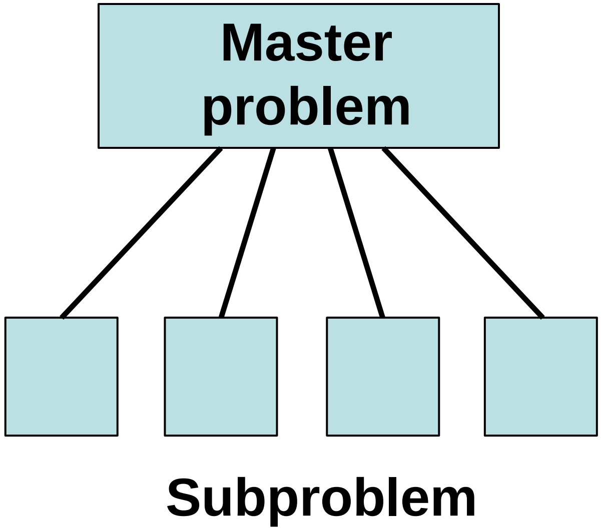
Logic-Based Benders Decomposition

John Hooker
Carnegie Mellon University

Benders Day Workshop
Eindhoven University
May 2024

Benders Decomposition

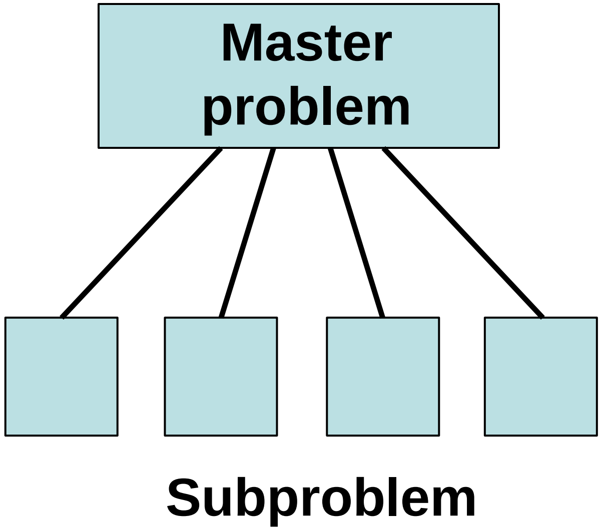
- One of the **best known** and **most successful** strategies for solving hard optimization problems.
  - Decomposes the problem **without sacrificing optimality**.



