Benders' decomposition: Fundamentals and implementations

Stephen J. Maher

University of Exeter,

@sj_maher

s.j.maher@exeter.ac.uk

24th September 2020

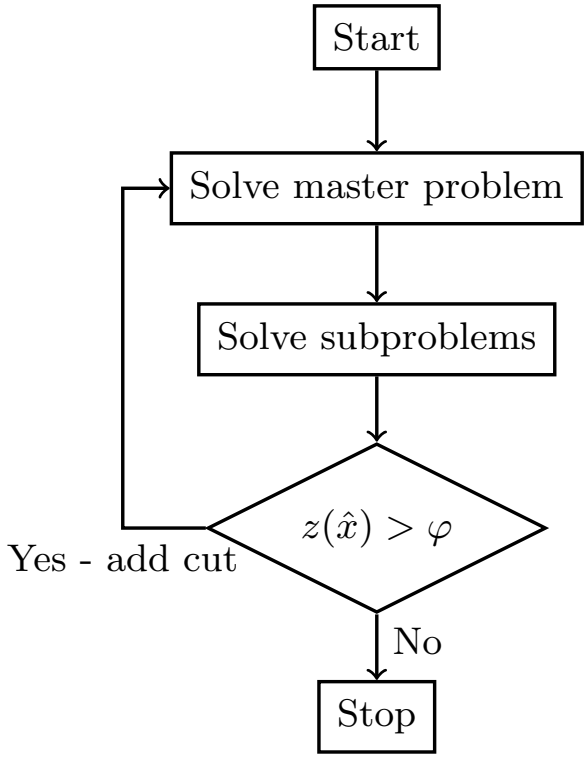
Part 2

Implementation

Resources

- P. Rubin. Benders Decomposition Then and Now. https://orinanobworld.blogspot.com/2011/10/benders-decomposition-then-and-now.html
- S. J. Maher. Implementing the branch-and-cut approach for a general purpose Benders' decomposition framework. http://www.optimization-online.org/DB_HTML/2019/09/7384.html
- P. Bonami, D. Salvagnin, and A. Tramontani. Implementing Automatic Benders Decomposition in a Modern MIP Solver http://www.optimization-online.org/DB_HTML/2019/12/7506.html

