JADE: Overview and applications.

Tony Downward & Andy Philpott

Electric Power Optimization Centre

a.downward@auckland.ac.nz

EPOC Winter Workshop 2022, 9 September 2022.

Outline

Overview of JADE

- Model / assumptions
- Inputs and training
- Setting up a JADE model

Lake Onslow

- 100% renewables and security of supply
- Specifications
- Results and Analysis

Other JADE applications

- Hydrogen integration
- Fuel availability
- Stagewise-dependent inflows

Outline

JADE model details

Model structure and assumptions

The JADE package is a Julia package that implements a multistage stochastic program, which formulated and solved using SDDP.jl[2].

[2] Dowson, O. and Kapelevich, L. SDDP.jl: a Julia package for stochastic dual dynamic programming, INFORMS Journal on Computing, 33(1) 27–33 (2021).

JADE model details

Model structure and assumptions

The JADE package is a Julia package that implements a multistage stochastic program, which formulated and solved using SDDP.jl[2].

When a JADE model is optimized, this involves finding a water value surface for each week in the year, and corresponding marginal water values for each reservoir.

[2] Dowson, O. and Kapelevich, L. SDDP.jl: a Julia package for stochastic dual dynamic programming, INFORMS Journal on Computing, 33(1) 27–33 (2021).

JADE model details

Model structure and assumptions

The JADE package is a Julia package that implements a multistage stochastic program, which formulated and solved using SDDP.jl[2].

When a JADE model is optimized, this involves finding a water value surface for each week in the year, and corresponding marginal water values for each reservoir.

The model seeks to determine optimal release policies as functions of the amount water in the reservoirs, accounting for future demand, as well as future expected inflows.

[2] Dowson, O. and Kapelevich, L. SDDP.jl: a Julia package for stochastic dual dynamic programming, INFORMS Journal on Computing, 33(1) 27–33 (2021).

JADE model details

Model structure and assumptions

The JADE package is a Julia package that implements a multistage stochastic program, which formulated and solved using SDDP.jl[2].

When a JADE model is optimized, this involves finding a water value surface for each week in the year, and corresponding marginal water values for each reservoir.

The model seeks to determine optimal release policies as functions of the amount water in the reservoirs, accounting for future demand, as well as future expected inflows.

When setting up a JADE model, you can specify risk aversion parameters and the set of historical years of inflows to include in the future outlook.

---

[2] Dowson, O. and Kapelevich, L. SDDP.jl: a Julia package for stochastic dual dynamic programming, INFORMS Journal on Computing, 33(1) 27–33 (2021).

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

These include:

- demand blocks, fixed generation and load shedding costs;

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

These include:

- demand blocks, fixed generation and load shedding costs;
- historical inflows, hydro network arcs + constraints;

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

These include:

- demand blocks, fixed generation and load shedding costs;
- historical inflows, hydro network arcs + constraints;
- hydro stations (capacity + efficiency, and reservoir limits);

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

These include:

- demand blocks, fixed generation and load shedding costs;
- historical inflows, hydro network arcs + constraints;
- hydro stations (capacity + efficiency, and reservoir limits);
- thermal stations (capacity + heat rate and fuel costs);

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

These include:

- demand blocks, fixed generation and load shedding costs;
- historical inflows, hydro network arcs + constraints;
- hydro stations (capacity + efficiency, and reservoir limits);
- thermal stations (capacity + heat rate and fuel costs);
- transmission network (line capacities + losses); and

JADE model details

Input data

JADE has inherited the input file format from DOASA, but is somewhat more flexible:

There are 14 core input files that JADE reads in when building the model of the electricity system.

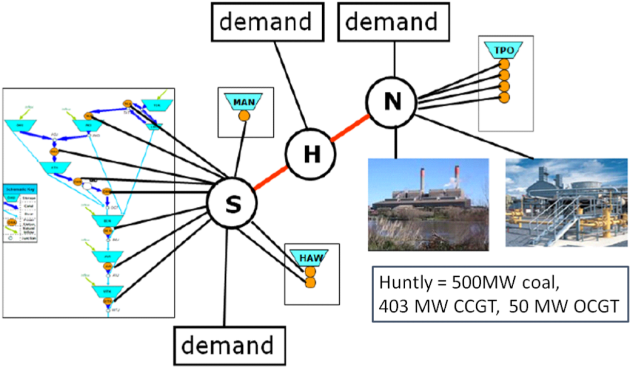
These include:

- demand blocks, fixed generation and load shedding costs;
- historical inflows, hydro network arcs + constraints;
- hydro stations (capacity + efficiency, and reservoir limits);
- thermal stations (capacity + heat rate and fuel costs);
- transmission network (line capacities + losses); and
- outages of stations and transmission lines.

JADE model details

Default model

The flexible input files enable you to define any hydro-dominant electricity system; however, the files that the EA will make available are calibrated to the current New Zealand electricity sector.



JADE model network.

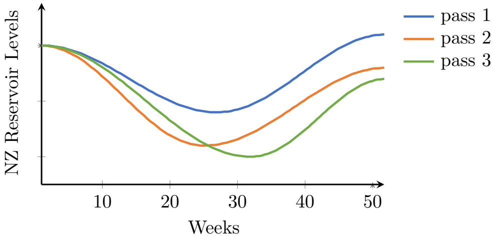
Finding optimal JADE policies

Steady-state policies

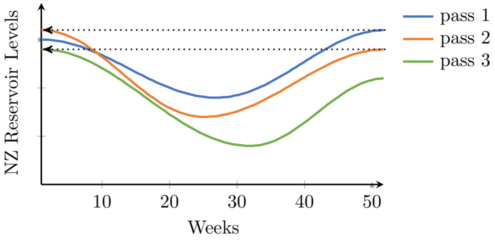
JADE utilises and builds upon the infinite-horizon features of SDDP.jl to enable the optimization and simulation of steady-state policies.

