Multistage and Multilevel Discrete Optimization: Algorithms and Complexity

Ted Ralphs[1]

Joint work with Aykut Bulut[1], Scott DeNegre[3], Menal Güzelsoy[2], Anahita Hassanzadeh[1]

[1]COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University
[2]SAS Institute, Advanced Analytics, Operations Research R & D
[3]The Chartis Group

Oakland University, November 11, 2013

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Our Starting Point

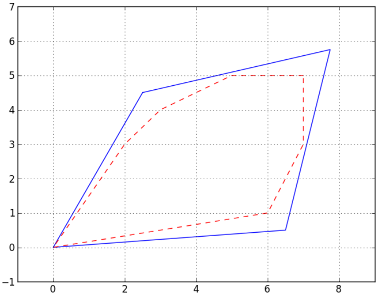
- We start by considering the following classical optimization problem.

Mixed Integer Linear Optimization Problem

$$\min\{c^\top x \mid x \in \mathcal{P} \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})\}, \tag{MILP}$$

where $\mathcal{P} = \{x \in \mathbb{R}^n_+ \mid Ax = b\}$, $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$.

- This optimization problem involves determining a (set of) decision(s) to be made *simultaneously* by a *single* decision-maker (i.e., with a *single* objective).



Generalizing

- Decision problems arising in sequential games and other real-world applications involve
  - multiple, independent decision-makers (DMs),
  - sequential/multi-stage decision processes, and/or
  - multiple, possibly conflicting objectives.
- Modeling frameworks
  - Multiobjective Optimization $\Leftarrow$ multiple objectives, single DM
  - Mathematical Optimization with Recourse $\Leftarrow$ multiple stages, single DM
  - Multilevel Optimization $\Leftarrow$ multiple stages, multiple objectives, multiple DMs
- *Multilevel optimization* generalizes standard mathematical optimization by modeling hierarchical decision problems, such as finite extensive-form games.
- Such models arises in a remarkably wide array of applications.

Game Theory

- Multilevel and multistage optimization is a tool for analyzing certain *finite extensive-form games*, sequential games involving n players.

Loose Definition

- The game is specified on a tree with each node corresponding to a move and the outgoing arcs specifying possible choices.
- The leaves of the tree have associated payoffs.
- Each player's goal is to maximize payoff.
- There may be *chance* players who play randomly according to a probability distribution and do not have payoffs (*stochastic games*).

- All players are rational and have perfect information.
- The problem faced by a player in determining the next move is a *multilevel/multistage* optimization problem.
- The move must be determined by taking into account the *responses of the other players*.

Multilevel and Multistage Games

- We use the term *multilevel* for competitive games in which there is no chance player.
- We use the term *multistage* for cooperative games in which all players receive the same payoff, but there are chance players.
- A *subgame* is the part of a game that remains after some moves have been made.

Stackelberg Game

- A Stackelberg game is a game with two players who make one move each.
- The goal is to find a *subgame perfect Nash equilibrium*, i.e., the move by each player that ensures that player's best outcome.

Recourse Game

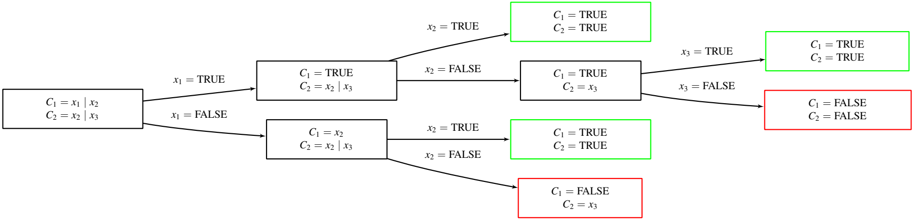
- A cooperative game in which play alternates between cooperating players and chance players.
- The goal is to find a *subgame perfect Markov equilibrium*, i.e., the move that ensures the best outcome in a probabilistic sense.

Canonical Example

- A canonical extensive-form game that illustrates many of the basic principles is the *k-player satisfiability game*.
  - k players determine the value of a set of Boolean variables with each in control of a specific subset.
  - In round i, player i determines the values of her variables.
  - Each player tries to choose values that force a certain end result, given that subsequent players may be trying to achieve the opposite result.
- Examples
  - k = 1: SAT
  - k = 2: The first player tries to choose values such that any choice by the second player will result in satisfaction.
  - k = 3: The first player tries to choose values such that the second player cannot choose values that will leave the third player without the ability to find satisfying values.
- Note that the odd players and the even players are essentially "working together" and the same game can be described with only two players.

Illustrating the Example

- This diagram illustrates the search for solutions to the problem as a tree.
- The nodes in green represent settings of the truth values that satisfy all the given clauses; red represents non-satisfying truth values.
  - With one player, the solution is any path to one of the green nodes.
  - With two players, the solution is a subtree in which there are no red nodes.
- The latter requires knowledge of *all* leaf nodes (important!).



Brief Overview of Practical Applications

- Hierarchical decision systems
  - Government agencies
  - Large corporations with multiple subsidiaries
  - Markets with a single "market-maker."
  - Decision problems with recourse
- Parties in direct conflict
  - Zero sum games
  - Interdiction problems
- Modeling "robustness": Chance player is external phenomena that cannot be controlled.
  - Weather
  - External market conditions
- Controlling optimized systems: One of the players is a system that is optimized by its nature.
  - Electrical networks
  - Biological systems

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Bilevel (Integer) Linear Optimization

Formally, a *bilevel linear optimization problem* is described as follows.

- $x \in X \subseteq \mathbb{R}^{n_1}$ are the *upper-level variables*
- $y \in Y \subseteq \mathbb{R}^{n_2}$ are the *lower-level variables*

Bilevel (Integer) Linear Optimization Problem

$$\max \left\{c^1 x + d^1 y \mid x \in \mathcal{P}_U \cap X, y \in \operatorname{argmin}\{d^2 y \mid y \in \mathcal{P}_L(x) \cap Y\}\right\} \quad \text{(MIBLP)}$$

The *upper-* and *lower-level feasible regions* are:

$$\mathcal{P}_U = \left\{x \in \mathbb{R}_+ \mid A^1 x \leq b^1\right\} \text{ and}$$
$$\mathcal{P}_L(x) = \left\{y \in \mathbb{R}_+ \mid G^2 y \geq b^2 - A^2 x\right\}.$$

We consider the general case in which $X = \mathbb{Z}^{p_1} \times \mathbb{R}^{n_1 - p_1}$ and $Y = \mathbb{Z}^{p_2} \times \mathbb{R}^{n_2 - p_2}$.

