# Duality, Multilevel Optimization, and Game Theory: Algorithms and Applications

Ted Ralphs[1]
Joint work with Sahar Tahernajad[1], Scott DeNegre[3], Menal Güzelsoy[2], Anahita Hassanzadeh[4]

[1]COR@L Lab, Department of Industrial and Systems Engineering, Lehigh University [2]SAS Institute, Advanced Analytics, Operations Research R & D
[3]The Hospital for Special Surgery
[4]Climate Corp

Norwegian School of Economics, Bergen, Norway, 22 September 2016

ISE Industrial and Systems Engineering

COR@L COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

# Outline

# Outline

# A Bit of Game Theory

- Our goal is to analyze certain *finite extensive-form games*, which are 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*.
- We are interested in problems with the number of possible moves is enormous, so brute force enumeration is not possible.

# Analyzing Games

- Categories
  - Multi-round vs. single-round
  - Zero sum vs. Non-zero sum
  - Winner take all vs. individual outcomes
- Goal of analysis
  - Find an equilibrium
  - Determine the optimal first move.

# 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.

# Outline

# Example: Coin Flip Game

## Coin Flip Game

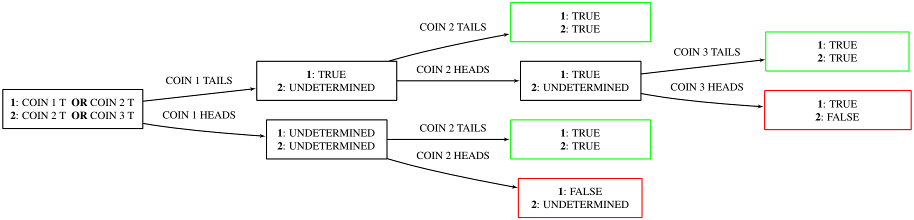
- *k* players take turns placing a set of coins heads or tails.
- In round *i*, player *i* places his/her coins.
- We have one or more logical expression that are of the form

  *COIN 1 is heads* OR *COIN 2 is tails* OR *COIN 3 is tails* OR . . .
- With even (resp. odd) *k*, “even” (resp. “odd”) players try to make all expressions true, while “odd” (resp. even) players try to prevent this.

## Examples

- k = 1: Player looks for a way to place coins so that all expressions are true.
- k = 2: The first player tries to flip her coins so that no matter how the second player flips his coins, some expression will be false.
- k = 3: The first player tries to flip his coins such that the second player cannot flip her coins in a way that will leave the third player without any way to flip his coins to make the expressions true.

# Coin Flip Game Tree

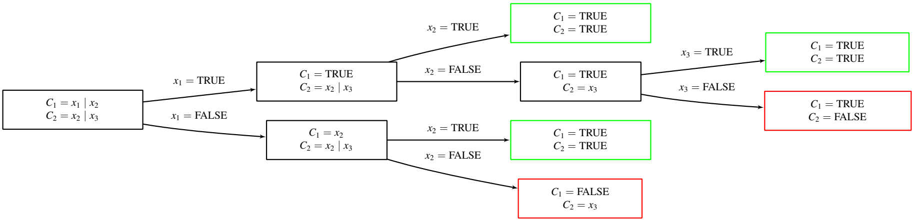


# Example: Stochastic Variant

- The coin flip game can be modified to a recourse problem if we make the even player a "chance player".
- In this variant, there is only one "cognizant" player (the odd player) who first chooses heads or tails for an initial set of coins.
- The even player is a chance player who randomly flips some of the remaining coins.
- Finally, the odd player tries to flip the remaining coins so as to obtain a positive outcome.
- The objective of the odd player's first move could then be, e.g., to maximize the probability of a positive outcome across all possible scenarios.
- Note that we still need to know what happens in all scenarios in order to make the first move optimally.

# The QBF Problem

- When expressed in terms of Boolean (TRUE/FALSE) variables, the problem is a special case of the so-called *quantified Boolean formula problem* (QBF).
- The case of $k = 1$ is the well-known Satisfiability Problem.
- This figure below 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!).



# 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

$$\exists X_1 \forall X_2 \exists X_3 \ldots ?X_k \tag{1}$$

- for even $k$, we have

$$\forall X_1 \exists X_2 \forall X_3 \ldots ?X_k \tag{2}$$

# Mathematical Optimization

- The general form of a *mathematical optimization problem* is:

**Form of a General Mathematical Optimization Problem**

$$
\begin{aligned}
z_{MP} = \min \quad & f(x) \\
\text{s.t.} \quad & g_i(x) \leq b_i, \quad 1 \leq i \leq m \\
& x \in X
\end{aligned}
\tag{MP}
$$

where $X \subseteq \mathbb{R}^n$ may be a discrete set.

- The function $f$ is the *objective function*, while $g_i$ is the *constraint function* associated with constraint $i$.
- Our primary goal is to compute the optimal value $z_{MP}$.
- However, we may want to obtain some auxiliary information as well.
- More importantly, we may want to develop parametric forms of (MP) in which the input data are the output of some other function or process.

# Multilevel and Multistage Optimization

- A (standard) mathematical optimization problem models a (set of) decision(s) to be made *simultaneously* by a *single* decision-maker (i.e., with a *single* objective).
- Decision problems arising in real-world sequential games can often be formulated as optimization problems, but they 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.

# From QBF to Multilevel Optimization

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

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

- (SAT) can be formulated as the optimization problem

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

- For $k = 2$, we then have

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

# How Difficult is the QBF?

- In general, we will focus on solving player one's decision problem, since this subsumes the solution of every other player's problem.
- No "efficient" algorithm exists for even the (single player) satisfiability problem.
- It is not surprising that the *k*-player satisfiability game is even more difficult (this can be formally proved).
  - 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 so-called *complexity classes* known as the *polynomial time hierarchy*.

# Roadmap for the Rest of the Talk

- We'll focus on simple games with two players (one of which may be a chance player) and two decision stages.
- We assume the determination of each player's move involves solution of an optimization problem.
- The optimization problem faced by the first player involves implicitly knowing what the second player's reaction will be to all possible first moves.
- The need for complete knowledge of the second player's possible reactions is what puts the complexity of these problems beyond that of standard optimization.

# Outline

# 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

# Example: Tunnel Closures [Bruglieri et al., 2008]

- The EU wishes to close certain international tunnels to trucks in order to increase security.
- The response of the trucking companies to a given set of closures will be to take the shortest remaining path.
- Each travel route has a certain “risk” associated with it and the EU’s goal is to minimize the riskiest path used after tunnel closures are taken into account.
- This is a classical Stackelberg game.

# Example: Robust Facility Location [Snyder, 2006])

- We wish to locate a set of facilities, but we want our decision to be robust with respect to possible disruptions.
- The disruptions may come from natural disasters or other external factors that cannot be controlled.
- Given a set of facilities, we will operate them according to the solution of an associated optimization problem.
- Under the assumption that at most $k$ of the facilities will be disrupted, we want to know what the worst case scenario is.
- This is a Stackelberg game in which the leader is not a cognizant DM.

