

# *SYMPHONY*

*an MILP solver*

Ashutosh Mahajan<sup>1</sup>, Menal Guzelsoy<sup>1</sup>, Ted Ralphs<sup>1</sup>

<sup>1</sup>Cor@l Lab

Department of Industrial and Systems Engineering  
Lehigh University

Informs 2008



## *What is SYMPHONY*

- ▶ SYMPHONY is an open-source solver for Mixed Integer Linear Programs (MILPs).
- ▶ Source code and binaries available at COIN (<http://www.coin-or.org>). Current version 5.1.10
- ▶ Compiles on Linux, Unix or Windows
- ▶ Uses OpenMP for shared memory parallelization
- ▶ PVM for distributed memory
- ▶ Includes solvers for specific problems: TSP, Multicriteria MILP, Matching Problem, VRP, Network Routing, Mixed Postman Problem, ...

## Features

- ▶ Implements LP based branch and cut (and price) algorithm at its core.
- ▶ Can use a variety of LP solvers through Osi (Clp, CPLEX, XPRESS).
- ▶ Valid inequalities are generated using Cut Generation Library (CGL).
- ▶ User can write a callback for LP, cuts, heuristics, branching etc.
- ▶ Many user controllable parameters to modify branch and cut procedures.
- ▶ The MILP solver is available as a shared library. Can be called from other programs.
- ▶ Capability to solve an instance after it has been modified.
- ▶ Capability to do *sensitivity analysis*.

### *In this talk*

Focus on SYMPHONY's core: a black box MILP solver on a single processor.

## *SYMPHONY's MIP solver*

Other *free* Solvers:

- ▶ Gnu Linear Programming Kit (GLPK, [www.gnu.org/software/glpk/](http://www.gnu.org/software/glpk/)).
- ▶ Coin Branch and Cut (Cbc, [www.coin-or.org](http://www.coin-or.org)).
- ▶ Solving Constraint Integer Programs (SCIP, <http://scip.zib.de>).
- ▶ lpsolve (<http://sourceforge.net/projects/lpsolve>).
- ▶ Minto (<http://coral.ie.lehigh.edu/~minto>)

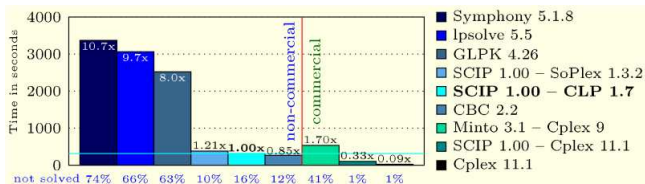
...and many any other non-free solvers.

### *SYMPHONY's versions*

Last release: version 5.1.10

Current stable: 5.2 (Available, To be released soon)

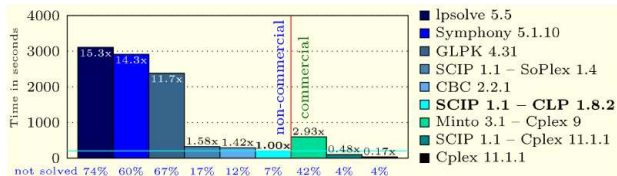
## Story so far: Version 5.1



September, 2008, Source: <http://scip.zib.de/>, based on tests by Hans Mittelmann



End September, Source: <http://scip.zib.de/>, based on tests by Hans Mittelmann



October 05, Source: <http://scip.zib.de/>, based on tests by Hans Mittelmann

## *New improvements in next release*

Available in version 5.2 (stable)

- ▶ Valid Inequalities
  - ▶ Check for duplicates
  - ▶ Check for quality, length
  - ▶ Time and node based strategy
- ▶ Branching rules
  - ▶ Reliability branching
  - ▶ Learning from strong branching
- ▶ Primal Heuristics: Feasibility pump
- ▶ Preprocessing
- ▶ Other minor improvements
  - ▶ Saving more sets of reduced costs at root node
  - ▶ Fine tuning diving strategy
  - ▶ Several code optimizations

## *Valid Inequalities: Duplicacy*

- ▶ Many cut generators used: Probing, Knapsack, Clique, Gomory, Two-MIR, Flow and cover.
- ▶ Cut generators (CGL) external to SYMPHONY
- ▶ Many cuts repeated upto 2-3 times even when generated from the same generator.
- ▶ High number of cuts checked or added at each iteration.

Instance	Total Cuts	Duplicates
10teams	12237	4844
air03	12680	5562
acc1	19601	3102
seymour	1203	361

Overall: 8% of cuts generated in at most 100 rounds of cuts at root node

- ▶ Generate hash using a random vector in  $[0, 1]^n$ .

## *Valid Inequalities: Quality*

$$\dots - 0.315939x_{268} - 0.000665x_{235} - 0.000665x_{317} - \\ 0.000000x_{109} - 0.000002x_{110} + 0.000006x_{113} - 0.000102x_{310} - \\ 0.000275x_{307} - 0.000275x_{315} \dots$$

- ▶ LP error tolerance is  $1e - 7$ .
- ▶ Gomory, MIR, two-MIR, Flow and Cover Cuts may have poor coefficients.
- ▶ We throw out cuts that have coefficients  $< 1e - 5$ .
- ▶ Or if ratio of Max/Min coefficients is  $> 1e4$ .
- ▶ Also throw out cuts that have too many coefficients.