These policies do not rely on any end-of-horizon marginal water values, and instead determine all the water values endogenously.

Computationally, modelling the steady-state policy requires a user-defined discount factor to be set. Values close to 1.0 require more iterations to converge.



Standard SDDP

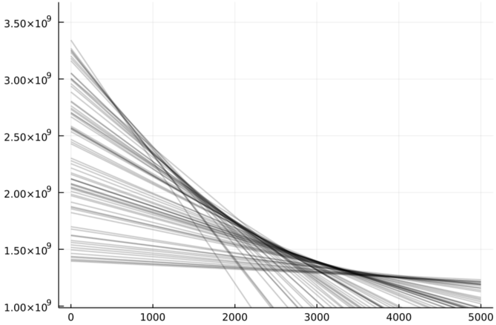


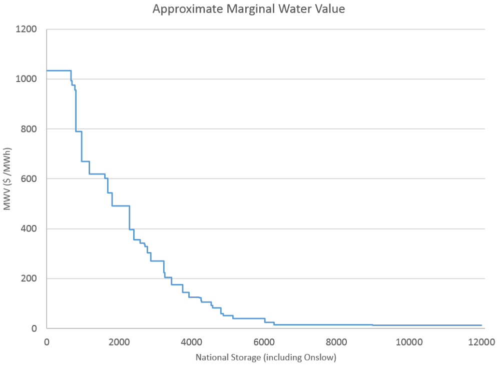
Steady-state SDDP

JADE model details

JADE future costs and marginal water values

A policy in JADE is defined by the future cost function (expected cost to go). The gradient of this function gives marginal water values that yield the optimal dispatch.





JADE package

Setting up a JADE model

The JADE package for Julia will be released (open-source) this month.

JADE package

Setting up a JADE model

The JADE package for Julia will be released (open-source) this month.

Since it is built using SDDP.jl, which uses JuMP, it can be used with a wide range of linear programming solvers.

JADE package

Setting up a JADE model

The JADE package for Julia will be released (open-source) this month.

Since it is built using SDDP.jl, which uses JuMP, it can be used with a wide range of linear programming solvers.

Although you can solve and simulate JADE models with open-source solvers, commercial solvers are recommended for large-scale models – e.g. Gurobi or CPLEX.

JADE package

Setting up a JADE model

The JADE package for Julia will be released (open-source) this month.

Since it is built using SDDP.jl, which uses JuMP, it can be used with a wide range of linear programming solvers.

Although you can solve and simulate JADE models with open-source solvers, commercial solvers are recommended for large-scale models – e.g. Gurobi or CPLEX.

We provide an interface so users can easily set up and run JADE models.

JADE package

Setting up a JADE model

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

Model Settings

Policy name	policy1
Scenario directory	
Problem start year	2019
Problem start week	1
Number of weeks	52
Steady state discount factor	0.92 (8.7%)
Sample start year	1970
Sample end year	2019
Inflow correlation length	3
LB flow penalty	500
UB flow penalty	50
Scale objective	1.00E+06
Scale reservoirs	1

Training Options

Iterations	100
Load saved cuts from	
Write EOH cuts file	FALSE
Load EOH cuts file	N/A
% MC in forward pass	100.0%
Custom inflow sequences	
Cut selection	

Simulation Settings

Simulation name	sim1
Simulation type	Historical (Sequential)
Reset initial storage levels	TRUE
Replications	N/A
Simulation start year	1970
Simulation end year	2019
Random seed	N/A

JADE

First: Find JADE Input Files
Last: Create JADE Run File

JADE package

Running up a JADE model

```julia
## This file should be in a directory containing an Input directory, which has a directory
## called <data_dir> containing the JADE input files.

using JADE, JuMP

## Choose your solver
using Gurobi
env = Gurobi.Env()
optimizer = optimizer_with_attributes(() -> Gurobi.Optimizer(env), "OutputFlag" => 0)

#using CPLEX
#optimizer = optimizer_with_attributes(CPLEX.Optimizer, "CPX_PARAM_SCRIND" => 0)

#using GLPK
#optimizer = optimizer_with_attributes(GLPK.Optimizer, "msg_lev" => 0)

## Set directory containing the Input / Output subdirectories
@eval JADE JADE_DIR = @__DIR__

## Modify these settings
data_dir = "test1"
runfile = "run"
training = true
simulation = true
```

