Multistage capacity planning using JuDGE

Andy Philpott[1]
Electric Power Optimization Centre
Department of Engineering Science,
University of Auckland.
(joint work with Anthony Downward, Regan Baucke and Sonja Wogrin)

[1] BIRS Workshop, Oaxaca, September 23, 2019.

Capacity expansion under uncertainty

- Where, when, how big to build capacity?
- Why? Planning 100% renewable electricity
- Multistage stochastic optimization with binary variables.
- Multi-horizon scenario trees [Kaut et al, 2014]
- Do we have to use progressive hedging? [Watson and Woodruff, 2011]
- If I had a hammer. . . [PS, PPM, 1962]

Summary

1. Introduction
2. The problem
3. A toy knapsack example
4. The general case
5. JuDGE.jl
6. Energy applications
   - Electricity distribution networks
   - Transmission expansion with imperfect competition
   - EMERALD: zero-carbon electricity
7. Conclusion

Summary

1. Introduction
2. The problem
3. A toy knapsack example
4. The general case
5. JuDGE.jl
6. Energy applications
   - Electricity distribution networks
   - Transmission expansion with imperfect competition
   - EMERALD: zero-carbon electricity
7. Conclusion

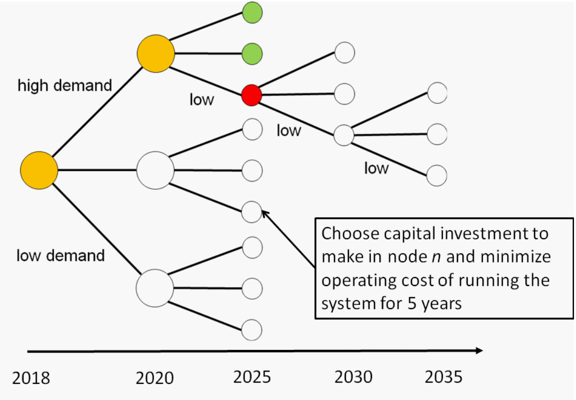
Multistage stochastic integer programming

- Multistage stochastic integer programming is hard.
- Deterministic equivalent problems (e.g. from SAA) become very large MIPs.
- The only hope is decomposition of some form.
  1. Stagewise decomposition e.g. SDDiP [Zou et al, 2018], MIDAS [P. et al, 2019];
  2. Nodal decomposition e.g. Lagrangian relaxation [Borison et al, 1984];
  3. Scenario decomposition [Rockafellar and Wets, 1989; Caroe and Schultz, 1999; Dentcheva and Romisch, 2004; Watson and Woodruff, 2011];
- For multihorizon scenario trees we can combine advantages of (2) and (3), for a restricted class of problems.

EMERALD: 100% renewable energy

- What to build to make NZ electricity 100 percent renewable by 2035.
- Scenario tree models states of the world with changes in demand, technology, government policy.
- Each subproblem computes optimal operation of electricity system in state of the world.
- State of the world lasts five years (giving 4 stages) and involves hydro inflow and wind uncertainty.
- Example model has 4 branches per stage giving 64 scenarios.

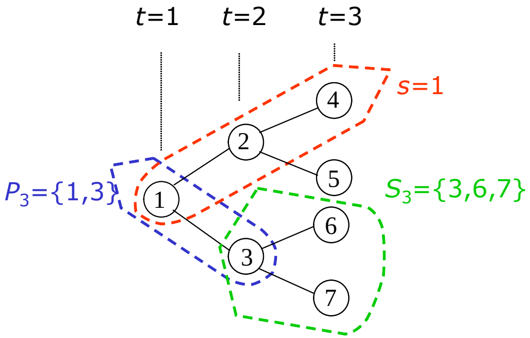
Multihorizon scenario tree



Multihorizon scenario tree for 100% renewable energy model. In each node of the tree we solve a two-stage operational subproblem given investments in capacity up to this time. The scenario tree for this subproblem is suppressed.

Scenario tree notation

Coarse-grain uncertainty has scenario tree $\mathcal{N}$ with N nodes. The probability of node n is ϕ_n. $\mathcal{P}_n$ denotes the set of predecessors of node n, and $\mathcal{S}_n$ the set of successors of node n.



A scenario tree with three stages and four scenarios.

