Multistage Integer Programming: 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

University of Houston, March 4, 2014

Outline

1. Introduction

2. Value Function

3. Algorithm

4. Conclusions

Outline

1 Introduction

2 Value Function

3 Algorithm

4 Conclusions

A Bit of Game Theory

- The optimization problems we address can be conceptualized as *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 *multistage* optimization problem.
- The move must be determined by taking into account the *uncertainty about future stages*.

Multilevel and Multistage Games

- In the literature, the term *multilevel* is used for competitive games in which there is no chance player.
- *Multistage* is used 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.

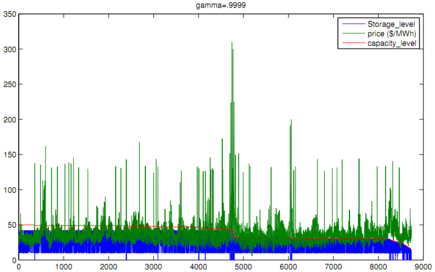
Stackelberg Game

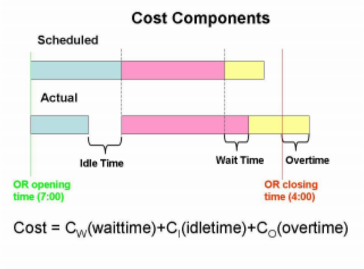
- A Stackelberg game is a game with two players who make one move each.
- The goal is to find a *subgame perfect Nash equilibrium*, i.e., the move by each player that ensures that player's best outcome.

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.

Quick Examples





Multistage Optimization

- A standard mathematical program 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 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 Programming $\Leftarrow$ multiple objectives, single DM
  - Mathematical Programming with Recourse $\Leftarrow$ multiple stages, single DM
  - Multilevel Programming $\Leftarrow$ multiple stages, multiple objectives, multiple DMs
- *Multilevel programming* generalizes standard mathematical programming by modeling hierarchical decision problems, such as finite extensive-form games.
- Such models arises in a remarkably wide array of applications.

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

Two-Stage Mixed Integer Linear Optimization

- With two stages, we hav the following general formulation:

$$z_{\mathrm{2SMILP}} = \min_{x \in \mathcal{P}_1} \Psi(x) = \min_{x \in \mathcal{P}_1} \left\{ c^\top x + \Xi(x) \right\}, \tag{1}$$

  where

$$\mathcal{P}_1 = \{x \in X \mid Ax = b, x \geq 0\} \tag{2}$$

  is the *first-stage feasible region* with $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$.

- Ξ represents the leader's expectation of the impact of future uncertainty.
- The canonical form employed in stochastic programming with recourse is

$$\Xi(x) = \mathbb{E}_{\omega \in \Omega} \left[\phi(h_\omega - T_\omega x) \right], \tag{3}$$

- ϕ is the second-stage *value function* to be defined shortly.
- $T_\omega \in \mathbb{Q}^{m_2 \times n_1}$ and $h_\omega \in \mathbb{Q}^{m_2}$ represent the input to the second-stage problem for scenario $\omega \in \Omega$.

The Second-Stage Value Function

- The structure of the objective function *Psi* depends primarily on the structure of the *value function*

$$\phi(\beta) = \min \left\{ d^\top y \mid y \in \operatorname{argmin}_{y \in \mathcal{P}_L(\beta)} q^\top y \right\}. \tag{4}$$

where

$$\mathcal{P}_2(\beta) = \{ y \in Y \mid Wy = \beta \} \tag{5}$$

is the *second-stage feasible region* with respect to a given right-hand side β and $Y = \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2}$.

- The second-stage problem is parameterized on the unknown value β of the right-hand side.
- This value is determined jointly by the realized value of ω and the values of the first-stage decision variables.
- The second-stage solution is evaluated with respect to two objective vectors, q and d, that represent the (possibly) differing valuations of the two players.

