EMERALD: a green GEM

Andy Philpott

Joint work with Anthony Downward and Regan Baucke

Electric Power Optimization Centre
University of Auckland
www.epoc.org.nz

EPOC Winter Workshop, Auckland, August 28, 2023.

What is EMERALD?

- EMERALD is EPOC's stochastic version of the Electricity Authority GAMS package GEM.

What is EMERALD?

- EMERALD is EPOC's stochastic version of the Electricity Authority GAMS package GEM.
- used to study the effect of New Zealand carbon policy on optimal investment in generation capacity over a 30 year planning horizon with future uncertainty.

What is EMERALD?

- EMERALD is EPOC's stochastic version of the Electricity Authority GAMS package GEM.
- used to study the effect of New Zealand carbon policy on optimal investment in generation capacity over a 30 year planning horizon with future uncertainty.
- applies the JuDGE.jl Julia package to solve a multistage stochastic program in a scenario tree.

What is EMERALD?

- EMERALD is EPOC's stochastic version of the Electricity Authority GAMS package GEM.
- used to study the effect of New Zealand carbon policy on optimal investment in generation capacity over a 30 year planning horizon with future uncertainty.
- applies the JuDGE.jl Julia package to solve a multistage stochastic program in a scenario tree.
- JuDGE visualizations enable one to explore the optimal expansion plan.

What is EMERALD?

- EMERALD is EPOC's stochastic version of the Electricity Authority GAMS package GEM.
- used to study the effect of New Zealand carbon policy on optimal investment in generation capacity over a 30 year planning horizon with future uncertainty.
- applies the JuDGE.jl Julia package to solve a multistage stochastic program in a scenario tree.
- JuDGE visualizations enable one to explore the optimal expansion plan.
- This talk presents a summary and some results of EMERALD.

Outline

Multi-horizon modelling and JuDGE

- Multi-horizon stochastic programming and capacity planning
- The JuDGE package

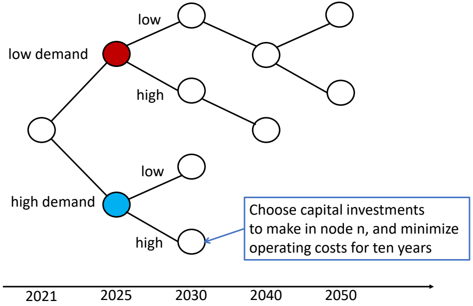
EMERALD demonstration

- The New Zealand model
- Demonstration
- Results

Central planning versus commercial investment

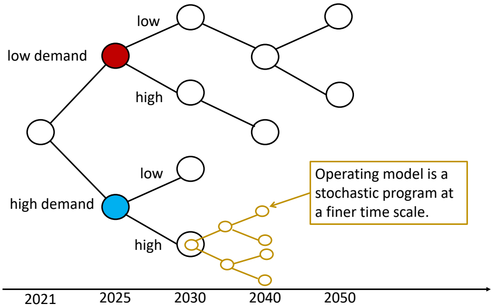
Outline

Multi-horizon planning

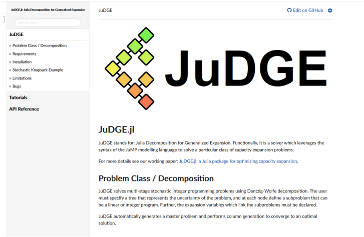


Capacity-expansion decisions over longer time scale (5 years or 10 years) result in lower operational costs, or higher revenue in the future.

Multi-horizon scenario trees



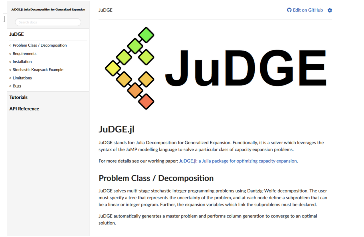
Operational decisions with short-term uncertainty optimized by a stochastic program.



`https://github.com/EPOC-NZ/JuDGE.jl`

JuDGE stands for Julia Decomposition for Generalized Expansion.).

