Lecture I - Welcome and Introduction

Applied Optimization with Julia

Dr. Tobias Vlćek
University of Hamburg - Fall 2025

About this Course

About me

- Field: Optimizing and simulating complex systems
- Languages of choice: Julia, Python and Rust
- Interest: Modelling, Simulations, Machine Learning
- Teaching: OR, Algorithms, and Programming
- Contact: tobias.vlcek@uni-hamburg.de

. . .

Tip

I really appreciate active participation and interaction!

Course Structure

Lectures

- Every Tuesday between 10.15 AM and 11.45 AM
- First four lectures repeat modelling and programming
- Later lectures discuss practical problems and implementation
- Lectures are interactive $\longrightarrow$ We discuss approaches!
- Communication takes place via OpenOlat and E-Mail

Tutorials

- Tutorials every Friday between 8.15 AM and 9.45 AM
- In these tutorials we are working on assignments
- Please bring a laptop with Windows, macOS, or Linux!
- This Friday there is no tutorial!

Assignments

- Based on applied problems of the lecture
- Up to 3 students can solve assignments together
- Submitted solutions earn bonus points for the exam
- Max. 0.5 point per tutorial

. . .

> **i Note**
>
> Bonus points only count if the mark is at least 4.0!

Course Objective

Applied Optimization

- Real-world problems can be addressed with models
- Our objective is to foster your interest in the topic
- Enable you to recognize and solve problem structures
- Includes problem understanding and implementation

Research in Operations Research

- Part of the University of Hamburg Business School
- Aiming to solve real-world problems
- Or improving our theoretical understanding
- Publication in journals and conferences

. . .

> **i Note**
>
> We are also happy to supervise Bachelor and Master theses!

National and international journals

- European Journal of OR
- Journal of the Operational Research Society
- Journal on Applied Analytics
- Management Science
- Operations Research
- OR Spectrum

Real-World Applications

Brewery Production Planning

M. Mickein, M. Koch, and K. Haase [1]

Police Service District Planning

T. Vlćek, K. Haase, M. Fliedner, and T. Cors [2]

Venue Seating under COVID-19

Usama Dkaidik and Matthes Koch; Current research with a likely paper submission in 2024

Metro Inflow Management

T. Vlćek, K. Haase, M. Koch, L. Dolz, A. Weygandt, and J. Pape [3]

Split-Order Minimization

T. Vlćek and G. Voigt [4]

Crowd Management

K. Haase et al. [5]

Lecture Preview

Part I

1. Welcome and Introduction
2. First Steps in Julia
3. Packages and Data Management
4. Modelling with JuMP

Part II

5. Beer Production Planning
6. Minimizing Split Orders in E-Commerce
7. Periodic Library Routing
8. Police Districting

Part III

9. Safety Planning for the Islamic Pilgrimage in Mecca
10. Intermission
11. Arena Seat Planning under Distancing Rules
12. Passenger Flow Control in Urban Rail
13. Recap and Discussion
14. ????

Julia Programming Language

Choice of Programming Language

. . .

Question: Have you ever heard of Julia?

Why Julia?

- Designed to be:
    - as general as Python
    - as statistics-friendly as R
    - as fast as C++!

. . .

> Tip
>
> Allows for fast data workflows, particularly in scientific computing!

Syntax

- Dynamically-typed syntax just like Python
- Similar to R, Matlab and Python - not like C++
- In comparison, accessible and easy to learn!
- No need to worry about memory management!

JuMP

- Package for algebraic modeling in Julia
- Simplifies solving complex optimization problems
- Provides a high-level, user-friendly interface
- Useful for operations research and data science

. . .

> Tip
>
> JuMP is an alternative to Pyomo, GAMS, and AMPL!

Must it be a new language?

- Yes, but no need to worry!
- Julia is quite similar to Python and R
- We will learn the syntax together in the first part
- It is helpful to switch languages from time to time

Algebraic Modeling

Do you have

experience with

algebraic modeling?

What is algebraic modeling?

- A "mathematical language" for optimization problems
- Allows for describing complex systems and constraints
- Based on linear algebra (Equations and Inequalities)

. . .

How to learn algebraic modeling?

- Practice, practice, and practice!
- Understand standard models and their approach
- Develop an understanding of constraints
- Understand the structure of a models solution space
- Use an available algorithms to determine solutions

Central Questions

- What is to be decided?
- What is relevant to the decision?
- What information is given and relevant?
- What parameters (data) are needed?
- Which variables and of which type are needed?

…

Tip

Modeling is a creative process!