Two-Stage Stochastic Program with Recourse

For the remainder of the talk, we consider the simpler case of two-stage stochastic programming:

$$\min \Psi(x) = \min_{x \in \mathcal{P}_1} c^\top x + \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x) \tag{SP}$$

where

$$\phi(\beta) = \min_{y \in \mathcal{P}_2(\beta)} q^\top y \tag{RP}$$

In this talk, we assume

- ω follows a discrete distribution with a finite support,
- W and q are fixed,
- $\mathcal{P}_1$ is compact, and
- $\mathbb{E}_{w \in \Omega}[\phi(h_\omega - T_\omega x)]$ is finite for all $x \in X$.

Unless otherwise indicated, all probability distributions will be uniform.

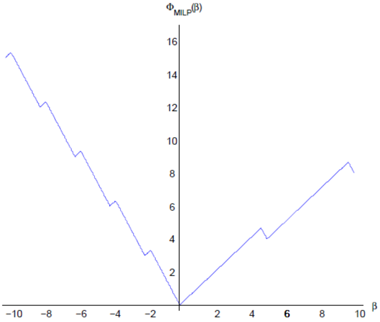
Outline

1. Introduction
2. Value Function
3. Algorithm
4. Conclusions

Illustrating the Value Function

Example 1

$$\begin{aligned}\phi(\beta) = \min\ & 6y_1 + 4y_2 + 3y_3 + 4y_4 + 5y_5 + 7y_6 \\ s.t.\ & 2y_1 + 5y_2 - 2y_3 - 2y_4 + 5y_5 + 5y_6 = \beta \\ & y_1, y_2, y_3 \in \mathbb{Z}_+, y_4, y_5, y_6 \in \mathbb{R}_+.\end{aligned} \qquad \text{(Ex.MILP)}$$

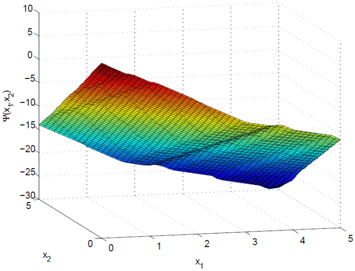


Illustrating the Objective Function

Example 2

$$\Psi(x) = -3x_1 - 4x_2 + \sum_{\omega \in \Omega} \phi(h_\omega - 2x_1 - 0.5x_2) \tag{Ex.SMP}$$

and $\Omega = \{1, 2\}, h_1 = 6, h_2 = 12$.



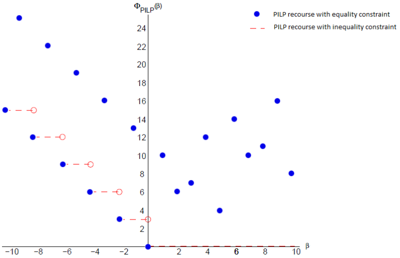
Note the similarity in structure of the objective function to the value function.

MILP Value Function (Pure Integer)

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

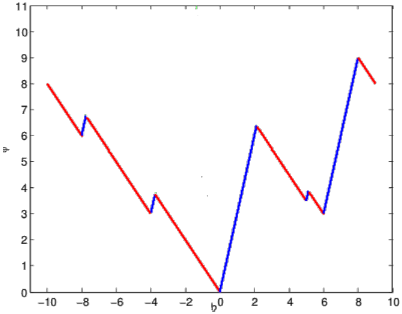
Example 3

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



Example 4

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



Continuous and Integer Restriction of an MILP

Consider the general form of the second-stage value function

$$\begin{aligned} \phi(\beta) = \min\, & q_I^\top y_I + q_C^\top y_C \\ \text{s.t. } & W_I y_I + W_C y_C = b, \\ & y \in \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2} \end{aligned} \tag{MILP}$$

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

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

and the similarly defined *integer restriction*:

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

Discrete Representation of the Value Function

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

$$\begin{aligned} \phi(\beta) = \min\ & q_I^\top y_I + \phi_C(\beta - W_I y_I) \\ \text{s.t. } & y_I \in \mathbb{Z}_+^{r_2} \end{aligned} \tag{6}$$

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

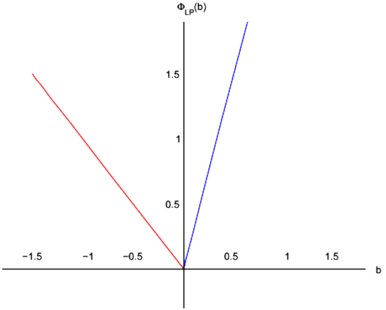
$$\phi_C(\beta, \hat{y}_I) = q_I^\top \hat{y}_I + \phi_C(\beta - W_I \hat{y}_I) \ \ \forall \beta \in \mathbb{R}^{m_2}. \tag{7}$$