Summary

1. Introduction
2. The problem
3. A toy knapsack example
4. The general case
5. JuDGE.jl
6. Energy applications
   - Electricity distribution networks
   - Transmission expansion with imperfect competition
   - EMERALD: zero-carbon electricity
7. Conclusion

Example: multistage knapsack problem

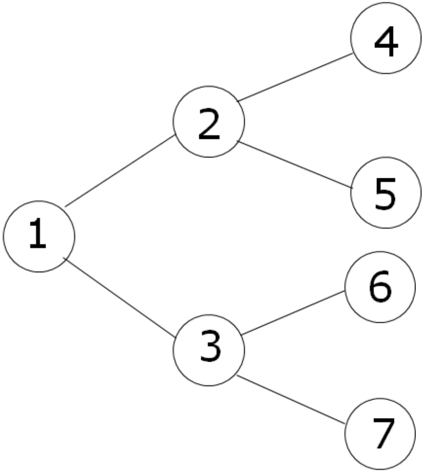
Consider random vectors (value) $\mathbf{v}_n \in \mathbb{R}^Y$ and (weight) $\mathbf{w}_n \in \mathbb{R}^Y$, $n \in \mathcal{N}$. Stochastic knapsack problem with fixed capacity u_0.

$$
\begin{aligned}
\text{KP: max} \quad & \textstyle\sum_{n\in\mathcal{N}} \phi_n \mathbf{v}_n^\top \mathbf{y}_n \\
\text{s.t.} \quad & \mathbf{w}_n^\top \mathbf{y}_n \leq u_0 && n \in \mathcal{N}, \\
& \mathbf{y}_n \in \{0,1\}^Y, && n \in \mathcal{N}.
\end{aligned}
$$

Capacity expansion version:

- expand knapsack capacity by parameter U with cost c_n, $n \in \mathcal{N}$.
- at most one expansion allowed in time horizon.
- minimize expected cost of expansion less expected value of knapsack contents.

Scenario tree of multistage knapsack problem



Scenario tree $\mathcal{N}$ of knapsack problem with 7 nodes.

Formulation of multistage knapsack problem

$$
\begin{aligned}
\text{KP: min} \quad & \textstyle\sum_{n\in\mathcal{N}} \phi_n \left(c_n z_n - \mathbf{v}_n^\top \mathbf{y}_n \right) & \\
\text{s.t.} \quad & \mathbf{w}_n^\top \mathbf{y}_n \le u_0 + \textstyle\sum_{h\in\mathcal{P}_n} U z_h, & n \in \mathcal{N}, \\
& \textstyle\sum_{h\in\mathcal{P}_n} z_h \le 1, & n \in \mathcal{N}, \\
& \mathbf{y}_n \in \{0,1\}^Y, & n \in \mathcal{N}, \\
& z_n \in \{0,1\}, & n \in \mathcal{N}.
\end{aligned}
$$

Instance of multistage knapsack problem for three stages

All scenarios are equally likely, $u_0 = 3$, $U = 4$.

cost	item				
node	1	2	3	4	5
1	60	20	10	15	10
2	8	10	20	20	10
3	8	10	15	10	30
4	40	40	35	10	20
5	15	35	15	15	20
6	70	30	15	15	10
7	25	50	25	15	20

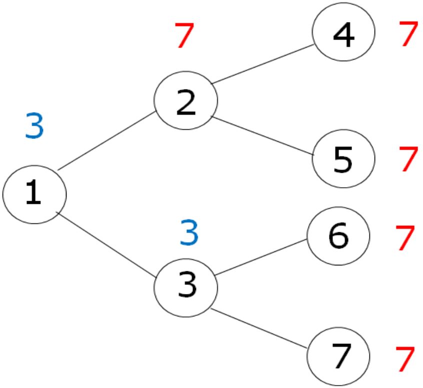
volume	item				
node	1	2	3	4	5
1	6	2	1	1	1
2	8	2	2	2	1
3	8	1	1	1	3
4	4	4	3	1	2
5	1	3	1	1	2
6	7	3	1	1	1
7	2	5	2	1	2

Solve as a large MIP

```
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Node    ItCnt     Gap

*     0+    0                          -5.0000                      8     ---
*     0+    0                        -121.2500                      8     ---
      0     0     -135.0000     5    -121.2500    -135.0000         8   11.34%
      0     0     -133.2500     4    -121.2500      Cuts: 7        15    9.90%
*     0+    0                        -130.0000    -133.2500        15    2.50%
*     0+    0                        -131.2500    -133.2500        15    1.52%
      0     0        cutoff          -131.2500    -131.2500        15    0.00%
Elapsed real time =   0.02 sec. (tree size =  0.00 MB)

GUB cover cuts applied:  1
Flow cuts applied:  2
Zero-half cuts applied:  1
Gomory fractional cuts applied:  3

Root node processing (before b&c):
  Real time             =    0.02
Parallel b&c, 4 threads:
  Real time             =    0.00
  Sync time (average)   =    0.00
  Wait time (average)   =    0.00
                          --------
Total (root+branch&cut) =    0.02 sec.
CPLEX 12.2.0.0: optimal integer solution; objective -131.25
15 MIP simplex iterations
0 branch-and-bound nodes
1 GUB-cover cut
2 flow-cover cuts
3 Gomory cuts
1 zero-half cut
ampl:
```

CPLEX 12.2 log file when solving knapsack problem on a 7 node scenario tree problem using branch and bound.