# Example: Fibrilation Ablation [Finta and Haines, 2004]

- Atrial fibrilation is a common form of heart arrhythmia that may be the result of impulse cycling within macroreentrant circuits.
- AF ablation procedures are intended to block these unwanted impulses from reaching the AV node.
- This is done by surgically removing some pathways.
- Since electrical impulses travel via the path of lowest resistance, we can model their flow using a mathematical optimization problem.
- If we wish to determine the least disruptive strategy for ablation, this is a Stackelberg game.
- In this case, the follower is not a cognizant DM.

# Example: Electricity Network [Bienstock and Verma, 2008]

- As we know, electricity networks operate according to principles of optimization.
- Given a network, determining the power flows is an optimization problem.
- Suppose we wish to know the minimum number of links that need to be removed from the network in order to cause a failure.
- This, too, can be viewed as a Stackelberg game.
- Note that neither the leader nor the follower is a cognizant DM in this case.

# Outline

# Setting: Two-Stage Mixed Integer Optimization

- We have the following general formulation:

**2SMILP**

$$z_{\text{2SMILP}} = \min_{x \in \mathcal{P}_1} \Psi(x) = \min_{x \in \mathcal{P}_1} \left\{ c^\top x + \Xi(x) \right\}, \qquad \text{(2SMILP)}$$

where

$$\mathcal{P}_1 = \left\{ x \in X \mid A^1 x = b^1 \right\}$$

is the *first-stage feasible region* with $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$, $A^1 \in \mathbb{Q}^{m_1 \times n_1}$, and $b^1 \in \mathbb{R}^{m_1}$.

- $\Xi$ is a "risk function" that represents the impact of future uncertainty.
- We'll refer to $\Xi$ as the *second-stage risk function*.
- The uncertainty can arise either due to stochasticity or due to the fact that $\Xi$ represents the reaction of a competitor.

# Special Case I: Bilevel (Integer) Linear Optimization

In the case of general bilevel optimization, we have

**Bilevel Risk Function**

$$\Xi(x) = \min_{y \in \mathcal{P}_2(b^2 - A^2x) \cap Y} \left\{ d^1 y \mid d^2 y = \phi(b^2 - A^2 x) \right\}$$

where $A^2 \in \mathbb{Q}^{m_2 \times n_1}$, and $b^2 \in \mathbb{R}^{m_2}$, $\mathcal{P}_2(\beta) = \left\{ y \in \mathbb{R}_+ \mid G^2 y \geq \beta \right\}$, and $Y = \mathbb{Z}^{p_2} \times \mathbb{R}^{n_2 - p_2}$.

Alternatively, the more familiar and equivalent form of the problem is

**Mixed Integer Bilevel Linear Optimization Problem (MIBLP)**

$$\min \left\{ cx + d^1 y \mid x \in \mathcal{P}_1 \cap X, y \in \operatorname{argmin}\{ d^2 y \mid y \in \mathcal{P}_2(b^2 - A^2 x) \cap Y \} \right\} \tag{MIBLP}$$

Note that this is well-defined to be the **optimistic case**.

# Special Case II: Recourse Problems

- Recourse problems are a special case in which the risk function has a certain simple form.
- For example, the canonical form of $\Xi$ employed in the case of two-stage stochastic integer optimization is

**Stochastic Risk Function**

$$\Xi(x) = \mathbb{E}_{\omega \in \Omega}\left[\phi(h_\omega - T_\omega x)\right],$$

where $\omega$ is a random variable from a probability space $(\Omega, \mathscr{F}, \mathcal{P})$.

- For each $\omega \in \Omega$, $T_\omega \in \mathbb{Q}^{m_2 \times n_1}$ and $h_\omega \in \mathbb{Q}^{m_2}$ is the realization of the input to the second-stage problem for scenario $\omega$.
- $\phi$ is the value function of the recourse MILP, to be defined later.

# Other Special Cases

- Pure integer.
- Positive constraint matrix at second stage.
- Binary variables at the first and/or second stage.
- Zero sum and interdiction problems.

## Mixed Integer Interdiction

$$\max_{x \in \mathcal{P}_1 \cap X} \min_{y \in \mathcal{P}_2(x) \cap Y} dy \qquad \text{(MIPINT)}$$

where

$$\mathcal{P}_1 = \left\{ x \in X \mid A^1 x \leq b^1 \right\} \qquad X = \mathbb{B}^n$$

$$\mathcal{P}_2(x) = \left\{ y \in Y \mid G^2 y \geq b^2, y \leq u(e-x) \right\} \qquad Y = \mathbb{Z}^p \times \mathbb{R}^{n-p}$$

- 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 second-stage systems described by general MILPs.

# Outline

# What is Duality?

- It is difficult to define precisely what is meant by "duality" in general mathematics, though the literature is replete with various examples of it.
  - Set Theory and Logic (De Morgan Laws)
  - Geometry (Pascal's Theorem & Brianchon's Theorem)
  - Combinatorics (Graph Coloring)
- We are interested in the notions of duality relevant to solving optimization problems.
- This duality manifests itself in different forms, depending on our point of view.

**Forms of Duality in Optimization**

- NP versus co-NP (computational complexity)
- Separation versus optimization (polarity)
- Inverse optimization versus forward optimization
- Weyl-Minkowski duality (representation theorem)
- Economic duality (pricing and sensitivity)
- Primal/dual functions in optimization

# Economic Interpretation of Duality

- The economic viewpoint interprets the variables as representing possible *activities* in which one can engage at specific numeric levels.
- The constraints represent available *resources* so that $g_i(\hat{x})$ represents how much of resource $i$ will be consumed at activity levels $\hat{x} \in X$.
- With each $\hat{x} \in X$, we associate a *cost* $f(\hat{x})$ and we say that $\hat{x}$ is *feasible* if $g_i(\hat{x}) \leq b_i$ for all $1 \leq i \leq m$.
- The space in which the vectors of activities live is the *primal space*.
- On the other hand, we may also want to consider the problem from the view point of the *resources* in order to ask questions such as

  - How much are the resources "worth" in the context of the economic system described by the problem?
  - What is the marginal economic profit contributed by each existing activity?
  - What new activities would provide additional profit?

- The *dual space* is the space of *resources* in which we can frame these questions.

# Outline

# Linear Optimization

- For this part of the talk, we focus on (single-level) mixed integer linear optimization problems (MILPs).

$$z_{IP} = \min_{x \in \mathcal{S}} c^\top x, \qquad \text{(MILP)}$$

where, $c \in \mathbb{R}^n$, $S = \{x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \mid Ax = b\}$ with $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{R}^m$.

- In this context, we can consider the concepts outlined previously more concretely.
- We can think of each row of $A$ as representing a resource and each each as representing an activity or product.
- For each activity, resource consumption is a linear function of activity level.
- We first consider the case $r = 0$, which is the case of the (continuous) linear optimization problem (LP).

# The LP Value Function

- Of central importance in duality theory for linear optimization is the *value function*, defined by

$$\phi_{LP}(\beta) = \min_{x \in \mathcal{S}(\beta)} c^\top x, \tag{LPVF}$$

for a given $\beta \in \mathbb{R}^m$, where $\mathcal{S}(\beta) = \{x \in \mathbb{R}^n_+ \mid Ax = \beta\}$.