Standard Benders' implementation

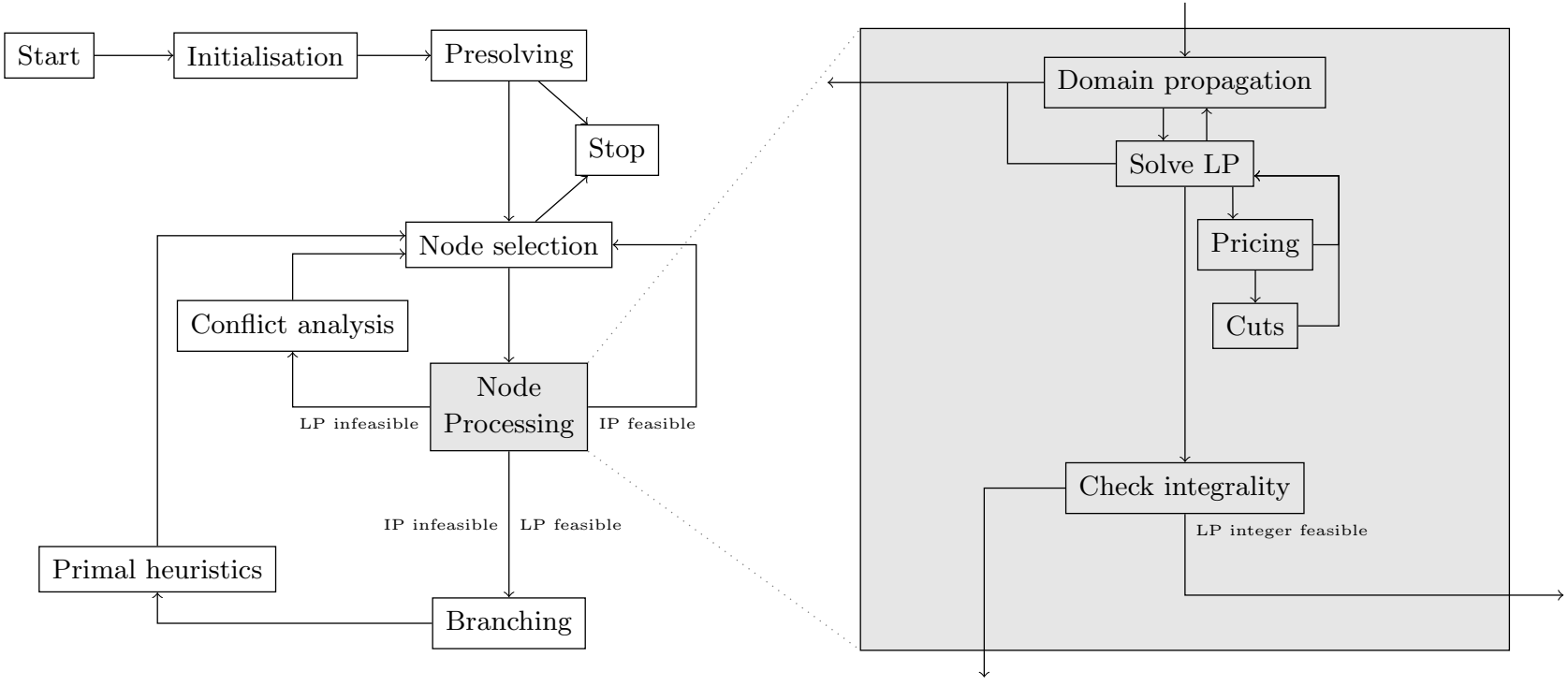


- Easy to understand and simple to implement.
- Not always effective, large overhead in repeatedly solving master problem.

Branch-and-cut

- Modern solvers pass through a number of different stages during node processing.
- Some of these stages can be used to generate Benders' cuts.
- By interrupting node processing, Benders' cuts are generated during the tree search.