Solution of multistage knapsack problem



Optimal expansion plan for instance of knapsack problem. Optimal expected cost = -131.25.

Summary

1. Introduction
2. The problem
3. A toy knapsack example
4. The general case
5. JuDGE.jl
6. Energy applications
   - Electricity distribution networks
   - Transmission expansion with imperfect competition
   - EMERALD: zero-carbon electricity
7. Conclusion

General class of problems

We have initial capacity $u_0 \in R^U$ that we increase with actions $z_n \in \{0,1\}^Z$ in node n of N. Expansion in node n costs $c_n^\top z_n$ and contributes additional capacity $U_n z_n \geq 0$ to the system in every node $h \in S_n$. So the capacity in node n is $u_0 + \sum_{h \in \mathcal{P}_n} U_h z_h$. In each state corresponding to node n we operate our capacity choosing actions $y_n \in Y_n$ to minimize cost $q_n^\top y_n$.

$$
\begin{aligned}
\text{SIP: min} \quad & \sum_{n \in \mathcal{N}} \phi_n \left(\mathbf{c}_n^\top \mathbf{z}_n + \mathbf{q}_n^\top \mathbf{y}_n \right) \\
\text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{u}_0 + \sum_{h \in \mathcal{P}_n} U_h \mathbf{z}_h, \quad n \in \mathcal{N} \\
& \mathbf{y}_n \in \mathcal{Y}_n, \quad n \in \mathcal{N}, \\
& \mathbf{z}_n \in \{0,1\}^Z, \quad n \in \mathcal{N}.
\end{aligned}
$$

Remarks

1. V_n and U_h are matrices of order $U \times Y$ and $U \times Z$.
2. The actions y_n are not affected by any actions in other nodes apart from capacity decisions z_h, $h \in P_n$. This precludes e.g. inventory being transferred from stage to stage.
3. The constraints $y_n \in Y_n$ can be very general, e.g. include binary variables, nonlinearities, competitive equilibrium.
4. We require an efficient method of solving the (almost) single-node subproblem:

$$\begin{aligned} \text{SP}(n): \ \min \quad & \phi_n \mathbf{q}_n^\top \mathbf{y}_n - \textstyle\sum_{h \in \mathcal{P}_n} \boldsymbol{\pi}_{hn}^\top \mathbf{z}_h - \boldsymbol{\mu}_n \\ \text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{u}_0 + \textstyle\sum_{h \in \mathcal{P}_n} U_h \mathbf{z}_h, \\ & \mathbf{y}_n \in \mathcal{Y}_n, \quad \mathbf{z}_h \in \{0,1\}^Z, \ h \in \mathcal{P}_n. \end{aligned}$$

SP(n) gives a column

Suppose we can solve SP(n) fast to give $z_h \in \{0,1\}^Z$, $h \in P_n$. Then populate $\hat{\mathbf{z}}_{hn} \in R^{NZ}$, a column of capacity decisions (one for each node $h \in \mathcal{N}$) with 0 if $h \notin P_n$ and z_h if $h \in P_n$. So

$$\hat{\mathbf{z}}_{hn} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Dantzig-Wolfe master problem

The columns $\widehat{\mathbf{z}}_{hn} \in R^{NZ}$ form the columns of a Dantzig-Wolfe master problem, that can be generated dynamically using π_{hn} and μ_n.

$$
\begin{array}{llll}
\text{MP:min} & \sum_{n\in\mathcal{N}} \phi_n \mathbf{c}_n^\top \mathbf{x}_n + \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}_n} \phi_n \mathbf{q}_n^\top \widehat{\mathbf{y}}_n^j w_n^j & & \\
\text{s.t.} & \sum_{j\in\mathcal{J}_n} \widehat{\mathbf{z}}_{hn}^j w_n^j \leq \mathbf{x}_h, \quad h \in \mathcal{P}_n, \quad n \in \mathcal{N}, & [\boldsymbol{\pi}_{hn}] \\
& \sum_{j\in\mathcal{J}_n} w_n^j = 1, \quad n \in \mathcal{N}, & [\boldsymbol{\mu}_n] \\
& w_n^j \in \{0,1\}, \quad n \in \mathcal{N}, \; j \in \mathcal{J}_n, & \\
& \mathbf{x}_n \in \{0,1\}^Z, \quad n \in \mathcal{N}. &
\end{array}
$$