Benders 1962

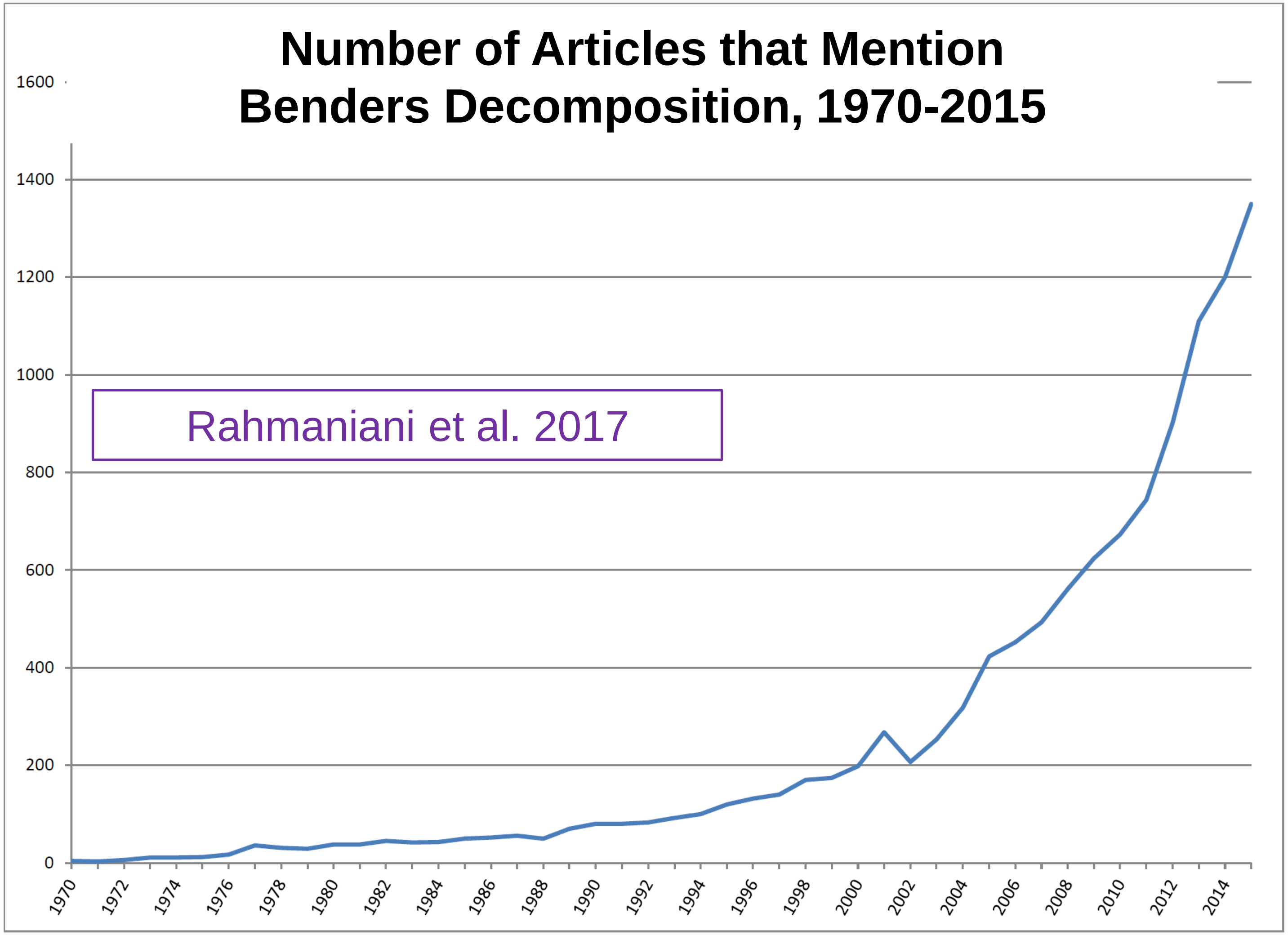
Benders Decomposition

- One of the **best known** and **most successful** strategies for solving hard optimization problems.
  - Decomposes the problem **without sacrificing optimality**.
    - **Master problem** contains **complicating variables**.
    - Problem simplifies to an easier **subproblem** when these variables are fixed.
    - Subproblem often **decouples** into smaller problems.
    - Key idea: **Benders cuts** provide feedback to master problem.



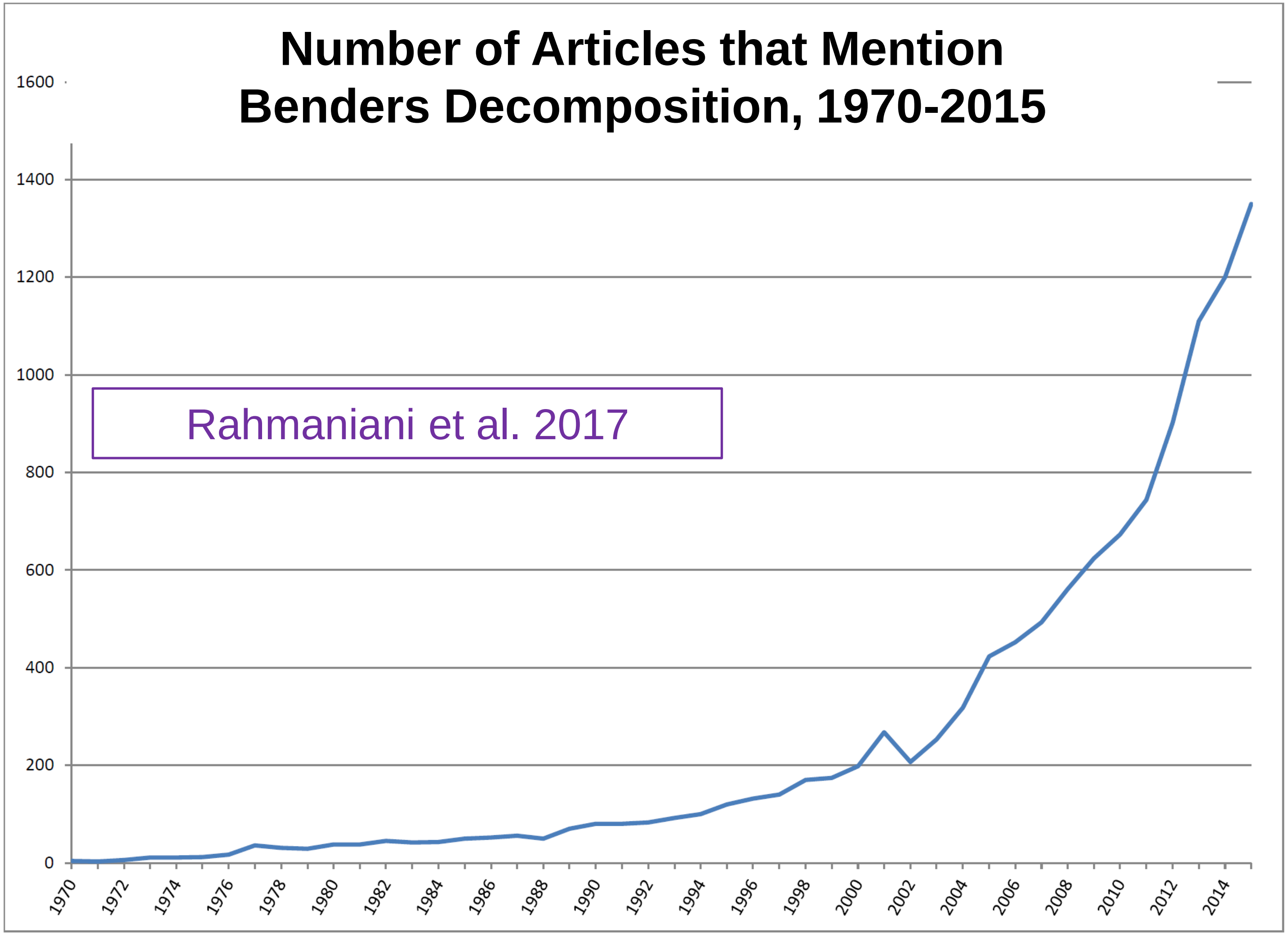
Benders 1962

Benders Decomposition



Based on Google Scholar

Benders Decomposition



Explosion continues after 2015

Based on Google Scholar

Benders Decomposition

- Classical Benders decomposition has a limitation.
  - The subproblem must be a **linear programming** problem.
    - Or a continuous nonlinear programming problem.
    - The linear programming **dual** provides the Benders cuts.

Benders 1962

Geoffrion 1972

Benders Decomposition

- Classical Benders decomposition has a limitation.
  - The subproblem must be a **linear programming** problem.
    - Or a continuous nonlinear programming problem.
    - The linear programming **dual** provides the Benders cuts.

