On modelling optimization problems via
Julia JuMP

Prof. Mayron César de O. Moreira

Department of Computer Science

Universidade Federal de Lavras (UFLA)

May 15 - May 17, 2019

Reggio Emilia, UNIMORE, Italy

Outlines

- Lecture 1: an overview of Julia Language
  - Simple examples, Handling input and output data;
  - JuMP: commands, examples, modeling classical problems: bin packing, TSP, cutting and packing, exercises;
- Lecture 2: Assembly Line Balancing Problems (ALBPs)
  - SALBP-1 and SALBP-2;
  - ALWABP;
  - Exercises.
- Lecture 3: Vehicle Routing Problem (VRP) and Production Planning
  - Modeling a variant of VRP;
  - Modeling a production planning problem;
  - Exercise.

Outlines

- Lecture 4: Problems on Graphs
  - Maximum Dispersion Problem (MaxDP);
  - Promoting Eulerian Districts for Arc Routing;
  - Exercise.
- Lecture 5: Final test

Universidade Federal de Lavras (UFLA)

- TOP 10 in the Brazilian Ranking of Courses (2018);
- Approximately 800 professors and 16,000 students
- 35 undergraduate courses;
- 43 graduate programs (21 MSc., and 23 MSc. and PhD)
- Prize "**Blue University**": Reference on *Water Sustainability*;
- New Campus: São Sebastião do Paraíso (Minas Gerais);
- The incentive to entrepreneurship: courses, business acceleration program, technological park (*LavrasTec*).

Universidade Federal de Lavras (UFLA)
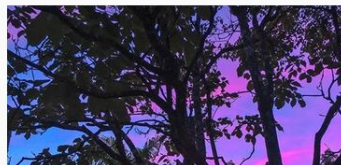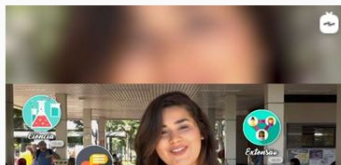
Universidade Federal de Lavras (UFLA)

Follow us:
www.instagram.com/uflabr

Department of Computer Science

- 40 professors (Ph.D.) and 8 administrative technicians;
- 2 undergraduate courses: Information Systems and Computer Science;
- MSc. Computer Science (13 advisors);
  - Information Recovery and Software Engineering;
  - Computational Intelligence, Computational Geometry and Operations Research;
  - Computer Networks and Embedded Systems.
- Teaching and Research Infrastructure.

Department of Computer Science

LLOOP: Lavras Laboratory of Optimization

- Professors: Andreza Beezão (DEX), Dilson L. Pereira (DCC), Júlio César Alves (DCC), and Mayron César de O. Moreira (DCC);
- 10 undergraduate students e 5 MSc. students;
- 4 projects funded by CNPq and FAPEMIG ;
- *Expertise*: Problems modeled on Graphs (Routing, Districting, Spanning Trees), Production Planning, and Cutting Problems;

Lavras
Lab of Optimization

LLOOP: since 2016

A short bio: Moreira, M. C. O.

- Professor at UFLA since 2015;
- Ph.D. at Universidade de São Paulo (USP);
  - Research stages: UPV (Valencia, Spain), UPC (Barcelona, Spain) and CIRRELT (Montreal, Canada);
- *Theory and applications of Operations Research*;
- *Funded by:* CNPq, FAPEMIG and UFLA.

Collaborators:

Why Julia?

"We want a language that's **open source**, (…) We want the **speed of C** with the **dynamism of Ruby**. (…) familiar **mathematical notation like Matlab**. (…) **general programming as Python**, as **easy for statistics as R**, as **natural for string processing as Perl,** (…)**.** We want it **interactive** and we want it **compiled**."