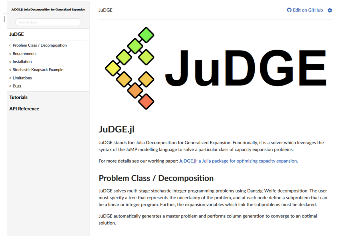
- allows users to easily implement multi-horizon optimization models using the JuMP modelling language;
- can apply end-of-horizon risk-measures in objective function and/or the constraints; and
- outputs an interactive view of the results over the scenario tree, enabling decision makers to explore the optimal expansion plan.



https://github.com/EPOC-NZ/JuDGE.jl

JuDGE stands for Julia Decomposition for Generalized Expansion.).

- allows users to easily implement multi-horizon optimization models using the JuMP modelling language;
- can apply end-of-horizon risk-measures in objective function and/or the constraints; and
- outputs an interactive view of the results over the scenario tree, enabling decision makers to explore the optimal expansion plan.



`https://github.com/EPOC-NZ/JuDGE.jl`

JuDGE stands for Julia Decomposition for Generalized Expansion.).

- allows users to easily implement multi-horizon optimization models using the JuMP modelling language;
- can apply end-of-horizon risk-measures in objective function and/or the constraints; and
- outputs an interactive view of the results over the scenario tree, enabling decision makers to explore the optimal expansion plan.

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

---

[2] Singh, P. & Wood, *Operations Research, 2009.*

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

[2] Singh, P. & Wood, *Operations Research, 2009.*

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

[2]Singh, P. & Wood, *Operations Research, 2009.*

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

[2]Singh, P. & Wood, *Operations Research, 2009.*

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

[2] Singh, P. & Wood, *Operations Research, 2009.*

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

[2] Singh, P. & Wood, *Operations Research, 2009.*

JuDGE modelling framework

To apply JuDGE we require...

- a tree with corresponding data and probabilities for each node;
- a subproblem defined as a JuMP model for each node in the tree; and
- expansion (and/or shutdown) decisions and costs;
- a choice of solver for master and subproblem.

JuDGE automatically forms a restricted master problem, and applies Dantzig-Wolfe decomposition.[2]

The LP relaxation of the restricted master problem is typically solved with an interior point method, and the subproblems are solved as mixed-integer programs.

JuDGE can formulate the deterministic equivalent problem directly as a JuMP model (mixed-integer program).

[2] Singh, P. & Wood, *Operations Research, 2009.*

Outline

The New Zealand model

Expansions and shutdowns

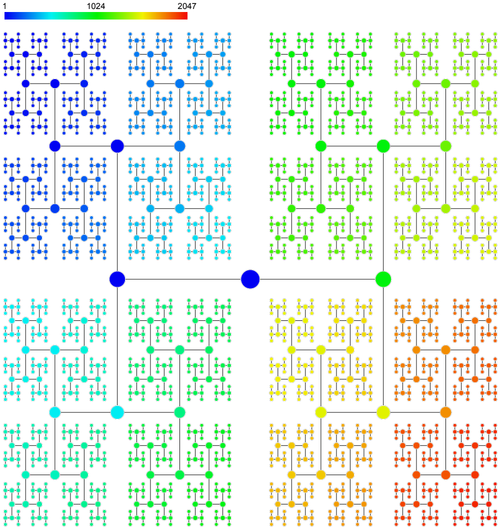
Optimize capacity expansion in response to uncertainty represented by a scenario tree.

Model is a risk-averse central-planning model minimizing discounted disbenefit Z summed from 2021-2050.

Risk is modelled using a convex combination of expected value and average value at risk, so Risk(λ, α) is

$$(1 - \lambda)\mathbb{E}[Z] + \lambda AVaR_{1-\alpha}[Z]$$

.



2047-node scenario tree.

The New Zealand model

Expansions and shutdowns

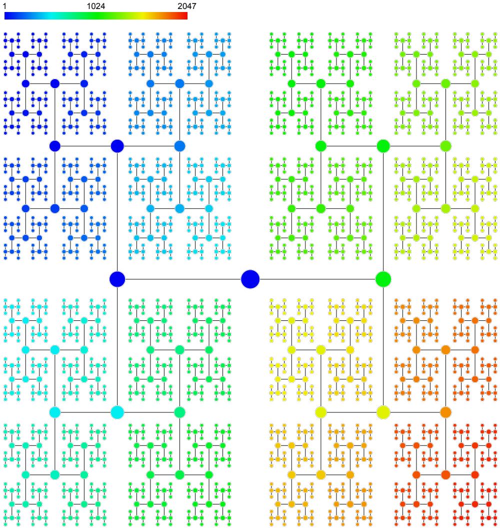
Optimize capacity expansion in response to uncertainty represented by a scenario tree.