- We let $\phi_{LP}(\beta) = \infty$ if $\beta \in \Omega = \{\beta \in \mathbb{R}^m \mid \mathcal{S}(\beta) = \emptyset\}$.
- The value function returns the optimal value as a parametric function of the right-hand side vector, which represents available resources.

# Economic Interpretation of the Value Function

- What information is encoded in the value function?
  - Consider the gradient $u = \phi'_{LP}(\beta)$ at $\beta$ for which $\phi_{LP}$ is continuous.
  - The quantity $u^\top \Delta b$ represents the marginal change in the optimal value if we change the resource level by $\Delta b$.
  - In other words, it can be interpreted as a vector of the *marginal costs of the resources*.
  - For reasons we will see shortly, this is also known as the *dual solution vector*.
- In the LP case, the gradient is a *linear under-estimator* of the value function and can thus be used to derive bounds on the optimal value for any $\beta \in \mathbb{R}^m$.

# Small Example: Fractional Knapsack Problem

- We are given a set $N = \{1, \dots n\}$ of items and a capacity $W$.
- There is a profit $p_i$ and a size $w_i$ associated with each item $i \in N$.
- We want a set of items that maximizes profit subject to the constraint that their total size does not exceed the capacity.
- In this variant of the problem, we are allowed to take a fraction of an item.
- For each item $i$, let variable $x_i$ represent the fraction selected.

**Fractional Knapsack Problem**

$$
\begin{aligned}
\min \sum_{j=1}^{n} p_j x_j & \\
\text{s.t.} \sum_{j=1}^{n} w_j x_j \leq W & \\
0 \leq x_i \leq 1 & \qquad \forall i
\end{aligned}
\tag{3}
$$

- What is the optimal solution?

# Generalizing the Knapsack Problem

- Let us consider the value function of a (generalized) knapsack problem.
- To be as general as possible, we allow sizes, profits, and even the capacity to be negative.
- We also take the capacity constraint to be an equality.
- This is a proper generalization.

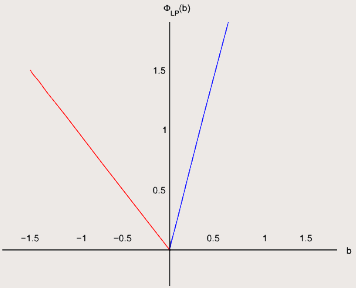
**Example 1**

$$\begin{aligned} \phi_{LP}(\beta) = \min\ & 6y_1 + 7y_2 + 5y_3 \\ s.t.\ & 2y_1 - 7y_2 + y_3 = \beta \\ & y_1, y_2, y_3, \in \mathbb{R}_+ \end{aligned}$$

# Value Function of the (Generalized) Knapsack Problem

- Now consider the value function of the example from the previous slide.
- What do the gradients of this function represent?

## Value Function for Example 1



# The Dual Optimization Problem

- Can we calculate the gradient of $\phi_{LP}$ at $b$ directly?
- Note that for any $u \in \mathbb{R}^m$, the following gives a lower bound on $\phi_{LP}(b)$.

$$\begin{aligned} g(u) = \min_{x \geq 0} \left[c^\top x + u^\top (b - Ax)\right] \quad &\leq \quad c^\top x^* \ + u^\top (b - Ax^*) \\ &= \quad c^\top x^* \\ &= \quad \phi_{LP}(b) \end{aligned}$$

- With some simplification, we can obtain an explicit form for this function.

$$\begin{aligned} g(u) \quad &= \quad min_{x \geq 0} \left[c^\top x + u^\top (b - Ax)\right] \\ &= \quad u^\top b + min_{x \geq 0} (c^\top - u^\top A) x \end{aligned}$$

- Note that

$$min_{x \geq 0} (c^\top - u^\top A) x = \begin{cases} 0, & \text{if } c^\top - u^\top A \geq \mathbf{0}^\top, \\ -\infty, & \text{otherwise,} \end{cases}$$

# The Dual Problem (cont'd)

- So we have

$$g(u) = \begin{cases} u^\top b, & \text{if } c^\top - u^\top A \geq \mathbf{0}^\top, \\ -\infty, & \text{otherwise,} \end{cases}$$

which is again a linear under-estimator of the value function.

- An LP dual problem is obtained by computing the strongest linear under-estimator with respect to $b$.

**LP Dual Problem**

$$\begin{aligned} \max_{u \in \mathbb{R}^m} g(u) = \max \ & b^\top u \\ \text{s.t. } & u^\top A \leq c^\top \end{aligned} \qquad \text{(LPD)}$$

# Representation of the LP Value Function

- Note that the feasible region of (LPD) does depend on $b$.
- Consider the *dual polyhedron* $\mathcal{D} = \{u \in \mathbb{R}^m \mid u^\top A \leq c^\top\}$.
- Then we have

**Representation of the LP Value Function**

$$\phi_{LP}(\beta) = \max_{u \in \mathcal{D}} u^\top \beta \qquad \text{(LPVF)}$$

for $\beta \in \mathbb{R}^m$.

- Using properties of the set $\mathcal{D}$, we can make this representation finite (combinatorial).
- This shows that it is convex and piecewise linear.

# What is the Importance in This Context?

- The dual problem is important is because it gives us a set of *optimality conditions*.
- For a given $b \in \mathbb{R}^m$, whenever we have
  - $x^* \in \mathcal{S}(b)$,
  - $u \in \mathcal{D}$, and
  - $c^\top x^* = u^\top b$,

  then $x^*$ is optimal!
- This means we can write down a set of constraints involving the value function that ensure optimality.
- This set of constraints can then be embedded inside another optimization problem.

# The MILP Value Function

- We now generalize the notions seen so far to the MILP case.
- The *value function* associated with the base instance (MILP) is

**MILP Value Function**

$$\phi(\beta) = \min_{x \in \mathcal{S}(\beta)} c^\top x \tag{VF}$$

for $\beta \in \mathbb{R}^m$, where $\mathcal{S}(\beta) = \{x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \mid Ax = \beta\}$.

- Again, we let $\phi(\beta) = \infty$ if $\beta \in \Omega = \{\beta \in \mathbb{R}^m \mid \mathcal{S}(\beta) = \emptyset\}$.

# Related Work on Value Function

## Duality

- Johnson [1973, 1974, 1979]
- Jeroslow [1979]
- Wolsey [1981]
- Güzelsoy and R [2007], Güzelsoy [2009]

## Structure and Construction

- Blair and Jeroslow [1977, 1982], Blair [1995]
- Kong et al. [2006]
- Hassanzadeh and R [2014b]

## Sensitivity and Warm Starting

- R and Güzelsoy [2005, 2006], Güzelsoy [2009]
- Gamrath et al. [2015]

# The (Mixed) Binary Knapsack Problem

- We now consider a further generalization of the previously introduced knapsack problem.
- In this problem, we must take some of the items either fully or not at all.
- In the example, we allow all of the previously introduced generalizations.