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

Illustrating the Continuous Restriction

Example 5

$$\phi_C(\beta) = \min 6y_1 + 7y_2 + 5y_3$$
$$s.t.\ 2y_1 - 7y_2 + y_3 = \beta$$
$$y_1, y_2, y_3 \in \mathbb{R}_+$$



Value Function of the Continuous Restriction

Recall the previously defined continuous restriction.

$$\begin{aligned} \phi_C(\beta) = \min\ & q_C^\top y_C \\ \text{s.t. } & W_C y_C = \beta \\ & y_C \in \mathbb{R}^n_+ \end{aligned} \tag{CR}$$

When the dual of (CR) is feasible, the epigraph of ϕ_C is the convex cone

$$\mathcal{L} := cone\{(W_1, q_1), (W_2, q_2), \ldots, (W_n, q_n), (0, 1)\} \tag{8}$$

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

$$\mathcal{L} := \{(\beta, z) : z \geq u_i^\top \beta, i = 1, \ldots, k, d_j^\top \beta \leq 0, j = 1, \ldots, p\}. \tag{9}$$

Properties of MILP Value Function

- We can improve on the previous representation by deriving a *minimal* discrete set that suffices to describe ϕ.

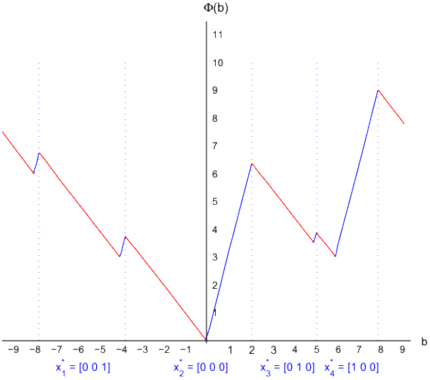
Theorem 1 *[Hassanzadeh et al., 2014]*
Under the assumption that $\{\beta \in \mathbb{R}^{m_2} \mid \phi_I(\beta) < \infty\}$ *is finite, there exists a finite set* $\mathcal{S} \subseteq Y$ *such that*

$$\phi(\beta) = \min_{y_I \in \mathcal{S}} \{q_I^\top y_I + \phi_C(\beta - W_I y_I)\}. \tag{10}$$

- The points in $\mathcal{S}$ are the points of *strict local convexity* of the value function.
- Associated with each of these points is a region (the *local stability set*) over which the integer part of the optimal solution remains constant.
- The value function of the MILP, when restricted to that region, is a translation of the value function of the continuous restriction (and thus convex).
- In [Hassanzadeh et al., 2014], we describe an algorithm for constructing a superset of $\mathcal{S}$ that is easy to implement.

Points of Strict Local Conexity

Example 6



The figure above shows the points of strict local convexity and the associated local stability sets for the previous example.

Outline

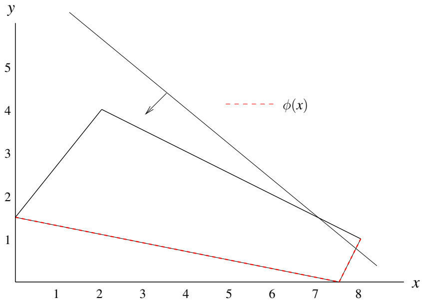
1. Introduction
2. Value Function
3. Algorithm
4. Conclusions

Benders' Principle (Linear Programming)

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

where

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

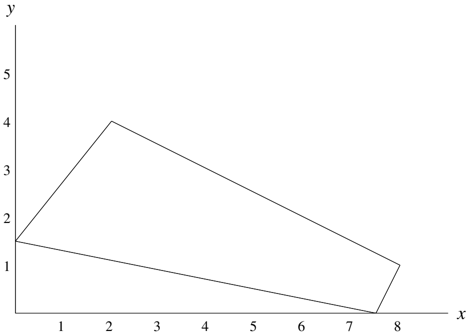


Basic Strategy:

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

Example

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

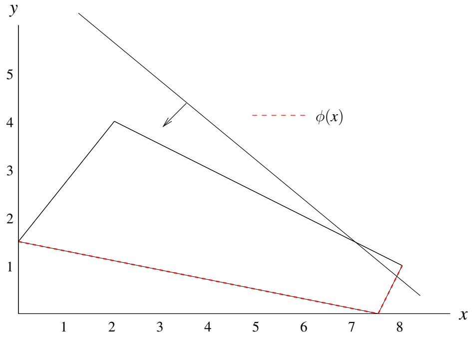


Value Function Reformulation

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

where

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

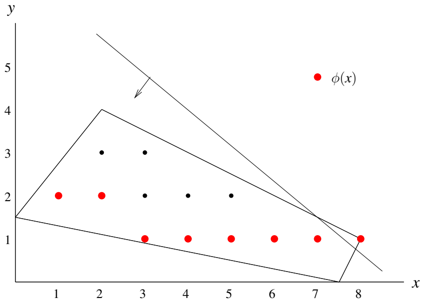


Benders' Principle (Integer Programming)

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

where

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

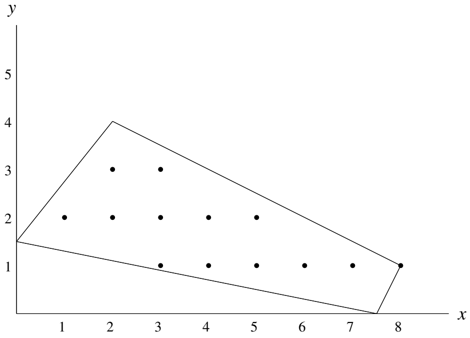


Basic Strategy:

- Here, ϕ is the value function of an *integer program*.
- 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

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

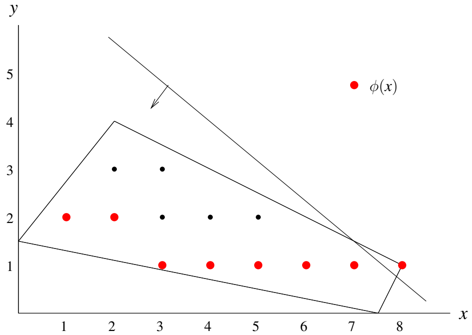


Value Function Reformulation

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

where

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



Related Algorithms

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

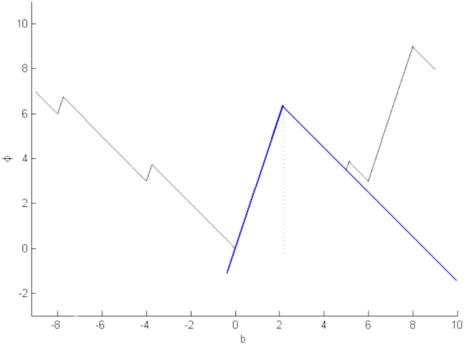
- Modification to the L-shaped framework [Laporte and Louveaux, 1993, Carøe and Tind, 1998, Sen and Higle, 2005]
  - Linear cuts in first stage for binary first stage
  - Optimality cuts from B&B and cutting plane, applied to pure integer second stage
  - Disjunctive programming approaches and cuts in the second stage
- Value function approaches: Pure integer case [Ahmed et al., 2004, Kong et al., 2006]
- Scenario decomposition [Carøe and Schultz, 1998]
- Enumeration/Gröbner basis reduction [Schultz et al., 1998]

Summary of Related Work

	First Stage			Second Stage			Stochasticity			
	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{B}$	$\mathbb{R}$	$\mathbb{Z}$	$\mathbb{B}$	$\mathbf{W}$	$\mathbf{T}$	$\mathbf{h}$	$\mathbf{q}$
[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]		*	*		*	*			*	
Current work	*	*	*	*	*	*		*	*	

Lower Bounds on the Value Function

We already observed that for an effective integer Benders' method, we need effective lower *bounding functions* to *approximate* the MILP value function.