Model is a risk-averse central-planning model minimizing discounted disbenefit Z summed from 2021-2050.

Risk is modelled using a convex combination of expected value and average value at risk, so Risk(λ, α) is

$$(1 - \lambda)\mathbb{E}[Z] + \lambda AVaR_{1-\alpha}[Z]$$

.



2047-node scenario tree.

The New Zealand model

Expansions and shutdowns

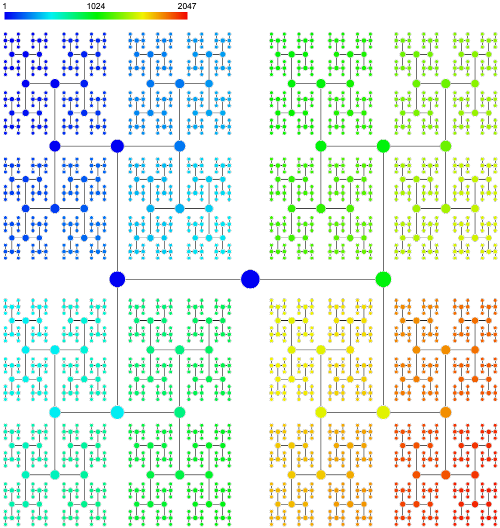
Optimize capacity expansion in response to uncertainty represented by a scenario tree.

Model is a risk-averse central-planning model minimizing discounted disbenefit Z summed from 2021-2050.

Risk is modelled using a convex combination of expected value and average value at risk, so Risk(λ, α) is

$$(1-\lambda)\mathbb{E}[Z] + \lambda AVaR_{1-\alpha}[Z]$$

.



2047-node scenario tree.

The New Zealand model

Defining the subproblems

Sets:

- seasons $t \in \mathcal{T}$;
- load blocks $b \in \mathcal{B}_t$, $t \in \mathcal{T}$;
- hydrological years $h \in \mathcal{H}$;
- technologies $k \in \mathcal{K}$.

The New Zealand model

Defining the subproblems

Sets:

- seasons $t \in \mathcal{T}$;
- load blocks $b \in \mathcal{B}_t$, $t \in \mathcal{T}$;
- hydrological years $h \in \mathcal{H}$;
- technologies $k \in \mathcal{K}$.

Variables:

- x_k capacity to build for technology k;
- g_k^{bh} generation from technology k in load block b, with hydrological year h.

The New Zealand model

Defining the subproblems

Sets:

- seasons $t \in \mathcal{T}$;
- load blocks $b \in \mathcal{B}_t$, $t \in \mathcal{T}$;
- hydrological years $h \in \mathcal{H}$;
- technologies $k \in \mathcal{K}$.

Variables:

- x_k capacity to build for technology k;
- g_k^{bh} generation from technology k in load block b, with hydrological year h.

Parameters:

- d^b demand in load block b;
- u_k initial capacity of technology k;
- U_k maximum capacity increment of each new technology k;
- θ_k^b is the capacity factor for technology k, in load block b.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;
ρ_h is the probability of hydrological year h;
c_k is the marginal cost of technology k;
e_k gives the emissions factor of technology k;
τ is the carbon tax.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;

ρ_h is the probability of hydrological year h;

c_k is the marginal cost of technology k;

e_k gives the emissions factor of technology k;

τ is the carbon tax.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;

ρ_h is the probability of hydrological year h;

c_k is the marginal cost of technology k;

e_k gives the emissions factor of technology k;

τ is the carbon tax.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;

- ρ_h is the probability of hydrological year h;
- c_k is the marginal cost of technology k;
- e_k gives the emissions factor of technology k;
- τ is the carbon tax.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t\in\mathcal{T}} \sum_{b\in\mathcal{B}_t} \Delta_b \sum_{h\in\mathcal{H}} \rho_h \sum_{k\in\mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;
ρ_h is the probability of hydrological year h;
c_k is the marginal cost of technology k;
e_k gives the emissions factor of technology k;
τ is the carbon tax.