Jeff Bezanson
JeffBezanson

Alan Edelman
Professor of Applied Mathematics
Computer Science and AI Laboratories
Applied Computing Group Leader

» Stefan Karpinski «

Viral B. Shah
ViralBShah
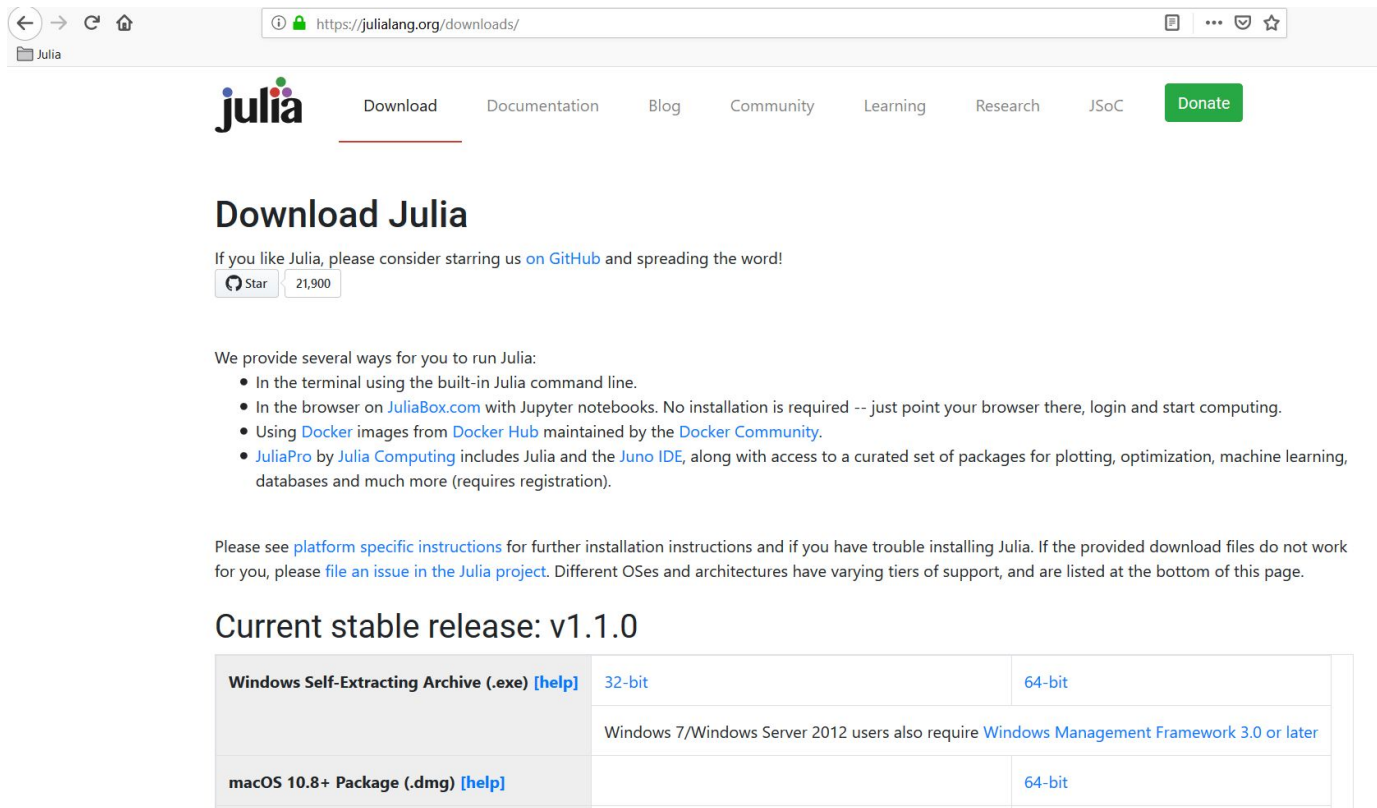
Why Julia?

- High Performance language;
- Efficient native code for multiple platforms via *Low Level Virtual Machines* (LLVM);
- Rich language of descriptive datatypes;
- Dynamically-typed (as a scripting language);
- Free and open source;
- Just-in-time compilation;
- Julia has: garbage collector, eager evaluation, and efficient libraries for floating-point calculations, linear algebra, random number generation, and regular expression matching.

Why Julia?

Download and install (Windows, MacOS and Linux)

*Remember to set up **path variable.***

https://julialang.org/downloads/

Julia

Download | Documentation | Blog | Community | Learning | Research | JSoC | **Donate**

Download Julia

If you like Julia, please consider starring us on GitHub and spreading the word!