Dual Functions

A dual function $\varphi : \mathbb{R}^{m_2} \to \mathbb{R} \cup \{\pm\infty\}$ is

$$\varphi(\beta) \leq \phi(\beta) \quad \forall \boldsymbol{b} \in \mathbb{R}^{m_2} \tag{11}$$

For a particular instance $\hat{\beta}$, the dual problem is

$$\phi_D = \max\{\varphi(\hat{\beta}) : \varphi(\beta) \leq \phi(\beta) \quad \forall \beta \in \mathbb{R}^{m_2},\ \varphi : \mathbb{R}^{m_2} \to \mathbb{R} \cup \{\pm\infty\}\} \tag{12}$$

Let $\mathcal{F}$ be a set of dual functions generated so far. Then Benders' master problem is

$$\begin{aligned} \min\ & c^\top x + \theta \\ & \theta \geq \sum_{\omega \in \Omega} \max_{f \in \mathcal{F}} f(h_\omega - T_\omega x)] \\ & x \in \mathcal{P}_1 \end{aligned} \tag{MP}$$

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:

$$\begin{aligned} \min\ & c^\top x \\ \text{s.t. } & Ax = b, \\ & l^t \le x \le u^t, x \ge 0 \end{aligned} \tag{13}$$

The dual at node t:

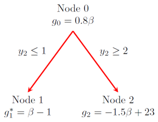
$$\begin{aligned} \max\ & \{\pi^t b + \underline{\pi}^t l^t + \bar{\pi}^t u^t\} \\ \text{s.t. } & \pi^t A + \underline{\pi}^t + \bar{\pi}^t \le c^\top \\ & \underline{\pi} \ge 0, \bar{\pi} \le 0 \end{aligned} \tag{14}$$

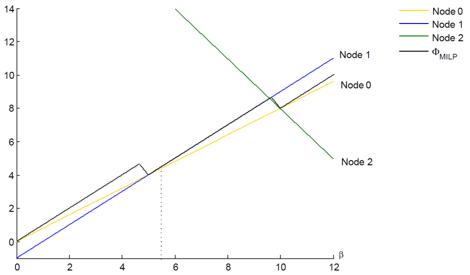
We obtain the following strong dual function:

$$\min_{t \in T} \{\pi^t b + \underline{\pi}^t l^t + \bar{\pi}^t u^t\} \tag{15}$$

Warm Starting the Solution Process

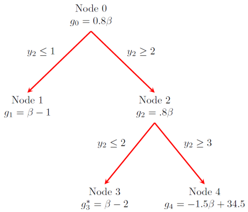
- Here, we illustrate the procedure.
- We can improve on the basic scheme by warm starting the solution of each subproblem from the tree generated during solution of the previous subproblem.

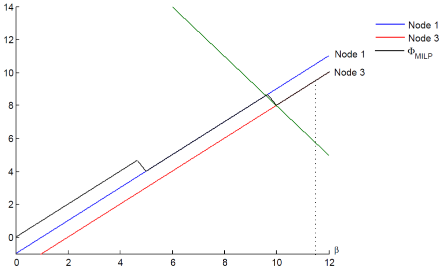




Warm Starting the Solution Process

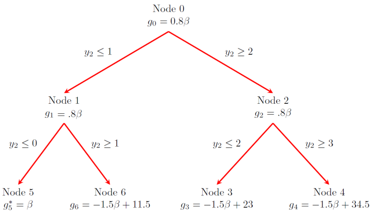
- Here, we illustrate the procedure.
- We can improve on the basic scheme by warm starting the solution of each subproblem from the tree generated during solution of the previous subproblem.