Cost of investments over the tree:

$$\min \quad \sum_{n\in\mathcal{N}} \phi_n \sum_{k\in\mathcal{K}} C_k x_k,$$

ϕ_n is the (discounted) probability of reaching node n;
C_k is the capital cost (per unit) of technology k;
$x_k \in [0,1]$ represents investment in technology k.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;
ρ_h is the probability of hydrological year h;
c_k is the marginal cost of technology k;
e_k gives the emissions factor of technology k;
τ is the carbon tax.

Cost of investments over the tree:

$$\min \quad \sum_{n \in \mathcal{N}} \phi_n \sum_{k \in \mathcal{K}} C_k x_k,$$

ϕ_n is the (discounted) probability of reaching node n;
C_k is the capital cost (per unit) of technology k;
$x_k \in [0, 1]$ represents investment in technology k.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;
ρ_h is the probability of hydrological year h;
c_k is the marginal cost of technology k;
e_k gives the emissions factor of technology k;
τ is the carbon tax.

Cost of investments over the tree:

$$\min \quad \sum_{n \in \mathcal{N}} \phi_n \sum_{k \in \mathcal{K}} C_k x_k,$$

ϕ_n is the (discounted) probability of reaching node n;
C_k is the capital cost (per unit) of technology k;
$x_k \in [0, 1]$ represents investment in technology k.

Medium-term Operational Model

Objective Functions

Subproblem at node n minimizes the operational costs of the electricity system:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{b \in \mathcal{B}_t} \Delta_b \sum_{h \in \mathcal{H}} \rho_h \sum_{k \in \mathcal{K}} (c_k + \tau e_k) g_k^{bh},$$

where Δ_b is the number of hours corresponding to load block b;
ρ_h is the probability of hydrological year h;
c_k is the marginal cost of technology k;
e_k gives the emissions factor of technology k;
τ is the carbon tax.

Cost of investments over the tree:

$$\min \quad \sum_{n \in \mathcal{N}} \phi_n \sum_{k \in \mathcal{K}} C_k x_k,$$

ϕ_n is the (discounted) probability of reaching node n;
C_k is the capital cost (per unit) of technology k;
$x_k \in [0, 1]$ represents investment in technology k.

Medium-term operations

Subproblem constraints

Load balance:

$$\sum_{k \in \mathcal{K}} g_k^{bh} = d^b, \quad \forall b \in \mathcal{B}, h \in \mathcal{H},$$

Generation capacity:

$$0 \leq g_k^{bh} \leq \theta_k^b (u_k + x_k U_k) \quad \forall b \in \mathcal{B}_t, t \in \mathcal{T}, h \in \mathcal{H}, k \in \mathcal{K},$$

Stored hydro generation:

$$\sum_{b \in \mathcal{B}_t} g_{\text{hydro}}^{bh} \times \Delta_b = \mu_t^h \quad \forall h \in \mathcal{H}, t \in \mathcal{T},$$

Expansions:

$$x_k \in [0, 1], \quad \forall k \in \mathcal{K}, i \in \{1, \ldots, N\}.$$

EMERALD demonstration

EMERALD case study uses...

- Three regions (NI, HAY, SI).
- Four seasons with 10 load blocks each.
- 16 load growth scenarios.
- 13 historical years model seasonal hydrological inflows.
- Data based on two-stage model of NZ system.[3]

[3] Ferris & Philpott, 100% renewable electricity with storage (2019) `http://www.epoc.org.nz`.

EMERALD input data

Demand and carbon price scenarios are related

- Annual total energy demand increases from
  - Electric vehicles;
  - Industrial load;
  - Consumer load;
  - Tiwau (or replacement).
- NZ CCC CO_2-e budgets in target years are assumed.
- CO_2-e budgets affect carbon prices.
- Carbon prices affect fossil fuels and electricity prices.
- Electric vehicle demand = f(gasoline price,electricity price).

EMERALD input data

Demand and carbon price scenarios are related

- Annual total energy demand increases from
  - Electric vehicles;
  - Industrial load;
  - Consumer load;
  - Tiwau (or replacement).
- NZ CCC CO_2-e budgets in target years are assumed.
- CO_2-e budgets affect carbon prices.
- Carbon prices affect fossil fuels and electricity prices.
- Electric vehicle demand = f(gasoline price,electricity price).