Benders 1962

Geoffrion 1972

- But the **underlying idea** is **more general** than it may appear.
  - This opens the door to **many new applications**.

Logic-Based Benders

- The **key idea**:
  - The subproblem dual multipliers encode a **proof of optimality**.
    - By proving a **bound** on the optimal value.
  - What kind of bound can be obtained **using the same proof** if the master problem solution changes?
    - The **Benders cut** answers this question.

Logic-Based Benders

- The **key idea**:
  - The subproblem dual multipliers encode a **proof of optimality**.
    - By proving a **bound** on the optimal value.
  - What kind of bound can be obtained **using the same proof** if the master problem solution changes?
    - The **Benders cut** answers this question.
- To exploit this idea:
  - Replace the LP dual with an **inference dual** whose solution is a **proof** that **logically deduces** a bound.
    - A **logic-based Benders cut** is derived from this proof.

JH 2000, JH & Ottosson 2003

Logic-Based Benders

- Result: **Logic-based** Benders decomposition (LBBD)
  - The subproblems can, in principle, be **any kind** of optimization problem.
    - Since the Benders cuts are obtained from an **inference dual**.
  - **Speedup** over state of the art can be several orders of magnitude.

JH 2000, JH & Ottosson 2003

Logic-Based Benders

- Result: **Logic-based** Benders decomposition (LBBD)
  - The subproblems can, in principle, be **any kind** of optimization problem.
    - Since the Benders cuts are obtained from an **inference dual**.
  - **Speedup** over state of the art can be several orders of magnitude.
  - The Benders cuts must often be **specifically designed** for a class of problems.
    - An inconvenience, but also an **advantage**.
    - It allows one to exploit **special structure** in the problem

JH 2000, JH & Ottosson 2003

Classical Benders Method

Solve the problem

$$\min\ f(\boldsymbol{x}) + \boldsymbol{c}\boldsymbol{y}$$
$$\boldsymbol{g}(\boldsymbol{x}) + A\boldsymbol{y} \geq \boldsymbol{b}$$

- Subproblem must be an LP.
- Benders cuts are based on classical duality.

Master problem

$$\min\ z$$
$$z \geq f(\boldsymbol{x}) + \boldsymbol{u}_k\big(\boldsymbol{b} - \boldsymbol{g}_k(\boldsymbol{x})\big),$$
$$\text{all } k \text{ (Benders cuts)}$$

Minimize cost *z* subject to bounds given by Benders cuts, obtained from values of ***x*** attempted in previous iterations *k*.

Trial value $\bar{x}$ that solves master →

← Benders cut

Subproblem

$$\min\ f(\bar{\boldsymbol{x}}) + \boldsymbol{c}\boldsymbol{y}$$
$$A\boldsymbol{y} \geq \boldsymbol{b} - \boldsymbol{g}(\bar{\boldsymbol{x}})$$

Obtain proof of optimality (solution ***u*** of LP dual). Use dual solution to obtain a Benders cut.

Repeat until the master problem and subproblem have the same optimal value.

Logic-based Benders Method

Solve the problem

$$\min \; f(\boldsymbol{x}, \boldsymbol{y})$$
$$(\boldsymbol{x}, \boldsymbol{y}) \in S$$
$$\boldsymbol{x} \in D$$

- Subproblem can be **any** optimization problem.
- View the subproblem dual as a **logical inference** problem.

Master problem

$$\min \; z$$
$$z \geq v_k(\boldsymbol{x}), \;\; \text{all } k \text{ (Benders cuts)}$$

Minimize cost *z* subject to bounds given by Benders cuts, obtained from values of ***x*** attempted in previous iterations *k*.

→ Trial value $\bar{x}$ that solves master

← Benders cut

Subproblem

$$\min \; f(\bar{\boldsymbol{x}}, \boldsymbol{y})$$
$$(\bar{\boldsymbol{x}}, \boldsymbol{y}) \in S$$

Obtain proof of optimality (solution of inference dual). Use **same proof** to deduce cost bounds for other values of ***x***, yielding a Benders cut

Repeat until the master problem and subproblem have the same optimal value.

Inference Duality

with LP duality as a special case

General optimization problem

$$\begin{aligned} &\min\ f(\boldsymbol{y}) \\ &\boldsymbol{y} \in S \end{aligned}$$

Inference dual

$$\begin{aligned} &\max\ z \\ &\big(\boldsymbol{y} \in S\big) \stackrel{P}{\Rightarrow} \big(z \le f(\boldsymbol{x})\big) \\ &P \in \mathcal{P} \end{aligned}$$

Inference Duality

with LP duality as a special case

General optimization problem

$$\min \ f(\boldsymbol{y})$$
$$\boldsymbol{y} \in S$$

Inference dual

$$\max \ z$$
$$(\boldsymbol{y} \in S) \stackrel{P}{\Rightarrow} (z \leq f(\boldsymbol{x}))$$
$$P \in \mathcal{P}$$

LP problem

$$\min \ \boldsymbol{cy}$$
$$A\boldsymbol{y} \geq \boldsymbol{b}$$
$$\boldsymbol{y} \geq \boldsymbol{0}$$

Its inference dual

$$\max \ z$$
$$\begin{pmatrix} A\boldsymbol{y} \geq \boldsymbol{b} \\ \boldsymbol{y} \geq \boldsymbol{0} \end{pmatrix} \Rightarrow (\boldsymbol{cy} \geq z)$$

Inference Duality

with LP duality as a special case

General optimization problem

$$\min \ f(\boldsymbol{y})$$
$$\boldsymbol{y} \in S$$

Inference dual

$$\max \ z$$
$$\big(\boldsymbol{y} \in S\big) \overset{P}{\Rightarrow} \big(z \le f(\boldsymbol{x})\big)$$
$$P \in \mathcal{P}$$