⬡ Star | 21,900

We provide several ways for you to run Julia:

- In the terminal using the built-in Julia command line.
- In the browser on JuliaBox.com with Jupyter notebooks. No installation is required -- just point your browser there, login and start computing.
- Using Docker images from Docker Hub maintained by the Docker Community.
- JuliaPro by Julia Computing includes Julia and the Juno IDE, along with access to a curated set of packages for plotting, optimization, machine learning, databases and much more (requires registration).

Please see platform specific instructions for further installation instructions and if you have trouble installing Julia. If the provided download files do not work for you, please file an issue in the Julia project. Different OSes and architectures have varying tiers of support, and are listed at the bottom of this page.

Current stable release: v1.1.0

Windows Self-Extracting Archive (.exe) [help]	32-bit	64-bit
	Windows 7/Windows Server 2012 users also require Windows Management Framework 3.0 or later	
macOS 10.8+ Package (.dmg) [help]		64-bit

Why Julia?

- Currently: Version 1.1.0;
- Other Resources
  - *Learning* (tutorials, texts and books):
    - https://julialang.org/learning/
    - *Coursera*: www.coursera.org/learn/julia-programming
  - *Official channel*: https://julialang.org/community/
  - *Documentation*: https://docs.julialang.org/en/v1/
- *Learn by examples (version 0.6.2)*:
  https://juliabyexample.helpmanual.io/
- IDE: Juno.

Why Julia?

- *Some ecosystems*
  - Bioinformatics and Computational Biology in Julia
    - BioJulia - https://github.com/BioJulia
  - Optimization packages
    - JuliaOpt - https://www.juliaopt.org/
  - Image Processing
    - JuliaImages - https://juliaimages.org/latest/
  - Quantum Systems
    - QuantumBFS - https://github.com/QuantumBFS
    - QuantumOptics - https://github.com/qojulia/QuantumOptics.jl
  - Complex Systems
    - JuliaDynamics - https://github.com/JuliaDynamics

Why Julia?

- *Some ecosystems*
  - Quantitative economic modeling
    - QuantEcon - https://github.com/QuantEcon
  - Astronomy and Astrophysics
    - Julia Astro - http://juliaastro.github.io/
  - Ecology
    - EcoJulia - https://github.com/EcoJulia

Why Julia?

JuliaBox: run in your browser!

Why Julia?

Julia Conference!

Why JuMP?

powered by NumFOCUS

JuMP is a domain-specific modeling language for **mathematical optimization** embedded in **Julia**. It currently supports a number of open-source and commercial solvers (Artelys Knitro, BARON, Bonmin, Cbc, Clp, Couenne, CPLEX, ECOS, FICO Xpress, GLPK, Gurobi, Ipopt, MOSEK, NLopt, SCS) for a variety of problem classes, including **linear programming**, **(mixed) integer programming**, **second-order conic programming**, **semidefinite programming**, and **nonlinear programming**.

Why JuMP?

- Syntax close to mathematical notation!
- "*easy to specify and **solve optimization problems without expert knowledge***" (https://github.com/JuliaOpt/JuMP.jl);
- Standard language: in a single line, we can change the *solver*;
- Many online examples:
  - https://github.com/JuliaOpt/JuMP.jl/tree/master/examples
  - https://github.com/JuliaOpt/juliaopt-notebooks/blob/master/notebooks/Shuvomoy%20-%20Column%20generation.ipynb

Let's practice!

We will use: Julia 1.1.0, JuMP 0.19.0, Cbc (*solver*), and Jupyter.

Go to:

https://github.com/mayronmoreira/unimore

Other References

- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, *59*(1), 65-98.
- Bezanson, J., Karpinski, S., Shah, V. B., & Edelman, A. (2012). Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*.
- Changhyun Kwon (2019). Julia Programming for Operations Research, 2nd Edition.
- Lubin, M., & Dunning, I. (2015). Computing in operations research using Julia. *INFORMS Journal on Computing*, *27*(2), 238-248.