JuMP: the year in review

Oscar Dowson

ORSNZ 2023

What is JuMP?

Part of the zoo of algebraic modeling languages

AIMMS
AMPL
CVX RESEARCH
CVXPY
GAMS
LINGO the modeling language and optimizer
MAXIMAL SOFTWARE
Mini Zn
PYOMO

CMPL, CPLEX Concert, GNU MathProg, Gurobi C++/Python API, linopy, MATLAB, Mosek Fusion, MOSEL, ompr, OPTMODEL, PuLP, Python-MIP, YALMIP, ZIMPL, …

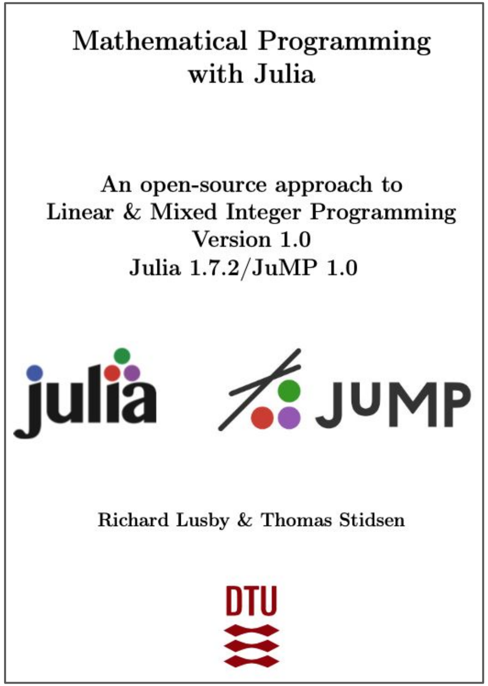
What is JuMP?

An open-source algebraic modeling language in Julia

```julia
using JuMP, Ipopt
function constrained_linear_regression(A::Matrix, b::Vector)
    model = Model(Ipopt.Optimizer)
    @variable(model, 0 <= x[1:size(A, 2)] <= 1)
    @variable(model, residuals[1:size(A, 1)])
    @constraint(model, residuals == A * x - b)
    @constraint(model, sum(x) <= 1)
    @objective(model, Min, sum(r^2 for r in residuals))
    optimize!(model)
    return value.(x)
end
A, y = rand(30, 20), rand(30)
x = constrained_linear_regression(A, b)
```

Who uses JuMP?

Academics



Roll over image to zoom in

Julia Programming for Operations Research

by Changhyun Kwon (Author)

4.5 50 ratings

See all formats and editions

Kindle	Paperback
$28.57	$28.86 - $28.90

Read with our free app

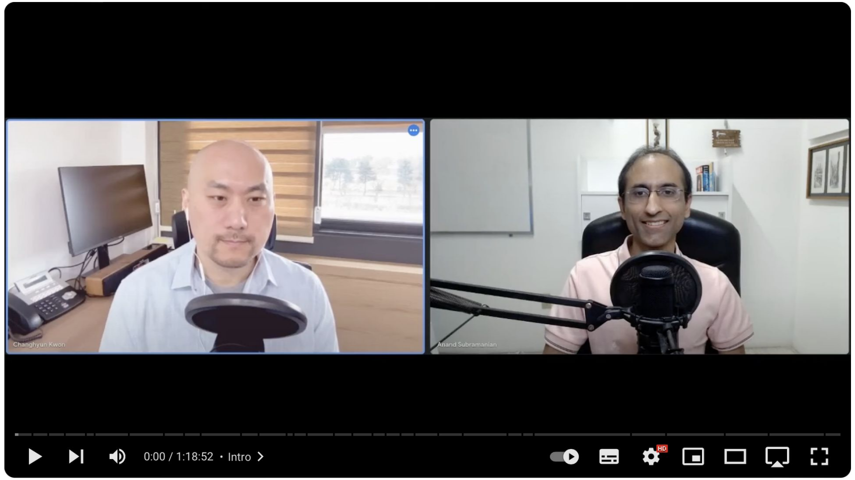
5 Used from $24.87

5 New from $28.90

Last Updated: December 2020

Based on Julia v1.3+ and JuMP v0.21+

The main motivation of writing this book was to help the author himself. He is a professor in the field of operations research, and his daily activities involve building models of mathematical optimization, developing algorithms for solving the problems, implementing those algorithms using computer programming languages, experimenting with data, etc. Three languages are involved: human language, mathematical language, and computer language. His team of students need to go over three different



Subject to: Changhyun Kwon

Subject to 2.28K subscribers Subscribed 23 Share Clip Save

Who uses JuMP?