Model Components

1. Objective function
2. Constraints
3. Variables

…

Note

We will go through these components step by step in each lecture!

Linear Optimization Model

Basic Model Formulation

$$\text{maximize} \quad F = \sum_{j \in \mathcal{J}} c_j \times X_j$$

subject to

$$\begin{aligned} \sum_{j \in \mathcal{J}} a_{i,j} \times X_j &\leq b_i \forall i \in \mathcal{I} \\ X_j \geq 0 & \qquad \forall j \in \mathcal{J} \end{aligned}$$

Model Components

$$\begin{aligned} \mathcal{I} &: \text{set of } \ i \in \mathcal{I}, \\ \mathcal{J} &: \text{set of } \ j \in \mathcal{J}, \\ F &: \text{Objective function variable}, \\ X_j &: \text{decision variables}, \\ c_j &: \text{objective function coefficients}, \\ a_{i,j} &: \text{parameters}, \\ b_i &: \text{parameters} \end{aligned}$$

…

Question: Have you ever seen something like this before?

What is this good for?

- Good Question! A lot of things:
  - Modeling real-world problems
  - Solving complex systems
  - Optimizing resource allocation
  - Decision-making under constraints
  - Simulation and prediction

From Abstract to Concrete

We've just seen the general structure:

- Sets ($\mathcal{I}$, $\mathcal{J}$)
- Parameters (c_j, $a_{i,j}$, b_i)
- Decision variables (X_j)
- Objective function and constraints

. . .

Now let's see how this works with a real problem!

. . .

Tip

Watch for these components as we build our first model together.

Solar Panel Transport

Case: Solar Panel Transport

Description

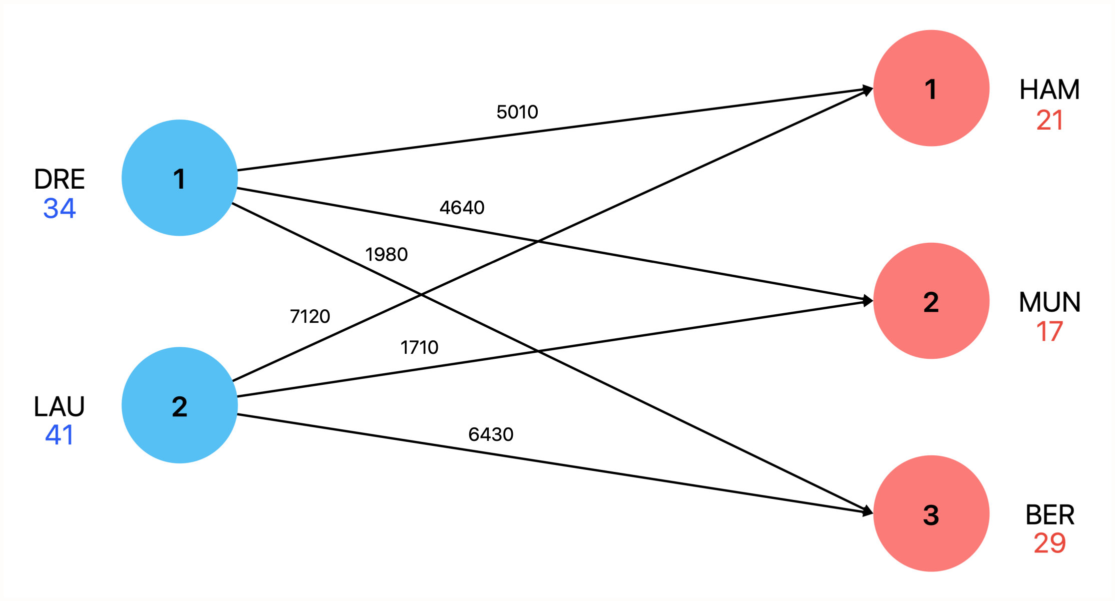
A company is producing solar panels in Dresden and Laupheim and has to transport them to new solar farms near Hamburg, Munich, and Berlin. The quantities offered and demanded (truckloads) and the transport costs per truckload in Euro are summarized in the following table.

Transport Costs

Origin/Destination	Hamburg	Munich	Berlin	Available
Dresden	5010	4640	1980	34
Laupheim	7120	1710	6430	41
Demand	21	17	29	

Example: A truckload from Dresden $i = 1$ to Munich $j = 2$ costs $c_{12} = 4640$ Euro. Moreover, it is necessary to fulfil all customer demands, as the contract has been signed.

Graphical Illustration



Understanding the Problem

What are we trying?

First, we always need to understand the objectives.

. . .

Question: What are our possible objectives?