Continuous Second Stage

- In general, if $Y = \mathbb{R}^{n_1}$, then the lower-level problem can be replaced with its optimality conditions.
- The optimality conditions for the lower-level optimization problem are

$$
\begin{aligned}
G^2 y &\geq b^2 - A^2 x \\
u G^2 &\leq d^2 \\
u(b^2 - G^2 - A^2 x) &= 0 \\
(d^2 - u G^2) y &= 0 \\
u, y &\in \mathbb{R}_+
\end{aligned}
$$

- When $X = \mathbb{R}^{n_1}$, this is a special case of a class of non-linear mathematical optimization problem known as *mathematical optimization problems with equilibrium constraints* (MPECs).
- An MPEC can be solved in a number of ways, including converting it to a standard discrete optimization problem.
- Note that in this case, the value function of the lower-level problem is piecewise linear, but not necessarily convex.

Some Special Cases

- Pure integer.
- Positive constraint matrix at lower level.
- Binary variables at the upper and/or lower level.
- *Interdiction problems*.

Mixed Integer Interdiction

$$\max_{x \in \mathcal{P}_U^I} \min_{y \in \mathcal{P}_L^I(x)} dy \tag{MIPINT}$$

where

$$\mathcal{P}_U^I = \left\{x \in \mathbb{B}^n \mid A^1 x \leq b^1\right\}$$
$$\mathcal{P}_L^I(x) = \left\{y \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \mid G^2 y \geq b^2, y \leq u(e-x)\right\}.$$

- The case where follower's problem has network structure is called the *network interdiction problem* and has been well-studied.
- The model above allows for lower-level systems described by general MILPs.

Recourse Problems

- If $d^1 = -d^2$, we call this as a *mathematical optimization problem with recourse*.
- We will focus primarily on this special case for the remainder of this section, but the general principles apply more broadly.
- We can reformulate the bilevel optimization problem as follows.

$$\min\{-c^1 x + Q(x) \mid x \in \mathcal{P}_U \cap X\}, \tag{1}$$

where

$$Q(x) = \min\{d^1 y \mid y \in \mathcal{P}_L(x) \cap Y\}. \tag{2}$$

- The function Q is known as the *value function* of the recourse problem and constructing/approximating it is the key to solving these problems.

Two-Stage Stochastic Optimization Problem with Recourse

- Two-stage stochastic mixed integer optimization problems form a special class of recourse problems in which the objective is the mean of second-stage values over a given probability distribution (note the shift in notation).

$$\min\{c^\top x + \mathbb{E}_\xi Q_\xi(x) \mid x \in X\}, \tag{3}$$

  where

$$Q_\xi(x) = \min\{q^\top y \mid y \in Y, Wy = h_\xi - Tx\}, \tag{4}$$

  ξ is a random variable from a probability space $(\Xi, \mathscr{F}, \mathcal{P})$, and for each $\xi \in \Xi$, $h_\xi \in \mathbb{R}^{m_2}$.

- If the distribution of ξ is discrete and has finite support, then (3) is a bilevel optimization problem.

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Solving Discrete Optimization Problems

- In general, *convex optimization problems* are “easy” to solve.
- In essence, this is because convex problems have only one local minimum.
- Discrete optimization problems are particularly challenging

  - the feasible region is nonconvex and
  - the description of the feasible region, though compact, is implicit.

- More computationally useful descriptions of the feasible set can be obtained by

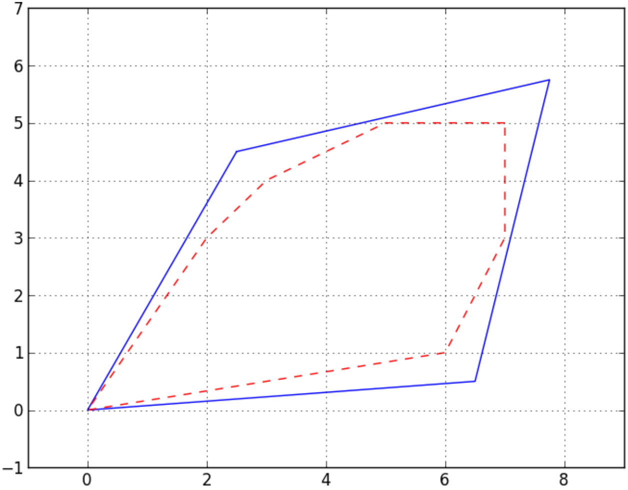
  - Convexification $\Rightarrow$ iteratively construct an explicit description of the convex hull of feasible solutions (cutting plane method)
  - Disjunction $\Rightarrow$ using logical disjunctions to represent the feasible set of as a finite union of polyhedra (branch and bound)

- In general, both of these approaches lead to descriptions of exponential size.
- In practice, we can dynamically generate only relevant parts of the description.

Discrete Optimization

$$z_{\mathrm{LP}} = \min_{x \in \mathbb{R}^n_+} \left\{ c^\top x \mid Ax \geq b \right\} \qquad \text{(LP)}$$

$$z_{\mathrm{IP}} = \min_{x \in \mathbb{Z}^n_+} \left\{ c^\top x \mid Ax \geq b \right\} \qquad \text{(MIP)}$$



Branch and Bound

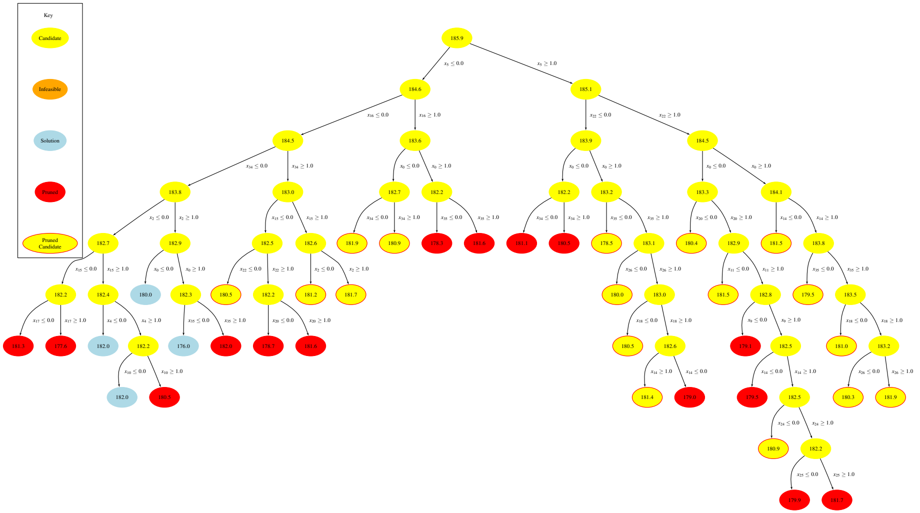
- A *relaxation* of an ILP is an auxiliary mathematical optimization problem for which
  - the feasible region contains the feasible region for the original ILP, and
  - the objective function value of each solution to the original ILP is not increased.
- Relaxations can be used to efficiently derive bounds on the optimal value.
  - Convex/Continuous relaxations
  - Combinatorial relaxations
  - Lagrangian relaxations