LP problem

$$\min \ \boldsymbol{c}\boldsymbol{y}$$
$$A\boldsymbol{y} \ge \boldsymbol{b}$$
$$\boldsymbol{y} \ge \boldsymbol{0}$$

Its inference dual

$$\max \ z$$
$$\begin{pmatrix} A\boldsymbol{y} \ge \boldsymbol{b} \\ \boldsymbol{y} \ge \boldsymbol{0} \end{pmatrix} \Rightarrow \big(\boldsymbol{c}\boldsymbol{y} \ge z\big)$$

Applying Farkas Lemma

$$\max \ z$$
$$\begin{pmatrix} z \le \boldsymbol{u}b \\ \boldsymbol{u}A \le \boldsymbol{c} \end{pmatrix} \text{ for some } \boldsymbol{u} \ge \boldsymbol{0}$$

Inference Duality

with LP duality as a special case

General optimization problem

$$\min \; f(\boldsymbol{y})$$
$$\boldsymbol{y} \in S$$

Inference dual

$$\max \; z$$
$$(\boldsymbol{y} \in S) \overset{P}{\Rightarrow} (z \le f(\boldsymbol{x}))$$
$$P \in \mathcal{P}$$

LP problem

$$\min \; \boldsymbol{c}\boldsymbol{y}$$
$$A\boldsymbol{y} \ge \boldsymbol{b}$$
$$\boldsymbol{y} \ge \boldsymbol{0}$$

Its inference dual

$$\max \; z$$
$$\begin{pmatrix} A\boldsymbol{y} \ge \boldsymbol{b} \\ \boldsymbol{y} \ge \boldsymbol{0} \end{pmatrix} \Rightarrow (\boldsymbol{c}\boldsymbol{y} \ge z)$$

Applying Farkas Lemma

$$\max \; z$$
$$\begin{pmatrix} z \le \boldsymbol{u}b \\ \boldsymbol{u}A \le \boldsymbol{c} \end{pmatrix} \text{ for some } \boldsymbol{u} \ge \boldsymbol{0}$$

This can be written

$$\max \; z$$
$$z \le \boldsymbol{u}\boldsymbol{b}$$
$$\boldsymbol{u}A \le \boldsymbol{c}$$
$$\boldsymbol{u} \ge \boldsymbol{0}$$

Inference Duality

with LP duality as a special case

General optimization problem

$$\min \ f(\boldsymbol{y})$$
$$\boldsymbol{y} \in S$$

Inference dual

$$\max \ z$$
$$(\boldsymbol{y} \in S) \overset{P}{\Rightarrow} (z \leq f(\boldsymbol{x}))$$
$$P \in \mathcal{P}$$

LP problem

$$\min \ \boldsymbol{cy}$$
$$A\boldsymbol{y} \geq \boldsymbol{b}$$
$$\boldsymbol{y} \geq \boldsymbol{0}$$

Its inference dual

$$\max \ z$$
$$\begin{pmatrix} A\boldsymbol{y} \geq \boldsymbol{b} \\ \boldsymbol{y} \geq \boldsymbol{0} \end{pmatrix} \Rightarrow (\boldsymbol{cy} \geq z)$$

Applying Farkas Lemma

$$\max \ z$$
$$\begin{pmatrix} z \leq \boldsymbol{u}\boldsymbol{b} \\ \boldsymbol{u}A \leq \boldsymbol{c} \end{pmatrix} \text{ for some } \boldsymbol{u} \geq \boldsymbol{0}$$

This can be written

$$\max \ z$$
$$z \leq \boldsymbol{ub}$$
$$\boldsymbol{u}A \leq \boldsymbol{c}$$
$$\boldsymbol{u} \geq \boldsymbol{0}$$

Remove z to get **LP dual**

$$\max \ \boldsymbol{ub}$$
$$\boldsymbol{u}A \leq \boldsymbol{c}$$
$$\boldsymbol{u} \geq \boldsymbol{0}$$

Inference Duality

Using various logical inference methods

Type of dual	**Inference method**	**Complete (strong dual)?**
Linear programming	Nonnegative linear combination + domination (linear inequalities)	Yes
Lagrangian	Same as LP dual (nonlinear or integer inequalities)	No
Surrogate	Nonnegative linear combination + material implication	No
Subadditive	Chvatal-Gomory cuts	Yes
Branching	Branch and bound	Yes

Branch and check

- Solve master problem only **once**
  - **Branch** on master problem variables
  - Invoke subproblem at **feasible nodes**
    - Add **Benders cut** at node, and continue branching.

JH 2000, Thorsteinsson 2001

Branch and check

- Solve master problem only **once**
  - **Branch** on master problem variables
  - Invoke subproblem at **feasible nodes**
    - Add **Benders cut** at node, and continue branching.

JH 2000, Thorsteinsson 2001

- Very **different** from branch and cut
  - Cuts are **not valid** for the problem solved by branching---only for the **subproblem**.
  - Cuts contain variables that have already been **fixed** by branching
    - Rather than variables not yet fixed
  - Cuts derived using **a different type of reasoning**.