JADE package

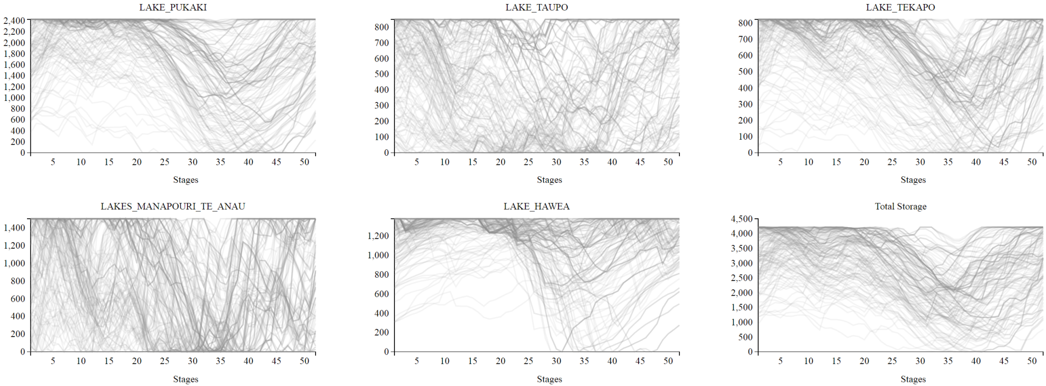
Running up a JADE model

```julia
## Set up inputs for JADE models
rundata = define_JADE_model(data_dir, run_file = runfile)
rundata.use_terminal_mwvs = true
## Create JADE model from the runfile data
model = create_JADE_model(rundata, optimizer)

if training
    ## Solve options
    solve_options = define_JADE_solve_options(data_dir, run_file = runfile)

    ## Optimize and store policies
    optimize_policy!(model, solve_options)
end

if simulation
    ## Perform simulation
    sim_settings = define_JADE_simulation(data_dir, run_file = runfile)
    results = simulate(model, sim_settings)
end

## Visualise simulation
JADE.plot_storage(results, data_dir * "-" * runfile * "-" * sim_settings.sim_dir)
JADE.plot_prices(results, data_dir * "-" * runfile * "-" * sim_settings.sim_dir)
```

JADE package

Raw SDDP.jl simulation output

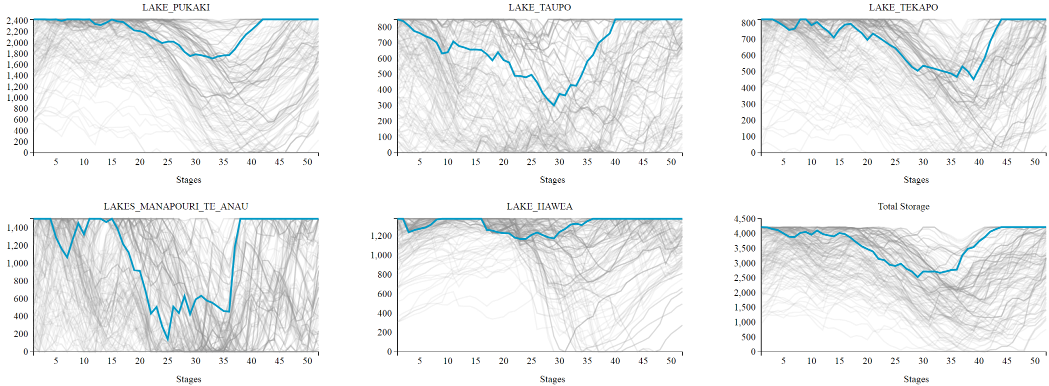
THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND



JADE package

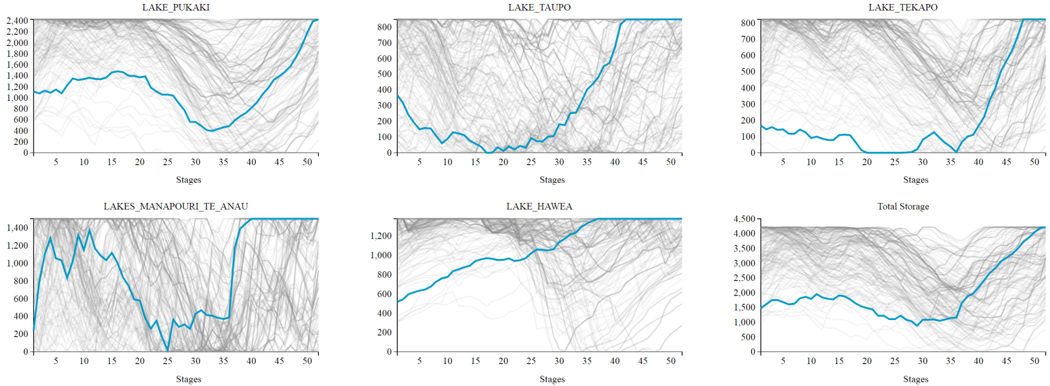
Raw SDDP.jl simulation output

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND



JADE package

Raw SDDP.jl simulation output



JADE package

API Documentation

JADE: a Julia DOASA Environment

Search docs

JADE

Edit on GitHub

JADE: a Julia DOASA Environment

JADE is a sophisticated hydro-thermal (and demand-response) scheduling tool calibrated to the New Zealand electricity system. JADE leverages the SDDP.jl package in order to find optimal waters values and release policies.

Requirements

JADE requires Julia 1.6+, JuMP 1.0+, SDDP.jl and appropriate optimizer(s).

Installation

JADE is installed by the `pkg` interface provided by Julia. In the Julia REPL, simply enter the following command.

```
] add "https://github.com/EPOC-NZ/JADE.git"
```

Julia will install all dependencies automatically, ensuring compatibility. If you encounter an issue during installation, it may be due to another package with dependencies that conflict with JADE's.

Running JADE

See the documentation for specifics about the JADE inputs and how to run JADE.

The Electricity Authority hosts a repository of JADE input files. These can be accessed from the EMI website. Any input files in the JADE Github repository are for testing purposes only.

Bugs

Please raise an issue if you experience an error while using JADE.