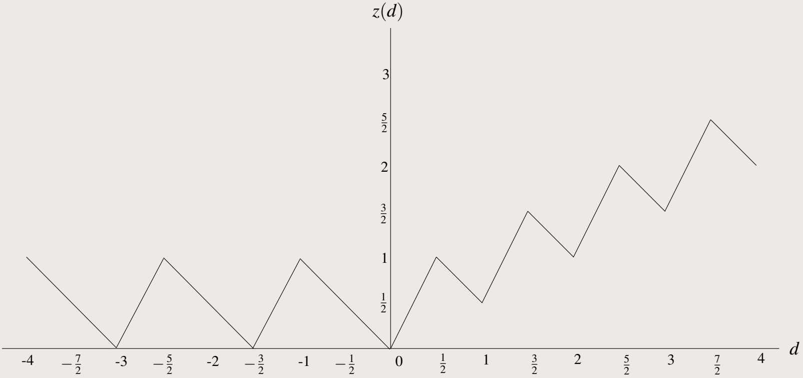
**Example 2**

$$\begin{aligned} \phi(\beta) = \min \quad & \tfrac{1}{2}x_1 + 2x_3 + x_4 \\ s.t \quad & x_1 - \tfrac{3}{2}x_2 + x_3 - x_4 = \beta \quad and \\ & x_1, x_2 \in \mathbb{Z}_+, x_3, x_4 \in \mathbb{R}_+. \end{aligned} \tag{4}$$

# Value Function for (Generalized) Mixed Binary Knapsack

- Below is the value function of the optimization problem in Example 2.
- How do we interpret the structure of this function?

## Value Function for Example 2

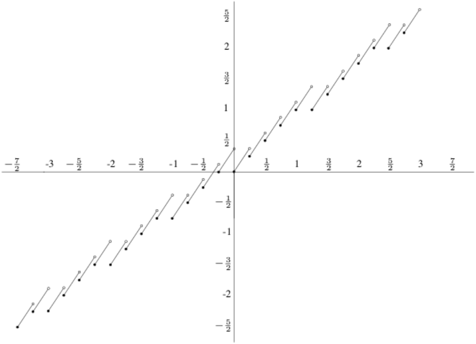


# Properties of the MILP Value Function

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

**Example 3**

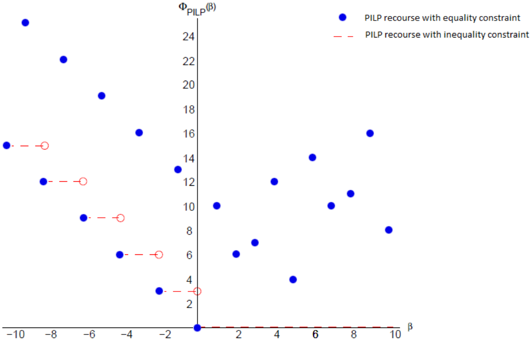
$$\begin{aligned} \phi(\beta) = \min\ & x_1 - \frac{3}{4}x_2 + \frac{3}{4}x_3 \\ s.t.\ & \frac{5}{4}x_1 - x_2 + \frac{1}{2}x_3 = \beta \\ & x_1, x_2 \in \mathbb{Z}_+,\ x_3 \in \mathbb{R}_+ \end{aligned} \qquad \text{(Ex2.MILP)}$$



# Example: MILP Value Function (Pure Integer)

**Example 4**

$$\phi(\beta) = \min 3x_1 + \frac{7}{2}x_2 + 3x_3 + 6x_4 + 7x_5 + 5x_6$$
$$s.t.\ 6x_1 + 5x_2 - 4x_3 + 2x_4 - 7x_5 + x_6 = \beta$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \in \mathbb{Z}_+$$



# Continuous and Integer Restriction of an MILP

Consider the general form of the second-stage value function

$$\begin{aligned} \phi_I(\beta) = \min\ & c_I^\top x_I + c_C^\top x_C \\ \text{s.t. } & A_I x_I + A_C x_C = \beta, \\ & y \in \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2} \end{aligned} \tag{MILP}$$

The structure is inherited from that of the *continuous restriction*:

$$\begin{aligned} \phi_C(\beta) = \min\ & c_C^\top x_C \\ \text{s.t. } & A_C x_C = \beta, \\ & x_C \in \mathbb{R}_+^{n_2 - r_2} \end{aligned} \tag{CR}$$

for $C = \{p_2 + 1, \ldots, n_2\}$ and the similarly defined *integer restriction*:

$$\begin{aligned} \phi_I(\beta) = \min\ & c_I^\top x_I \\ \text{s.t. } & A_I x_I = \beta \\ & x_I \in \mathbb{Z}_+^{r_2} \end{aligned} \tag{IR}$$

for $I = \{p_2 + 1, \ldots, n_2\}$.

# Discrete Representation of the Value Function

For $\beta \in \mathbb{R}^{m_2}$, we have that

$$\begin{aligned} \phi(\beta) = \min\ & c_I^\top x_I + \phi_C(\beta - A_I x_I) \\ \text{s.t. } & x_I \in \mathbb{Z}_+^{r_2} \end{aligned} \tag{5}$$

- From this we see that the value function is comprised of the minimum of a set of translations of $\phi_C$.
- The set of shifts, along with $\phi_C$ describe the value function exactly.
- For $\hat{x}_I \in \mathbb{Z}_+^{r_2}$, let

$$\phi_C(\beta, \hat{x}_I) = c_I^\top \hat{x}_I + \phi_C(\beta - A_I \hat{x}_I)\ \forall \beta \in \mathbb{R}^{m_2}. \tag{6}$$

- Then we have that $\phi(\beta) = \min_{x_I \in \mathbb{Z}_+^{r_2}} \phi_C(\beta, \hat{x}_I)$.

# Related Results

- From the basic structure outlined, we can derive many other useful results.

**Proposition 1** *[Hassanzadeh and R, 2014b] The gradient of $\phi$ on a neighborhood of a differentiable point is a unique optimal dual feasible solution to* (CR).

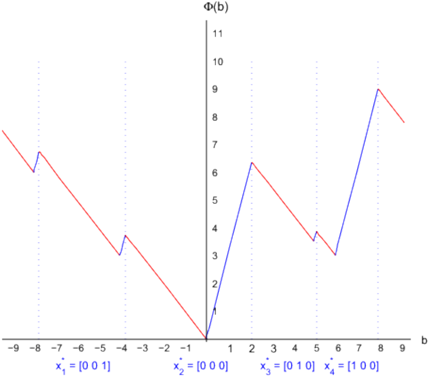
**Proposition 2** *[Hassanzadeh and R, 2014b] Consider $\mathcal{N} \subseteq \mathbb{R}^m$ over which $\phi$ is differentiable. Then, there exist an integral part of the solution $x_I^* \in \mathbb{Z}^r$ and $E \in \mathcal{E}$ such that $\phi(b) = c_I^\top x_I^* + \nu_E^\top (b - A_I x_I^*)$ for all $b \in \mathcal{N}$.*

- This last result can be extended to a subset of the domain over which $\phi$ is convex.
- Over such a region, $\phi$ coincides with the value function of a translation of the continuous restriction.
- Putting all of together, we get a practical finite representation.

