Python Concept of Project

Implementing the Master Production Schedule, Lot Sizing, and Traveling Salesman Problem in Python

---

Group 4 - Names of Contributors (alphabetical order):

- Rosanelli Simon - a12108938@unet.univie.ac.at
- Telek Márton - a12142811@unet.univie.ac.at
- Zadina Selina - a12021659@unet.univie.ac.at

Content of Project Proposal

Vienna, 25th of November 2025,

1. Introduction

The project at hand aims to connect the content taught in the course '*ABWL Produktion und Logistik II (2025W)*' at the University of Vienna with the Python knowledge discussed in '*Introduction to Programming for Business Students*'. The main topics that will be covered in this project will be (1) the Master Production Schedule (MPS), (2) Lot Sizing, and (3) the Traveling Salesman Problem (TSP).

By linking these two classes - Production and Logistics, and Programming in Python-, this project aims to create a programme that aids managers in the decision-making of various questions in the areas mentioned above. Furthermore, the selection of topics aims at modelling three chronologically occurring real-world issues that business companies face: Firstly, the amounts to be produced are calculated, factoring in the constraints of limited resources. Then, lot sizes are computed while minimizing setup and storage costs. Lastly, the order of delivery is decided upon. These three areas will be united by a menu system that enables the user to not only calculate the production schedule, for instance, but also alter and update the data input/output. Additionally, the interactive menu is supported by various data-visualizations.

The theoretical base for this project will be the course materials from 'ABWL Produktion und Logistik II', more precisely chapters one, four, and seven. The datasets for the Master Production Schedule (1) and lot sizing parts (2) of this project will be generated by us and inspired by exercises from the aforementioned class. The dataset for the Traveling Salesman Problem (3) will consist of real-life data.

2. Problem definition

As introduced above, this project aims to implement three fundamental concepts from Production and Logistics in Python and combine them in an interactive menu. The issues at hand are real-world planning problems faced by companies.

(1) Master Production Schedule (MPS):

The MPS determines how much of each product should be produced to maximise profit while bound by limited resources, for instance, stock levels of raw materials or machine capacity. While problems containing only two products can be easily computed by hand via linear programming, real-life examples show that companies usually manage numerous products at a time, rendering manual solutions impossible. Implementing the MPS in Python allows us to compute profitable production plans for more realistic scenarios.

(2) Lot Sizing:

Here, with a given demand, lot sizes are computed in order to minimize set-up and storage costs. We will be implementing two methods:

(a) Wagner-Whitin is an exact solution method that guarantees to find optimal lot sizes.

(b) Just-In-Time (JIT): Here, we have maximum set-up costs and a non-optimal result. Nevertheless, we wanted to implement this heuristic because the user might want to know how high costs would be if production occurred in every period. This enables the user to take into account other factors, such as the flexibility of production.

(3) Traveling Salesman Problem (TSP):

The TSP models routing decisions: a company must visit multiple customer/supplier locations while trying to keep the trip as short as possible. A problem with a realistic number of locations cannot be solved for one exact, optimal solution. Therefore, we will be implementing a heuristic approach in Python.

The three topics discussed represent important decision-making areas in production planning: ‘How much should/can I produce?’, ‘When should I produce?’, ‘How can I maximize profits while minimizing costs?’ and ‘In which order should I deliver to my clients?’. The goal of this project is to implement a programme supporting real-world logistics and operational planning. A programme like this may especially be useful for young entrepreneurs and small businesses.

3. Methodology & Data

This class newly introduced us to the programming language Python. Since we are far from experts, we will be taking an iterative approach, meaning that firstly, we will implement the mechanisms behind the three topics mentioned in the introduction separately. Only after running these parts successfully will we begin putting them together and constructing the menu that will link these areas.

3.1 Data

The data used in our project will be provided through Excel files. For the first two topics (Master Production Schedule and Lot Sizing), we will create our own datasets inspired by examples from the class 'Production and Logistics'. For the Traveling Salesman Problem, we will receive a real-world dataset supplied by a member of the Production and Logistics department.

3.2 Work packages/Libraries in Python

Task: Data Handling & Computation

- **Pandas:** This library will be used for reading, writing, and manipulating datasets in classical data formats such as csv or xlsx. Additionally, we can use the package for storing data on MPS tables (1), lot sizing tables (2), as well as demand and cost parameters. By executing inherent functions of the library, we create summary tables with, for example, production quantities and costs.

- **NumPy:** Built-in functions of this library support us with fast numerical calculations, performing vectorized cost calculations in MPS (1) and lot sizing (2), to program dynamic algorithms like Wagner-Whitin for lot sizing (2), and creating matrices concerning the distance matrix for the Travelling salesman problem (3).

Task: Optimization & Algorithms

- **SciPy:** This package supports the project with its inherent optimization functions (cost minimization tasks, (2) lot sizing optimization).