API Reference »

Version docs

JADE package

API Documentation

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

JADE: a Julia DOASA Environment

Search docs

JADE

API Reference

Version docs

User interface functions

These functions are to be utilised by JADE users to set up and run JADE models.

Functions to set up JADE model / data

Using run files

`JADE.define_JADE_model` — Method

```
define_JADE_model(inputdir::String; run_file::String = "run")
```

This function reads a JADE run file in the Input/<inputdir> folder. In particular, this function reads the parameters that are needed in order to define the SDDP model.

Required Arguments

`inputdir` is the name of the subdirectory (within `Input`) that contains the JADE data files.

Keyword Arguments

`run_file` is the name of the csv file that contains the parameters we wish to load.

`JADE.define_JADE_solve_options` — Method

```
define_JADE_solve_options(inputdir::String; run_file = "run")
```

This function loads the training options for JADE.

Required Arguments

`inputdir` is the name of the subdirectory (within `Input`) where the `run_file` is located.

Keyword Arguments

Outline

Overview of JADE

- Model / assumptions
- Inputs and training
- Setting up a JADE model

Lake Onslow

- 100% renewables and security of supply
- Specifications
- Results and Analysis

Other JADE applications

- Hydrogen integration
- Fuel availability
- Stagewise-dependent inflows

Lake Onslow Pumped Hydro Proposal

100% renewables and security of supply?

Over the past 5 years, there have been several targets set for the decarbonisation of the electricity market.

Lake Onslow Pumped Hydro Proposal

100% renewables and security of supply?

Over the past 5 years, there have been several targets set for the decarbonisation of the electricity market.

An initial proposal of 100% renewables (in a normal hydrological year) by 2035 was put forward by the government. The Interim Climate Change Committee (ICCC) was tasked at looking at the government's policy and concluded that reaching a 100% renewable electricity sector would be more expensive than equivalent CO_2 reductions from other sectors.

Lake Onslow Pumped Hydro Proposal

100% renewables and security of supply?

Over the past 5 years, there have been several targets set for the decarbonisation of the electricity market.

An initial proposal of 100% renewables (in a normal hydrological year) by 2035 was put forward by the government. The Interim Climate Change Committee (ICCC) was tasked at looking at the government's policy and concluded that reaching a 100% renewable electricity sector would be more expensive than equivalent CO_2 reductions from other sectors.

Nevertheless, in the 2020 election campaign, the Labour party announced a more ambitious policy: to reach 100% renewable electricity by 2030. Since being re-elected they have backed away from 100% renewables, however, a cornerstone of this policy is the NZ Battery project, which may be critical ensure security of supply in dry winters.

Lake Onslow Pumped Hydro Proposal

100% renewables and security of supply?

Over the past 5 years, there have been several targets set for the decarbonisation of the electricity market.

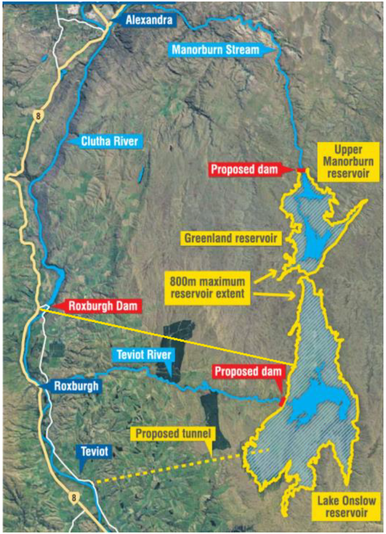
An initial proposal of 100% renewables (in a normal hydrological year) by 2035 was put forward by the government. The Interim Climate Change Committee (ICCC) was tasked at looking at the government's policy and concluded that reaching a 100% renewable electricity sector would be more expensive than equivalent CO_2 reductions from other sectors.

Nevertheless, in the 2020 election campaign, the Labour party announced a more ambitious policy: to reach 100% renewable electricity by 2030. Since being re-elected they have backed away from 100% renewables, however, a cornerstone of this policy is the NZ Battery project, which may be critical ensure security of supply in dry winters.

A pumped hydro station at Lake Onslow is a front-runner for this, with the government intending to spend $30 million to develop the business case, and an estimated $4 billion to construct it.

Lake Onslow Pumped Hydro Proposal

Possible Specifications



There are several options being considered for Lake Onslow:

- capacity of the reservoir;
- capacity of the pump / generator; and
- location of the tunnel.

In addition, how it would operate in the market is also undecided.

We have made some coarse assumptions about the Lake Onslow pumped hydro station:

- 1300 MW capacity for both pumping and generating;
- 77% overall efficiency;
- 8000 GWh capacity;
- negligible natural inflows;
- Onslow will operate as a competitive station.

Results and Analysis

Scenarios

We have developed a number of scenarios with varying amounts of wind generation, and investigated how a 100% renewable system would operate with and without Onslow. We compute the steady-state policy, and simulate this using the inflows from 200 randomly-ordered historical years.

Results and Analysis

Scenarios

We have developed a number of scenarios with varying amounts of wind generation, and investigated how a 100% renewable system would operate with and without Onslow. We compute the steady-state policy, and simulate this using the inflows from 200 randomly-ordered historical years.

The wind is assumed to have a 40% capacity factor, so a 100MW wind farm provides the same energy as a 40MW geothermal plant over a week. We examine scenarios with between 3200MW and 4800MW of new wind capacity. For simplicity, the demand scenario we are using throughout is 2020 demand.