# Interpretation

- It is only possible to get a unique linear price function for resource vectors where the value function is differentiable.
- This only happens when the continuous restriction has a unique dual solution at the current resource vector.
- Otherwise, there is no linear price function that will be valid in an epsilon neighborhood of the current resource vector.
- When the dual solution does exist, its value is determined by only the continuous part of the problem!
- Thus, these prices reflect behavior over only a very localized region for which the discrete part of the solution remains constant.
- In the case of the generalized knapsack problem, the differentiable points have the following two properties:
  - the continuous part of the solution is non-zero (and unique); and
  - The discrete part of the solution is unique.

**Example 5**




**Theorem 1** *[Hassanzadeh and R, 2014b]*
*Under the assumption that* $\{\beta \in \mathbb{R}^{m_2} \mid \phi_I(\beta) < \infty\}$ *is finite, there exists a finite set* $\mathcal{S} \subseteq Y$ *such that*

$$\phi(\beta) = \min_{x_I \in \mathcal{S}} \{c_I^\top x_I + \phi_C(\beta - A_I x_I)\}. \tag{7}$$

# Outline

# Dual Functions

- A *dual function* $F : \mathbb{R}^m \rightarrow \mathbb{R}$ is one that satisfies $F(\beta) \leq \phi(\beta)$ for all $\beta \in \mathbb{R}^m$.
- What is it useful for and how do we choose/construct one?
- In many application, we want one for which $F(b) \approx \phi(b)$.
- This results in the following generalized *dual problem* associated with the base instance (MILP).

$$\max \{F(b) : F(\beta) \leq \phi(\beta),\ \beta \in \mathbb{R}^m, F \in \Upsilon^m\} \tag{D}$$

where $\Upsilon^m \subseteq \{f \mid f : \mathbb{R}^m \rightarrow \mathbb{R}\}$

- We call $F^*$ *strong* for this instance if $F^*$ is a *feasible* dual function and $F^*(b) = \phi(b)$.
- This dual instance always has a solution $F^*$ that is strong if the value function is bounded and $\Upsilon^m \equiv \{f \mid f : \mathbb{R}^m \rightarrow \mathbb{R}\}$. Why?

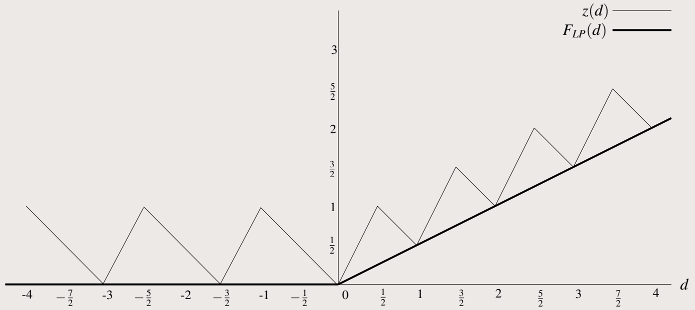
# Example: LP Relaxation Dual Function

**Example 6**

$$F_{LP}(d) = \min \quad vd, \\ s.t \quad 0 \geq v \geq -\tfrac{1}{2}, \; and \\ v \in \mathbb{R}, \tag{8}$$

which can be written explicitly as

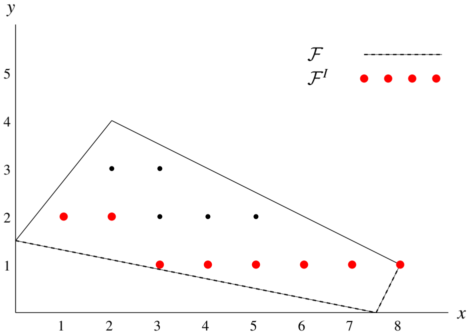
$$F_{LP}(\beta) = \begin{cases} 0, & \beta \leq 0 \\ -\tfrac{1}{2}\beta, & \beta > 0 \end{cases} .$$



# Outline

# Value Function Reformulation

More generally, we can reformulate (MIBLP) as




$$
\begin{aligned}
\min \quad & c^1 x + d^1 y \\
\text{subject to} \quad & A^1 x \leq b^1 \\
& G^2 y \geq b^2 - A^2 x \\
& d^2 y \leq \phi(b^2 - A^2 x) \\
& x \in X, y \in Y,
\end{aligned}
$$

where $\phi$ is the value function of the second-stage problem.

- This is, in principle, a standard mathematical optimization problem.
- Note that the second-stage variables need to appear in the formulation in order to enforce feasibility.

# Value Function Reformulation (Recourse)

An important special case is when $d^1 = d^2$. We can reformulate (MIBLP) as

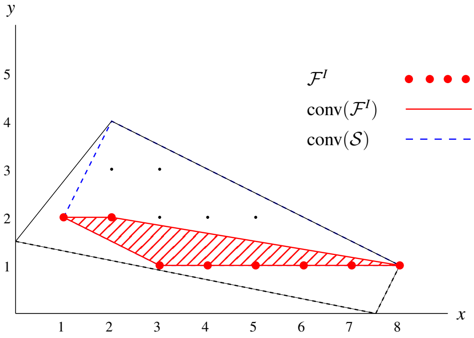
$$\begin{aligned} \min \quad & c^1 x + w \\ \text{subject to} \quad & A^1 x \leq b^1 \\ & w \geq \phi(b^2 - A^2 x) \\ & x \in X \end{aligned}$$

where $\phi$ is again the value function of the second-stage problem.

- In these special cases, we can project out the second-stage variables, as in a traditional Benders algorithm.
- This leads to a generalized Benders algorithm obtained by constructing approximations of $\phi$ dynamically.
- Non-linear "cuts" arising from lower-bounding functions are added in each step to improve the approximations.

# Polyhedral Reformulation

Convexification considers the following conceptual reformulation.




$$\begin{aligned} \min \quad & c^1 x + d^1 y \\ \text{s.t.} \quad & (x, y) \in \operatorname{conv}(\mathcal{F}^I) \end{aligned}$$

where $\mathcal{F}^I = \{(x, y) \mid x \in \mathcal{P}_1 \cap X, y \in \operatorname{argmin}\{d^2 y \mid y \in \mathcal{P}_2(x) \cap Y\}\}$

- To get bounds, we'll optimize over a relaxed feasible region.
- We'll iteratively approximate the true feasible region with linear inequalities.

# Continuous Second Stage

- In general, if $Y = \mathbb{R}^{n_1}$, then the second-stage problem can be replaced with its optimality conditions.
- The optimality conditions for the second-stage 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 problems 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 integer optimization problem.
- Note that in this case, the value function of the second-stage problem is piecewise linear, but not necessarily convex.

# Outline

# Overview of Algorithms

- There are two main classes of algorithms

### Dual

- Generalized Benders approach
- Approximate the value function from **below**.
- "Benders cuts" are (non-linear, non-convex) "dual functions".
- Can be combined with branching to get "local convexity".

### Primal

- Generalized branch-and-cut approach
- Approximate the value function from **above**
- With linear "optimality cuts", we need to branch to achieve convergence in the general case.

- Naturally, we can also have hybrids.
- Any convergent algorithm for bilevel optimization must somehow construct an approximation of the value function, usually by intelligent "sampling."