A simpler version of the problem

- Suppose at most one expansion per capital item is allowed over the time horizon.
- In addition suppose $U_{hn} = U$, so capacity increment matrix is deterministic and constant with time.
- Simpler subproblem:

$$\begin{aligned} \text{SP}(n)\text{: } \min \quad & \phi_n \mathbf{q}_n^\top \mathbf{y}_n - \boldsymbol{\pi}_n^\top \mathbf{z}_n - \boldsymbol{\mu}_n \\ \text{s.t.} \quad & V_n \mathbf{y}_n \leq \mathbf{u}_0 + U\mathbf{z}_n, \\ & \mathbf{y}_n \in \mathcal{Y}_n, \quad \mathbf{z}_n \in \{0,1\}^Z. \end{aligned}$$

Dantzig-Wolfe master problem (at most one expansion)

$$
\begin{aligned}
\text{MP:}\min \quad & \textstyle\sum_{n\in\mathcal{N}} \phi_n \mathbf{c}_n^\top \mathbf{x}_n + \sum_{n\in\mathcal{N}} \sum_{j\in\mathcal{J}_n} \phi_n \mathbf{q}_n^\top \widehat{\mathbf{y}}_n^j w_n^j \\
\text{s.t.} \quad & \textstyle\sum_{j\in\mathcal{J}_n} \widehat{\mathbf{z}}_n^j w_n^j \le \sum_{h\in\mathcal{P}_n} \mathbf{x}_h, \quad n\in\mathcal{N}, \qquad [\pi_n] \\
& \textstyle\sum_{h\in\mathcal{P}_n} \mathbf{x}_h \le 1, \quad n\in\mathcal{L}, \\
& \textstyle\sum_{j\in\mathcal{J}_n} w_n^j = 1, \quad n\in\mathcal{N}, \qquad [\mu_n] \\
& w_n^j \in \{0,1\}, \quad j\in\mathcal{J}_n, \quad n\in\mathcal{N}, \\
& \mathbf{x}_n \in \{0,1\}^F, \quad n\in\mathcal{N}.
\end{aligned}
$$

Master problem matrix for 7 nodes (one expansion)

Node	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6	x1	x2	x3	x4	x5	x6
1	-1																																									
1		-1																																								
1			-1																																							
1				-1																																						
1					-1																																					
1						-1																																				
2	-1						-1																																			
2		-1						-1																																		
2			-1						-1																																	
2				-1						-1																																
2					-1						-1																															
2						-1						-1																														
3	-1												-1																													
3		-1												-1																												
3			-1												-1																											
3				-1												-1																										
3					-1												-1																									
3						-1												-1																								
4	-1						-1												-1																							
4		-1						-1												-1																						
4			-1						-1												-1																					
4				-1						-1												-1																				
4					-1						-1												-1																			
4						-1						-1												-1																		
5	-1						-1																		-1																	
5		-1						-1																		-1																
5			-1						-1																		-1															
5				-1						-1																		-1														
5					-1						-1																		-1													
5						-1						-1																		-1												
6	-1												-1																		-1											
6		-1												-1																		-1										
6			-1												-1																		-1									
6				-1												-1																		-1								
6					-1												-1																		-1							
6						-1												-1																		-1						
7	-1												-1																								-1					
7		-1												-1																								-1				
7			-1												-1																								-1			
7				-1												-1																								-1		
7					-1												-1																								-1	
7						-1												-1																								-1

Node	Generated columns giving feasible capacity expansion plans																													
1	0	1	0	0																									≤	0
1	0	0	0	0																									≤	0
1	0	0	0	0																									≤	0
1	0	0	0	0																									≤	0
1	1	1	1	0																									≤	0
1	0	1	0	0																									≤	0
2					0	0	1	0																					≤	0
2					0	0	0	1																					≤	0
2					1	0	0	1																					≤	0
2					0	1	0	1																					≤	0
2					0	0	0	0																					≤	0
2					0	1	0	0																					≤	0
3									0	0	0	0																	≤	0
3									0	0	1	0																	≤	0
3									0	1	0	1																	≤	0
3									1	0	1	0																	≤	0
3									0	0	1	0																	≤	0
3									1	0	1	0																	≤	0
4													0	1	0	0													≤	0
4													0	1	1	1													≤	0
4													0	0	1	0													≤	0
4													1	0	0	0													≤	0
4													0	0	0	0													≤	0
4													0	0	0	1													≤	0
5																	1	0	1	1									≤	0
5																	0	0	0	0									≤	0
5																	0	0	1	0									≤	0
5																	0	0	1	0									≤	0
5																	0	0	0	1									≤	0
5																	1	0	0	0									≤	0
6																					0	0	0	0					≤	0
6																					0	0	0	0					≤	0
6																					0	0	0	1					≤	0
6																					0	0	0	1					≤	0
6																					1	0	0	0					≤	0
6																					0	0	1	1					≤	0
7																									0	0	1	1	≤	0
7																									0	0	0	1	≤	0
7																									0	1	0	0	≤	0
7																									0	0	0	0	≤	0
7																									1	0	0	0	≤	0
7																									0	0	0	0	≤	0
	1	1	1	1																									=	1
					1	1	1	1																					=	1
									1	1	1	1																	=	1
													1	1	1	1													=	1
																	1	1	1	1									=	1
																					1	1	1	1					=	1
																									1	1	1	1	=	1