Branch and Bound

Put the root subproblem into the queue and while the queue is nonempty, do

1. Remove a subproblem and solve its relaxation.
2. The relaxation is infeasible $\Rightarrow$ subproblem is infeasible.
3. Solution is feasible for the MILP $\Rightarrow$ subproblem solved.
4. Solution is not feasible for the MILP $\Rightarrow$ lower bound.
   - If the lower bound exceeds the global upper bound, we can *prune the node*.
   - Otherwise, we *branch* and add the resulting subproblems to the queue.

Branch and Bound



Outline

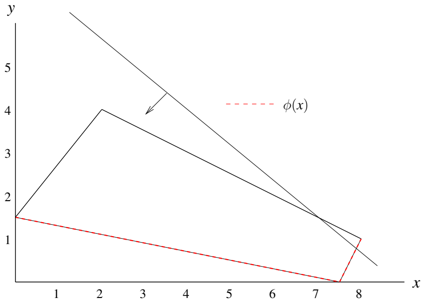
1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Benders' Principle (Linear Optimization)

$$z_{\rm LP} = \min_{(x,y)\in\mathbb{R}^n} \left\{ c'x + c''y \;\middle|\; A'x + A''y \geq b \right\}$$
$$= \min_{x\in\mathbb{R}^{n'}} \left\{ c'x + \phi(b - A'x) \right\},$$

where

$$\phi(d) = \min c''y \quad \text{s.t. } A''y \geq d, \quad y \in \mathbb{R}^{n''}$$

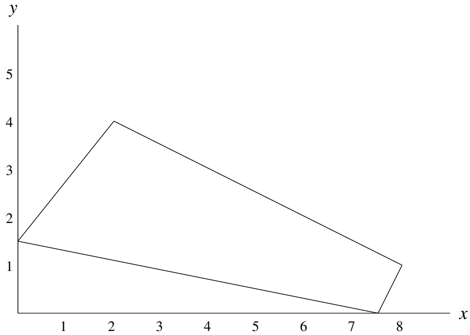


Basic Strategy:

- The function ϕ is the *value function* of a linear optimization problem.
- The value function is piecewise linear and convex.
- We iteratively generate a lower approximation by sampling the domain.

Example

$$
\begin{aligned}
z_{LP} \quad = \quad \min \quad & x + y \\
\text{s.t.} \quad & 25x - 20y \geq -30 \\
& -x - 2y \geq -10 \\
& -2x + y \geq -15 \\
& 2x + 10y \geq 15 \\
& x, y \in \mathbb{R}
\end{aligned}
$$

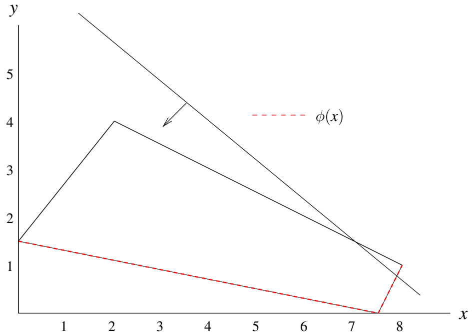


Value Function Reformulation

$$z_{LP} = \min_{x \in \mathbb{R}} x + \phi(x),$$

where

$$\begin{aligned} \phi(x) = \min \quad & y \\ \text{s.t.} \quad & -20y \geq -30 - 25x \\ & -2y \geq -10 + x \\ & y \geq -15 + 2x \\ & 10y \geq 15 - 2x \\ & y \in \mathbb{R} \end{aligned}$$

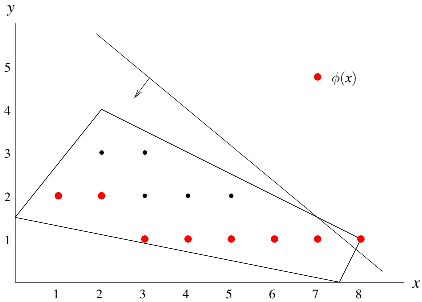


Benders' Principle (Discrete Optimization)

$$z_{\text{IP}} = \min_{(x,y)\in\mathbb{Z}^n} \left\{ c'x + c''y \;\middle|\; A'x + A''y \geq b \right\}$$
$$= \min_{x\in\mathbb{R}^{n'}} \left\{ c'x + \phi(b - A'x) \right\},$$

where

$$\phi(d) = \min c''y$$
$$\text{s.t. } A''y \geq d$$
$$y \in \mathbb{Z}^{n''}$$

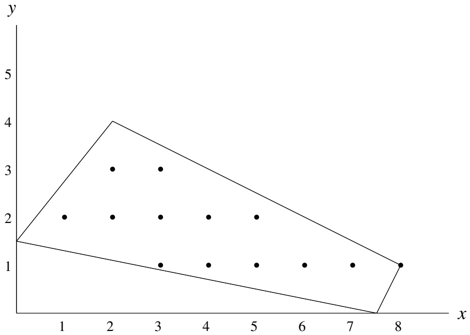


Basic Strategy:

- Here, ϕ is the value function of an *discrete optimization problem*.
- In the general case, the function ϕ is piecewise linear but not convex.
- Here, we also iteratively generate a lower approximation by evaluating ϕ.