# Related Work On Bilevel Optimization

## General Nonconvex

- Mitsos [2010]
- Kleniati and Adjiman [2014a,b]

## Discrete Linear

- Moore and Bard [1990]
- DeNegre [2011], DeNegre and R [2009], DeNegre et al. [2016]
- Xu [2012]
- Caramia and Mari [2013]
- Caprara et al. [2014]
- Fischetti et al. [2016]
- Hemmati and Smith [2016], Lozano and Smith [2016]

# Related Work on Stochastic Optimization with Recourse

| | First Stage | | | Second Stage | | | Stochasticity | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathbb{R}$ | $\mathbb{Z}$ | $\mathbb{B}$ | $\mathbb{R}$ | $\mathbb{Z}$ | $\mathbb{B}$ | **W** | **T** | **h** | **q** |
| Laporte and Louveaux [1993] | | | * | * | * | * | * | * | * | |
| Carøe and Tind [1997] | * | | * | * | | * | * | * | * | * |
| Carøe and Tind [1998] | * | * | * | | * | * | | * | * | |
| Carøe and Schultz [1998] | * | * | * | * | * | * | | * | * | * |
| Schultz et al. [1998] | * | | | | * | * | | | * | |
| Sherali and Fraticelli [2002] | | | * | * | | * | * | * | * | * |
| Ahmed et al. [2004] | * | * | * | | * | * | * | | * | * |
| Sen and Higle [2005] | | | * | * | | * | | * | * | |
| Sen and Sherali [2006] | | | * | * | * | * | | * | * | |
| Sherali and Zhu [2006] | * | | * | * | | * | * | * | * | |
| Kong et al. [2006] | | * | * | | * | * | * | * | * | * |
| Sherali and Smith [2009] | | | * | * | | * | * | * | * | * |
| Yuan and Sen [2009] | | | * | * | | * | | * | * | * |
| Ntaimo [2010] | | | * | * | | * | * | | | * |
| Gade et al. [2012] | | | * | | * | * | * | * | * | * |
| Trapp et al. [2013] | | * | * | | * | * | | | * | |
| Hassanzadeh and R [2014a] | * | * | * | * | * | * | | * | * | |

# Generalized Benders for the Recourse Case

- As an illustration of how we might approach the solution of one particular class, we consider the case of recourse problems once again.
- Recall the earlier definition of a recourse problem.

**Stochastic Risk Function**

$$\Xi(x) = \mathbb{E}_{\omega \in \Omega} \left[ \phi(h_\omega - T_\omega x) \right],$$

where $\omega$ is a random variable from a probability space $(\Omega, \mathscr{F}, \mathcal{P})$.

- For each $\omega \in \Omega$, $T_\omega \in \mathbb{Q}^{m_2 \times n_1}$ and $h_\omega \in \mathbb{Q}^{m_2}$ is the realization of the input to the second-stage problem for scenario $\omega$.
- $\phi$ is the value function of the recourse MILP that we defined earlier. .

# The Second-Stage Risk Function

- The structure of the objective function $\Psi$ depends primarily on the structure of the *risk function*:

**Second-stage Risk Function**

$$\Xi(x) = \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x), \qquad \text{(2S-VF)}$$

where $\phi$ is again the value function of the second-stage problem.

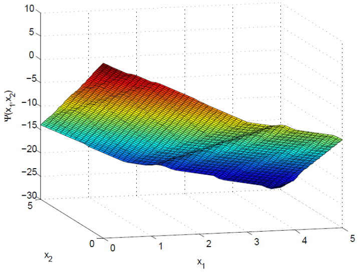
- The risk function is parameterized on the unknown value $x$ of the first-stage solution.
- The value $\Xi(x)$ for a fixed $x$ is the mean over $\Omega$ of the value of the second stage recourse/value function in each scenario.

# Example

**Example 7**

$$\min \Psi(x_1, x_2) = \min \; -3x_1 - 4x_2 + \mathbb{E}[\phi(\omega - 2x_1 - 0.5x_2)]$$
$$s.t.\; x_1 \leq 5, x_2 \leq 5 \qquad \text{(Ex.SMP)}$$
$$x_1, x_2 \in \mathbb{R}_+,$$

*and* $\omega \in \{6, 12\}$ *with a uniform probability distribution.*



# Generalized Benders [Hassanzadeh and R, 2014a]

**Benders' Master Problem**

$$\begin{aligned} \min \quad & c'x + w \\ \text{subject to} \quad & A'x \leq b' \\ & w \geq \underline{\Xi}(x) \\ & x \in X \end{aligned}$$

- $\underline{\Xi}$ is a lower approximation of the risk function $\Xi$.
- This lower approximation can be obtained, in turn from a lower approximation $\underline{\phi}$ of $\phi$, as follows:

$$\underline{\Xi}(x) = \sum_{\omega \in \Omega} p_\omega \underline{\phi}(h_\omega - T_\omega x) \qquad \text{(U-2S-VF)}$$

- Where do we get $\underline{\phi}$?

# Constructing the Dual Function

- $\underline{\phi}$ is a *dual function* that we construct dynamically by evaluating $\Xi(x)$ for different values of $x$.
- Each evaluation of $\Xi$ requires the evaluation $\phi(\beta)$ for multiple different values of $\beta$ (one for each scenario).
- Each evaluation of $\phi$ yields information that we can use to build up our approximation.
- The algorithm gives us a natural way of sampling the domain.

# Implementation Overview

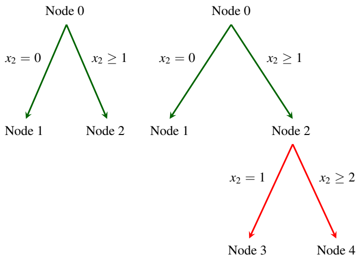
## Basic Scheme

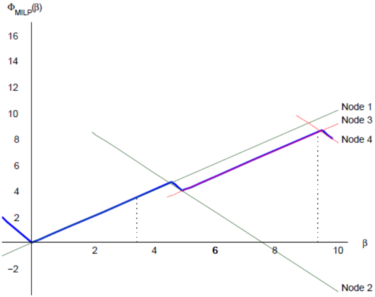
1. Solve master problem to obtain new first stage solution and lower bound.
2. Solve scenario subproblems to update value function approximation and obtain new upper bound.
3. Terminate when upper bound equals lower bound.

- As in the classical algorithm, we alternate between solving the master problem and subproblems that update the current approximation.
- The approximation may come from a single tree or a set of trees (more shortly).
- We require a solver capable of exporting the dual function resulting from the solve process.
- Ideally, the solver should also be capable of iterative refinement and warm-starting, though this is not necessary.
- The SYMPHONY MILP solver has this capability.

# Dual Functions from Branch-and-Bound [Wolsey, 1981]

- The most widely used algorithm for evaluating $\phi(\beta)$ is *branch and bound*.
- We iteratively apply *valid disjunctions* to the *LP relaxation* of an MILP.
- The disunctions partition the feasible region into a collection of *subproblems*.
- In each iteration, we derive a linear dual function for each subproblem and then take the minimum to derive a valid dual function.
- We approximate the value function as the max of all individual dual functions.