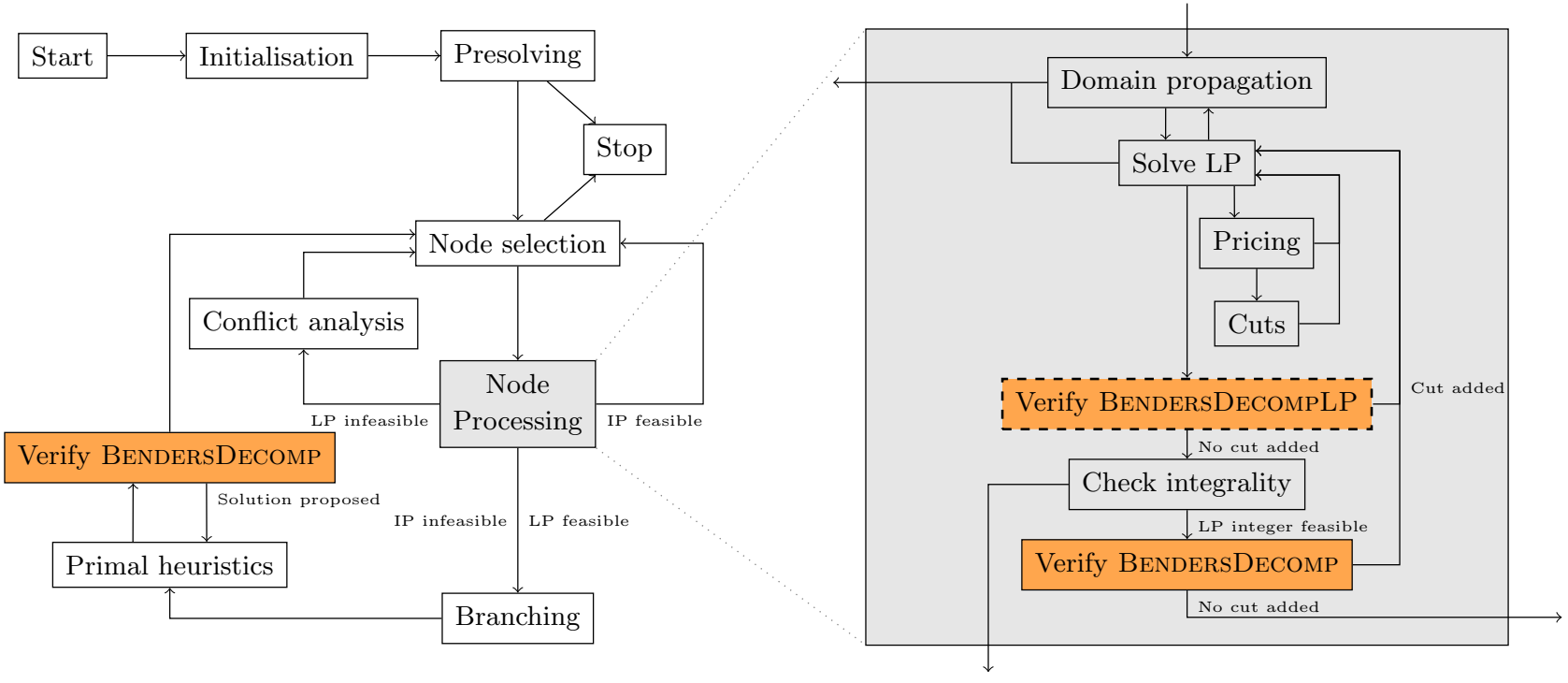
Solving process



Branch-and-cut

- Modern solvers pass through a number of different stages during node processing.
- Some of these stages can be used to generate Benders' cuts.
- By interrupting node processing, Benders' cuts are generated during the tree search.

Cut generation - Branch-and-cut



Key implementation details

- Constraint handlers are fundamental for the implementation of Benders' decomposition in SCIP. Constraint handlers in SCIP are: `BendersDecomp` and `BendersDecompLP`.
  - These provide callbacks to verify solutions during *node processing* and found by *primal heuristics*
- General framework requires a solve and cut loop to solve the subproblems and generate Benders' cuts. This loop is required for both constraint handlers.
- Flexibility in the solve and cut loop is necessary for the implementation of enhancement techniques.

Key implementation details

- Constraint handlers are fundamental for the implementation of Benders' decomposition in SCIP. Constraint handlers in SCIP are: `BendersDecomp` and `BendersDecompLP`.
  - These provide callbacks to verify solutions during *node processing* and found by *primal heuristics*
- General framework requires a solve and cut loop to solve the subproblems and generate Benders' cuts. This loop is required for both constraint handlers.
- Flexibility in the solve and cut loop is necessary for the implementation of enhancement techniques.

For more details please look at

- S. J. Maher. Implementing the branch-and-cut approach for a general purpose Benders' decomposition framework. `http://www.optimization-online.org/DB_HTML/2019/09/7384.html`

Enhancements for Benders' decomposition - Resources

- S. J. Maher. So you have decided to use Benders' decomposition. Be prepared for what comes next!!! `http://www.stephenjmaher.com/blog/blog-entry.php?blogfile=bendersDecomp`
- S. J. Maher. Benders' decomposition in practice. `http://www.stephenjmaher.com/blog/blog-entry.php?blogfile=rruflp`
- Santoso, T., Ahmed, S., Goetschalckx, M. and Shapiro, A. A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research, 2005, 167, 96-115.