- **PuLP (Linear Programming):** This library can also be used for formulating the optimization problem of the MPS (1) and solving lot sizing models (2) with respect to production quantities/costs under budget constraints while using the Wagner-Whitin-Algorithm.

Task: User Interface Layer of our menu system

- **Streamlit/Shiny for Python:** One of these packages will be used to build a simple menu-based system/interactive dashboard. The libraries allow us to implement buttons that contain functions for calculations and might help with the file upload of the dataset. The goal of using these libraries is to generate a modern, more professionally looking interface for our menu system.

Task: Visualization of (input/output) data

- **Matplotlib/Seaborn:** Additionally, we might include graphical visualizations on pre-chosen input/output data within our menu-based system.

3.3 Collaboration between group members

To ensure efficient cooperation and transparency within our group, we will make use of GitHub, a version-control platform that supports collaboration. On GitHub, we will coordinate our project development and manage all code sections related to the three core business problems: the MPS (1), lot sizing (2), and the TSP (3).

The advantages of using this platform are that every modification is recorded transparently, thus we ensure that no code work is lost or overwritten. Since all project members will soon be familiar with the (basic) usage of GitHub, we also benefit from working independently on different branches/tasks (see '*3.2 Work packages/Libraries in Python*'), before merging all code sections together into one final main project. Altogether, GitHub provides a good platform for executing our iterative approach mentioned before.

4. Project Plan

Timetable - Project in Python (Group 4)					
Due by:	25. Nov	02. Dec	13. Jan	20. Jan	27. Jan
Orientation & Planning					
Group formation and collection of topic ideas					
Choose project + dataset, assigning roles					
Concept draft: Intro, Methodology, Project Plan					
Submission of Proposal for Group Project in Python (6 pages)					
Submission of peer review form					
Constructive feedback to other partner group					
Coding Phase 1					
Basic functions, data loading, menu structure					
Coding Phase 2					
Visualizations, exception handling, final code					
Peer Review presentation					
Presentation of our project					
Finalization & Submission					
Proofreading/testing code structure					
Upload of final code-version					
Final project presentation					
Final Presentation of our project					

5. Realization (until 13th of January)

6. References of work packages/libraries

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

Hess, C. (2025). ABWL Produktion und Logistik II [Vorlesungsfolien]. Universität Wien. https://ufind.univie.ac.at/de/course.html?lv=040137&semester=2025W

Hunter, J. D. (2007). *Matplotlib: A 2D graphics environment*. *Computing in Science & Engineering*, 9(3), 90–95.

Mitchell, S., & OSullivan, M. (2011). *PuLP: A linear programming toolkit for Python*. The University of Auckland, Python Software Foundation. Available at: https://coin-or.github.io/pulp/

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) **SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python**. *Nature Methods*, 17(3), 261-272. DOI: 10.1038/s41592-019-0686-2.

RStudio, PBC. (2023). *Shiny for Python: Interactive web applications with Python*. Posit Documentation. Retrieved from https://shiny.posit.co/py/