Results and Analysis

Scenarios

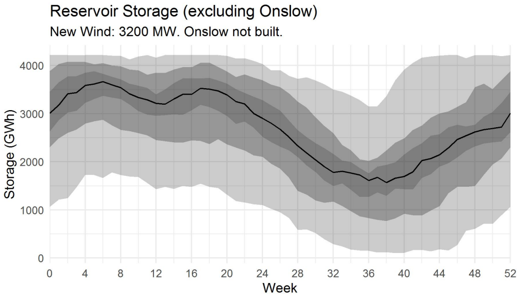
We have developed a number of scenarios with varying amounts of wind generation, and investigated how a 100% renewable system would operate with and without Onslow. We compute the steady-state policy, and simulate this using the inflows from 200 randomly-ordered historical years.

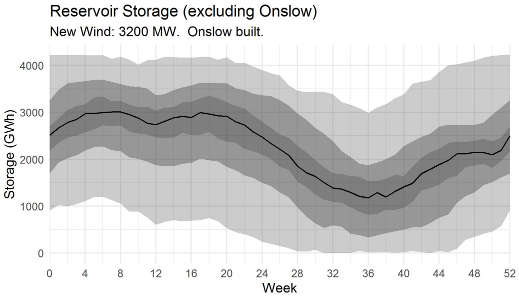
The wind is assumed to have a 40% capacity factor, so a 100MW wind farm provides the same energy as a 40MW geothermal plant over a week. We examine scenarios with between 3200MW and 4800MW of new wind capacity. For simplicity, the demand scenario we are using throughout is 2020 demand.

Given the large number of rough assumptions made, the results are intended as an demonstration of the functionality of JADE, and shouldn't be taken as an accurate prediction on the impact of Onslow on the market.

Results and Analysis

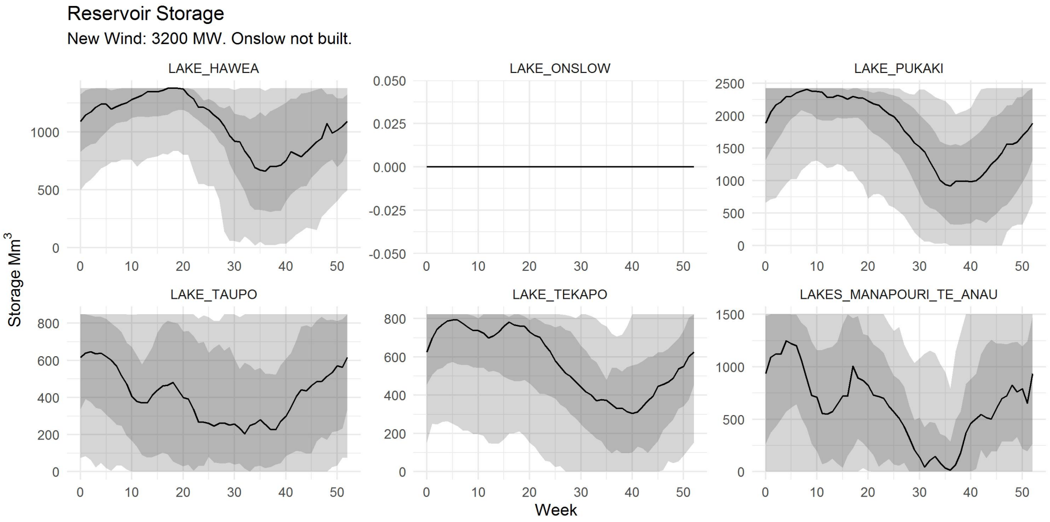
Total storage excluding Onslow: 3200MW wind