Companies

Home > Transportation Science > Ahead of Print >

Solving a Continent-Scale Inventory Routing Problem at Renault

Louis Bouvier, Guillaume Dalle, Axel Parmentier, Thibaut Vidal

Published Online: 31 Oct 2023 | https://doi.org/10.1287/trsc.2022.0342

Abstract

This paper is the fruit of a partnership with Renault. Their reverse logistic requires solving a continent-scale multiattribute inventory routing problem (IRP). With an average of 30 commodities, 16 depots, and 600 customers spread across a continent, our instances are orders of magnitude larger than those in the literature. Existing algorithms do not scale, so we propose a large neighborhood search (LNS). To make it work, (1) we generalize existing split delivery vehicle routing problems and IRP neighborhoods to this context, (2) we turn a state-of-the-art matheuristic for medium-scale IRP into a large neighborhood, and (3) we introduce two novel perturbations: the reinsertion of a customer and that of a commodity into the IRP solution. We also derive a new lower bound based on a flow relaxation. In order to stimulate the research on large-scale IRP, we introduce a library of industrial instances. We benchmark our algorithms on these instances and make our code open source. Extensive numerical experiments highlight the relevance of each component of our LNS.

A two-step linear programming model for energy-efficient timetables in metro railway networks

Shuvomoy Das Gupta [a], J. Kevin Tobin [b], Lacra Pavel [a]

Show more

+ Add to Mendeley Share Cite

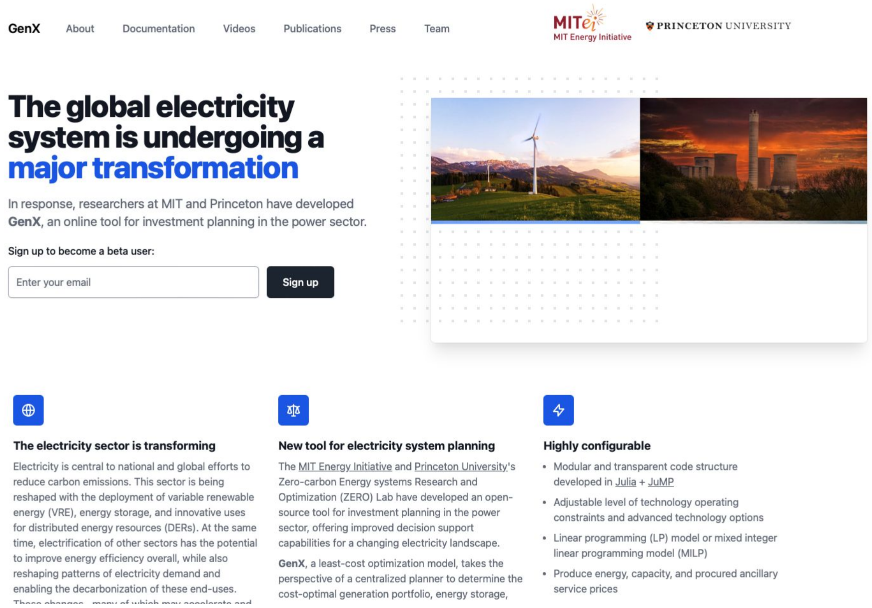
https://doi.org/10.1016/j.trb.2016.07.003 Get rights and content

Abstract

In this paper we propose a novel two-step linear optimization model to calculate energy-efficient timetables in metro railway networks. The resultant timetable minimizes the total energy consumed by all trains and maximizes the utilization of regenerative energy produced by braking trains, subject to the constraints in the railway network. In contrast to other existing models, which are $\mathcal{NP}$-hard, our model is computationally the most tractable one being a linear program. We apply our optimization model to different instances of service PES2-SFM2 of line 8 of Shanghai Metro network spanning a full service period of one day (18 h) with thousands of active trains. For every instance, our model finds an optimal timetable very quickly (largest runtime being less than 13 s) with significant reduction in effective energy consumption (the worst case being 19.27%). Code based on the model has been integrated with Thales Timetable Compiler - the industrial timetable compiler of Thales Inc that has the largest installed base of communication-based train control systems worldwide.