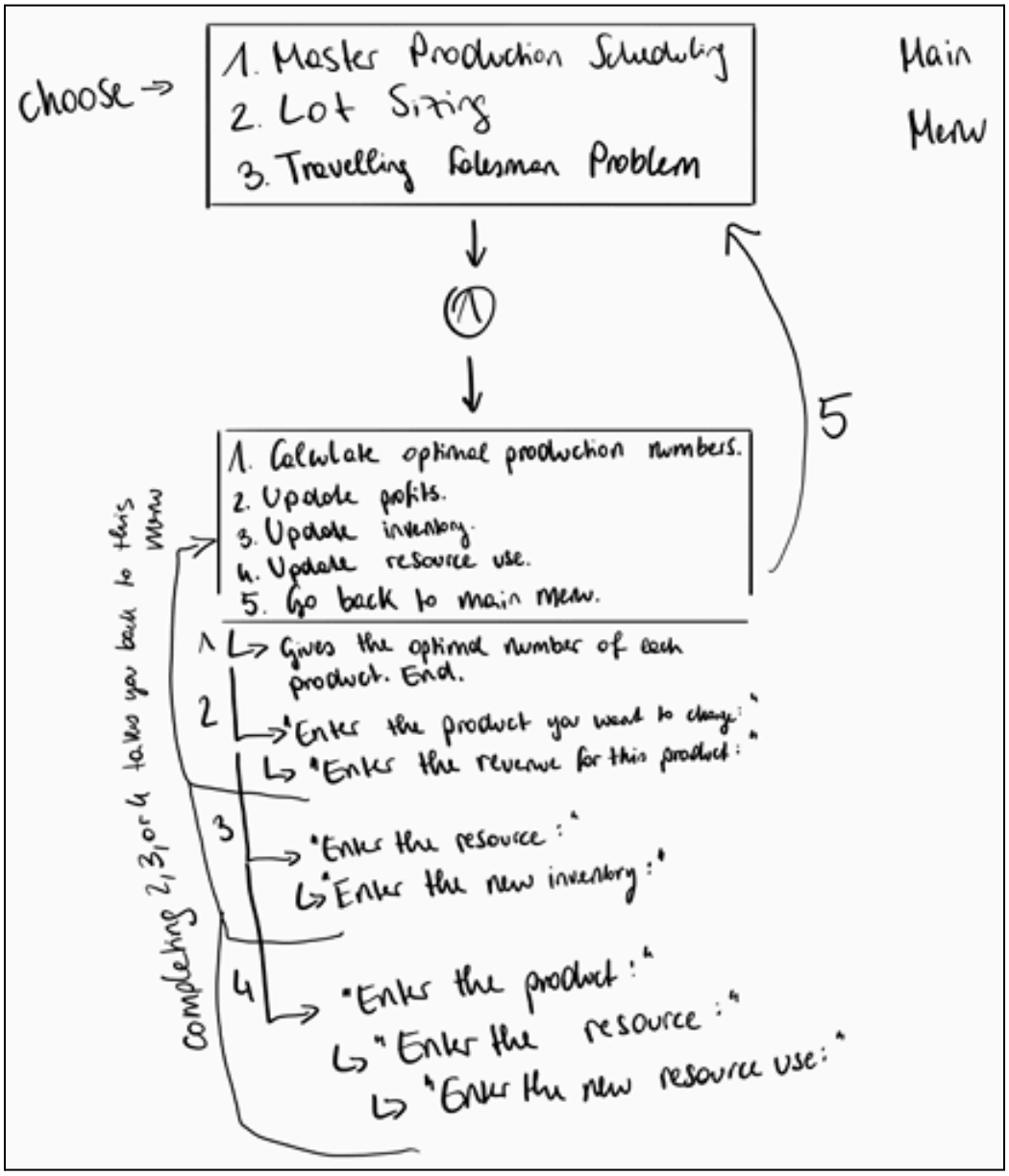
Streamlit Inc. (2023). *Streamlit: The fastest way to build data apps in Python*. Streamlit Documentation. Retrieved from: https://docs.streamlit.io/

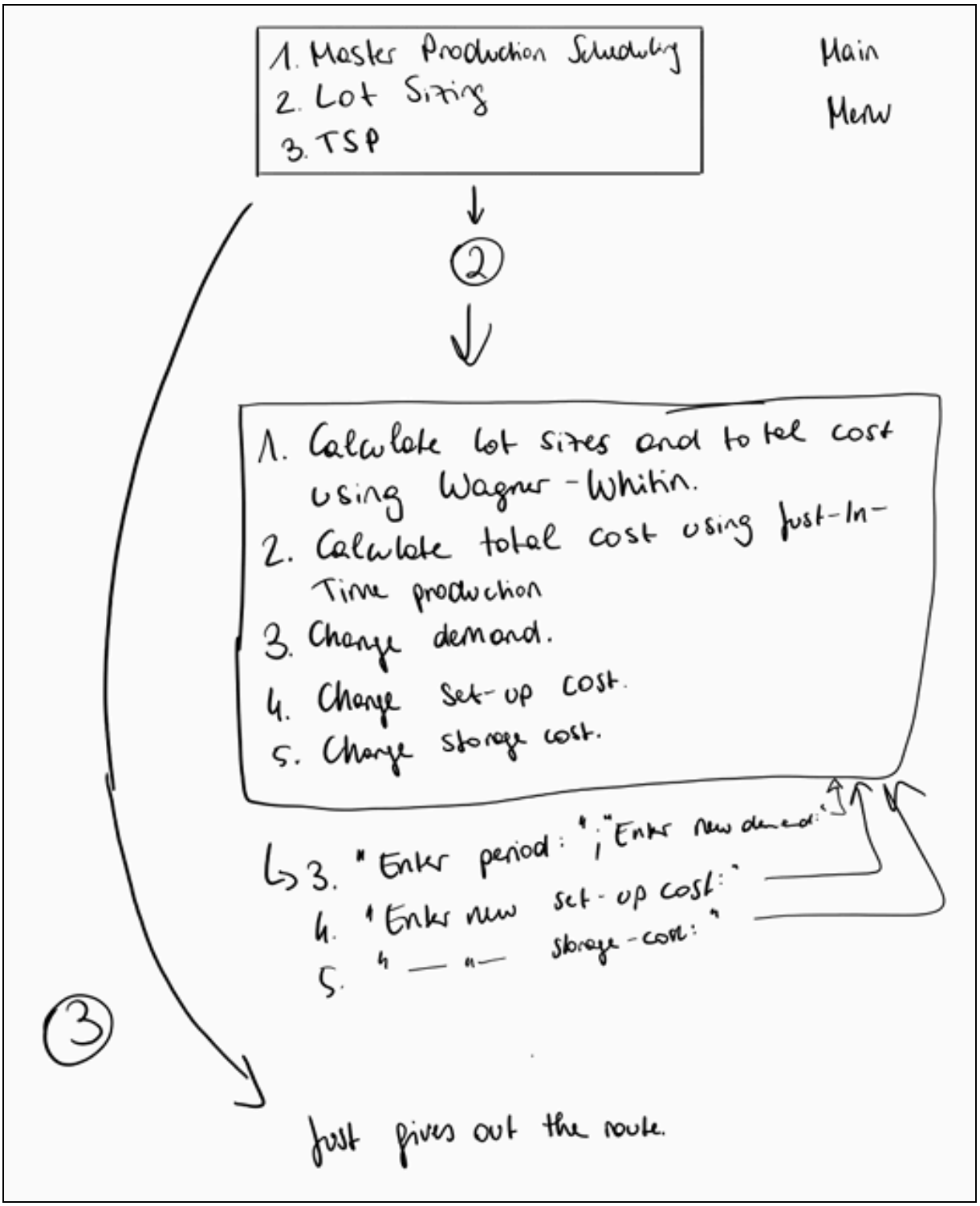
The pandas development team. (2025). pandas-dev/pandas: Pandas (v2.3.3). Zenodo. https://doi.org/10.5281/zenodo.17229934

Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021, https://doi.org/10.21105/joss.03021.

7. Appendix

This is a rough sketch of a system architecture diagram for our Python group project:




1. Master Production Scheduling
2. Lot Sizing
3. TSP
Main Menu
2
1. Calculate lot sizes and total cost using Wagner-Whitin.
2. Calculate total cost using just-in-Time production
3. Change demand.
4. Change set-up cost.
5. Change storage cost.
3. " Enter period: " ; " Enter new demand: "
4. " Enter new set-up cost: "
5. " — " storage-cost: "
3
just gives out the route.

❓ *INFO on all Assignments*

Python Project

---

1. INFO Group Projects:

-You will be assigned to the groups of 4-5
-Groups will be announced on October 20
-Each group will submit a project proposal
-Deadline: November 25, Midnight
-At least 6 pages including the Concept of Projects:

-Introduction (topic, data), Methodology (work packages, tasks), Project Plan

-Dataset: Use your own data file (from practice, uni, hobby, etc.) or select a dataset from Kaggle (www.kaggle.com)

-Realization: Write corresponding programs/analyses, Menu system, Visualizations, Try to cover as many topics as possible from this course*(Write corresponding programs/analyses, Menu system, Visualizations, Try to cover as many topics as possible from this course.* potential solution design using the Python environment.)

2. INFO Peer review presentations (20%):

-Each group presents their project and provide useful and constructive feedback to the partner group
-Each group will have 10 minutes (7 min. presentation + 3 min. feedback)
-Each group submits a peer review form

3. INFO Final project presentation (50%):

-Each group will have 20 minutes
-Each group member should speak during the presentation

Python Project Ideas:

https://www.ccbp.in/blog/articles/python-projects-for-final-year-students

1. Using Leaflet (package: folium):
https://leafletjs.com/examples/choropleth/
https://leafletjs.com/examples.html
https://courses.spatialthoughts.com/python-dataviz.html#interactive-maps-with-folium
https://www.broadlyepi.com/python/pyfriday-tutorial-creating-leaflet-maps-in-python-with-folium-and-geopandas/
https://realpython.com/python-folium-web-maps-from-data/

2. Webscraping in Python:

3. Interactive Dashboard (Time-Series Forecasting) with Shiny/Streamlit:

https://towardsdatascience.com/applied-models-of-production-planning-systems-in-python-942a2c91e6ad/

https://medium.com/data-science/production-fixed-horizon-planning-with-python-8dd38b468e86

https://medium.com/@mb20261/python-by-examples-mastering-capacity-planning-for-optimal-business-performance-2-of-2-8e64e02d25c7

Introduction to Python:
Why is Programming vital for modern businesses?
-Data: Evidence-based decisions
-Automation: Enabling efficiency by handling repetitive tasks
-Innovation: Creating new products and processes
-Finance: Analysis of large datasets and forecasting, development of fintech applications and risk management
-Marketing: Building interactive websites to track customer behavior and analyzing user data for targeted campaigns
-HR: Digital transformation by automating the hiring process, modeling employee retention and enabling data-driven talent managementstrategies

Benefits of Programming in Business:
-Automation increases the efficiency and cost-effectiveness
-Innovation with new products, services and business models
-Logical thinking and problem-solving skills for complex problems
-Improves communication and collaboration between business and IT teams

---