Matrix structure of restricted master problem

Master problem matrix for expanding knapsack problem

	0	1	0	0	0	1	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0			
Node	180	25	30	10	15	2.5	2.5	-75	-35	-30	-15	-33	-18	-19	-9	-21	-11	-18	-10	-21	-10	-131.25		
1	-1							1	0													0.00	≤	0
2	-1	-1								1	0											0.00	≤	0
3	-1		-1									1	0									0.00	≤	0
4	-1	-1		-1										1	0							0.00	≤	0
5	-1	-1			-1											1	0					0.00	≤	0
6	-1		-1			-1												1	0			0.00	≤	0
7	-1		-1				-1													1	0	0.00	≤	0
								1	1													1.00	=	1
										1	1											1.00	=	1
												1	1									1.00	=	1
														1	1							1.00	=	1
																1	1					1.00	=	1
																		1	1			1.00	=	1
																				1	1	1.00	=	1

Example showing all columns enumerated for expanding knapsack problem for 4 scenarios (7 nodes), and optimal master solution. Each column corresponds to the original optimal knapsack or an expanded knapsack. The problem has integer extreme points.

Bounds

Let

$$V^* = \text{optimal solution value}$$

$$U^* = \text{optimal solution value to restricted master problem with binary decisions}$$

$$R^* = \text{optimal solution value to LP relaxation of restricted master problem}$$

$$r_n = \text{optimal solution value for subproblem at node } n$$

Theorem

$$R^* - \sum_{n \in \mathcal{N}} r_n \leq V^* \leq U^*$$

Stabilize dual variables

- Dual variables from master problem often poorly behaved.
- Regularization improves performance.
- Best performance obtained using
  - cplex_options 'baropt';
  - cplex_options 'crossover = 0';

Summary

1 Introduction

2 The problem

3 A toy knapsack example

4 The general case

5 JuDGE.jl

6 Energy applications
- Electricity distribution networks
- Transmission expansion with imperfect competition
- EMERALD: zero-carbon electricity

7 Conclusion

JuDGE.jl

- JuDGE is a Julia package [Baucke, Downward, P. 2019]
- Complement to sddp.jl developed by Oscar Dowson. [Dowson and Kapelevich, 2017]
- User provides the stage problem and the scenario tree.
- JuDGE sets up the problem and solves it using Dantzig-Wolfe decomposition.
- Easily parallelized.

JuDGE data for the knapsack example

```
using JuDGE
numinvest = 1
numitems = 5
mydegree = 2
mydepth = 3
mutable struct Knapsack
 itemreward::Array{Float64,1}
 volume::Array{Float64,1}
 investcost::Array{Float64,1}
end
mytree = buildtree(depth=mydepth,degree=mydegree)
for i in 1:totalnodes
 mytree.nodes[i].data = Knapsack(itemcost[i,:], itemvolume[i,:],
investcost[i,:])
end
```

JuDGE output for knapsack: 4 scenarios

```
Documentation: https://docs.julialang.org
Type "?help" for help.

Version 0.6.2 (2017-12-13 18:08 UTC)
Official http://julialang.org/ release
x86_64-w64-mingw32

julia> cd("D:\\docs\\2019\\JuDGE\\example")

julia> include("generateSmallDataSet.jl")

julia> include("solveSmallKnapsack.jl")
----------------------------------------
Building model...  built.
----------------------------------------
Solving model...
----------------------------------------
-Inf Inf
-232.5 -36.25
-246.25 -131.25
-131.25 -131.25
----------------------------------------
Convergence criterion satisfied.
----------------------------------------
Expansion decisions: Node[1]
(1,) 0.0
Expansion decisions: Node[1, 1]
(1,) 1.0
Expansion decisions: Node[1, 1, 1]
(1,) 0.0
Expansion decisions: Node[1, 1, 2]
(1,) 0.0
Expansion decisions: Node[1, 2]
(1,) 0.0
Expansion decisions: Node[1, 2, 1]
(1,) 1.0
Expansion decisions: Node[1, 2, 2]
(1,) 1.0
```

JuDGE applied to the knapsack problem

Summary

1. Introduction
2. The problem
3. A toy knapsack example
4. The general case
5. JuDGE.jl
6. Energy applications
   - Electricity distribution networks
   - Transmission expansion with imperfect competition
   - EMERALD: zero-carbon electricity
7. Conclusion

Expansion of electricity distribution networks
[Singh, 2009]

- Electricity distribution networks operate as trees each rooted on a substation.
- Line faults dealt with by switching using a spare backstop line linking to another tree.
- Expand lines at least cost so that network can handle any line failure by switching ($N-1$ security).
- Two-stage stochastic MIP with N MIP subproblems.

