Decision Diagrams for Integer Linear and Nonlinear Programming

Willem-Jan van Hoeve (Carnegie Mellon University)

Joint work with:
Danial Davarnia (Iowa State University)
Christian Tjandraatmadja (Google)

EURO, June 2019
Decision Diagrams for Integer Linear and Nonlinear Programming

Willem-Jan van Hoeve (Carnegie Mellon University)

Joint work with:
Danial Davarnia (Iowa State University)
Christian Tjandraatmadja (Google)

EURO, June 2019
Overview

- Motivation
- Decision Diagrams for Integer Programming
  - incorporate DD bounds in MIP search
  - cut generation
  - outer approximation for MINLP
- Conclusions
Decision Diagrams and Integer Feasible Sets
$S = \{(0,0,0),$
    
    $(0,0,1),$
    
    $(0,1,0),$  
    
    $(1,0,0)\}$
$S = \{(0,0,0),
(0,0,1),
(0,1,0),
(1,0,0)\}$

$x_1 + x_2 + x_3 \leq 1$
Decision Diagrams and Integer Feasible Sets

\[ S = \{(0,0,0),\]
\[ (0,0,1),\]
\[ (0,1,0),\]
\[ (1,0,0)\}\]
Decision Diagrams and Integer Feasible Sets

$S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\}$

$x_1 + x_2 + x_3 \leq 1$
Decision Diagrams and Integer Feasible Sets

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \]

BDD: binary decision diagram

MDD: multi-valued decision diagram

\[ x_1 + x_2 + x_3 \leq 1 \]
S = \{(0,0,0),
(0,0,1),
(0,1,0),
(1,0,0)\}
Optimizing Over Decision Diagrams

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \]

\[ x_1 + x_2 + x_3 \leq 1 \]

\[ \max x_1 + 2x_3 \]
Optimizing Over Decision Diagrams

\[ S = \{(0,0,0), \quad (0,0,1), \quad (0,1,0), \quad (1,0,0)\} \]
Optimizing Over Decision Diagrams

$S = \{(0,0,0), \newline
(0,0,1), \newline
(0,1,0), \newline
(1,0,0)\}$

$x_1 \quad 0 \quad 1 \quad \newline
x_2 \quad 0 \quad 0 \quad 0 \quad \newline
x_3 \quad 2 \quad 0 \quad 0 \quad \newline
max x_1 + 2x_3 \newline
x_1 + x_2 + x_3 \leq 1
Optimizing Over Decision Diagrams

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \]
Optimizing Over Decision Diagrams

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \]

optimal objective value: 2
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

[Andersen et al. 2007]

[Bergman et al. 2011, 2014]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \]

\[ x_1 + x_2 + x_3 \leq 1 \]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \]

\[ x_1 + x_2 + x_3 \leq 1 \]

[Andersen et al. 2007]

[Bergman et al. 2011, 2014]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[
S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\}
\]

\[x_1 + x_2 + x_3 \leq 1\]

[Andersen et al. 2007]

[Bergman et al. 2011, 2014]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \cup \{(0,1,1), (1,0,1)\} \]

\[ x_1 + x_2 + x_3 \leq 1 \]

[Andersen et al. 2007]
[Bergman et al. 2011, 2014]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{ (0,0,0), (0,0,1), (0,1,0), (1,0,0) \} \cup \{ (0,1,1), (1,0,1) \} \]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \cup \{(0,1,1), (1,0,1)\} \]

\[ \text{max} \ x_1 + 2x_3 \]

[Andersen et al. 2007]

[Bergman et al. 2011, 2014]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \cup \{(0,1,1), (1,0,1)\} \]

[Andersen et al. 2007] [Bergman et al. 2011, 2014]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[ S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \cup \{(0,1,1), (1,0,1)\} \]
Relaxed Decision Diagrams

- Relaxed Decision Diagrams have limited width: polynomial size
- Over-approximation of feasible set: dual bound

\[
S = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0)\} \cup \{(0,1,1), (1,0,1)\}
\]

\[
\text{max } x_1 + 2x_3
\]

upper bound: 3

[Andersen et al. 2007]

[Bergman et al. 2011, 2014]
Categories of Successful Applications

• Sequencing and routing problems
  – single machine scheduling with setup times, time windows, precedence constraints (including TSPTW) [Cire & vH, OR2013], [Knable et al. EJOR 2017] [O’Neil & Hoffman, ORL2019]

• Decomposition and embedding in MIP models
  – nonlinear objective functions [Bergman&Cire, MgtSc 2018]
  – column generation [Morrison et al. IJOC 2016] [Kowalczyk & Leus IJOC 2018]

• Combinatorial optimization

• Constraint Programming
  – DD-based constraint propagation [Andersen et al. CP2007] [Hoda et al. CP2010]
Application to Integer Programming