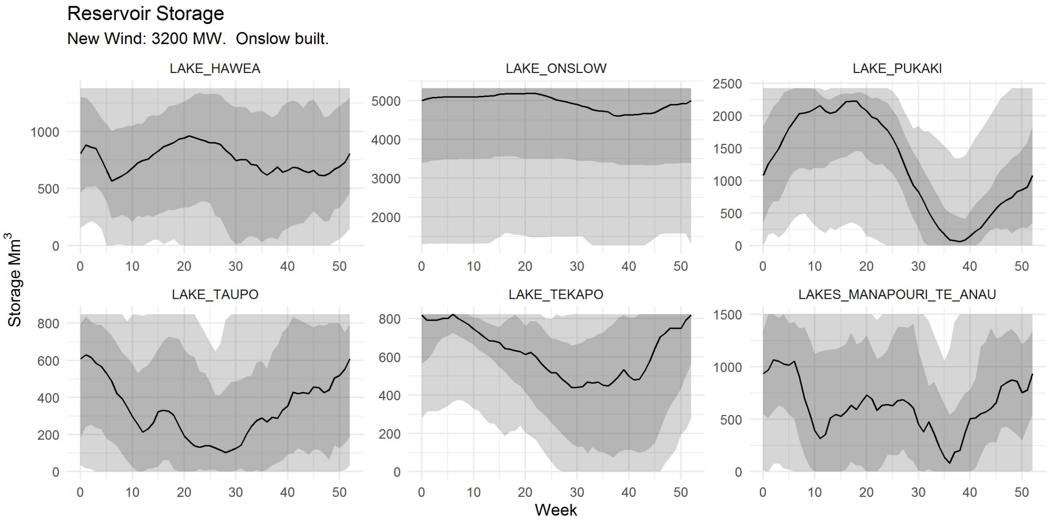
Results and Analysis

Storage by reservoir: 3200MW wind



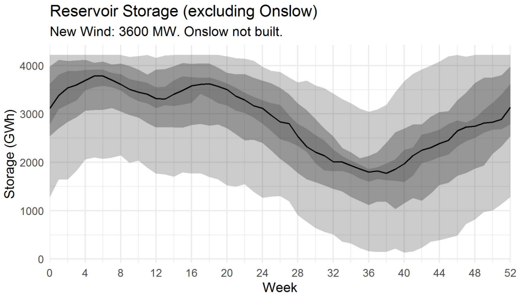
Results and Analysis

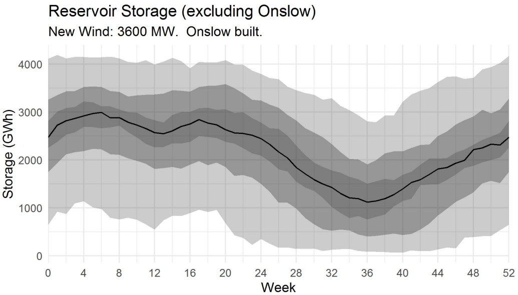
Storage by reservoir: 3200MW wind



Results and Analysis

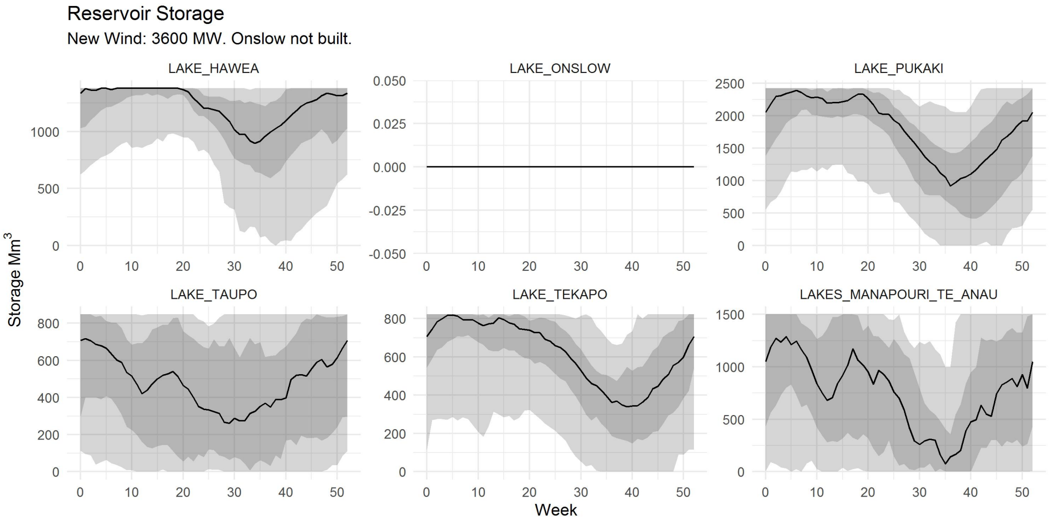
Total storage excluding Onslow: 3600MW wind