Example

$$
\begin{aligned}
z_{IP} \quad = \quad \min \quad & x + y \\
\text{s.t.} \quad 25x - 20y &\geq -30 \\
-x - 2y &\geq -10 \\
-2x + y &\geq -15 \\
2x + 10y &\geq 15 \\
x, y &\in \mathbb{Z}
\end{aligned}
$$

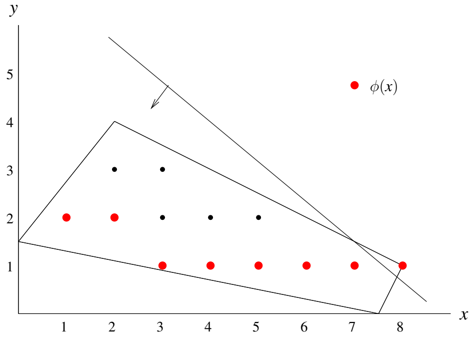


Value Function Reformulation

$$z_{IP} = \min_{x \in \mathbb{Z}} x + \phi(x),$$

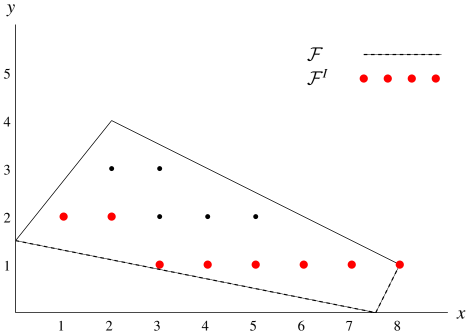
where

$$\begin{aligned} \phi(x) = \min \quad & y \\ \text{s.t.} \quad & -20y \geq -30 - 25x \\ & -2y \geq -10 + x \\ & y \geq -15 + 2x \\ & 10y \geq 15 - 2x \\ & y \in \mathbb{Z} \end{aligned}$$



Value Function Reformulation (General Case)

More generally, we can reformulate (MIBLP) as



$$
\begin{aligned}
\max \quad & c^1 x + d^1 y \\
\text{subject to} \quad & A^1 x \le b^1 \\
& G^2 y \ge b^2 - A^2 x \\
& d^2 y = z_{LL}(b^2 - A^2 x) \\
& x \in X, y \in Y,
\end{aligned}
$$

where z_{LL} is the value function of the lower-level problem.

- This is, in principle, a standard mathematical optimization problem.
- Note that relaxing integrality does not yield a valid bound on the optimum, as in the single-level case.
- Relaxing integrality effectively replaces z_{LL} with the value function of the LP relaxation.

Two-Stage Stochastic Optimization Problems with Recourse

- For most of the remainder of the talk, we consider the two-stage stochastic mixed integer optimization problem (note the shift in notation)

$$\min\{c^\top x + \mathbb{E}_\xi Q_\xi(x) \mid x \in X\}, \tag{5}$$

  where

$$Q_\xi(x) = \min\{q^\top y \mid y \in Y, Wy = h_\xi - Tx\}, \tag{6}$$

  ξ is a random variable from a probability space $(\Xi, \mathscr{F}, \mathcal{P})$, and for each $\xi \in \Xi$, $h_\xi \in \mathbb{R}^{m_2}$.
- If the distribution of ξ is discrete and has finite support, then (3) is a bilevel optimization problem.

Value Function Reformulation of the Two-Stage Problem

Let us assume the following.

- The probability space Ξ is finite.
- q, T, and W are fixed.
- The dual of the LP relaxation of the recourse problem is feasible, i.e.,

$$\{\nu \in \mathbb{R}^{m_2} \mid W_I^\top \nu \leq q_I, W_C^\top \nu \leq q_C\} \neq \emptyset.$$

- X is non-empty and bounded.

Let

- $\mathcal{S}(\beta) = \{y \in Y \mid Wy = \beta\}$
- $\phi(\beta) = \{q^\top y \mid y \in \mathcal{S}(\beta)\}$

Then our problem is to determine $\min_{x \in X} c^\top x + \mathbb{E}_\xi[\phi(h_\xi - Tx)]$. Here, ϕ is the value function of the second-stage problem.

Generalized Benders' Algorithm for the Two-stage Recourse Problem

Step 0. Initialize

a) Let $x^0 \in \text{argmin}\{c^\top x \mid x \in X\}$ and $\gamma^0 = Tx^0$
b) Set $\mathcal{F} = \emptyset$, $\underline{\phi}^0 = -\infty$, and $k = 1$.

Step 1. Update the lower approximation and solve master problem

a) Construct functions f_ξ^k dual to ϕ and strong at $h_\xi - Tx^{k-1}$ for $\xi \in \Xi$.
b) Set $\mathcal{F} = \mathcal{F} \cup \left(\cup_{\xi \in \Xi} f_\xi^k\right)$ and $\underline{\phi}^k = \max_{f \in \mathcal{F}} f$
c) Determine $x^k \in \text{argmin}_{x \in X}\{c^\top x + \mathbb{E}_\xi[\underline{\phi}^k(h_\xi - Tx)]\}$

Step 2. Check termination condition

a) If $\min_{x \in X}\{c^\top x + \mathbb{E}_\xi[\underline{\phi}^k(h_\xi - Tx)]\} = c^\top x^k + \mathbb{E}_\xi[\underline{\phi}^k(h_\xi - Tx^k)]$ then stop, x^k is an optimal solution.
b) Set k = k+1 and go to Step 1.

Related Algorithms

The algorithmic framework we utilize builds on a number of previous works.

- Modification to the L-shaped framework (**???**)
  - Linear cuts in first stage for binary first stage
  - Optimality cuts from B&B and cutting plane, applied to pure integer second stage
  - Disjunctive programming approaches and cuts in the second stage
- Value function approaches: Pure integer case (**??**)
- Scenario decomposition (**?**)
- Enumeration/Gröbner basis reduction (**?**)

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Approximating the Value Function

- In general, it is difficult to construct the value function explicitly.
- We therefore propose to approximate the value function by either upper or lower bounding functions

Lower bounds

Derived by considering the value function of *relaxations* of the original problem or by constructing *dual functions* ⇒ Relax constraints.

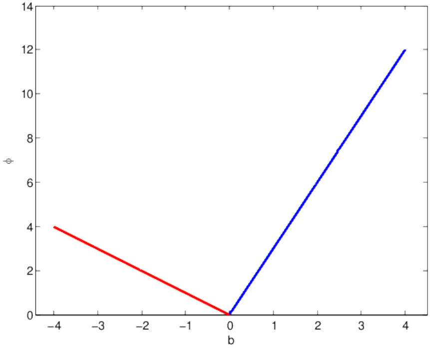
Upper bounds

Derived by considering the value function of *restrictions* of the original problem ⇒ Fix variables.