Minimizing the transport costs over all truckloads while meeting the demand based on the available solar panels adhering to the available panels.

Let's break it down

step by step!

Sets

Remember, sets are collections of elements

. . .

Question: What sets are needed?

. . .

$$\begin{aligned} \mathcal{I} &: \text{Set of production sites, indexed by } \; i \; \text{ with } \; i \in \{1, ..., |\mathcal{I}|\}, \\ \mathcal{J} &: \text{Set of customers, indexed by } \; j \; \text{ with } \; j \in \{1, ..., |\mathcal{J}|\}. \end{aligned}$$

. . .

Tip

We often use plural names for sets and a caligraphic letter, e.g., $\mathcal{I}$ and $\mathcal{J}$.

Parameters

Parameters are fixed values that are given.

. . .

Question: What parameters are needed?

. . .

$$
\begin{aligned}
c_{i,j} &: \text{Costs per truck load for transport from } i \text{ to } j, \\
a_i &: \text{Available truck loads at } i, \\
b_j &: \text{Customer demands at } j.
\end{aligned}
$$

. . .

Tip

We usually use the corresponding lower-case letter, e.g., c, a, and b.

Decision Variable

- Decision variables are the values we are trying to find
- Here, our objective is to minimize the transport costs

. . .

Question: What decision variables are needed?

. . .

$$X_{i,j} \text{ Trucks that deliver panels from site } i \text{ to customer } j.$$

. . .

Tip

We use upper-case letters to distinguish variables from parameters, e.g., X.

These are our

building blocks!

Objective Function

- The objective function is the value we are trying to minimize (or maximize)
- Formalized as a sum of decision variables and parameters

. . .

Question: Do you remember the objective?

Minimizing the Transport Costs

Minimize the transport costs over all truckloads while meeting customer demand within the available supply from production sites.

. . .

Question: How can we write this down?

. . .

$$\text{Minimize} \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{i,j} \times X_{i,j}$$

Constraints

- Constraints are conditions that must be met
- They limit the solution space!

Question: Objective value without any constraints?

. . .

- The value is zero
- We can transport any number of panels

. . .

Question: What constraints are needed?

Supply Constraints

Ensure that the number of panels transported from a location does not exceed the available panels.

. . .

Question: How can we formalize this?

. . .

$$\sum_{j \in \mathcal{J}} X_{i,j} \leq a_i \quad \forall i \in \mathcal{I}$$

Demand Constraints

Ensure that the demand of each customer is covered.

. . .

Question: Any ideas?

. . .

$$\sum_{i \in \mathcal{I}} X_{i,j} = b_j \quad \forall j \in \mathcal{J}$$

Non-negativity Constraints

Ensure no negative number of truckloads are transported.

. . .

Question: Has anyone an idea how to write this down?

. . .

$$X_{i,j} \geq 0 \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}$$

Transport Problem

The complete model can then be written as:

$$\begin{aligned}
\text{Minimize} \quad & F = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{i,j} \times X_{ij} \\
\text{subject to:} \quad & \sum_{j \in \mathcal{J}} X_{i,j} \leq a_i & \forall i \in \mathcal{I} \\
& \sum_{i \in \mathcal{I}} X_{i,j} = b_j & \forall j \in \mathcal{J} \\
& X_{i,j} \geq 0 & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}
\end{aligned}$$

Inequality Constraints

Question: Could we replace $=$ by $\geq$ in the demand constraint?

. . .

- Yes, we could!
- We could deliver more than the demand
- But this would not happen here

. . .

Question: Why won't we transport more than the demand?

. . .

- Due to the associated costs!

Any

questions?

Profit Maximization

Description

Unfortunately, the margins on solar panels are low. After the previous contract has been fulfilled, the company produced the same number of panels as before. In addition, all three customers want to order the same number of truckloads with solar panels again. The revenue per truckload of panels is 11,000 Euros. The complete production of a truckload of solar panels, including materials, costs 6,300 Euros.

New Objective

In the new contract, the company wants to maximize its profits while the demand does not have to be fulfilled.

. . .

Question: What changes are necessary?

. . .

- We need to change the objective function
- We need to change some parameters
- We need to adjust some constraints

. . .

Question: Does our decision variable change?

. . .

- No, we still transport truckloads of solar panels

New Parameters

$$\begin{aligned} r &: \text{Revenue per truckload of solar panels}, \\ c &: \text{Production costs per truckload of solar panels}. \end{aligned}$$

. . .

Question: What is the profit per truckload of solar panels?