- IP model
- Presolve
- Cutting planes
- Branch-and-bound
Application to Integer Programming

IP model \[\downarrow\] Presolve \[\downarrow\] Cutting planes \[\uparrow\] DD construction from IP model \[\uparrow\] Branch-and-bound

[Tjandraatmadja, PhD 2018]
DD Compilation for IP Models
DD Compilation for IP Models

• Option 1: use linear constraints from model
  – single DD for (subset) of constraints; usually weaker than LP bound
  – (using *multiple* DDs can be quite effective, for nonlinear problems)
DD Compilation for IP Models

- Option 1: use linear constraints from model
  - single DD for (subset) of constraints; usually weaker than LP bound
  - (using multiple DDs can be quite effective, for nonlinear problems)
- Option 2: identify structure in model
  - e.g. set covering, set packing, independent set,…
  - dedicated DD representing substructure of the model
  - can be stronger than LP bound, and faster to compute
DD Compilation for IP Models

- Option 1: use linear constraints from model
  - single DD for (subset) of constraints; usually weaker than LP bound
  - (using multiple DDs can be quite effective, for nonlinear problems)
- Option 2: identify structure in model
  - e.g. set covering, set packing, independent set,…
  - dedicated DD representing substructure of the model
  - can be stronger than LP bound, and faster to compute
- Option 3: use structure inferred by solver
  - conflict graph/clique table
Conflict Graph for Binary Problems

\[
x_1 + x_2 + x_3 \leq 1
\]
\[
x_2 + (1 - x_3) \leq 1
\]
\[
(1 - x_1) + (1 - x_2) \leq 1
\]
Conflict Graph for Binary Problems

\[ x_1 + x_2 + x_3 \leq 1 \]
\[ x_2 + (1 - x_3) \leq 1 \]
\[ (1 - x_1) + (1 - x_2) \leq 1 \]
Conflict Graph for Binary Problems

\[ x_1 + x_2 + x_3 \leq 1 \]
\[ x_2 + (1 - x_3) \leq 1 \]
\[ (1 - x_1) + (1 - x_2) \leq 1 \]
Conflict Graph for Binary Problems

\[
\begin{align*}
  x_1 + x_2 + x_3 &\leq 1 \\
  x_2 + (1 - x_3) &\leq 1 \\
  (1 - x_1) + (1 - x_2) &\leq 1
\end{align*}
\]
Conflict Graph for Binary Problems

\[ x_1 + x_2 + x_3 \leq 1 \]
\[ x_2 + (1 - x_3) \leq 1 \]
\[ (1 - x_1) + (1 - x_2) \leq 1 \]
Conflict Graph for Binary Problems

Conflict graphs are inferred and constructed by most modern MIP solvers

\[ x_1 + x_2 + x_3 \leq 1 \]
\[ x_2 + (1 - x_3) \leq 1 \]
\[ (1 - x_1) + (1 - x_2) \leq 1 \]
Decision Diagram Compilation

• State: variable domains
• Transition: propagate decision

\[ x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{0, 1\} \]

- \( x_1 \)
- \( x_2 \)
- \( x_3 \)

\( \overline{x}_1 \)
\( \overline{x}_2 \)
\( \overline{x}_3 \)
Decision Diagram Compilation

- **State**: variable domains
- **Transition**: propagate decision

\[ x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{0, 1\} \]

\[ x_2 \in \{0, 1\}, x_3 \in \{0, 1\} \quad x_2 \in \{0, 1\}, x_3 \in \{1\} \]
Decision Diagram Compilation

• State: variable domains
• Transition: propagate decision
Decision Diagram Compilation

- State: variable domains
- Transition: propagate decision

\[
\begin{array}{c}
\overline{x_1} \\
\end{array}
\begin{array}{c}
x_1 \\
\end{array}
\begin{array}{c}
\overline{x_2} \\
x_2 \\
\end{array}
\begin{array}{c}
\overline{x_3} \\
x_3 \\
\end{array}
\]

\[
x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{0, 1\}
\]

\[
x_2 \in \{0, 1\}, x_3 \in \{0, 1\}
\]

\[
x_3 \in \{1\}
\]
Decision Diagram Compilation

- State: variable domains
- Transition: propagate decision

Theorem: If root state is domain consistent, then this approach yields a reduced exact DD
Decision Diagram Compilation

• State: variable domains
• Transition: propagate decision

• Theorem: If root state is domain consistent, then this approach yields a reduced exact DD
Decision Diagram Compilation

- State: variable domains
- Transition: propagate decision

\[ x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{\emptyset, 1\} \]

- Theorem: If root state is domain consistent, then this approach yields a reduced exact DD
Decision Diagram Compilation

- State: variable domains
- Transition: propagate decision

- Theorem: If root state is domain consistent, then this approach yields a reduced exact DD
Stronger DD relaxation via Lagrangian
Stronger DD relaxation via Lagrangian