A Bit of History

▪ **Classical** Benders decomposition	*Benders (1962)*
▪ "Generalized" BD (for continuous **nonlinear** inequalities)	*Geoffrion (1972)*
▪ Connection between **LP duality** and **logic** (unit resolution proof)	*Jeroslow & Wang (1990)*
▪ Logic circuit verification – **First clear application** of LBBD (in retrospect)	*JH and Yan (1995)*
▪ **General statement** of LBBD and branch & check	*JH (2000)*
▪ Computational **testing** of LBBD	*Jain and Grossmann (2001)*
▪ Computational **testing** of branch & check	*Thorsteinsson (2001)*
▪ **Further development** and testing	*JH and Ottosson (2003)*
▪ Combinatorial Benders cuts (branch and check applied to **MILP**)	*Codato and Fischetti (2006)*

Example: Machine Scheduling

- Assign tasks to machines.
- Then schedule tasks assigned to each machine.
  - Subject to **time windows**.
  - Scheduling problem **decouples** into a separate problem for each machine.
- Objective is to minimize **makespan**.

Example: Machine Scheduling

- Assign tasks in master, schedule in subproblem.
  - Combine mixed integer programming and constraint programming

Master problem

Assign tasks to resources to minimize cost.

Solve by **mixed integer programming**.

Trial assignment $\bar{x}$ →

← Benders cut $z \geq g_k(x)$

Subproblem

Schedule jobs on each machine, subject to time windows.

Constraint programming obtains proof of optimality (dual solution).

Use same proof to deduce cost for some other assignments, yielding Benders cut.

Nogood Cuts

- Logic-based Benders cuts (optimality cuts)
  - Simplest cut is a **strengthened nogood cut**

$$z \geq z_i^* - M \sum_{j \in J_i} (1 - x_{ij})$$

  - where $x_{ij} = 1$ when task j assigned to machine i
  - M = large number
  - J_i is a reduced set of tasks that result in optimal makespan z_i^* when assigned to machine i.
  - Various algorithms reduce J_i by repeatedly re-solving subproblem.

Nogood Cuts

- Strengthening methods for nogood cuts

Method	Yields irreducible cut?
Greedy	No
Deletion filter	Yes
Heuristic binary search	No
Depth-first binary search	Yes
Quick Xplain	Yes
Binary search Quick Xplain	Yes

Analytical Cuts

- **Analytical cuts** for machine scheduling
  - Stronger cuts that exploit problem structure:

$$z \geq z_i^* - \sum_{j \in J_i} (1 - x_{ij})\big(p_{ij} + \max\{0,\ r_j - r_{\min} - p_{\min}\}\big) - (d_{\max} - d_{\min})$$

$$z \geq z_i^* - \sum_{j \in J_i} (1 - x_{ij})\big(p_{ij} + \max\{0,\ r_j - r_{\min} - p_{\min}\}\big) + (d_{\max} - d_{\min})\big)$$

where

$$z_i^* = \text{current min makespan for resource } i$$

$$r_j = \text{ release time, } p_{ij} = \text{ processing time, } d_j = \text{ deadline}$$