Overall: 22% of cuts generated in at most 100 rounds of cuts at root node

## Feasibility Pump

- ▶ Primal Heuristic (Fischetti, Glover, Lodi, 2005).
- ▶ Round the LP solution to the nearest integer.
- ▶ The rounded point is usually infeasible.
- ▶ Minimize the  $L_1$  norm of distance of this point from the LP polytope.
- ▶ Iterate.

### Performance

- ▶ Found a feasible solution for 93/176 instances.
- ▶ Solution value is usually not close to optimal.

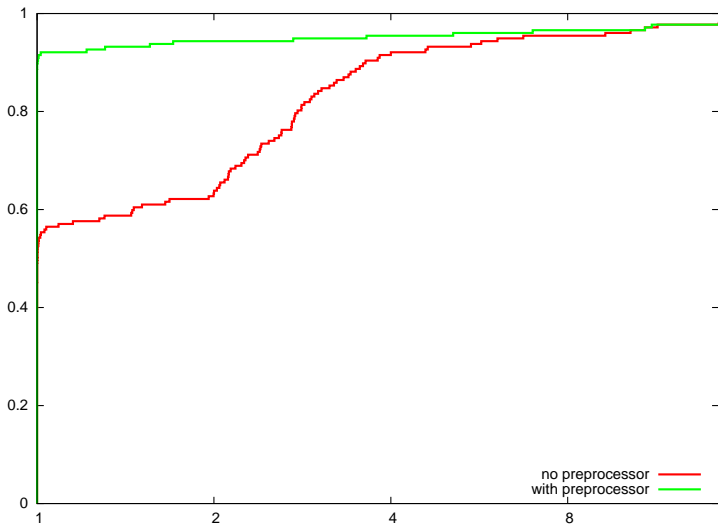
Instance	version 5.1.10	version 5.2
acc0	7200	24.20
acc1	7200	70.94
disctom	7200	50.65

Time (seconds) taken to solve the instance.

## *Preprocessing*

- ▶ Basic preprocessing (Savelsbergh, 1994).
- ▶ Tightening variable bounds.
- ▶ Tightening right hand side.
- ▶ Improving coefficients in constraint matrix.
- ▶ Detects infeasibility.
- ▶ Collects information like step size in objective function, types of variables, constraints etc.
- ▶ Only done in root node.

# Preprocessing



Effect of Basic Preprocessing. 177 instances, time limit 7200s

## Reliability Branching

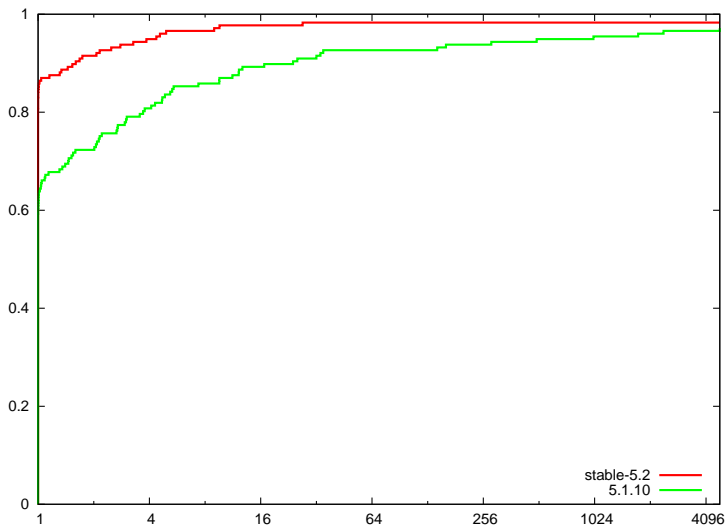
- ▶ SYMPHONY's default branching strategy uses strong branching.
- ▶ Good for increasing the lower bound but consumes large time.
- ▶ Version 5.2 has reliability branching (Achterberg, 2005).
- ▶ Do strong branching on a particular variable only a few ( $\eta$ ) times and then use the *score* in any further branching decision.

Instance	strong br	rel br
10teams	0.59	0.2
30_05_100	0.92	2.9
A2C1S1	2.21	0.05
UMTS	3.44	0.21
arki001	3.73	0.02

Ratio of time spent in Branching and to that in LP

Overall: 2.51 in strong branching, 1.51 in reliability branching

## Overall Progress



Performance of 5.1.10 against 5.2. 177 instances, time limit 7200s

Instances solved with version 5.1.10: 49, with 5.2: **75**

## *Future Work*

- ▶ Many more heuristics.
- ▶ Branching rules for problems where lower bound does not change with branching.
- ▶ Probing, implications and conflict graphs

## *Future Work*

- ▶ Many more heuristics.
- ▶ Branching rules for problems where lower bound does not change with branching.
- ▶ Probing, implications and conflict graphs

## **Conclusion**

- ▶ Version 5.2 will have significant improvements over 5.1.10
- ▶ It is stable and available from COIN-OR
- ▶ Not yet released as an official release. But will be soon.

Please try it!