LP Value Function

Example

$$
\begin{aligned}
\phi_{LP}(b) = \min\ & 6x_1 + 7x_2 + 5x_3 \\
\text{s.t. } & 2x_1 - 7x_2 + x_3 = b \qquad \text{(Ex.LP)} \\
& x_1, x_2, x_3 \in \mathbb{R}_+
\end{aligned}
$$



LP Value Function Structure

$$\phi_{LP}(b) = \min c^\top x \quad \text{s.t. } Ax = b \quad x \in \mathbb{R}^n_+ \tag{LP}$$

- Assume the dual of (LP) is feasible.
- The epigraph of ϕ_{LP} is a convex cone, call it $\mathcal{L}$:

$$\mathcal{L} := cone\{(A_1, c_1), (A_2, c_2), \ldots, (A_n, c_n), (0, 1)\}$$

- Let $u_1, \ldots, u_k$ be extreme points of the feasible region of the dual of (LP) and $d_1, \ldots, d_p$ be its extreme directions. Then

$$\mathcal{L} := \{(b, z) : z \geq u_i^\top b, i = 1, \ldots, k, d_j^\top b \leq 0, j = 1, \ldots, p\}.$$

- Note that the value function has an underlying discrete structure.

MILP Value Function

Now we consider the MILP value function $\phi : \mathbb{R}^m \to \mathbb{R} \cup \{\pm\infty\}$

$$\begin{aligned} \phi(b) = \min\ & c^\top x \\ \text{s.t. } & Ax = b \\ & x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \end{aligned} \tag{MILP}$$

We define

- $\mathcal{S}(b) = \{x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \mid Ax = b\}$.
- $B = \{b \in \mathbb{R}^m \mid S(b) \neq \emptyset\}$.

Example: MILP Value Function

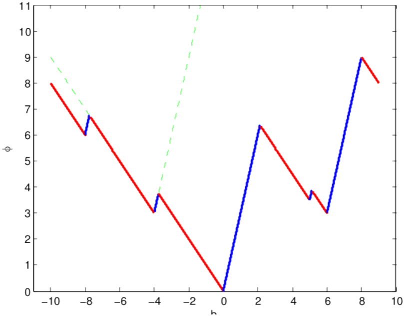
The value function of a MILP is non-convex and discontinuous piecewise polyhedral.

Example

$$\phi(d) = \min 3x_1 + \frac{7}{2}x_2 + 3x_3 + 6x_4 + 7x_5 + 5x_6$$

$$\text{s.t. } 6x_1 + 5x_2 - 4x_3 + 2x_4 - 7x_5 + x_6 = d$$

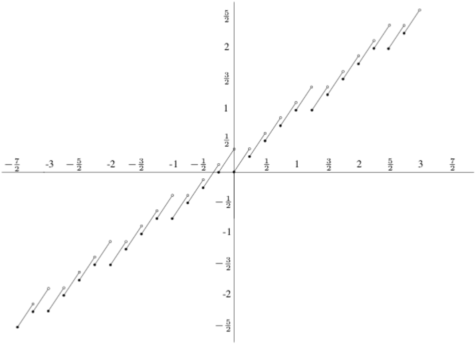
$$x_1, x_2, x_3 \in \mathbb{Z}_+,\ x_4, x_5, x_6 \in \mathbb{R}_+$$



Example: MILP Value Function

Example

$$\phi(b) = \min x_1 - \frac{3}{4}x_2 + \frac{3}{4}x_3$$
$$\text{s.t. } \frac{5}{4}x_1 - x_2 + \frac{1}{2}x_3 = b \qquad \text{(Ex2.MILP)}$$
$$x_1, x_2 \in \mathbb{Z}_+,\ x_3 \in \mathbb{R}_+$$



Continuous and Integer Restriction of an MILP

Consider

$$\begin{aligned} \phi(b) = \min\, & c_I^\top x_I + c_C^\top x_C \\ \text{s.t. } & A_I x_I + A_C x_C = b, \\ & x \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} \end{aligned} \tag{MILP}$$

Define the *continuous restriction* of (MILP) as

$$\begin{aligned} \phi_C(b) = \min\, & c_C^\top x_C \\ \text{s.t. } & A_C x_C = b, \\ & x \in \mathbb{R}_+^{n-r} \end{aligned} \tag{CR}$$

and its *integer restriction* as

$$\begin{aligned} \phi_I(b) = \min\, & c_I^\top x_I \\ \text{s.t. } & A_I x_I = b \\ & x_I \in \mathbb{Z}_+^r \end{aligned} \tag{IR}$$

Discrete Representation of the Value Function

For $b \in \mathbb{R}^m$, we have that

$$\begin{aligned} \phi(b) = \min\ & c_I^\top x_I + \phi_C(b - A_I x_I) \\ \text{s.t. } & x_I \in \mathbb{Z}_+^r \end{aligned} \tag{7}$$

- From this we see that the value function is comprised of the minimum of a set of translations of ϕ_C.
- The set of translations, along with ϕ_C describe the value function exactly.
- For $\hat{x}_I \in \mathbb{Z}_+^r$, let

$$\phi_C(b, \hat{x}_I) = \phi_C(b - A_I \hat{x}_I) + c_I^\top x_I \ \ \forall b \in \mathbb{R}^m.$$

- Then we have that $\phi(b) = \min_{x_I \in \mathbb{Z}_+^r} \phi_C(b, \hat{x}_I)$.

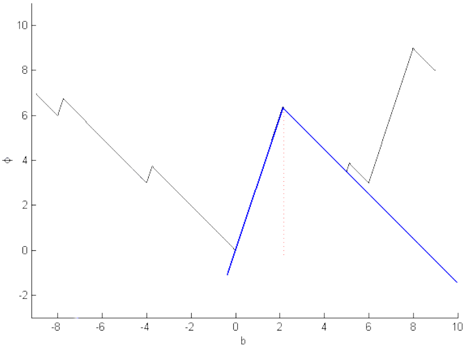
Bounding the Value Function From Below

A *dual function* $\varphi : \mathbb{R}^m \to \mathbb{R} \cup \{\pm\infty\}$ is and function such that

$$\varphi(b) \leq \phi(b) \quad \forall b \in \Lambda$$

For a particular value of $\hat{b}$, the dual problem is

$$\phi_D = \max\{\varphi(\hat{b}) : \varphi(b) \leq \phi(b) \quad \forall b \in \mathbb{R}^m, \; \varphi : \mathbb{R}^m \to \mathbb{R} \cup \{\pm\infty\}\}$$



MILP Duals from Branch-and-Bound

Let T be set of the terminating nodes of the tree. Then in a terminating node $t \in T$ we solve:

$$\begin{aligned} \min\ & c^\top x \\ \text{s.t. } & Ax = b, \\ & l^t \leq x \leq u^t, x \geq 0 \end{aligned} \tag{8}$$

The dual at node t:

