Julia

JuMP

Julia for Mathematical Programming

Julia

- Development started in 2009
- 2012
- 2017 – 250k users
- Version 1.0 target date

Why Julia?

“We want a language that’s open source, with a liberal license. We want the speed of C with the dynamism of Ruby. We want a language that’s homoiconic, with true macros like Lisp, but with obvious, familiar mathematical notation like Matlab. We want something as usable for general programming as Python, as easy for statistics as R, as natural for string processing as Perl, as powerful for linear algebra as Matlab, as good at gluing programs together as the shell. Something that is dirt simple to learn, yet keeps the most serious hackers happy. We want it interactive and we want it compiled.

(Did we mention it should be as fast as C?)”

Julia Packages - https://pkg.julialang.org/

- Symbolic – Reduce, Symata, Maxima
- Numerical methods
- Differential Equations (ODE’s, PDE’s, Stochastic, …)
- Statistics
- Graphs
- Neural nets
- Specialist Applications – Finance, bio-informatics, geospatial, …
- …

JuMP - Julia for Mathematical Programming

https://jump.readthedocs.io/

- Built-in package
- Solver independent

JuliaPro and JuliaBox

- Atom editor
- Jupyter

Simple Problem

```
species=("white rhino", "buffalo", "zebra")
au=[2.4, 1, 0.5]
growth = [0.08, 0.22, 0.15]
price = [60, 12, 2]
K=100    #Carrying Capacity
```

What numbers of each species to maximise profit?

Require

- twice as many buffalo as rhino
- four times as many zebra as rhino

The game ranch problem

Part A

Solve the ranch problem

File – RanchA.jl

Part B

Name the following constraint ‘resource’ as follows:

```
@constraint(ranch, resource,  sum(au[i]∗x[i]  for  i=1:3)  <=  100.0  )
```

What is the value of another unit of land?

Try: print(getdual(resource))

(See RanchB.jl)

Problem 2 – Animal Food

Exercise A

- AnimalFoodA.jl – Make the objective function more concise
- Check your results with AnimalFoodB.jl

Exercise B

- Modify AnimalFoodB.jl to find the shadow prices

Exercise C

- Note the alternative form of writing constraints in AnimalFoodC.jl

Examples

- Applications of optimization with Xpress-MP
- http://www.maths.ed.ac.uk/hall/Xpress/FICO_Docs/Xpress-booka4.pdf

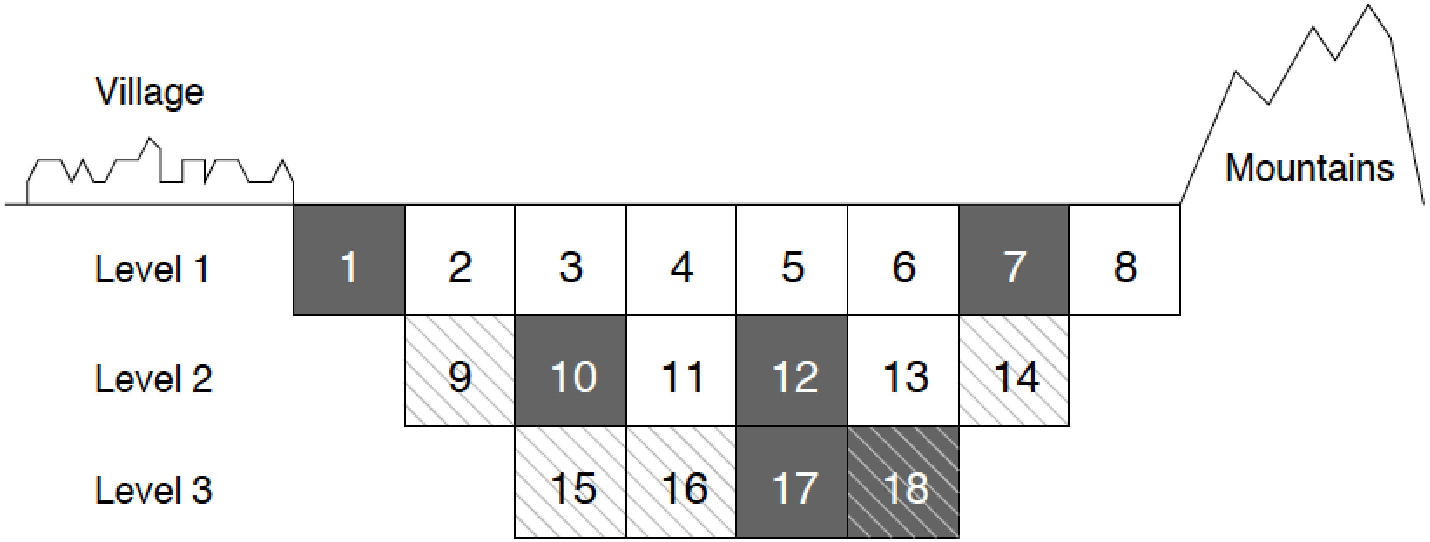
Problem 3 - Depot location

- Problem: Choose from 12 sites where to build
- Run the model in file **depotLocation.jl**
- Note the binary variables and hence the change to CbcSolver

Problem 4 - Mining

Uranium in dark
(1, 7, 10, 12, 17 and 18)

Striped blocks very hard
(9, 14, 15, 16, and 18)



Which blocks should be mined?

Mining

Exercise A

- Load and run file **miningArcsA.jl**
- Note precedence constraints
- Note the use of tuples: (i,j) in arcs
- Note that the use of continuous variables

Exercise B

- Load and run mineAlt.jl
- Note the declaration of an Array of tuples
- Note “Vararg” to allow elements of the Array to have different numbers of tuples
- Does the use of binary variables with Cbc give the same result as Clp?

Writing and reading to files

- Open the file TaxOfficesCBC.jl
- Similarities to the depot location problem
- Note how to print the solution to a file

Exercise A

- Print the solution to a file with your preferred directory

Exercise B

- Save the data file **AMSIOptRSA.csv** to a suitable directory
- Open the file **readXLS.jl** and read in the data from the CSV file
- Get to know the format of the data by printing some extracts as shown as a sample.

Travelling Salesman Problem

- Planning a flight tour
- 7 cities to visit
- File **flightTour3.jl**
- Note the use of a function
- Exercise
- Solve as given and identify sub-cycles
- Add constraints to remove sub-cycle and solve again. Repeat and until problem solved.

Flight Tour

- **TSP Challenge (requires Gurobi)**
- Read up on callbacks and lazy constraints in the JuMP docs
- Extend or modify flightTour3.jl to automatically eliminate sub-cycles

Simulated Annealing

- See the file SA_flightTour.jl
- Solves the same problem

Resources

- https://julialang.org/
- Julia Programming for Operations Research: A Primer on Computing by Changhyun Kwon (246 pages; published: 2016-05; ISBN: 978-1533328793) This book aims to teach how one can solve various optimization problems arising in operations research and management science.
- JuMP **https://jump.readthedocs.io/**