Results

Fault Scenarios (number)	Extensive Models			D.W. Decomposition		
	SNDR-0 (CPU sec.)	SNDR-SN (CPU sec.)	SNDR-SN$_S$ (CPU sec.)	CG-0 (CPU sec.)	CG-SN$_S$ (CPU sec.)	CG-SN$_S$-I (CPU sec.)
1	**5.1**	4.5	2.1	11.3	4.1	**28.8**
2	**102.8**	7.6	7.5	116.9	35.4	**57.9**
3	**458.8**	135.1	21.9	224.6	92.9	**89.0**
4	-	2249.3	211.4	1122.0	381.5	**184.6**
5	-	1789.9	2039.0	-	617.4	**237.8**
6	-	-	762.3	-	441.1	**260.6**
7	-	-	3331.5	-	683.4	**316.2**
8	-	-	5285.1	-	2378.5	**389.1**
9	-	-	-	-	2240.4	**413.4**
10	-	-	-	-	4612.4	**551.3**
50	-	-	-	-	-	* **2487.0**
100						**9075.7**
150						**17881.8**
179						**22653.9**

Results from tests on 179 line distribution network. Blue results from cplex_options 'baropt'; cplex_options 'crossover = 0';

Sylvia Park instance

Capacity expansion in electricity networks with competition [Wogrin, Downward & P. 2019]

- Social planner seeks transmission capacity that maximizes expected social welfare.
- Consumers choose demand to maximize their welfare.
- Given transmission network, competing generators choose quantities to maximize their own profit.
- Generators act as oligopolists: conjectural variation μ varies from Cournot ($\mu = 2$) to perfect competition ($\mu = 1$).

Consumer welfare

The consumer at node k in transmission network chooses $d_k \geq 0$ to maximize consumer surplus

$$a_k d_k - \frac{b_k}{2} d_k^2 - \pi_k d_k$$

This gives KKT conditions at node k

$$0 \leq \pi_k - (a_k - b_k d_k) \perp d_k \geq 0.$$

Inverse demand function sets price

$$P_k(d_k) = a_k - b_k d_k$$

Generator profit

Generator i has a marginal production cost c_i, a given production capacity u_i and offers an amount x_i to the market. The optimization problem faced by agent i (who is located at node k) is

$$\begin{aligned} \mathrm{P}(i): \quad \min \quad & x_i(c_i - P_k(\textstyle\sum_{j\in k} x_j - \sum_l f_{kl} + \sum_l f_{lk})) \\ \text{s.t.} \quad & x_i \leq u_i, \\ & x_i \geq 0. \end{aligned}$$

$$\begin{aligned} 0 \quad \leq \quad & c_i - a_k + b_k(\sum_{j\neq i\in k} x_j - \sum_l f_{kl} + \sum_l f_{lk}) + \mu b_k x_i + \lambda_i \\ \perp \quad & x_i \geq 0 \\ 0 \quad \leq \quad & u_i - x_i \perp \lambda_i \geq 0. \end{aligned}$$

System operator

Given prices π, the system operator solves

$$
\begin{aligned}
\text{SO:}\ \min_{f,\theta} \quad & \textstyle\sum_{k,l} (\pi_k - \pi_l) f_{kl} \\
\text{s.t.} \quad & X_{kl} f_{kl} = \theta_k - \theta_l \\
& f_{kl} \le t_{kl}, && [\sigma] \\
& f_{kl} \ge -t_{kl}. && [\rho]
\end{aligned}
$$

$$
\begin{aligned}
0 &= \sigma_{kl} - \rho_{kl} - (\pi_k - \pi_l) \perp \theta_k \\
0 &\le t_{kl} - \frac{\theta_k - \theta_l}{X_{kl}} \perp \sigma_{kl} \ge 0 \\
0 &\le t_{kl} + \frac{\theta_k - \theta_l}{X_{kl}} \perp \rho_{kl} \ge 0.
\end{aligned}
$$

Market clearing

$$0 \leq \sum_{i \in k} x_i - \sum_{l} f_{kl} + \sum_{l} f_{lk} - d_k \perp \pi_k \geq 0.$$

Formulation in JuDGE

- In node $n \in \mathcal{N}$, each generator chooses an operating level that satisfies their KKT conditions, assuming that transmission quantities are fixed.
- In node $n \in \mathcal{N}$, each consumer chooses a demand level that satisfies their KKT conditions, assuming that transmission quantities are fixed.
- In node $n \in \mathcal{N}$, system operator chooses transmission quantities that satisfy their KKT conditions (to maximize surplus) assuming nodal prices are fixed.
- Subproblem is a MIP that seeks transmission capacities that maximize social welfare in node $n \in \mathcal{N}$, while paying or being paid for transmission capacity.
- JuDGE integrates transmission expansions into scenario tree $\mathcal{N}$, and maximizes expected social welfare.