Original IP model

\[
\max \ c^T x
\]

\[
Fx \leq f \quad \leftarrow \text{Structured constraints for DD}
\]

\[
Ax \leq b \quad \leftarrow \text{Any set of linear constraints}
\]

\[
x \in \mathbb{Z}^n, \ \ell \leq x \leq u
\]
Stronger DD relaxation via Lagrangian

Original IP model

\[
\begin{align*}
\max & \quad c^T x \\
\text{s.t.} & \quad Fx \leq f \\
& \quad Ax \leq b \\
& \quad x \in \mathbb{Z}^n, \quad \ell \leq x \leq u
\end{align*}
\]

Structured constraints for DD

Any set of linear constraints

Lagrangian model

\[
\begin{align*}
\min_{\lambda \geq 0} \max & \quad c^T x + \lambda^T (b - Ax) \\
\text{s.t.} & \quad Fx \leq f \\
& \quad x \in \mathbb{Z}^n, \quad \ell \leq x \leq u
\end{align*}
\]
Stronger DD relaxation via Lagrangian

Original IP model

\[
\max \ c^T x \\
F x \leq f \quad \leftarrow \text{Structured constraints for DD} \\
A x \leq b \quad \leftarrow \text{Any set of linear constraints} \\
x \in \mathbb{Z}^n, \ell \leq x \leq u
\]

Lagrangian model

\[
\min_{\lambda \geq 0} \max \ c^T x + \lambda^T (b - Ax) \\
F x \leq f \\
x \in \mathbb{Z}^n, \ell \leq x \leq u
\]

Lagrangian subproblem is longest path in DD (efficient)
Stronger DD relaxation via Propagation

- Propagate linear constraints
- Additional state information
  - variable domains
  - constraint right-hand sides

\[
3x_1 + x_2 + 2x_3 \leq 4
\]
\[
x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{0, 1\}
\]
\[
x_1 = 0
\]
\[
x_1 = 1
\]
Stronger DD relaxation via Propagation

- Propagate linear constraints
- Additional state information
  - variable domains
  - constraint right-hand sides

\[ 3x_1 + x_2 + 2x_3 \leq 4 \]
\[ x_1 \in \{0, 1\}, \ x_2 \in \{0, 1\}, \ x_3 \in \{0, 1\} \]

\[ x_2 + 2x_3 \leq 4 \]
\[ x_2 \in \{0, 1\}, \ x_3 \in \{0, 1\} \]
Stronger DD relaxation via Propagation

- Propagate linear constraints
- Additional state information
  - variable domains
  - constraint right-hand sides

\[ 3x_1 + x_2 + 2x_3 \leq 4 \]
\[ x_1 \in \{0, 1\}, x_2 \in \{0, 1\}, x_3 \in \{0, 1\} \]

\[ x_1 = 0 \]
\[ x_1 = 1 \]

\[ x_2 + 2x_3 \leq 4 \]
\[ x_2 \in \{0, 1\}, x_3 \in \{0, 1\} \]

\[ x_2 + 2x_3 \leq 1 \]
\[ x_2 \in \{0, 1\}, x_3 \in \{0\} \]
Experimental evaluation

• Experimental setup
  – Independent set problem on random graphs (Watts-Strogatz)
  – Add set of random knapsack constraints $\sum_{i \in S} a_i x_i \leq b$
  – Vary number of variables $n$
  – Vary number of knapsack constraints $m$
Experimental evaluation

• Experimental setup
  – Independent set problem on random graphs (Watts-Strogatz)
  – Add set of random knapsack constraints \( \sum_{i \in S} a_i x_i \leq b \)
  – Vary number of variables \( n \)
  – Vary number of knapsack constraints \( m \)

• Implemented in SCIP 5.0.1
  – Only IP model is given to solver
  – DD compiled automatically
Random Graphs + Knapsack Constraints

On average: 65.5% node reduction
1.59x speedup

$n = 300, 350, 400, 450$
$m = 0.1 n$
Random Graphs + Knapsack Constraints

$n = 300, 350, 400, 450$
$m = 0.1n$

On average: 65.5% node reduction
1.59x speedup
Deriving Cutting Planes from Decision Diagrams

LP relaxation

max

cut

facet-defining cut
Related work:

• Becker et al. [2005], Behle [2007]: Lagrangian cut generation using exact decision diagrams
• Buchheim et al. [2008]: Target cuts
Cut-Generating Linear Program
Cut-Generating Linear Program

\[ \begin{align*}
\text{Cut} & = \text{Generating Linear Program} \\
\end{align*} \]
Cut-Generating Linear Program
Cut-Generating Linear Program

\[ \begin{align*}
  x_1 & - x_2 - 1 = 0 \\
  x_1 - x_2 - 2 = 0 \\
  s_1 & = 0 \\
  s_2 & = 0 \\
  s_3 & = 0 \\
  x^* & = 0
\end{align*} \]
Cut-Generating Linear Program

