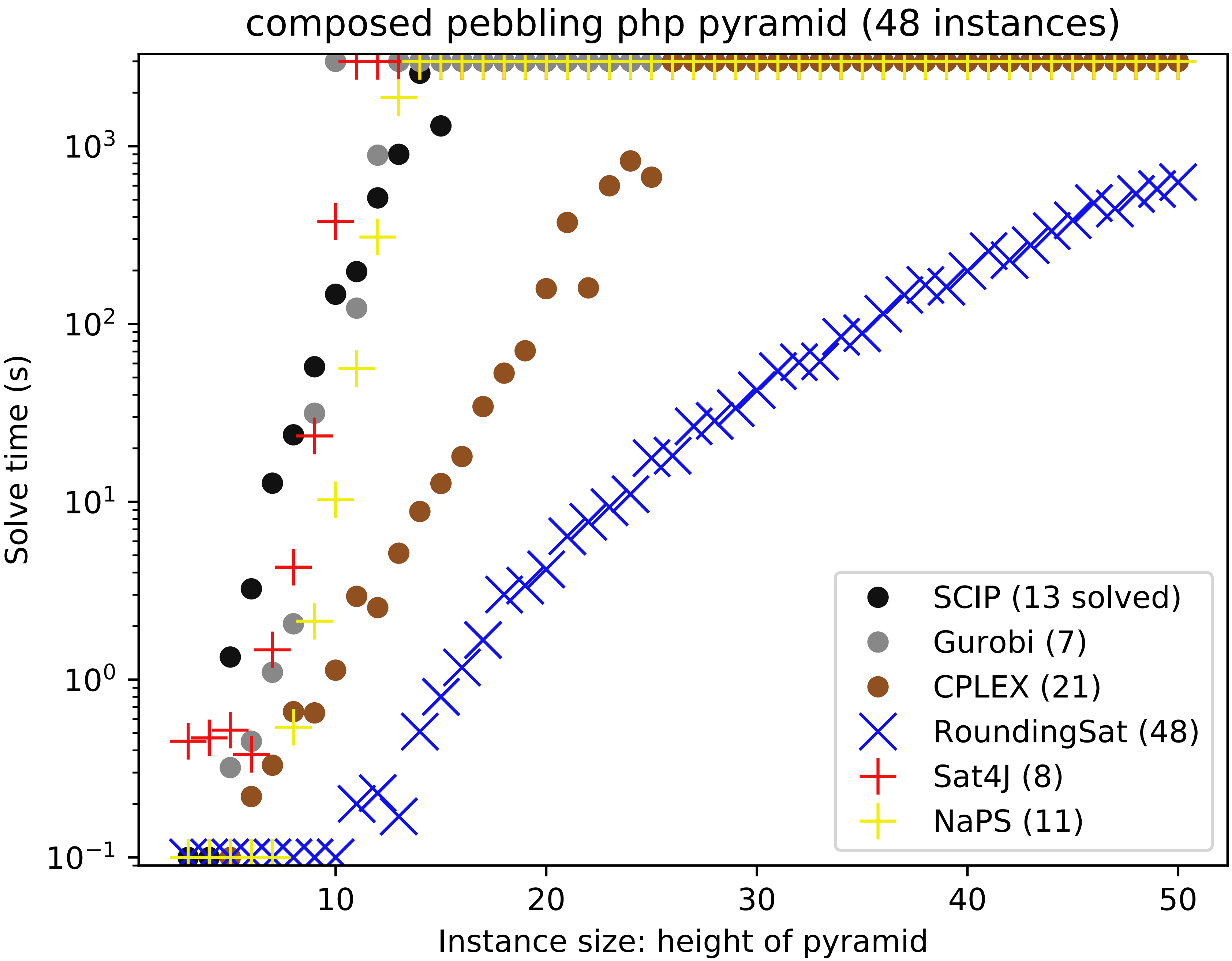


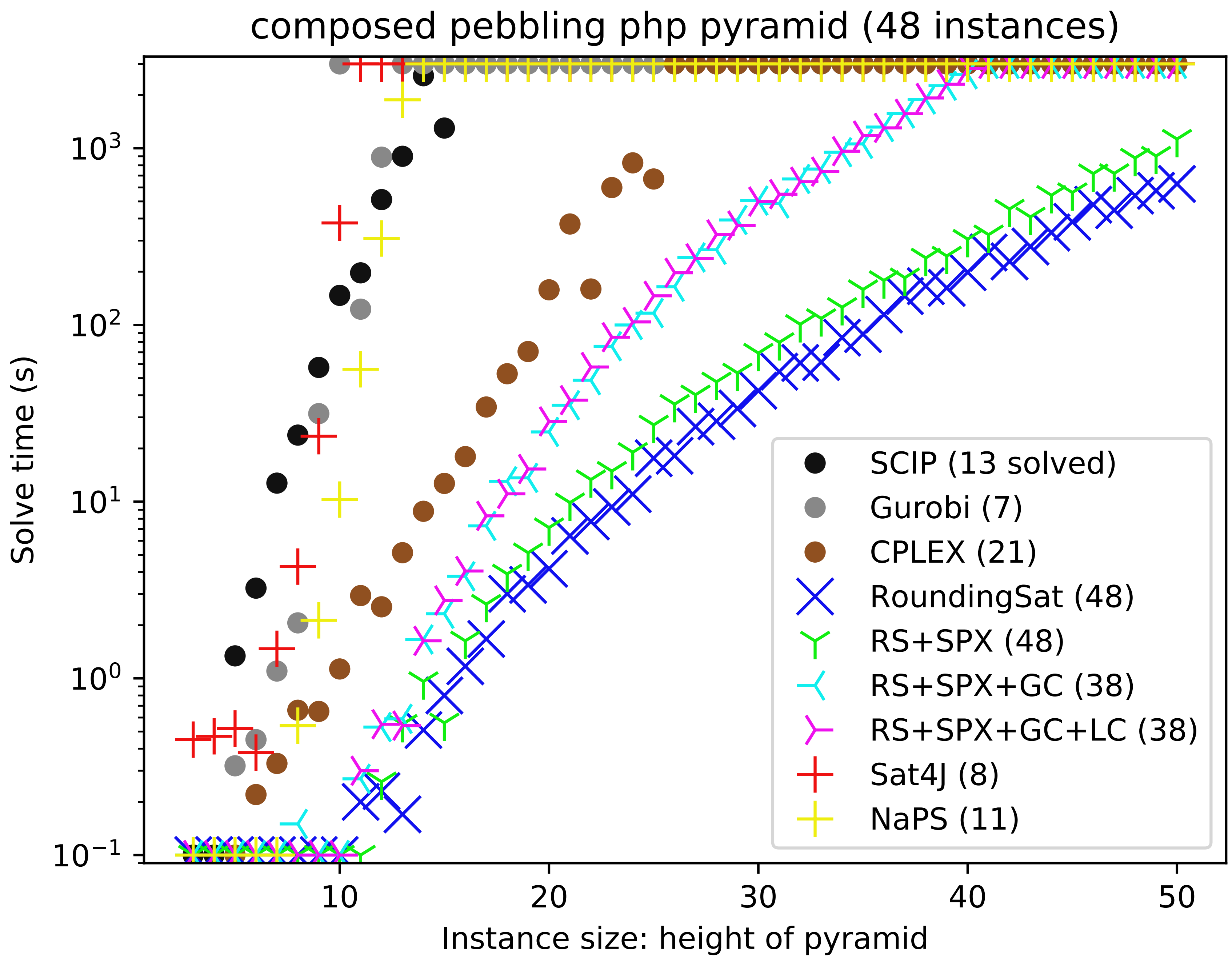
Miplib01Opt (higher is better, 291 instances)
SCIP (121 solved)
RoundingSat (62)
Sat4J (56)
NaPS (64)
Number of solved instances
Timeout limit (s)
0
20
40
60
80
100
120
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}

[MIPLIB]