Enhancements for Benders' decomposition

- Cut strengthening
- Cutting on all solutions
- Large Neighbourhood Benders' search
- Trust region heuristic
- Three-phase method
- Presolving – auxiliary variable bounds

Enhancements for Benders' decomposition

- **Cut strengthening**
- Cutting on all solutions
- Large Neighbourhood Benders' search
- Trust region heuristic
- **Three-phase method**
- Presolving – auxiliary variable bounds

Cut strengthening techniques - Resources

- Magnanti, T. and Wong, R. Accelerating Benders' decomposition: Algorithmic enhancement and model selection criteria. Operations Research, 1981, 29, 464-484
- Papadakos, N. Practical enhancements to the Magnanti-Wong method. Operations Research Letters, 2008, 36, 444-449
- Fischetti, M., Ljubić, I. and Sinnl, M. Redesigning Benders Decomposition for Large-Scale Facility Location. Management Science, 2017, 63, 2146-2162.

Cut strengthening

A simple in-out cutting methods described by Fischetti et al. (2017).

Cut strengthening

A simple in-out cutting methods described by Fischetti et al. (2017).

Given a corepoint x_o and the current LP solution x, the separation solution is given by

$$\hat{x} = \lambda x + (1 - \lambda) x_o.$$

Cut strengthening

A simple in-out cutting methods described by Fischetti et al. (2017).

Given a corepoint x_o and the current LP solution x, the separation solution is given by

$$\hat{x} = \lambda x + (1 - \lambda) x_o.$$

After k iterations without lower bound improvements

$$\hat{x} = x + \delta.$$

Cut strengthening

A simple in-out cutting methods described by Fischetti et al. (2017).

Given a corepoint x_o and the current LP solution x, the separation solution is given by

$$\hat{x} = \lambda x + (1 - \lambda) x_o.$$

After k iterations without lower bound improvements

$$\hat{x} = x + \delta.$$

After a further k iterations without lower bound improvement

$$\hat{x} = x.$$

Cut strengthening

A simple in-out cutting methods described by Fischetti et al. (2017).

Given a corepoint x_o and the current LP solution x, the separation solution is given by

$$\hat{x} = \lambda x + (1 - \lambda) x_o.$$

After k iterations without lower bound improvements

$$\hat{x} = x + \delta.$$

After a further k iterations without lower bound improvement

$$\hat{x} = x.$$

After each iteration the core point is updated by

$$x_o = \lambda x + (1 - \lambda) x_o.$$

Cut strengthening - initial core point

Five different options for initialising the core point

- First LP solution
- First primal solution
- Relative interior point
- Vector of all ones
- Vector of all zeros

- Reinitialise core point with each incumbent change

Three-phase method - Resources

- McDaniel, D. and Devine, M. A Modified Benders' Partitioning Algorithm for Mixed Integer Programming. Management Science, 1977, 24, 312-319.
- Laporte, G. and Louveaux, F. V. The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters, 1993, 13, 133-142.
- Mercier, A., Cordeau, J., and Soumis, F. A computational study of Benders' decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research, 2005, 32, 1451-1476.
- Angulo, G., Ahmed, S. and Dey, S. S. Improving the Integer L-Shaped Method. INFORMS Journal on Computing, 2016, 28, 483-499.

Three-phase method

$$
\begin{aligned}
\min \quad & c^\top x + d^\top y, \\
\text{subject to} \quad & Ax \geq b, \\
& Bx + Dy \geq g, \\
& x \in \mathbb{Z}_+^{p_1} \times \mathbb{R}_+^{n_1 - p_1}, \\
& y \in \mathbb{Z}_+^{p_2} \times \mathbb{R}_+^{n_2 - p_2}.
\end{aligned}
$$

- Classical approach used to improve the convergence of the BD algorithm.
- First proposed by McDaniel and Devine (1977), rediscovered by many other researchers.
  1. Relax integrality on x and y
  2. Solve relaxed master problem to optimality by BD
  3. Reintroduce integrality on x, solve master problem to optimality
  4. *Reintroduce integrality on y, solve master problem to optimality.*

Three-phase method in branch-and-cut

- Different implementation to the original algorithm
  1. Generate BD cuts from fractional LP solutions while solving the master problem root node.
  2. Generate BD cuts from integral LP solutions using a relaxed subproblem throughout the tree.
  3. Generate BD cuts from integral LP solutions using a integer subproblem throughout the tree.

- Option: Perform the first phase at nodes deeper than the root node.