# Tree Representation of the Value Function

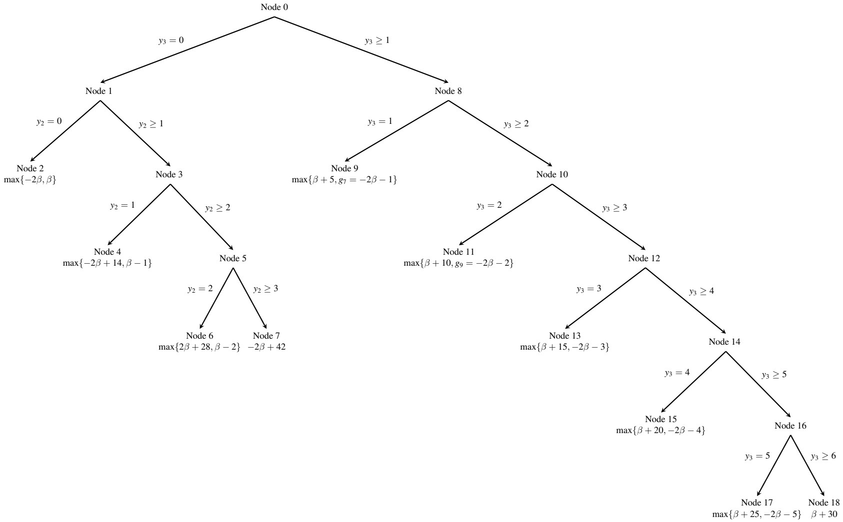
- Continuing the process, we eventually generate the entire value function.
- Consider the strengthened dual

$$\underline{\phi}^*(\beta) = \min_{t \in T} q_{I_t}^\top y_{I_t}^t + \phi_{N \setminus I_t}^t(\beta - W_{I_t} y_{I_t}^t), \tag{9}$$

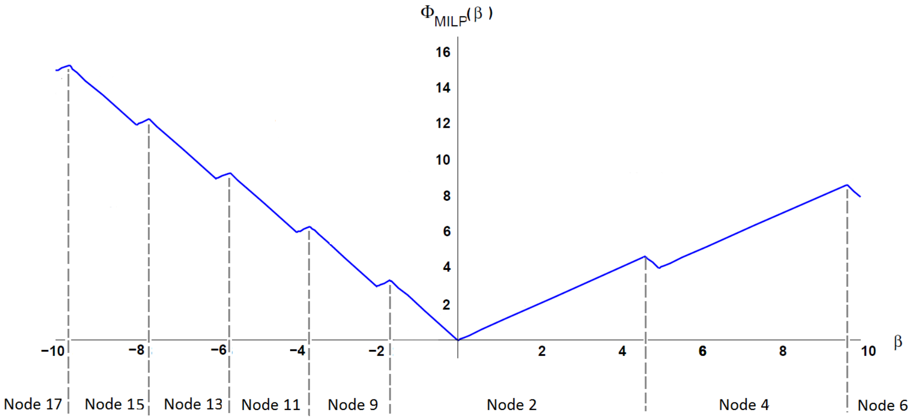
- $I_t$ is the set of indices of fixed variables, $y_{I_t}^t$ are the values of the corresponding variables in node $t$.
- $\phi_{N \setminus I_t}^t$ is the value function of the linear optimization problem at node $t$ including only the unfixed variables.

**Theorem 2** *[Hassanzadeh and R, 2014a] Under the assumption that* $\{\beta \in \mathbb{R}^{m_2} \mid \phi_I(\beta) < \infty\}$ *is finite, there exists a branch-and-bound tree with respect to which* $\underline{\phi}^* = \phi$.

# Example of Value Function Tree



# Correspondence of Nodes and Local Stability Regions



# Quick Example

Consider

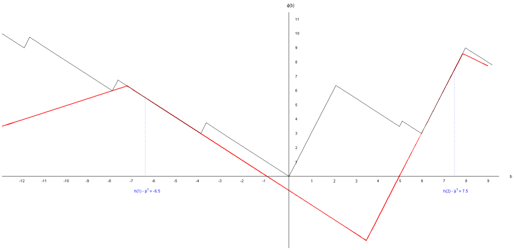
$$\begin{aligned} \min \Psi(x) = \min\ & -3x_1 - 4x_2 + \sum_{\omega=1}^{2} 0.5\phi(h_\omega - T_\omega x) \\ \text{s.t. } & x_1 + x_2 \leq 5 \\ & x \in \mathbb{Z}_+ \end{aligned} \tag{10}$$

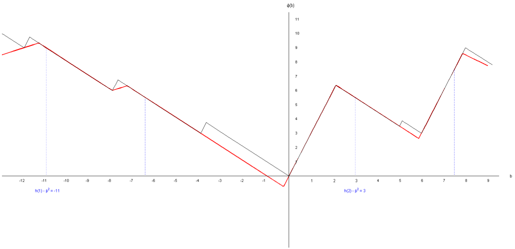
where

$$\begin{aligned} \phi(\beta) = \min\ & 3y_1 + \frac{7}{2}y_2 + 3y_3 + 6y_4 + 7y_5 \\ \text{s.t. } & 6y_1 + 5y_2 - 4y_3 + 2y_4 - 7y_5 = \beta \\ & y_1, y_2, y_3 \in \mathbb{Z}_+,\ y_4, y_5 \in \mathbb{R}_+ \end{aligned} \tag{11}$$

with $h_\omega = [-4 \;\; 10]$ and $T_\omega = [-2 \;\; \frac{1}{2}]^\top$ (technology matrix is constant)).

# Example



# Software: SYMPHONY and MibS

- The *Mixed Integer Bilevel Solver* (MibS) is a solver for bilevel integer programs available open source on github (`http://github.com/tkralphs/MibS`.
- It depends on a number of other projects available through the COIN-OR repository (`http://www.coin-or.org`).

**COIN-OR Components Used**

- The COIN High Performance Parallel Search (CHiPPS) framework to manage the global branch and bound.
- The SYMPHONY framework for checking bilevel feasibility..
- The COIN LP Solver (CLP) framework for solving the LPs arising in the branch and cut.
- The Cut Generation Library (CGL) for generating cutting planes within both SYMPHONY and MibS itself.
- The Open Solver Interface (OSI) for interfacing with SYMPHONY and CLP.

- SYMPHONY implements the procedures for constructing and exporting dual functions from branch and bound.

# Conclusions

- This general class of problems is extremely challenging computationally.
- There are special cases (interdiction/zerp sum) that are substantially easier and much progress has been made on solving these.
- Both the theory and methodology for the general case is maturing slowly.
- Many of the computational tools necessary for experimentation now also exist.
- We are currently focusing on the general case, developing the software and methodology necessary.
- There are many avenues for contribution, so please join us!

# References I


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

D. Bienstock and A. Verma. The n-k problem in power grids: New models, formulations and computation. Available at `http://www.columbia.edu/~dano/papers/nmk.pdf`, 2008.

C.E. Blair. A closed-form representation of mixed-integer program value functions. *Mathematical Programming*, 71:127–136, 1995.

C.E. Blair and R.G. Jeroslow. The value function of a mixed integer program: I. *Discrete Mathematics*, 19:121–138, 1977.