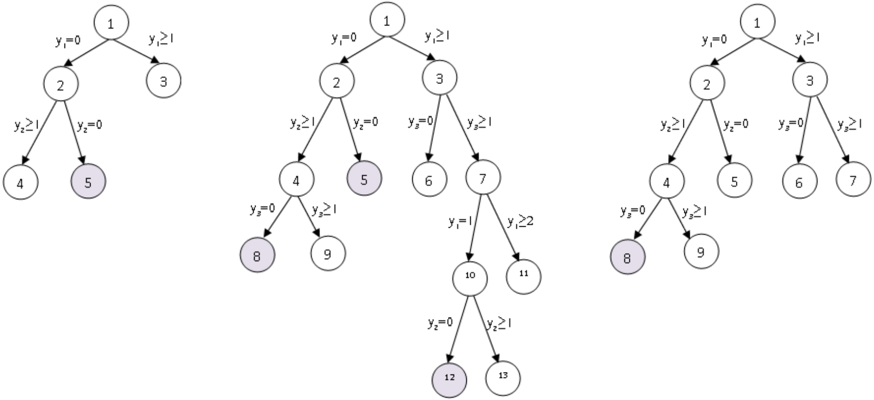
$$\begin{aligned} \max\ & \{\pi^t b + \underline{\pi}^t l^t + \bar{\pi}^t u^t\} \\ \text{s.t. } & \pi^t A + \underline{\pi}^t + \bar{\pi}^t \leq c^\top \\ & \underline{\pi} \geq 0, \bar{\pi} \leq 0 \end{aligned} \tag{9}$$

We obtain the following strong dual function:

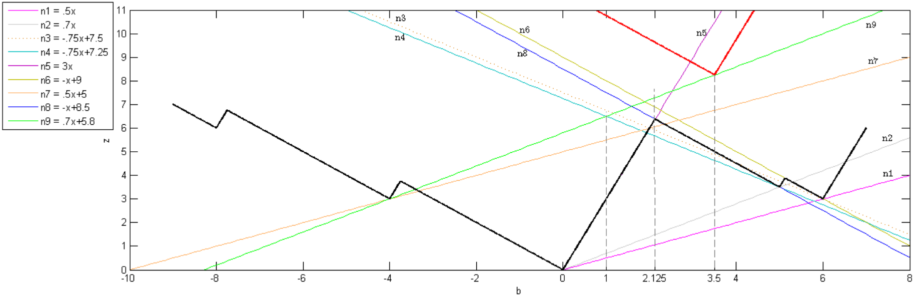
$$\min_{t \in T} \{\pi^t b + \underline{\pi}^t l^t + \bar{\pi}^t u^t\} \tag{10}$$

MILP Duals from Branch-and-Bound

Figure : Dual Functions from B&B for right hand sides 1, 2.125, 3.5



MILP Duals from Branch-and-Bound



Two-Stage Stochastic Optimization Problems with Recourse

- For most of the remainder of the talk, we consider the two-stage stochastic mixed integer optimization problem.

$$\min\{c^\top x + \mathbb{E}_\xi Q_\xi(x) \mid x \in X\}, \tag{11}$$

where

$$Q_\xi(x) = \min\{d^\top y \mid y \in Y, Wy = h_\xi - Tx\}, \tag{12}$$

ξ is a random variable from a probability space $(\Xi, \mathscr{F}, \mathcal{P})$, and for each $\xi \in \Xi$, $h_\xi \in \mathbb{R}^{m_2}$.

- If the distribution of ξ is discrete and has finite support, then (3) is a bilevel optimization problem.

Example

Consider

$$\begin{aligned} \min z(x) = \min \ & -3x_1 - 4x_2 + \sum_{s=1}^{2} 0.5\phi(h_s - \gamma) \\ \text{s.t. } & x_1 + x_2 \leq 5 \\ & x \in \mathbb{Z}_+ \\ & 2x_1 + \frac{1}{2}x_2 = \gamma \end{aligned} \tag{13}$$

where

$$\begin{aligned} \phi(\beta) = \min \ & 3y_1 + \frac{7}{2}y_2 + 3y_3 + 6y_4 + 7y_5 \\ \text{s.t. } & 6y_1 + 5y_2 - 4y_3 + 2y_4 - 7y_5 = \beta \\ & y_1, y_2, y_3 \in \mathbb{Z}_+, \ y_4, y_5 \in \mathbb{R}_+ \end{aligned} \tag{14}$$

with $h = [-4, 10]$.

Example

Iteration 0

- $\mathcal{F} = \emptyset$
- $\underline{\phi}^0 = -\infty$

Solve

$$\begin{aligned} \min\ & -3x_1 - 4x_2 \\ \text{s.t. } & x_1 + x_2 \leq 5 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{aligned}$$

to obtain $x^0 = [0, 5]$ and $\gamma^0 = 2.5$.

Example

Iteration 1

Step 1

- Solve the second stage problem for each scenario, e.g., evaluate $\phi(h_1 - \gamma^0) = \phi(-6.5)$ and $\phi(h_2 - \gamma^0) = \phi(7.5)$.
- The respective strong dual functions are

$$f_1^1(\beta) = \min\{-\beta - 1, 0.5\beta + 10\} \text{ and } f_2^1(\beta) = \min\{3\beta - 15, -0.75\beta + 14.5\}.$$

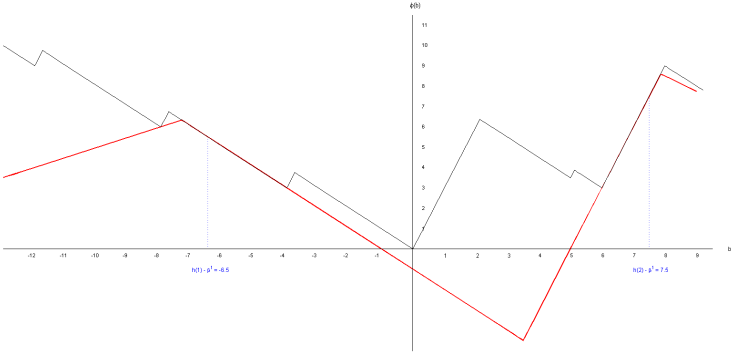
- Then, $\underline{\phi}^1(\beta) = \max\{f_1^1, f_2^1\}$.
- Solve the master problem

$$\begin{aligned} \min\ & -3x_1 - 4x_2 + 0.5(\underline{\phi}^1(-4-\gamma) + \underline{\phi}^1(10-\gamma)) \\ \text{s.t. } & x_1 + x_2 \leq 5 \\ & 2x_1 + \frac{1}{2}x_2 = \gamma \\ & x_1, x_2 \in \mathbb{Z}_+ \end{aligned}$$

- The solution to the master problem yields $\gamma^1 = 7$.