. . .

$$p = r - c$$

Former Model

$$\begin{aligned} \text{Minimize} \quad F = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{i,j} \times X_{i,j} & \\ \text{subject to:} \quad \sum_{j \in \mathcal{J}} X_{i,j} \leq a_i & \qquad \forall i \in \mathcal{I} \\ \sum_{i \in \mathcal{I}} X_{i,j} \geq b_j & \qquad \forall j \in \mathcal{J} \\ X_{i,j} \geq 0 & \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \end{aligned}$$

. . .

Question: What do we need to change here?

New Model

$$
\begin{aligned}
\text{Maximize} \quad & F = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (p - c_{i,j}) \times X_{i,j} & \\
\text{subject to:} \quad & \sum_{j \in \mathcal{J}} X_{i,j} \leq a_i & \forall i \in \mathcal{I} \\
& \sum_{i \in \mathcal{I}} X_{i,j} \leq b_j & \forall j \in \mathcal{J} \\
& X_{i,j} \geq 0 & \forall i \in \mathcal{I}, \forall j \in \mathcal{J}
\end{aligned}
$$

Model Reflection

Take a moment to think about what we just built:

Complexity

How many decision variables does this problem have?

$|\mathcal{I}| \times |\mathcal{J}| = 2 \times 3 = 6$ variables

Each represents a shipping route from a production site to a customer.

Scalability

What happens if we add more production sites?

The number of variables grows as $|\mathcal{I}| \times |\mathcal{J}|$

- 5 sites × 10 customers = 50 variables
- 10 sites × 20 customers = 200 variables
- The model structure stays the same, but computational complexity increases!

Assumptions

What real-world factors are we ignoring?

- Vehicle capacity limits per route
- Time windows for delivery
- Driver working hours and breaks
- Traffic conditions and travel time
- Fuel costs vs. distance relationship
- Possibility of multi-stop routes

Variations

How would the model change for different scenarios?

- Air transport: Higher costs, faster delivery, weight limits
- Multiple vehicle types: Different capacities and costs
- Time-sensitive deliveries: Add scheduling constraints
- Partial shipments: Allow fractional truckloads

What Did We Learn?

Modeling Process:

- Define the problem clearly

- Identify decision variables
- Formulate obj. function
- Add necessary constraints
- Verify completeness

Key Insights:

- Sets organize our indices
- Parameters hold data
- Variables = decisions
- Constraints limit feasibility
- Obj. drives optimization

. . .

Tip

This systematic approach works for any optimization problem!

Do you have

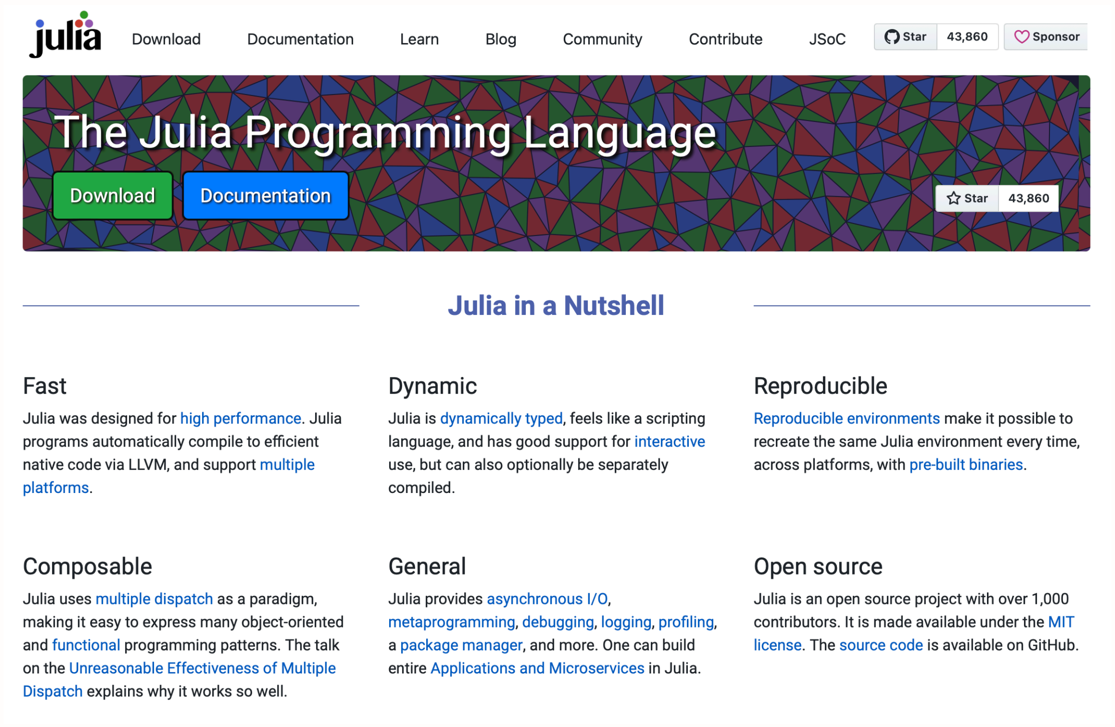
any questions?