$$r_{\min} = \min_{j \in J_i}\{r_j\},\ \ p_{\min} = \min_{j \in J_i}\{p_{ij}\}$$

$$d_{\min} = \min_{j \in J_i}\{d_j\},\ \ d_{\max} = \max_{j \in J_i}\{d_j\}$$

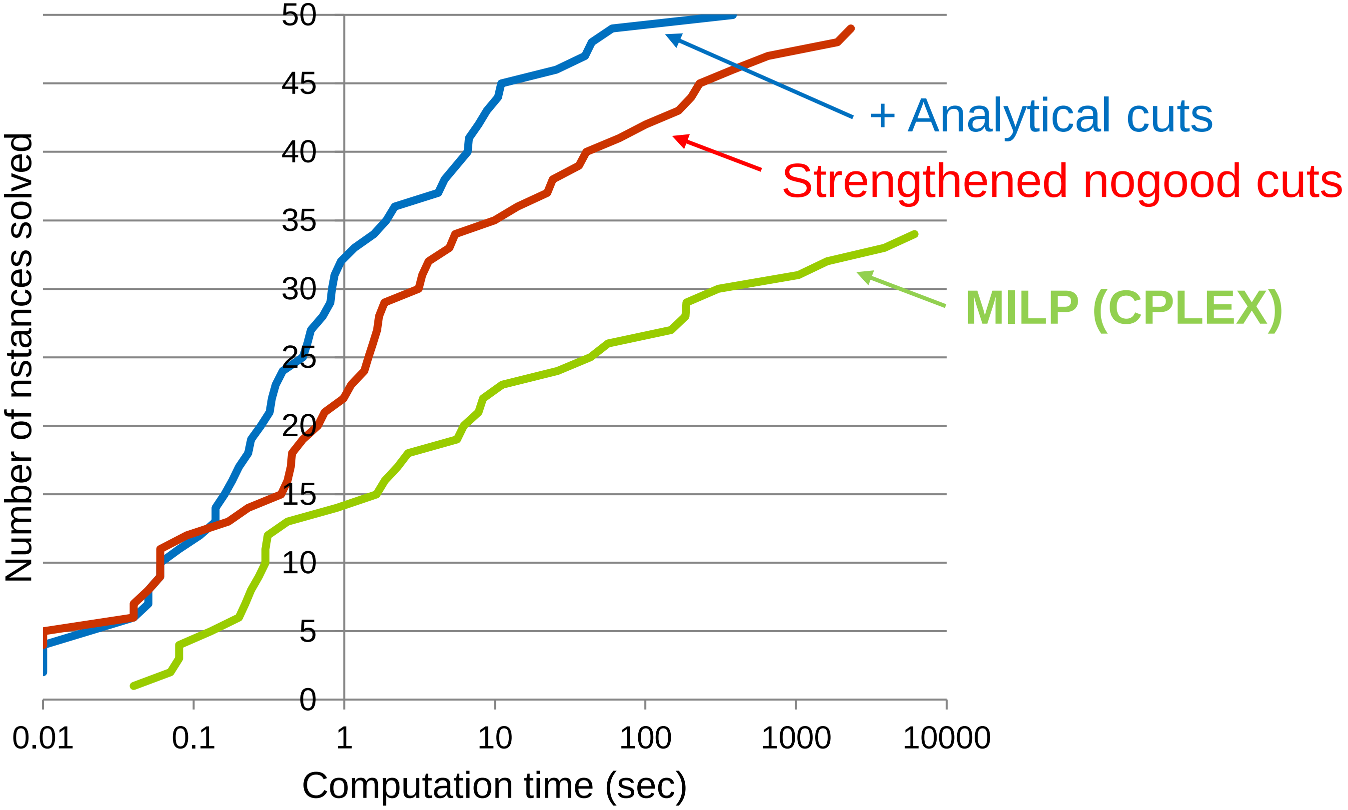
Logic-based Cuts

- Types of logic-based cuts

Type of cut	**Function**
Nogood cut	Excludes **most recent** master problem solution
Strengthened nogood cut	Excludes a **class** of solutions based on re-solving subproblem
Analytical cut	Exploits most recent solution and **problem structure** to exclude a class of solutions
Explanation-based cut	Cut based directly on solution of the **inference dual** of subproblem
Combinatorial Benders cut	Special case of strengthened nogood cut designed for **MILP**

Example: Machine Scheduling

LBBD performance profile for 50 problem instances



Ciré, Coban, JH (2015)

Stochastic Machine Scheduling

- **Random** processing times
  - Represented by multiple scenarios.
  - Processing times revealed after machine assignment but before scheduling on each machine.
  - Solve subproblem by CP
- Previous state of the art
  - **Integer L-shaped** method.
  - Classical Benders cuts based on LP relaxation of MILP subproblem.
  - Weak “integer cuts” to ensure convergence.

Stochastic Machine Scheduling

Computation time

10 jobs, 2 machines, processing times drawn from uniform distribution

Each time (seconds) is average over 3 instances

Scenarios	*Integer L-shaped*	*Branch & Check*
1	127	1
5	839	2
10	2317	3
50	> 3600	17
100	> 3600	37
500	> 3600	279

Elçi and JH (2022)

Example: Home Healthcare

- Caregiver assignment and routing
  - Focus on regular hospice care
  - Qualifications matched to patient needs
  - Time windows, breaks, etc., observed
  - Weekly schedule
- Rolling time horizon
  - New patients every week.
  - Minimal schedule change for existing patients.
- Efficient staff utilization
  - Maximize number of patients served by given staff level.
  - Optimality important, due to cost of taking on staff.

Heching, JH, Kimura (2019)

Example: Home Healthcare

Master problem

Assign patients to healthcare aides and days of the week

$$\max \sum_j \delta_j$$

$$\sum_i x_{ij} = \delta_j, \quad \text{all } j$$

$$\sum_{i,k} y_{ijk} = v_j \delta_j, \quad \text{all } j$$

$$y_{ijk} \leq x_{ij}, \quad \text{all } i, j, k$$

Spacing constraints on visit days

Benders cuts

Relaxation of subproblem

$$\delta_j, x_{ij}, y_{ijk} \in \{0, 1\}$$

= 1 if patient *j* scheduled

Required number of visits per week

= 1 if patient *j* assigned to aide *I* on day *k*

= 1 if patient *j* assigned to aide *i*

MILP model

Example: Home Healthcare

Subproblem

Sequence and schedule visits for each healthcare aide *j* separately.

$$\text{all-different}\{\pi_{k\nu} \mid \nu = 1, \ldots, |P_i|\}$$

$$[s_j, s_j + p_j] \subseteq [r_j, d_j]$$

$$s_{\pi_{k\nu}} + p_{\pi_{k\nu}} + t_{\pi_{k\nu}\pi_{k,\nu+1}} \leq s_{\pi_{k,\nu+1}}, \quad \text{all } k, \nu$$

*n*th patient in sequence

Patients assigned to aide *i*

Start time

Visit duration

Travel time

CP model
(or use interval variables)

Example: Home Healthcare

Strengthened nogood cuts

If no feasible schedule for aide *j*, generate a cut requiring that at least one patient be assigned to another aide.

$$\sum_{j \in \bar{P}_{ik}} (1 - y_{ijk}) \geq 1$$

Reduced set of patients whose assignment to aide *i* on day *k* creates infeasibility, obtained by re-solving subproblem with fewer aides. This excludes many assignments that cannot be feasible.

Branch and check

Variant of LBBD that generates Benders cuts during branch-and-bound solution of master problem. Master problem solved only once.