Step 2: Since $\phi(-6.5) + \phi(7.5) < \underline{\phi}^1(-6.5) + \underline{\phi}^1(7.5)$, we continue.

Example



Example

Iteration 2

Step 1

- Evaluate $\phi(-11)$ and $\phi(3)$ to obtain the dual functions

  $f_1^2(\beta) = \min\{-\beta - 2, 0.5\beta + 15\}$ and

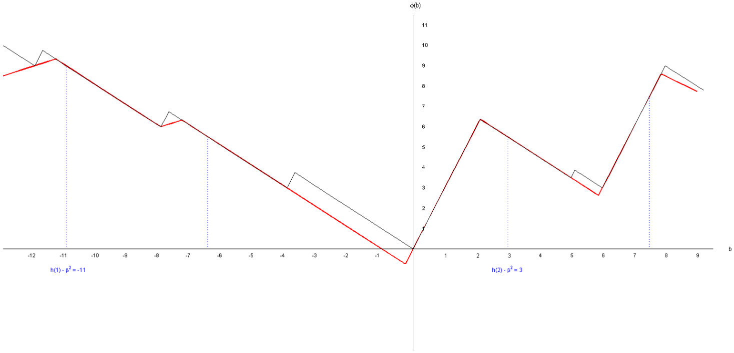
  $f_2^2(\beta) = \min\{3\beta, -\beta + 8.5, 0.7\beta + 5.8\}$.

- Update $\underline{\phi}^2(\beta) = \max\{f_1^1, f_2^1, f_1^2, f_2^2\}$.

- Solve the updated master problem to obtain $\gamma^2 = 4$.

Step 2: Since $\phi(-11) + \phi(3) < \underline{\phi}^2(-11) + \underline{\phi}^2(3)$, we continue.

Example



Example

Iteration 3

Step 1

- Evaluate $\phi(-8)$ and $\phi(6)$ to obtain the dual functions

  $f_1^3(\beta) = -0.75\beta$ and

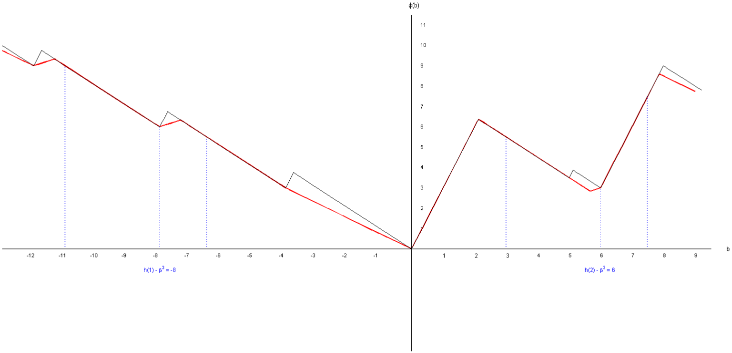
  $f_2^3(\beta) = 0.5\beta$.

- Update $\underline{\phi}^3(\beta) = \max\{f_1^1, f_2^1, f_1^2, f_2^2, f_1^3, f_2^3\}$.

- Solve the updated master problem to obtain $\gamma^3 = 4$.

Step 2: Since $\phi(-8) + \phi(6) = \underline{\phi}^3(-8) + \underline{\phi}^3(6)$, we stop.

Example



ϕ(b)
11
10
9
8
7
6
5
4
3
2
1
b
-12
-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
1
2
3
4
5
6
7
8
9
h(1) - β3 = -8
h(2) - β3 = 6

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Master Problem Formulation

Notation:

- $s, r \in \{1, \ldots, S\}$ where S is the number of scenarios
- $p \in \{1, \ldots, k\}$ where k is the iteration number
- $n \in \{1, \ldots, N(s)\}$ where $N(s)$ is the number of terminating nodes in the B&B tree solved for scenario s.
- $\theta_s = \mathcal{F}_s(\beta)$
- $t_{spr} = F_r^p(h(s) - \beta)$
- a_{prn}, ν_{prn} respectively, the dual vector and intercept obtained from node n of the B&B tree solved for scenario r in iteration p.
- p_s probability of scenario s
- $M > 0$ an appropriate large number

Master Problem Formulation

Solving the second stage problem with B&B, in Step 2, the following problem is solved to get β^{k+1} :

$$
\begin{aligned}
f^k = \min\ & c^\top x + \sum_{s=1}^{S} p_s \theta_s \\
\text{s.t. } & Tx = \beta \\
& \theta_s \geq t_{spr} && \forall s, p, r \\
& t_{spr} \leq a_{prn} + \nu_{prn}^\top (h(s) - \beta) && \forall s, r, p, n \\
& t_{spr} \geq a_{prn} + \nu_{prn}^\top (h(s) - \beta) - M u_{sprn} && \forall s, p, r, n \\
& \sum_{n=1}^{N} u_{sprn} = N(s) - 1 && \forall s, p, r \\
& x \in X, u_{sprn} \in \mathbb{B} && \forall s, p, r, n
\end{aligned}
\tag{master}
$$

Implementation Challenges

- To make the algorithm practical, several issues need to be solved.
- The master problem includes a piecewise linear function which grows in dimensions.
- In each iteration, for a scenario s, $S \times N(s)$ binary variables are added, where $N(s)$ is the number of new pieces of the function.
- Therefore, some "cut pool management" techniques need to be used to keep the size of the master problem manageable.
- This requires using an appropriate database.
- The examined right hand sides and their corresponding dual functions also need to be stored in an efficient manner.

Algorithms for General Bilevel Optimization Problems

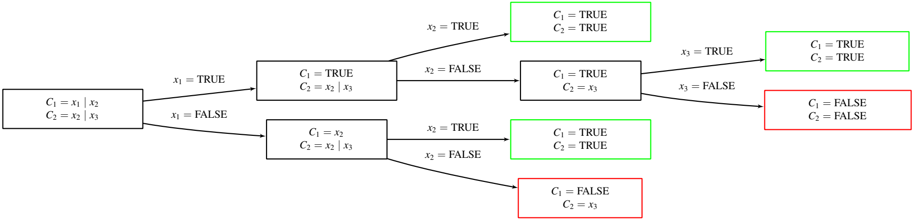
- The general case is much more difficult because we need the *solution* to the lower-level problem, not just the *value*.
- Algorithms must involve some kind of relaxation of the problem.
- Relaxations are inherently weak.
- Some progress has been made, but incorporating knowledge of the value function into the relaxation has proven exceptionally challenging.