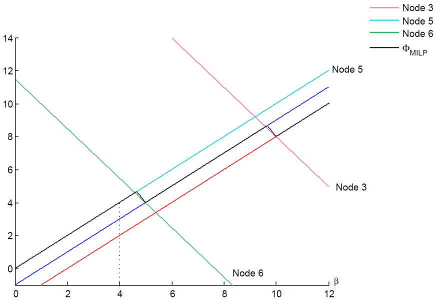




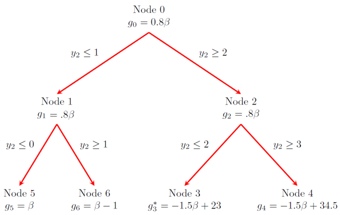
Warm Starting the Solution Process

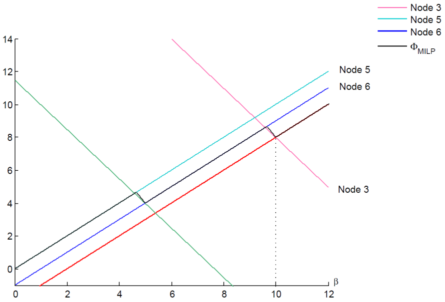
- Here, we illustrate the procedure.
- We can improve on the basic scheme by warm starting the solution of each subproblem from the tree generated during solution of the previous subproblem.





- Here, we illustrate the procedure.
- We can improve on the basic scheme by warm starting the solution of each subproblem from the tree generated during solution of the previous subproblem.





Generating the Value Function in a Single Tree

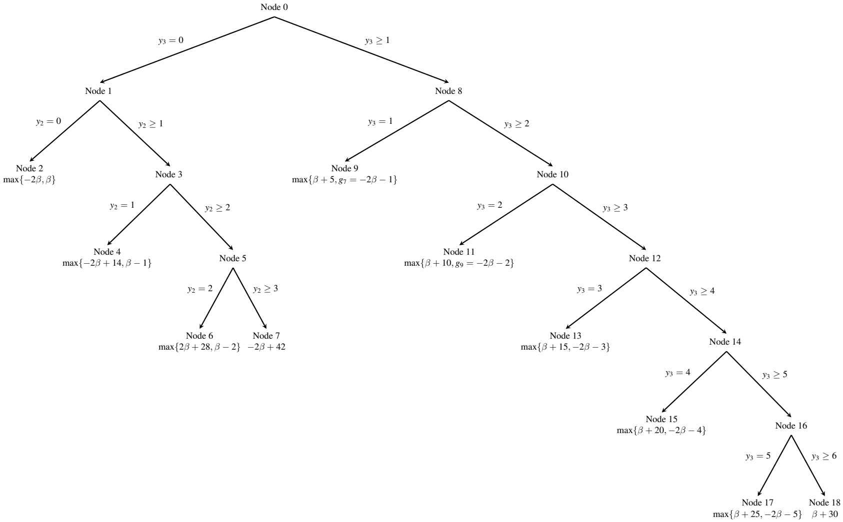
- Continuing the process, we eventually generate the entire value function.
- Consider the strengthened dual

$$\underline{\phi}^*(\beta) = \min_{t \in T} q_{I_t}^\top y_{I_t}^t + \phi_{N \setminus I_t}(\beta - W_{I_t} y_{I_t}^t), \tag{16}$$

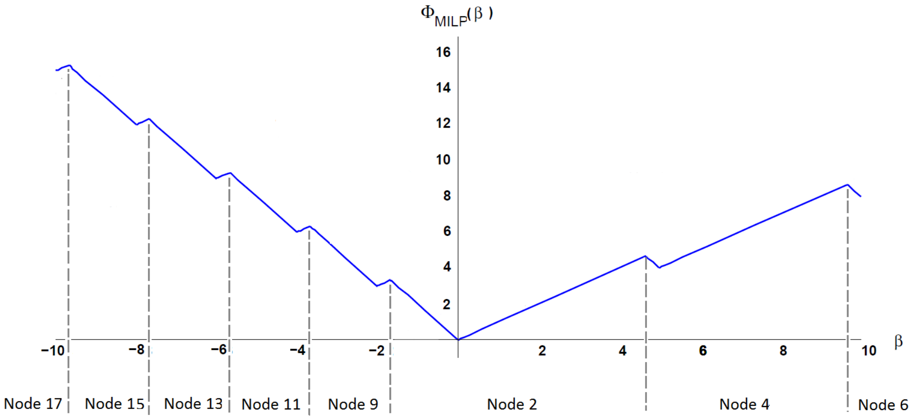
- I_t is the set of indices of fixed variables, $y_{I_t}^t$ are the values of the corresponding variables in node t.
- $\phi_{N \setminus I_t}$ is the value function of the linear program including only the unfixed variables.