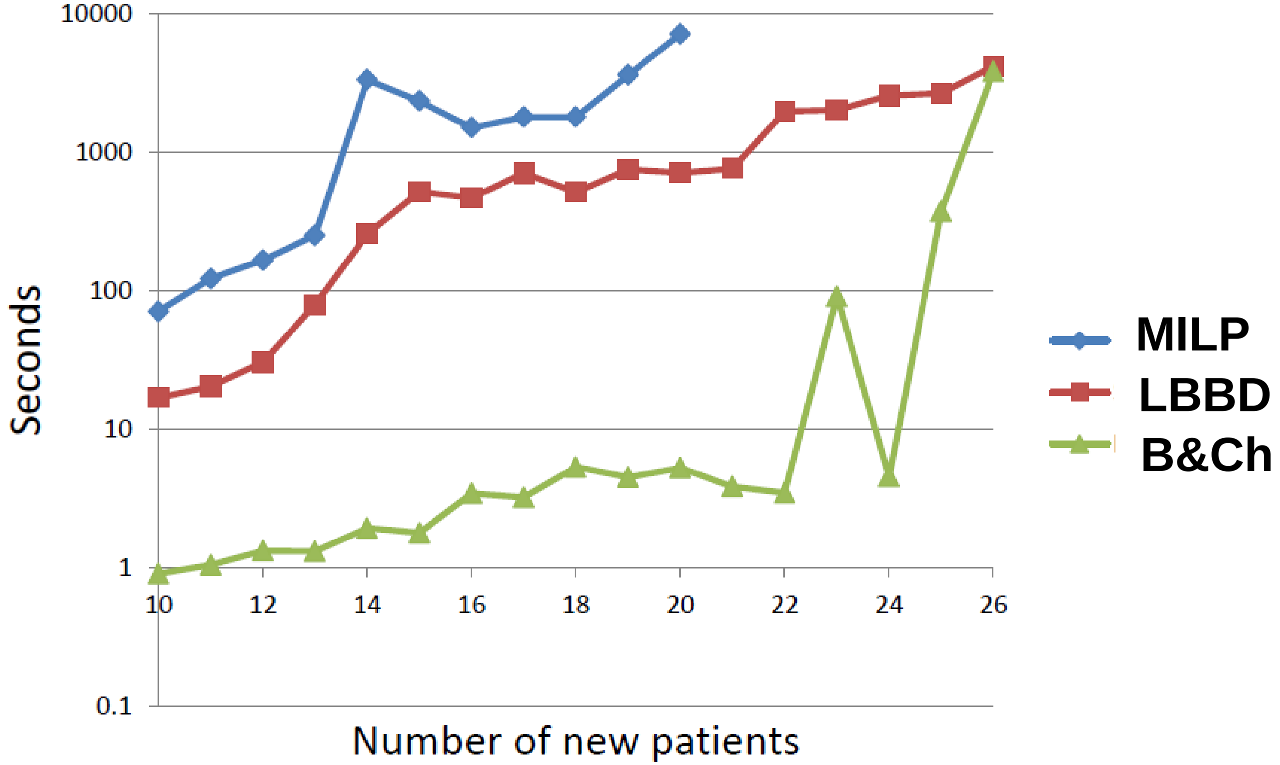
JH (2000), Thorsteinsson (2001)

Example: Home Healthcare

Computational results

Heching, JH, Kimura (2019)

Data from home hospice care firm.



Example: Home Healthcare

Computational results

Heching, JH, Kimura (2019)

Data from Danish home care agency.

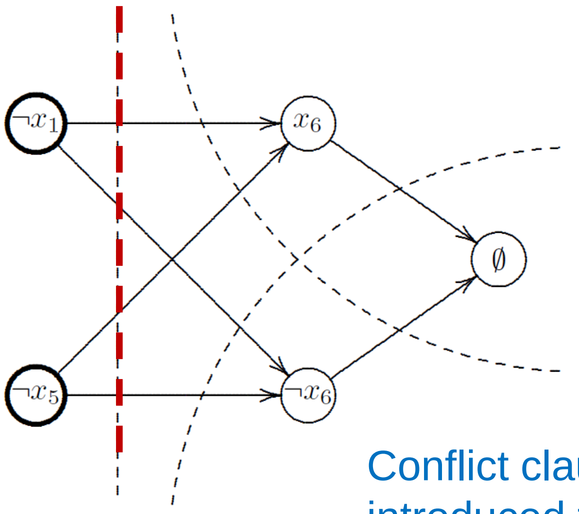
Instance	Patients	Crews	Weighted objective MILP	Weighted objective LBBD	Weighted objective B&Ch	Covering objective MILP	Covering objective LBBD	Covering objective B&Ch
hh	30	15	*	3.16	**1.41**	*	**23.3**	441
ll1	30	8	*	1.74	**0.43**	*	108	**1.41**
ll2	30	7	2868	1.56	**0.32**	*	**1.38**	6.45
ll3	30	6	1398	2.16	**0.30**	*	**3.07**	5.98

*Computation time exceeded one hour.

Example: SAT

- **Conflict clauses** as logic-based Benders cuts.
  - The **subproblem** is the problem at a node of the DPLL search tree.
  - The **inference dual** is defined by **unit resolution**.
  - The **dual solution** is a unit refutation, encoded in a **conflict graph**.

The conflict graph is a solution of the **inference dual**



The resulting **conflict clause** $x_1 \vee x_5$ is a **Benders cut**

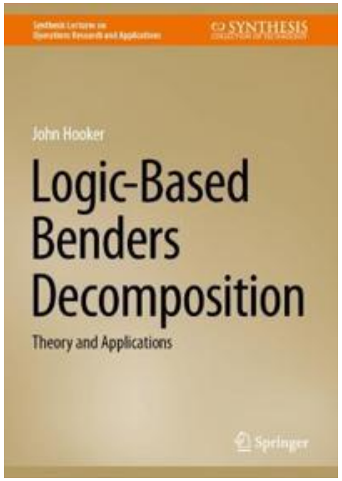
Conflict clauses were introduced for Benders **a year before** they appeared in SAT literature

LBBD Applications

- Recent book describes LBBD applications to **147 problem classes**, described in **226 publications**.
  - Many use domain-specific analytical cuts.