Subproblem in node n

$$
\begin{aligned}
\text{SP}(n,\mu): \quad & \min_{t,x,d,\sigma,\rho,\pi,\lambda,} \textstyle\sum_{k,l} C_{kl} t_{kl} \\
\text{s.t.} \quad & 0 = \sigma_{kl} - \rho_{kl} - (\pi_k - \pi_l) \\
& 0 \leq T_{kl} + t_{kl} - \tfrac{\theta_k - \theta_l}{X_{kl}} \perp \sigma_{kl} \geq 0 \\
& 0 \leq T_{kl} + t_{kl} + \tfrac{\theta_k - \theta_l}{X_{kl}} \perp \rho_{kl} \geq 0. \\
& 0 \leq \pi_k - (a_k - b_k d_k) \perp d_k \geq 0 \\
& 0 \leq c_i - a_k + b_k \left(\textstyle\sum_{j \neq i \in k} x_j - \sum_l f_{kl} + \sum_l f_{lk}\right) + \mu b_k x_i + \lambda \\
& \perp x_i \geq 0 \\
& 0 \leq u_i - x_i \perp \lambda_i \geq 0. \\
& 0 \leq \textstyle\sum_{i \in k} x_i - \sum_l f_{kl} + \sum_l f_{lk} - d_k \perp \pi_k \geq 0.
\end{aligned}
$$

Subproblem in node n as a MIP

Complementarity constraints e.g.

$$0 \leq u_i - x_i \perp \lambda_i \geq 0$$

can be written as

$$\begin{aligned} \lambda_i &\leq Mz_i \\ u_i - x_i &\leq u_i(1 - z_i) \\ z_i &\in \{0, 1\} \end{aligned}$$

where M is an upper bound on λ_i.

EMERALD: 100% renewable energy

- What to build to make NZ electricity 100 percent renewable by 2035.
- Scenario tree models states of the world with changes in demand, technology, government policy.
- Each subproblem computes optimal operation of electricity system in state of the world.
- State of the world lasts five years (giving 4 stages) and involves hydro inflow and wind uncertainty.
- Example model has 4 branches per stage giving 64 scenarios.

JuDGE output for EMERALD: 64 scenarios

```
julia
               _
   _       _ _(_)_     |  Type "?help" for help.
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Version 0.6.2 (2017-12-13 18:08 UTC)
  | | | | | | |/ _` |  |  Official http://julialang.org/ release
  | | |_| | | | (_| |  |  x86_64-w64-mingw32
 _/ |\__'_|_|_|\__'_|  |
|__/                   |

julia> cd("D:\\docs\\2019\\JuDGE\\emerald")

julia> include("emerald5.jl")
---------------------------------------
Building model...  built.
---------------------------------------
Solving model...
---------------------------------------
LB = -Inf          UB = Inf
LB = 364508          UB = 862473
LB = 507504          UB = 821656
LB = 461194          UB = 821214
LB = 692815          UB = 816813
LB = 730664          UB = 816813
LB = 767779          UB = 816299
LB = 782914          UB = 812698
LB = 790045          UB = 805830
LB = 793798          UB = 804134
LB = 797269          UB = 802166
LB = 799554          UB = 802166
LB = 801303          UB = 802166
LB = 800668          UB = 802166
LB = 801224          UB = 802166
LB = 802166          UB = 802166
---------------------------------------
Convergence criterion satisfied.
---------------------------------------
Expansion decisions: Node[1]
((:NI, 4, 1),) 1.0
((:NI, 5, 1),) 1.0
Expansion decisions: Node[1, 1]
((:SI, 3, 1),) 1.0
```

JuDGE solution for 64 scenarios, 85 nodes.

Summary

1. Introduction
2. The problem
3. A toy knapsack example
4. The general case
5. JuDGE.jl
6. Energy applications
   - Electricity distribution networks
   - Transmission expansion with imperfect competition
   - EMERALD: zero-carbon electricity
7. Conclusion

Conclusion

- Stochastic capacity expansion via column generation.
- Leads to strong formulations of master problems, often naturally integer.
- Seldom need branch and price; manual branching often sufficient.
- MIP subproblems smaller than in scenario decomposition.
- Solving MIP subproblems fast is key to success.
- Subproblems can (easily) be solved in parallel to speed up the computation.
- JuDGE package (for Julia v0.6.2) available for free at https://github.com/reganbaucke/JuDGE.jl.