Theorem 2 *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



Example of Value Function Tree



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
- $n \in \{1, \ldots, N(p, r)\}$ where $N(p, r)$ is the number of terminating nodes in the B&B tree solved for scenario r at iteration p.
- $\theta_s = \mathcal{F}(h(s) - \beta)$
- $t_{spr} = F_r^p(h(s) - \beta)$ the approximation of scenario s's recourse obtained from the optimal dual function of iteration p and scenario r.
- ν_{prn}, a_{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

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

Example

Consider

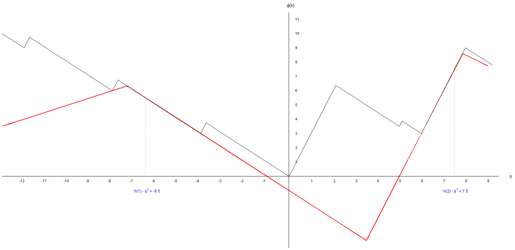
$$\min f(x) = \min \; -3x_1 - 4x_2 + \sum_{s=1}^{2} 0.5Q(x,s) \quad \text{s.t. } x_1 + x_2 \leq 5, \quad x \in \mathbb{Z}_+ \tag{17}$$

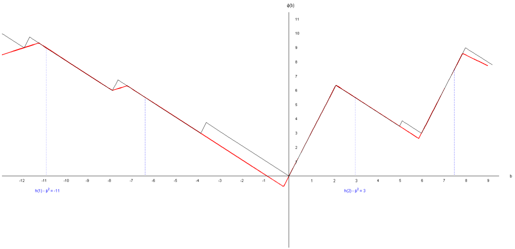
where

$$Q(x,s) = \min 3y_1 + \frac{7}{2}y_2 + 3y_3 + 6y_4 + 7y_5 \quad \text{s.t. } 6y_1 + 5y_2 - 4y_3 + 2y_4 - 7y_5 = h(s) - 2x_1 - \frac{1}{2}x_2, \quad y_1, y_2, y_3 \in \mathbb{Z}_+,\; y_4, y_5 \in \mathbb{R}_+ \tag{18}$$

with $h(s) \in \{-4, 10\}$.

Example



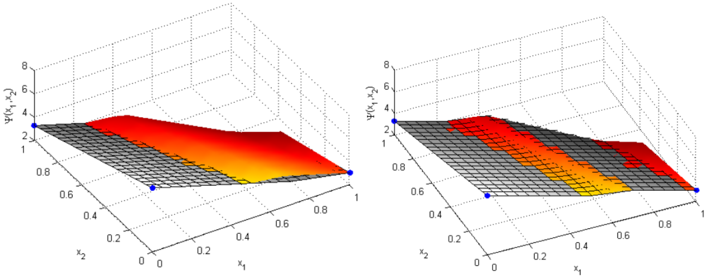


Outline

1. Introduction
2. Value Function
3. Algorithm
4. Conclusions

Conclusions

Non-convex optimality cuts are ugly. But they may be worthwhile!



- We have developed an algorithm for the two-stage problem with general mixed integer in both stages.
- The algorithm uses the Benders' framework with B&B dual functions as the optimality cuts.
- Such cuts have computationally desirable properties such as warm-starting.
- We need to keep the size of approximations small. This can be done through warm-starting trees and scenario bunching.

Future Work

- We have implemented the algorithm using SYMPHONY as our mixed-integer linear optimization solver.
- Warm-starting a B&B tree is possible in the solver.
- We so far have a fairly “naive” implementation and anticipate much improvment is possible.
- In particular, we shuld be able to exploit parallelism much more easily here than in the traditional MILP case.
- We also need to develop a scenario bunching scheme. Doing this, we decide on the local area of the tree to examine.
- Finally, we hope to move on soon to the more general case of multilevel programming.

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.

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.

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.

A Hassanzadeh, T K Ralphs, and M Güzelsoy. On the value function of a mixed integer linear optimization problem and an algorithm for construction. Technical report, COR@L Laboratory, Lehigh University, 2014.

References II

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 e for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

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

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.

References III

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.

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.

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.

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.