Miplib01Opt (higher is better, 291 instances)
SCIP (121 solved)
RoundingSat (62)
RS+SPX (93)
RS+SPX+GC (92)
RS+SPX+GC+LC (93)
Sat4J (56)
NaPS (64)
Number of solved instances
Timeout limit (s)
0
20
40
60
80
100
120
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}

[MIPLIB]


composed pebbling php pyramid (48 instances)
Solve time (s)
10^3
10^2
10^1
10^0
10^{-1}
10
20
30
40
50
Instance size: height of pyramid
SCIP (13 solved)
Gurobi (7)
CPLEX (21)
RoundingSat (48)
Sat4J (8)
NaPS (11)



composed pebbling php pyramid (48 instances)
Solve time (s)
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10
20
30
40
50
Instance size: height of pyramid
SCIP (13 solved)
Gurobi (7)
CPLEX (21)
RoundingSat (48)
RS+SPX (48)
RS+SPX+GC (38)
RS+SPX+GC+LC (38)
Sat4J (8)
NaPS (11)

Related work

- Some SAT-based solvers use LP solver to decide specialized constraints
  - **LCG-Glucose**: network-flow propagation
  - **SMT**: deciding linear theory

Related work

- Some SAT-based solvers use LP solver to decide specialized constraints
  - **LCG-Glucose**: network-flow propagation
  - **SMT**: deciding linear theory
- In the end, SAT-based solvers only learn **clauses**
  - **exponentially weaker** than learning 0-1 linear constraints
  - Farkas constraints, Chvátal-Gomory cuts, PB learned constraints: **all are used for conflict-driven learning**

Summary

Conflict-driven search for 0-1 ILPs

Summary

Conflict-driven search for 0-1 ILPs

- Generates cuts from search conflicts

Summary

Conflict-driven search for 0-1 ILPs

- Generates cuts from search conflicts
- Does not always find short refutations for rational infeasibility

Summary

Conflict-driven search for 0-1 ILPs

- Generates cuts from search conflicts
- Does not always find short refutations for rational infeasibility
- Allows sound & efficient integration with LP solver

Summary

Conflict-driven search for 0-1 ILPs

- Generates cuts from search conflicts
- Does not always find short refutations for rational infeasibility
- Allows sound & efficient integration with LP solver
- Is further improved by value heuristic & Gomory cut generation

Summary

Conflict-driven search for 0-1 ILPs

- Generates cuts from search conflicts
- Does not always find short refutations for rational infeasibility
- Allows sound & efficient integration with LP solver
- Is further improved by value heuristic & Gomory cut generation

Experiments approach best of both worlds

Summary

Conflict-driven search for 0-1 ILPs

- Generates cuts from search conflicts
- Does not always find short refutations for rational infeasibility
- Allows sound & efficient integration with LP solver
- Is further improved by value heuristic & Gomory cut generation

Experiments approach best of both worlds

Thanks for your attention!

References

[MS96] GRASP - a new search algorithm for satisfiability - Marques-Silva, Sakallah

[BS97] Using CSP lookback techniques to solve real-world SAT instances - Bayardo, Schrag

[MMZZM01] Chaff: Engineering an efficient SAT solver - Moskewicz, Madigan, Zhao, Zhang, Malik

[P04] Where are the hard knapsack problems? - Pisinger

[CK05] A fast pseudo-Boolean constraint solver - Chai, Kuehlmann

[SS06] Pueblo: A hybrid pseudo-Boolean SAT solver - Sheini, Sakallah

[LP10] The Sat4j library, release 2.2 - Le Berre, Parrain

[EN18] Divide and conquer: Towards faster pseudo-boolean solving - Elffers, Nordström

[EGNV18] Using Combinatorial Benchmarks to Probe the Reasoning Power of pseudo-Boolean Solvers - Elffers, Giráldez-Cru, Nordström, Vinyals

[ZIB] SoPlex - soplex.zib.de

[PBCOMP] Latest PB competition - www.cril.univ-artois.fr/PB16/

[MIPLIB] The Mixed Integer Programming Library - miplib.zib.de