Who uses JuMP?

Energy system modellers



Tulipa

Electricity Market and Sector-Coupling Modelling Tools from TNO

6 followers · Netherlands

mopo

Modelling tools for the energy transition

Who is (some of) JuMP?

https://github.com/jump-dev/JuMP.jl/graphs/contributors

In the last 12 months of github.com/jump-dev…

>600,000
downloads of jump-dev packages

1,104
pull requests opened

49
unique contributors

+131,132
lines added

We worked on

Rational and BigFloat

Nonlinear programming

Multi-objective

Constraint programming

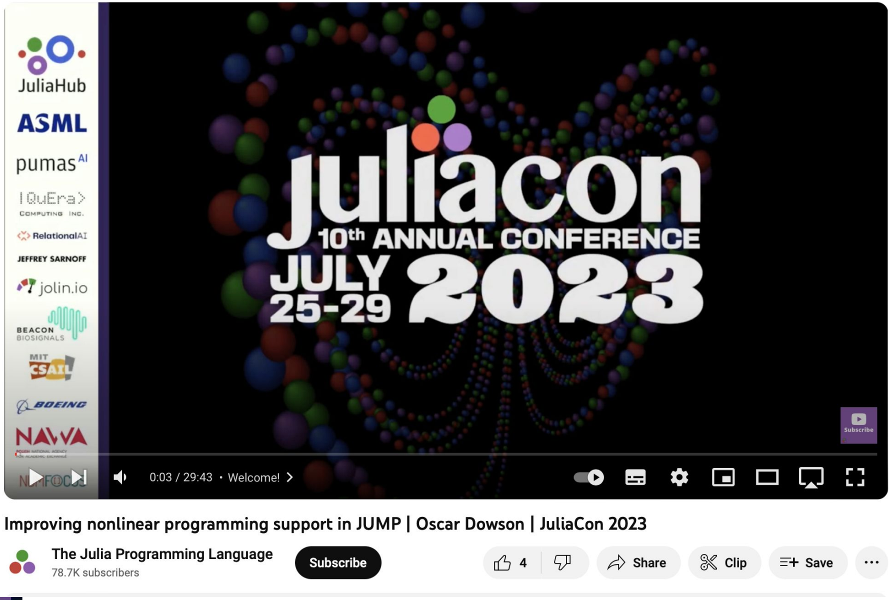
Time-To-First-Solve

Nonlinear complementarity

Complex numbers

Improving nonlinear programming support in JuMP

https://jump.dev/JuMP.jl/stable/manual/nonlinear/



Improving nonlinear programming support in JuMP

https://jump.dev/JuMP.jl/stable/manual/nonlinear/

```
using JuMP
model = Model()
@variable(model, x[1:2])
@objective(model, Min, x[2]^3 * sin(x[1])^x[2])
my_func(y) = sum(2^y[1] .+ exp.(y))
@expression(model, expr, 2 * my_func(x))
@constraint(model, expr <= 100)
@constraint(model, sqrt(x' * x) <= 1)
```

Carleton Coffrin
ccoffrin

Nonlinear complementarity

https://jump.dev/JuMP.jl/stable/tutorials/nonlinear/complementarity/

```
using JuMP, PATHSolver
model = Model(PATHSolver.Optimizer)
@variable(model, 0 <= x[1:2] <= 1, start = 0.5)
# To add a constraint of the form `F(x) ⊥ x` do
@constraint(model, x[2]^3 - x[1] ⊥ x[1])
@constraint(model, 1 - exp(x[1]) ⊥ x[2])
optimize!(model)
value.(x)
```