EMERALD input data

Demand and carbon price scenarios are related

- Annual total energy demand increases from
  - Electric vehicles;
  - Industrial load;
  - Consumer load;
  - Tiwau (or replacement).
- NZ CCC CO_2-e budgets in target years are assumed.
- CO_2-e budgets affect carbon prices.
- Carbon prices affect fossil fuels and electricity prices.
- Electric vehicle demand = f(gasoline price,electricity price).

EMERALD input data

Demand and carbon price scenarios are related

- Annual total energy demand increases from
  - Electric vehicles;
  - Industrial load;
  - Consumer load;
  - Tiwau (or replacement).
- NZ CCC CO_2-e budgets in target years are assumed.
- CO_2-e budgets affect carbon prices.
- Carbon prices affect fossil fuels and electricity prices.
- Electric vehicle demand = f(gasoline price,electricity price).

EMERALD input data

Demand and carbon price scenarios are related

- Annual total energy demand increases from
  - Electric vehicles;
  - Industrial load;
  - Consumer load;
  - Tiwau (or replacement).
- NZ CCC CO_2-e budgets in target years are assumed.
- CO_2-e budgets affect carbon prices.
- Carbon prices affect fossil fuels and electricity prices.
- Electric vehicle demand = f(gasoline price,electricity price).

EMERALD input data

Scenario tree for demand and carbon price

```
mytree, data = tree_with_data(myscenariotree.csv)

 n,p,EVTWh,industryTWh,consumerTWh,TiwauTWh,carbon
 1,-,0.1,8.525,27.727,5.475,50
 11,1,0.1389,10.750025,32.16332,5.475,50
 12,1,0.1389,11.50875,29.168804,5.475,50
 111,11,0.55,12.276,35.49056,5.475,200
 112,11,0.55,11.227425,28.55881,5.475,200
 121,12,0.55,13.14555,32.191047,5.475,200
 122,12,0.55,12.028775,25.897018,5.475,200
 1111,111,5,15.8565,39.566429,5.475,500
 1121,112,5,14.50955,31.802869,5.475,500
 ....
```

EMERALD input data

Scenario tree for demand and carbon price

```
mytree, data = tree_with_data(myscenariotree.csv)
```

```
n,p,EVTWh,industryTWh,consumerTWh,TiwauTWh,carbon
1,-,0.1,8.525,27.727,5.475,50
11,1,0.1389,10.750025,32.16332,5.475,50
12,1,0.1389,11.50875,29.168804,5.475,50
111,11,0.55,12.276,35.49056,5.475,200
112,11,0.55,11.227425,28.55881,5.475,200
121,12,0.55,13.14555,32.191047,5.475,200
122,12,0.55,12.028775,25.897018,5.475,200
1111,111,5,15.8565,39.566429,5.475,500
1121,112,5,14.50955,31.802869,5.475,500
....
```

```
JuDGE.visualize_tree(mytree, data)
```

Scenario tree

Creating the JuDGE model

```
model = JuDGEModel(mytree,
          ConditionallyUniformProbabilities,
          sub_problems,
          JuDGE_MP_Solver,
          discount_factor=0.92)
          risk=Risk(0.95,(1/16))
```

Creating the JuDGE model

```
model = JuDGEModel(mytree,
          ConditionallyUniformProbabilities,
          sub_problems,
          JuDGE_MP_Solver,
          discount_factor=0.92)
          risk=Risk(0.95,(1/16))
```

Creating the JuDGE model

```
model = JuDGEModel(mytree,
          ConditionallyUniformProbabilities,
          sub_problems,
          JuDGE_MP_Solver,
          discount_factor=0.92)
          risk=Risk(0.95,(1/16))
```

Creating the JuDGE model

```
model = JuDGEModel(mytree,
          ConditionallyUniformProbabilities,
          sub_problems,
          JuDGE_MP_Solver,
          discount_factor=0.92)
          risk=Risk(0.95,(1/16))
```

Creating the JuDGE model

```
model = JuDGEModel(mytree,
          ConditionallyUniformProbabilities,
          sub_problems,
          JuDGE_MP_Solver,
          discount_factor=0.92)
          risk=Risk(0.95,(1/16)))
```