Target cut
Cut-Generating Linear Program

\[
\begin{align*}
\text{min } & y_1 + y_2 \\
\text{s.t. } & -y_1 + 2y_2 = 2 \\
& -2y_3 + y_4 + y_5 = -1 \\
& \text{flow conservation}
\end{align*}
\]
Cut-Generating Linear Program

\[
\begin{align*}
\text{min } & \quad y_1 + y_2 \\
\text{s.t. } & \quad -y_1 + 2y_2 = 2 \\
& \quad -2y_3 + y_4 + y_5 = -1 \\
& \quad + \text{ flow conservation}
\end{align*}
\]

Solution:
\[y_1 = y_3 = 4/3, \quad y_2 = y_5 = 5/3, \quad y_4 = 0\]
Cut-Generating Linear Program

- **Solution methods**
  - solve CGLP as LP (facet defining cuts)
  - or use subgradient method (iteratively finds longest path in DD)  

  ![Target cut diagram](image)

  \[
  \begin{align*}
  \text{min } & \quad y_1 + y_2 \\
  \text{s.t. } & \quad -y_1 + 2y_2 = 2 \\
  & \quad -2y_3 + y_4 + y_5 = -1 \\
  & \quad \text{flow conservation}
  \end{align*}
  \]

  **Solution:**
  \[y_1=y_3=4/3, \quad y_2=y_5=5/3, \quad y_4=0\]  

  [Tjandraatmadja & vH, IJOC 2019]
  [Davarnia & vH]
Outer Approximation Scheme for MINLP

• Solve Integer Linear Programming relaxation: $x^*$
• For all constraints that are violated by $x^*$: add linearization cut
• Repeat until $x^*$ is feasible

• Requires that all functions are convex and sufficiently smooth (continuously differentiable)

[Westerlund & Pettersson, 1995]
[Duran and Grossmann, 1986]
Outer Approximation with DDs

• Generate a DD (relaxed or exact) for each individual constraint
  – Done once in pre-processing phase
Outer Approximation with DDs

- Generate a DD (relaxed or exact) for each individual constraint
  - Done once in pre-processing phase

- Outer Approximation with DD cuts:
  - Solve Integer Linear Programming relaxation: \( x^* \)
  - For all constraints that are violated by \( x^* \): add DD cut
  - Repeat until \( x^* \) is feasible
Outer Approximation with DDs

• Generate a DD (relaxed or exact) for each individual constraint
  – Done once in pre-processing phase

• Outer Approximation with DD cuts:
  – Solve Integer Linear Programming relaxation: $x^*$
  – For all constraints that are violated by $x^*$: add DD cut
  – Repeat until $x^*$ is feasible

• Requires that all functions are factorable
  – Can be non-convex
Outer Approximation Example

Linearization cut

DD cut
Experimental Evaluation: Polynomial Knapsack

\[
\begin{align*}
\max & \quad \sum_{i=1}^{n} c_i x_i \\
\text{s.t.} & \quad \sum_{i=1}^{n} a_{ij} x_i^{k_{ij}} \leq b_j \quad \forall j \in J \\
x & \in [l, u] \cap \mathbb{Z}^n
\end{align*}
\]

n=500, |J| = 5, bounds [0,5]
degree k of monomial in \{1,\ldots,10\}

5 randomly generated instances

maximum DD width is 3000

time limit is 300s

Gap closure for various outer approximation methods
Experimental Evaluation: Penetration Pricing

Find discrete prices for n products subject to minimum revenue constraints

\[
\begin{align*}
\min & \quad \sum_{i=1}^{n} c_i x_i \\
\text{s.t.} & \quad \sum_{i=1}^{n} a_{ij}^j x_i e^{-x_i^{k_i}} \geq b_j, \quad \forall j \in J \\
& \quad x \in [l, u] \cap \mathbb{Z}^n.
\end{align*}
\]

Find discrete prices for n products subject to minimum revenue constraints

|J| = 5, prices \{0, 0.1, ..., 1.0\}
degree k of monomial in \{1, 2, 3\}

maximum DD width is 5000
time limit is 300s

Gap closure for various sizes and MINLP solvers
Conclusion

• Decision Diagrams can be applied to Integer Programming
• Incorporate DD bounds in MIP search
  – conflict graph represented as DD
  – strengthened by Lagrangian relaxation and constraint propagation
  – up to 65.5% node reduction (1.59x speedup)
• Outer approximation for MINLP
  – applies to non-convex factorable functions
  – can outperform state-of-the-art approaches on certain problem classes


http://www.andrew.cmu.edu/user/vanhoeve/mdd/