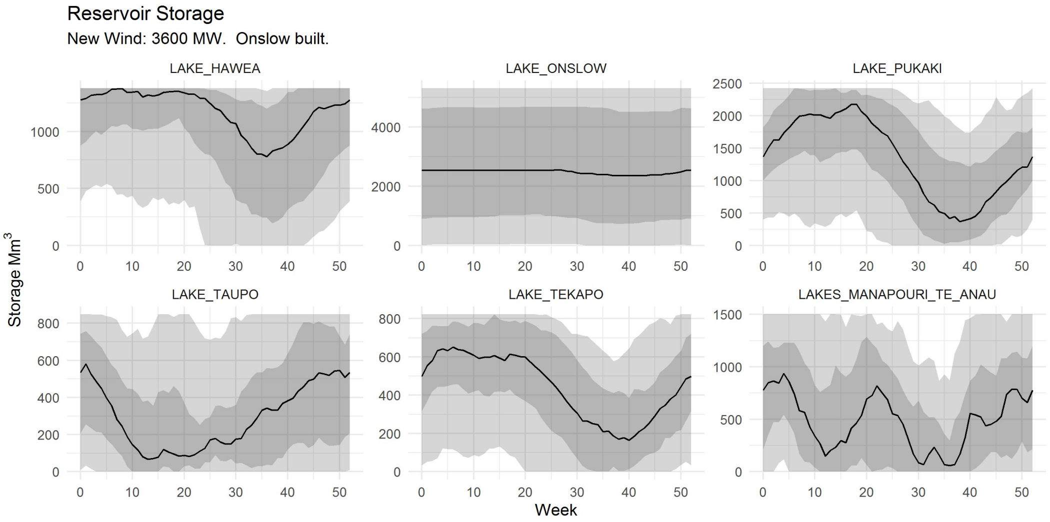
Results and Analysis

Storage by reservoir: 3600MW wind



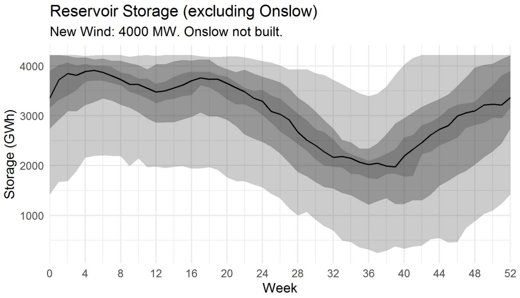
Results and Analysis

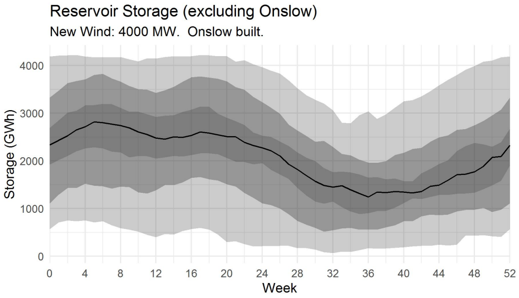
Storage by reservoir: 3600MW wind



Results and Analysis

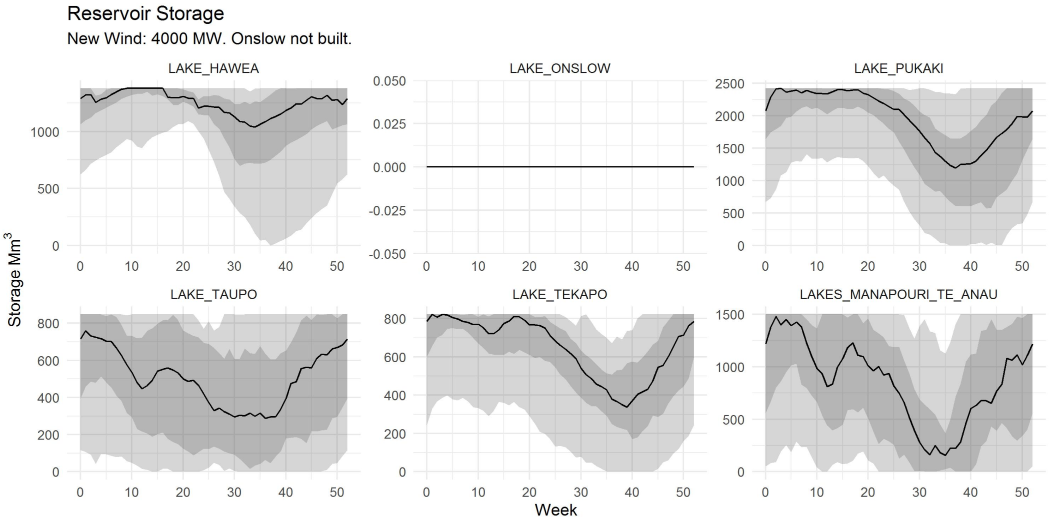
Total storage excluding Onslow: 4000MW wind