Outline

1. Introduction
   - General Setting
   - Problem Classes
2. Algorithms
   - Branch and Bound
   - Benders' Technique
   - Approximating the Value Function
   - Implementation
3. Complexity
   - Canonical Example
   - Analysis
4. Final Remarks

Back to SAT

- Recall the earlier example of the satisfiability problem.
- The nodes in green represent settings of the truth values that satisfy all the given clauses; red represents non-satisfying truth values.
  - With one player, the solution is any path to one of the green nodes.
  - With two players, the solution is a subtree in which there are no red nodes.



More Formally

- More formally, we are given a Boolean formula with variables partitioned into k sets $X_1, \ldots, X_k$.
- For k odd, the SAT game can be formulated as

$$\exists X_1 \forall X_2 \exists X_3 \ldots ?X_k \tag{15}$$

- for even k, we have

$$\forall X_1 \exists X_2 \forall X_3 \ldots ?X_k \tag{16}$$

- A more general form of this problem, known as the *quantified Boolean formula problem* (QBF) allows an arbitrary sequence of quantifiers.

From SAT Game to Multilevel Optimization

- For $k = 1$, SAT can be formulated as the (feasibility) problem

$$\exists x \in \{0,1\}^n : \sum_{i \in C_j^0} x_i + \sum_{i \in C_j^1} (1 - x_i) \geq 1 \ \forall j \in J. \tag{SAT}$$

- (**??**) can be formulated as the optimization problem

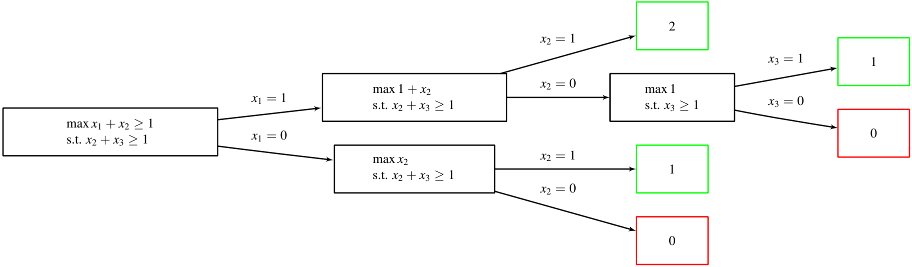
$$\begin{aligned} \max_{x \in \{0,1\}^n} & \sum_{i \in C_0^0} x_i + \sum_{i \in C_0^1} (1 - x_i) \\ \text{s.t.} & \sum_{i \in C_j^0} x_i + \sum_{i \in C_j^1} (1 - x_i) \geq 1 \ \forall j \in J \setminus \{0\} \end{aligned}$$

- For $k = 2$, we then have

$$\begin{aligned} \max_{x_{I_1} \in \{0,1\}^{I_1}} \min_{x_{I_2} \in \{0,1\}^{I_2}} & \sum_{i \in C_0^0} x_i + \sum_{i \in C_0^1} (1 - x_i) \\ \text{s.t.} & \sum_{i \in C_j^0} x_i + \sum_{i \in C_j^1} (1 - x_i) \geq 1 \ \forall j \in J \setminus \{0\} \end{aligned}$$

Branch and Bound for Optimization Version of SAT

- Consider the earlier example of the SAT game, now as an optimization problem.
- In the one player version, the goal is simply to maximize payoff.
- The two player game is zero-sum with the first player attempting to maximize while the second player attempts to minimize.
- The complexity of the two-player game comes from the requirement to account for the payoff at *all* leaf nodes.



Outline

1 Introduction
- General Setting
- Problem Classes

2 Algorithms
- Branch and Bound
- Benders' Technique
- Approximating the Value Function
- Implementation

3 Complexity
- Canonical Example
- Analysis

4 Final Remarks

How Difficult is the SAT Game?

- Fundamentally, we would like to know how difficult it is to solve player one's decision problem.
- It is well-known that the (single player) satisfiability problem is is in the complexity class *NP*-complete.
- It is perhaps to be expected that the k-player satisfiability game is in a different class.
  - The k^{th} player to move is faced with a satisfiability problem.
  - The $(k-1)^{th}$ player is faced with a 2-player subgame in which she must take into account the move of the k^{th} player.
  - And so on . . .
- Each player's decision problem appears to be exponentially more difficult than the succeeding player's problem.
- This complexity is captured formally in the hierarchy of complexity classes known as the *polynomial time hierarchy*.

Where Does the Complexity Come From?

- Notice that the size of the solution space for a multilevel optimization problem with n variables is *independent of* k!
- The complexity comes from the fact that we are forced to enumerate a larger portion of that solution space in order to prove optimality/feasibility.
- It is the size of the *certificate* that grows, not the size of the solution space.
- In multilevel optimization, the increased enumeration is forced by the competitive nature of the problem.
- In multistage optimization, the increased enumeration is forced by the uncertainty in the problem.

Complexity in Two-stage Stochastic Optimization

- In two-stage stochastic optimization, complexity grows with the number of scenarios.
- This is what forces us to examine more of the solution space than we would otherwise have to.
- We are tempted by existence of the extensive form to think that the size of the solution space grows.
- However, the extensive form merely explicitly accounts for the required additional enumeration.

Conclusions

- This has been a high-level overview of a very wide swath of problems that present immense computational challenges.
- There is much work to be done and many opportunities.
- Our aim is not just to develop the theory, but also to put it into practice.
- Please join us!

http://www.coin-or.org

Questions?

References I

Ahmed, S., M. Tawarmalani, and N. Sahinidis 2004. A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming* **100**(2), 355–377.

Carøe, C. and R. Schultz 1998. Dual decomposition in stochastic integer programming. *Operations Research Letters* **24**(1), 37–46.

Carøe, C. and J. Tind 1998. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming* **83**(1), 451–464.

Kong, N., A. Schaefer, and B. Hunsaker 2006. Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Mathematical Programming* **108**(2), 275–296.

Laporte, G. and F. Louveaux 1993. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters* **13**(3), 133–142.

Schultz, R., L. Stougie, and M. Van Der Vlerk 1998. Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis. *Mathematical Programming* **83**(1), 229–252.

References II

Sen, S. and J. Higle 2005. The C 3 theorem and a D 2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming* **104**(1), 1–20. ISSN 0025-5610.