Michael C Ferris
MichaelCFerris

sdirkse67

Complex number support

https://jump.dev/JuMP.jl/stable/manual/complex/

```
using JuMP
model = Model()
@variable(model, x in ComplexPlane())
#     real(x) + imag(x) im

@variable(model, y[1:2, 1:2] in HermitianPSDCone())
#     2×2 LinearAlgebra.Hermitian{...}:
#      real(y[1,1])                    real(y[1,2])+imag(y[1,2])*im
#      real(y[1,2])-imag(y[1,2])*im    real(y[2,2])
```

Benoît Legat
blegat

Constraint programming

https://jump.dev/JuMP.jl/stable/tutorials/linear/constraint_programming/

```
using JuMP, MiniZinc
model = Model(() -> MiniZinc.Optimizer{Float64}("highs"))
@variable(model, 1 <= x[1:3] <= 3, Int)
@variable(model, 0 <= z <= 1, Bin)
@constraint(model, x in MOI.AllDifferent(3))
@constraint(model, z <--> {x[1] == 1.0})
@objective(model, Max, sum(i * x[i] for i in 1:3))
optimize!(model)
value.(x)
```

Chris Coey
chriscoey

RelationalAI

Multi-objective support

https://jump.dev/JuMP.jl/stable/tutorials/linear/multi_objective_examples/

```julia
using JuMP, HiGHS
import MultiObjectiveAlgorithms as MOA
model = Model(() -> MOA.Optimizer(HiGHS.Optimizer))
set_attribute(model, MOA.Algorithm(), MOA.Dichotomy())
@variable(model, 0 <= x[1:2] <= 3)
@objective(model, Min, [3x[1] + x[2], -x[1] - 2x[2]])
@constraint(model, 3x[1] - x[2] <= 6)
optimize!(model)
pareto_frontier = [
    value.(x; result = i) for i in 1:result_count(model)
]
```

Gökhan Kof
kofgokhan

XavierG
xgandibleux

Generic number support

https://jump.dev/JuMP.jl/stable/tutorials/conic/arbitrary_precision/

```julia
using JuMP, CDDLib
model = GenericModel{Rational{BigInt}}(
    CDDLib.Optimizer{Rational{BigInt}},
)
@variable(model, 1 // 7 <= x[1:2] <= 2 // 3)
@constraint(model, c1, (2 // 1) * x[1] + x[2] <= 1)
@constraint(model, c2, x[1] + 3x[2] <= 9 // 4)
@objective(model, Max, sum(x))
optimize!(model)
value.(x)   # Returns [1 // 6, 2 // 3]
```

Benoît Legat
blegat

Generic number support

https://jump.dev/JuMP.jl/stable/tutorials/conic/arbitrary_precision/

```julia
using JuMP, Clarabel
model = GenericModel{BigFloat}(
    Clarabel.Optimizer{BigFloat},
)
@variable(model, x[1:2, 1:2] in PSDCone())
@variable(model, t)
y = rand(2, 2)
@constraint(model, [t; vec(x .- y)] in SecondOrderCone()
@objective(model, Min, t)
optimize!(model)
value.(x)   # Returns Vector{BigFloat}
```

Clarabel.jl

Paul Goulart
goulart-paul

Time-To-First-Solve

```
% julia +1.6 bench.jl
VERSION = v"1.6.7"
 5.689157 seconds
 9.472270 seconds
```

```
% julia +1.9 bench.jl
VERSION = v"1.9.3"
 4.332634 seconds
 0.190987 seconds
```

```
@show VERSION
@time using JuMP, HiGHS
@time begin
  model = Model(HiGHS.Optimizer)
  set_silent(model)
  @variable(model, x >= 0)
  @variable(model, 0 <= y <= 3)
  @objective(model, Min, 12x + 20y)
  @constraint(model, c1, 6x + 8y >= 100)
  @constraint(model, c2, 7x + 12y >=
120)
  optimize!(model)
end
```

Tim Holy
timholy

Plans for next year

- Better support for Xpress and KNITRO, MINLP from Gurobi
- Modeling with scientific units

```
@variable(model, x, u"m/s")
```

- Better engagement with users

  jump-teaching@googlegroups.com
- Finding a sustainable funding source
- JuMP-dev 2024 in Montréal (before ISMP)

  https://jump.dev/meetings/jumpdev2024/