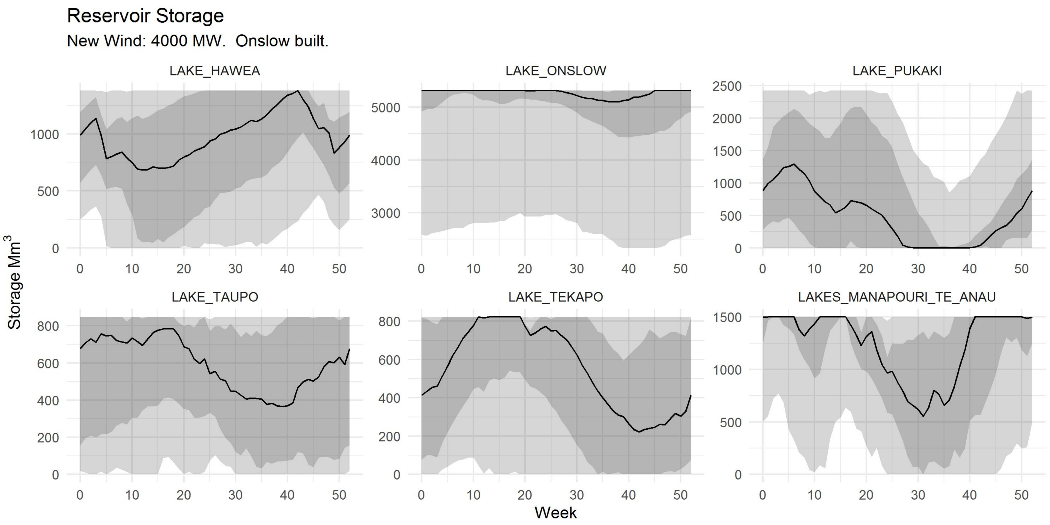
Results and Analysis

Storage by reservoir: 4000MW wind



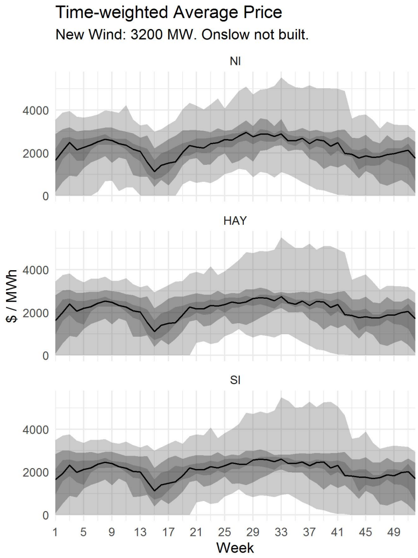
Results and Analysis

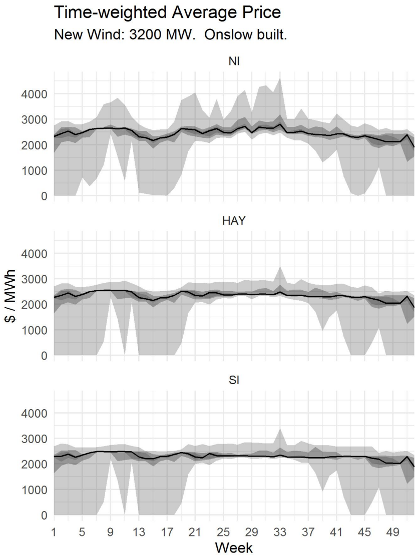
Storage by reservoir: 4000MW wind



Results and Analysis

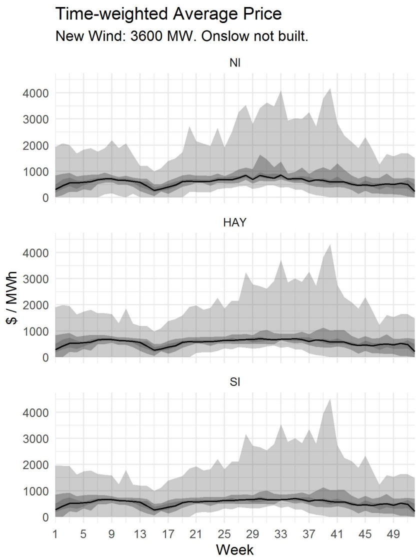
TWAP Prices: 3200MW wind

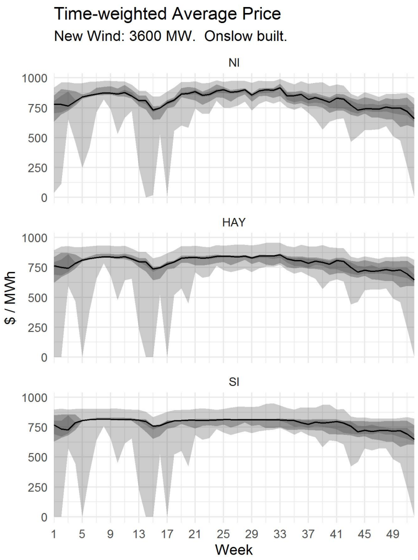




Results and Analysis

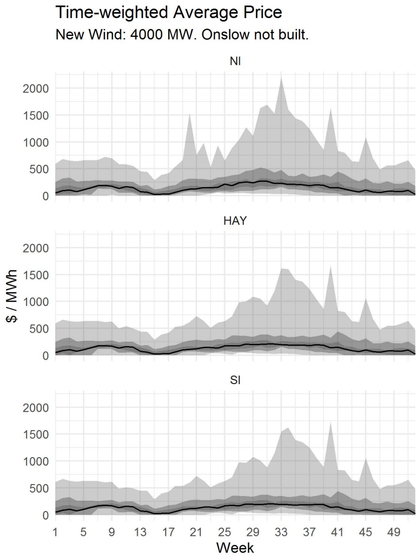
TWAP Prices: 3600MW wind

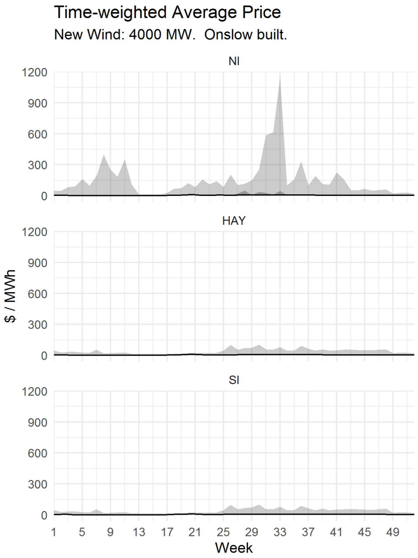




Results and Analysis

TWAP Prices: 4000MW wind





Improvements to the Onslow model

There are a number of improvements needed to this model to make it more realistic, including:

- explicitly dealing with wind variability;
- model and compare different tunnel options;
- demand growth estimates;
- comparing different pumping, generation and storage capacities;
- more realistic transmission network;
- comparing different operating policies; and
- modelling emissions of geothermal generators.

Outline

Other JADE Applications

Hydrogen Integration

EPOC is working on the HINT (Hydrogen Integration) project with Canterbury University, which is funded by MBIE.

We have an ongoing project with two Part 4 Engineering Science students (Connor Roulston and Harry Thurman) who are currently exploring how a flexible hydrogen (production) plant (that can respond to price) may help New Zealand reach 100% renewables.

To examine this, the thermal plants have been removed from JADE, and wind is being added. An (850MW) hydrogen plant is added in Otago that sheds load cheaply.

The key to this is that the hydrogen plant is shedding GWh of energy, during dry years, not just shifting load to offpeak periods.

Other JADE Applications

Investigating the effect of fuel shortages

We are testing a version of JADE that incorporates fuel storage for thermal plants.

This can be used to model things such as a coal stockpile, variable biofuels availability, or the production and usage of hydrogen.

Within JADE, these fuels are treated in a similar way to water in a dam, and for each fuel, a marginal value will be determined, which will be a function of the amount of fuel available each week.

Other JADE Applications

Stagewise-dependent inflows

Although JADE 1.0 uses stagewise-independent inflows for the training of the SDDP model, it uses the DIA (Dependent Inflow Adjustment) technique to reflect the higher medium-term variation, exhibited by stagewise-dependent inflow processes.

We have an ongoing project with two Part 4 Engineering Science students (Adam Clifford and Sebastian Ayson Macfarlane) who are currently evaluating the affect of modelling inflows using auto-regressive processes.

We are interested in the difference in release policies and prices, and also the increased computational burden associated with training the model.

Thanks for your attention.

Any questions?

Contact me: `a.downward@auckland.ac.nz`