Running EMERALD

Solving and producing output

```
JuDGE.solve(model,termination=Termination(reltol=0.001))

resolve_subproblems(model)

solution = JuDGE.solution_to_dictionary(model)

(some code to set up custom_plots using plotly)

JuDGE.visualize_tree(mytree, solution,
custom=custom_plots)
```

Running EMERALD

Solving and producing output

```
JuDGE.solve(model,termination=Termination(reltol=0.001))
resolve_subproblems(model)
solution = JuDGE.solution_to_dictionary(model)
(some code to set up custom_plots using plotly)
JuDGE.visualize_tree(mytree, solution,
custom=custom_plots)
```

EMERALD results

Outline

Models for optimal planning

1. Computable general equilibrium models (e.g. C-PLAN);

Models for optimal planning

1. Computable general equilibrium models (e.g. C-PLAN);
2. Agent simulation models (e.g. ENZ);

Models for optimal planning

1. Computable general equilibrium models (e.g. C-PLAN);
2. Agent simulation models (e.g. ENZ);
3. Deterministic planning models (e.g. GEM);

Models for optimal planning

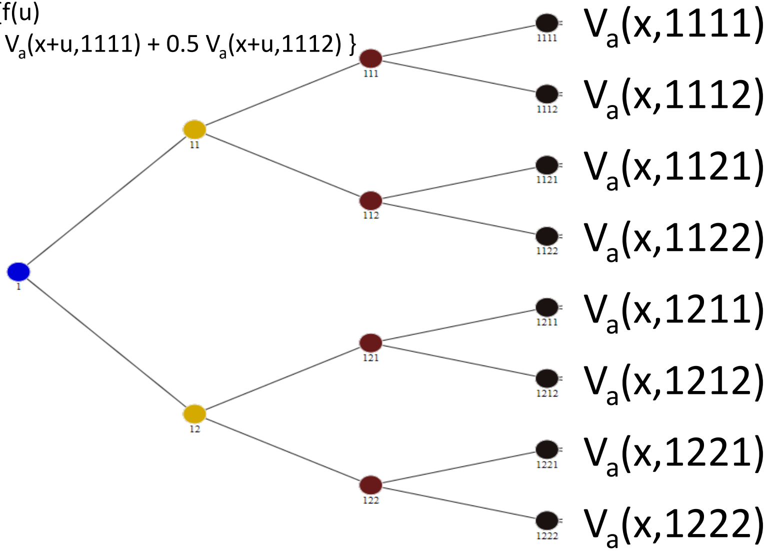
1. Computable general equilibrium models (e.g. C-PLAN);
2. Agent simulation models (e.g. ENZ);
3. Deterministic planning models (e.g. GEM);
4. Stochastic, risk-averse planning models (e.g. EMERALD);

Models for optimal planning

1. Computable general equilibrium models (e.g. C-PLAN);
2. Agent simulation models (e.g. ENZ);
3. Deterministic planning models (e.g. GEM);
4. Stochastic, risk-averse planning models (e.g. EMERALD);
5. Do models (2), (3), (4) yield dynamic investment (partial) equilibrium?

Dynamic investment equilibrium by backward induction

$$V_a(x,111) = \max\{f(u) + 0.5\, V_a(x+u,1111) + 0.5\, V_a(x+u,1112)\}$$



Dynamic investment equilibrium = EMERALD

- Optimal risk-averse plan from EMERALD matches partial equilibrium when risk measures are coherent and risk-trading instruments are available.[4]
- Each agent in EMERALD has their own coherent risk measure. This corresponds to a nested risk measure with single-stage risk sets that vary with node.
- What model of social risk model should we use in EMERALD? JuDGE uses an end-of-horizon risk measure.
- NOTE: the intersection of agent risk sets define a nested risk measure for the social planner that might not be an end-of-horizon measure.

[4] Ralph & Smeers, SIOPT, 2015, Ferris & P., Operations Research, 2022.

The End

`JuDGE.jl` Julia Library downloadable from

`https://github.com/EPOC-NZ/JuDGE.jl`

My contact: `a.philpott@auckland.ac.nz`
Technical questions to: `a.downward@auckland.ac.nz`