Conclusion

THE END

References

- Borison, A. B., Morris, P. A., & Oren, S. S. (1984). A state-of-the-world decomposition approach to dynamics and uncertainty in electric utility generation expansion planning. Operations Research, 32(5), 1052-1068.
- Carøe, C. C., & Schultz, R. (1999). Dual decomposition in stochastic integer programming. Operations Research Letters, 24(1-2), 37-45.
- Dentcheva, D., & Römisch, W. (2004). Duality gaps in nonconvex stochastic optimization. Mathematical Programming, 101(3), 515-535.
- Dowson, O., & Kapelevich, L. (2017). SDDP. jl: A Julia package for Stochastic Dual Dynamic Programming. Optimization Online.
- Fisher, E. B., O'Neill, R. P., & Ferris, M. C. (2008). Optimal transmission switching. IEEE Transactions on Power Systems, 23(3), 1346-1355.
- Galati, M. (2009). Coin-or dip (decomposition for integer programming).

References

- Hedman, K. W., O'Neill, R. P., Fisher, E. B., & Oren, S. S. (2009). Optimal transmission switching with contingency analysis. IEEE Transactions on Power Systems, 24(3), 1577-1586
- Kaut, M., Midthun, K. T., Werner, A. S., Tomasgard, A., Hellemo, L., & Fodstad, M. (2014). Multi-horizon stochastic programming. Computational Management Science, 11(1-2), 179-193.
- Philpott, A. B., Wahid, F., & Bonnans, J. F. (2019). MIDAS: A mixed integer dynamic approximation scheme. Mathematical Programming, 1-32.
- Rockafellar, R. T., & Wets, R. J. B. (1991). Scenarios and policy aggregation in optimization under uncertainty. Mathematics of operations research, 16(1), 119-147.
- Singh, K. J., Philpott, A. B., & Wood, R. K. (2009). Dantzig-Wolfe decomposition for solving multistage stochastic capacity-planning problems. Operations Research, 57(5), 1271-1286.

References

- Villumsen, J. C., & Philpott, A. B. (2012). Investment in electricity networks with transmission switching. European Journal of Operational Research, 222(2), 377-385.
- Watson, J. P., & Woodruff, D. L. (2011). Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. Computational Management Science, 8(4), 355-370.
- Zou, J., Ahmed, S., & Sun, X. A. (2018). Stochastic dual dynamic integer programming. Mathematical Programming, 1-42.

Knapsack example

```
  (_)     | (_) (_)    |  Documentation: https://docs.julialang.org
   _ _   _| |_  __ _   |  Type "?help" for help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 0.6.2 (2017-12-13 18:08 UTC)
 _/ |\__'_|_|_|\__'_|  |  Official http://julialang.org/ release
|__/                   |  x86_64-w64-mingw32

julia> cd("D:\\docs\\2019\\JuDGE\\example")

julia> include("generateSmallDataSet.jl")

julia> include("solveSmallKnapsack.jl")
---------------------------------------
Building model...  built.
---------------------------------------
Solving model...
---------------------------------------
-Inf Inf
-232.5 -36.25
-246.25 -131.25
-131.25 -131.25
---------------------------------------
Convergence criterion satisfied.
---------------------------------------
Expansion decisions: Node[1]
(1,) 0.0
Expansion decisions: Node[1, 1]
(1,) 1.0
Expansion decisions: Node[1, 1, 1]
(1,) 0.0
Expansion decisions: Node[1, 1, 2]
(1,) 0.0
Expansion decisions: Node[1, 2]
(1,) 0.0
Expansion decisions: Node[1, 2, 1]
(1,) 1.0
```

Capacity expansion in transmission networks with switches [Villumsen & P. 2012]

- In electricity transmission networks switching out transmission lines can lower dispatch cost.
- MIP models for this have been developed [Fisher *et al*, 2008], [Hedman, 2010]
- Two-stage model
  - first stage invests in transmission switches on existing lines;
  - second stage switches to decrease cost in each scenario;
  - minimize overall annual capital cost minus expected annual saving from switching.
- To resolve any fractions we use the COIN DIP branch-and-price code. [Galati, 2009]

Computational results of column generation

Instance		Branch-and-price-and-cut					Branch-and-bound		
		time (s)		price-	cut-	gap	time (s)	gap	lower
$\lvert\omega\rvert$	k	total	master	passes	passes				bound
2	3	96	0	10	0	0	547	0	1351.22
2	5	1427	1	20	0	0	330	0	1338.55
4	3	257	4	16	0	0	2310	0	1036.93
4	5	3172	6	38	0	0	7133	0	1033.95
4	10	22444	30	98	0	0	3055	0	1009.60
8	3	978	8	26	1	0	2070 †	2.26	846.17
16	3	3213	11	36	0	0	3722 †	0	775.01
32	3	25477	279	129	2	0	4968 †	9.68	690.81
64	3	11126	72	38	0	0	8589 †	28.84	678.44

Installing transmission switches with random demand on IEEE118-bus network (185 lines) in a two-stage model (Villumsen and P., 2012)