C.E. Blair and R.G. Jeroslow. The value function of an integer program. *Mathematical Programming*, 23:237–273, 1982.

M. Bruglieri, R. Maja, G. Marchionni, and P.L. Da Vinci. Safety in hazardous material road transportation: State of the art and emerging problems. In *Advanced Technologies and Methodologies for Risk Management in the Global Transport of Dangerous Goods*, chapter 6, pages 88–129. IOS Press, 2008.

# References II


A. Caprara, M. Carvalho, A. Lodi, and G.J. Woeginger. Bilevel knapsack with interdiction constraints. Technical Report OR-14-4, University of Bologna, 2014.

M. Caramia and R. Mari. Enhanced exact algorithms for discrete bilevel linear problems. *Optimization Letters*, ??:??–??, 2013.

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

C.C. Carøe and J. Tind. A cutting-plane approach to mixed 0-1 stochastic integer programs. *European Journal of Operational Research*, 101(2):306–316, 1997.

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

S DeNegre. *Interdiction and Discrete Bilevel Linear Programming*. Phd, Lehigh University, 2011. URL `http://coral.ie.lehigh.edu/{~}ted/files/papers/ScottDeNegreDissertation11.pdf`.

# References III


S DeNegre and T K R. A branch-and-cut algorithm for bilevel integer programming. In *Proceedings of the Eleventh INFORMS Computing Society Meeting*, pages 65–78, 2009. doi: 10.1007/978-0-387-88843-9_4. URL `http://coral.ie.lehigh.edu/~ted/files/papers/BILEVEL08.pdf`.

S.T. DeNegre, T.K. R, and S.A. Tahernejad. Mibs: An open source solver for mixed integer bilevel optimization problems. Technical report, COR@L Laboratory, Lehigh University, 2016.

B. Finta and D.E Haines. Catheter ablation therapy for atrial fibrillation. *Cardiology Clinics*, 22(1):127–145, 2004.

M. Fischetti, I. Ljubic, M. Monaci, and M. Sinnl. Intersection cuts for bilevel optimization. In *Proceedings of the 18th Conference on Integer Programming and Combinatorial Optimization*, 2016.

Dinakar Gade, Simge Küçükyavuz, and Suvrajeet Sen. Decomposition algorithms with parametric gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, pages 1–26, 2012.

# References IV


G. Gamrath, B. Hiller, and J. Witzig. Reoptimization techniques for mip solvers. In *Proceedings of the* $14^{th}$ *International Symposium on Experimental Algorithms*, 2015.

M Güzelsoy. *Dual Methods in Mixed Integer Linear Programming*. Phd, Lehigh University, 2009. URL http://coral.ie.lehigh.edu/{~}ted/files/papers/MenalGuzelsoyDissertation09.pdf.

M Güzelsoy and T K R. Duality for mixed-integer linear programs. *International Journal of Operations Research*, 4:118–137, 2007. URL http://coral.ie.lehigh.edu/~ted/files/papers/MILPD06.pdf.

A Hassanzadeh and T K R. A generalized benders' algorithm for two-stage stochastic program with mixed integer recourse. Technical report, COR@L Laboratory Report 14T-005, Lehigh University, 2014a. URL http://coral.ie.lehigh.edu/~ted/files/papers/SMILPGenBenders14.pdf.

# References V


A Hassanzadeh and T K R. On the value function of a mixed integer linear optimization problem and an algorithm for its construction. Technical report, COR@L Laboratory Report 14T-004, Lehigh University, 2014b. URL http://coral.ie.lehigh.edu/~ted/files/papers/MILPValueFunction14.pdf.

Mehdi Hemmati and Cole Smith. A mixed integer bilevel programming approach for a competitive set covering problem. Technical report, Clemson University, 2016.

Robert G Jeroslow. Minimal inequalities. *Mathematical Programming*, 17(1):1–15, 1979.

Ellis L Johnson. Cyclic groups, cutting planes and shortest paths. *Mathematical programming*, pages 185–211, 1973.

Ellis L Johnson. On the group problem for mixed integer programming. In *Approaches to Integer Programming*, pages 137–179. Springer, 1974.

Ellis L Johnson. On the group problem and a subadditive approach to integer programming. *Annals of Discrete Mathematics*, 5:97–112, 1979.

# References VI


P. Kleniati and C. Adjiman. Branch-and-sandwich: A deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development. *Journal of Global Optimization*, 60:425–458, 2014a.

P. Kleniati and C. Adjiman. Branch-and-sandwich: A deterministic global optimization algorithm for optimistic bilevel programming problems. part ii: Convergence analysis and numerical results. *Journal of Global Optimization*, 60: 459–481, 2014b.

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

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

L. Lozano and J.C. Smith. A backward sampling framework for interdiction problems with fortification. Technical report, Clemson University, 2016.

# References VII


A. Mitsos. Global solution of nonlinear mixed integer bilevel programs. *Journal of Global Optimization*, 47:557–582, 2010.

J.T. Moore and J.F. Bard. The mixed integer linear bilevel programming problem. *Operations Research*, 38(5):911–921, 1990.

Lewis Ntaimo. Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. *Operations research*, 58(1):229–243, 2010.

T K R and M Güzelsoy. The symphony callable library for mixed-integer linear programming. In *Proceedings of the Ninth INFORMS Computing Society Conference*, pages 61–76, 2005. doi: 10.1007/0-387-23529-9_5. URL `http://coral.ie.lehigh.edu/~ted/files/papers/SYMPHONY04.pdf`.

T K R and M Güzelsoy. Duality and warm starting in integer programming. In *The Proceedings of the 2006 NSF Design, Service, and Manufacturing Grantees and Research Conference*, 2006. URL `http://coral.ie.lehigh.edu/~ted/files/papers/DMII06.pdf`.

# References VIII

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

S. Sen and J.L. Higle. The C3 theorem and a D2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104 (1):1–20, 2005. ISSN 0025-5610.

S. Sen and H.D. Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006. ISSN 0025-5610.

Hanif D Sherali and J Cole Smith. Two-stage stochastic hierarchical multiple risk problems: models and algorithms. *Mathematical programming*, 120(2):403–427, 2009.

H.D. Sherali and B.M.P. Fraticelli. A modification of Benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342, 2002.

# References IX

H.D. Sherali and X. Zhu. On solving discrete two-stage stochastic programs having mixed-integer first-and second-stage variables. *Mathematical Programming*, 108 (2):597–616, 2006.

L. V. Snyder. Facility location under uncertainty: A review. *IIE Transactions*, 38(7): 537–554, 2006.

Andrew C Trapp, Oleg A Prokopyev, and Andrew J Schaefer. On a level-set characterization of the value function of an integer program and its application to stochastic programming. *Operations Research*, 61(2):498–511, 2013.

L.A. Wolsey. Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming*, 20(1):173–195, 1981. ISSN 0025-5610.

P. Xu. *Three Essays on Bilevel Optimization and Applications*. PhD thesis, Iowa State University, 2012.

Yang Yuan and Suvrajeet Sen. Enhanced cut generation methods for decomposition-based branch and cut for two-stage stochastic mixed-integer programs. *INFORMS Journal on Computing*, 21(3):480–487, 2009.