JH (2023)

- Some examples…



Some LBBD applications

- Transportation
  - Vehicle routing and scheduling
  - Traffic diversion
  - Train scheduling
  - Railroad gantry crane scheduling
  - Pipeline scheduling
  - Container ship routing and scheduling
  - Lock scheduling
  - Ride sharing
  - Bicycle sharing
  - Electric bus routing
  - Location and scheduling of charging stations

Some LBBD applications

- Container port management
  - Container stacking and drayage
  - Yard crane scheduling
  - Gantry crane assignment and scheduling
  - Berth allocation
  - Ship loader scheduling

Some LBBD applications

- Production and maintenance:
  - Task assignment and scheduling
  - Machine assignment and scheduling
  - Job/flow shop scheduling
  - Assembly line balancing
  - Work cell assignment and scheduling
  - Employee shift assignment
  - Lot sizing
  - Blast furnace scheduling
  - Mine scheduling
  - Chemical batch scheduling
  - Aircraft maintenance
  - Wind turbine maintenance

Some LBBD applications

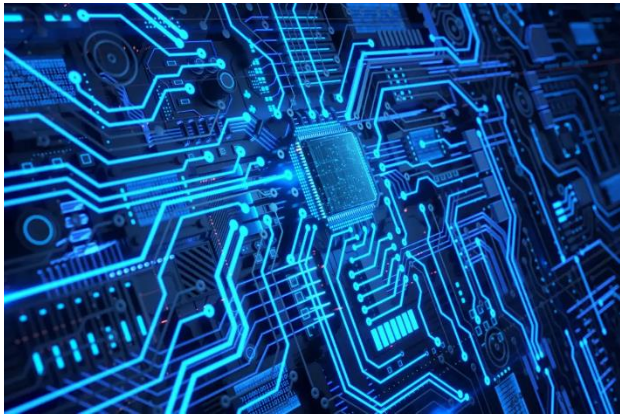
- Supply chain logistics
  - Plant location and truck allocation
  - Packing and cutting
  - Distribution center location & vehicle routing
  - Concrete delivery
  - Wheat supply chain
  - Intermodal transport
  - Supply chain reconfiguration

Some LBBD applications

- End user delivery
  - Shelf space allocation in warehouse
  - Order picking in warehouse
  - Robotic pod repositioning
  - Order consolidation
  - Crowdshipping
  - Packing orders into parcels
  - Package delivery with drones

Some LBBD applications

- Telecommunications and computing
  - Allocation of frequency spectrum
  - Local are network design
  - Optical network regenerator locations
  - Network reliability
  - Network upgrade
  - Edge computing
  - Allocation of tasks to processors in multicore computing
  - Information flow for autonomous driving
  - Logic circuit verification



Some LBBD applications

- Medical applications
  - Clinical outpatients scheduling
  - Operating room scheduling
  - Hospital therapist scheduling
  - Home healthcare routing and scheduling
  - Kidney exchange
  - DNA sequence alignment
  - Radiation therapy control
  - Cancer screening
  - Covid test center location
  - Vaccine distribution

Some LBBD applications

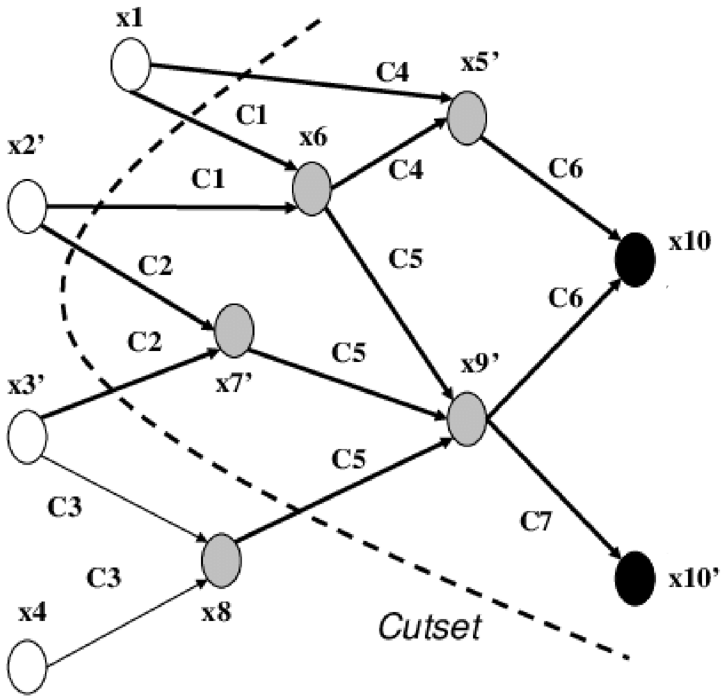
- Disaster management
  - Robust disaster preparedness
  - Earthquake infrastructure risk management
  - Search and rescue after earthquake
  - Fortifying service facilities
  - Electric grid restoration
  - Wildfire suppression
  - Pipeline damage monitoring

Some LBBD applications

- Other applications
  - Tournament scheduling
  - Baseball umpire scheduling
  - Course timetabling
  - Call center scheduling
  - Network interdiction
  - Decision tree learning
  - Military flow diversion
  - Energy policy analysis
  - Electricity price equilibration

Some LBBD applications

- Abstract problem classes:
  - SAT, maxSAT, SATMT (conflict clauses)
  - 0-1 programming with subproblem decoupling
  - General optimal control
  - Linear complementarity and quadratic programming
  - Operator counts in automated planning
  - Modular arithmetic
  - Minimal chord completion
  - Piecewise linear regression
  - Robust optimization



49

LBBD general-purpose software

- Automatic LBBD in **MiniZinc**
  - Uses SAT-style conflict clauses
- **Nutmeg**
  - Uses **branch and check**

ed-lam/**nutmeg**

Nutmeg – a MIP and CP branch-and-check solver

Conclusion

The inherent potential of Benders decomposition continues to unfold after 60 years.


THE
END