Installing Julia

Download and Install Julia

To prepare for the upcoming lectures, we start by installing the Julia Programming Language and an Integrated Development Environment (IDE) to work with Julia.

Installating Julia



Julia in a Nutshell

Fast

Julia was designed for high performance. Julia programs automatically compile to efficient native code via LLVM, and support multiple platforms.

Dynamic

Julia is dynamically typed, feels like a scripting language, and has good support for interactive use, but can also optionally be separately compiled.

Reproducible

Reproducible environments make it possible to recreate the same Julia environment every time, across platforms, with pre-built binaries.

Composable

Julia uses multiple dispatch as a paradigm, making it easy to express many object-oriented and functional programming patterns. The talk on the Unreasonable Effectiveness of Multiple Dispatch explains why it works so well.

General

Julia provides asynchronous I/O, metaprogramming, debugging, logging, profiling, a package manager, and more. One can build entire Applications and Microservices in Julia.

Open source

Julia is an open source project with over 1,000 contributors. It is made available under the MIT license. The source code is available on GitHub.

- Head to julialang.org and follow the instructions.
- The easiest way to install Julia is via the shell/terminal
- Later, you can then manage Julia with `juliaup`

. . .

Tip

If you are ever asked to add something to your "PATH", do so!

VS Code

- Next, we are going to install VS Code
- Head to the website code.visualstudio.com
- Download and install the latest release

Verify the Installation

- Start the IDE and take a look around
- Search for the field “Extensions” on the left sidebar
- Click it and search for “Julia”
- Download and install “Julia (Julia Language Support)”

. . .

⚠ Warning

Any problems? Ask me for help!

Create a new file

- Create a new file with a ".jl" ending
- Save it somewhere on your computer
- e.g., in a folder that you will use in the course

```
print("Hello World!")
```

```
Hello World!
```

. . .

- Run the file by clicking "run" in the upper right corner
- OR by pressing "Control+Enter" or "STRG+Enter"

Everything working?

- If the terminal opens with a `Hello World!` → perfect!
- If not, the IDE likely cannot find the path to Julia
- Try to determine the path and save it to VS Code
- After saving it, try to run the file again

. . .

> Tip
>
> Don't worry if it is not running right away. We will fix this together!

Installation Checklist

Before the next lecture, try to ensure you can:

- ☐ Open VS Code
- ☐ See Julia extension in the extensions panel
- ☐ Create a new `.jl` file
- ☐ See syntax highlighting in your Julia file
- ☐ Run code and see output in the terminal

. . .

> Warning
>
> Having trouble? We will fix your issues together in the next lecture!

Starting with Julia

How to get started?

- Learning a new programming language is a daunting task

- It is best to start with some small, interactive problems
- Then, slowly increase the scope of the tasks
- We will do this together in class!

. . .

Note

And that's it for todays lecture!
We now have covered a first introduction and are ready to start solving some problems in the upcoming lectures.

Questions?

Literature

Literature I

For interesting literature to learn more about Julia, take a look at the literature list of this course.

Literature II

Bibliography

[1] M. Mickein, M. Koch, and K. Haase, "A Decision Support System for Brewery Production Planning at Feldschlösschen," INFORMS Journal on Applied Analytics, vol. 52, no. 2, pp. 158–172, 2022.

[2] T. Vlćek, K. Haase, M. Fliedner, and T. Cors, "Police service district planning," OR Spectrum, Feb. 2024, doi: 10.1007/s00291-024-00745-3.

[3] T. Vlćek, K. Haase, M. Koch, L. Dolz, A. Weygandt, and J. Pape, "Controlling Passenger Flows into Metro Systems to Mitigate Overcrowding during large-scale Events," Submitted to Transportation Research: Part B, 2024.

[4] T. Vlćek and G. Voigt, "Optimizing SKU-Warehouse Allocations to minimize Split Parcels in E-Commerce Environments," To be Submitted Soon, 2024.

[5] K. Haase et al., "Improving pilgrim safety during the hajj: an analytical and operational research approach," Interfaces, vol. 46, no. 1, pp. 74–90, 2016.