Hochschule Aalen
Beethovenstraße 1
73430 Aalen
Germany

Niklas Hess, Marco Nentwig, Christian Nguyen, Max Weixler

Management of Logistics Processes

Development of a Mixed Model Line Balancing in Python

Student Register Numbers: 82905, 82913, 83493, 83029

Module: Management of Logistics Processes

Semester: 7

Professor: Prof. Dr.-Ing. Nicole Stricker

Submission Date: February 12, 2024

Contents

List of Figures

Acronyms

GUI **G**raphical **U**ser **I**nterface

MMLB **M**ixed **M**odel **L**ine **B**alancing

VSC **V**isual **S**tudio **C**ode

1 Topic and Introduction

In manufacturing companies, the proper assembly line plays a crucial role in ensuring efficient production processes and meeting customer demand. Nevertheless, with wrong assembly line configurations, companies may face challenges such as high production costs, low productivity, and delays. In the shopfloor of a manufacturing facility, assembly line balancing is therefore a crucial process that aims to optimise the allocation of tasks among workers and machines. This process ensures that the assembly line operates efficiently and effectively, maximising productivity while minimising bottlenecks and idle time. There are several models in order to optimise and balance assembly lines, including the mixed model line balancing approach. This approach aims to balance the workload and maximise efficiency in assembly lines where multiple product models are being produced. The mixed model line balancing approach takes into consideration various factors such as workstation requirements, cycle times, and demand variability to determine the optimal allocation of tasks across workstations in order to prevent idle time. To address this challenge, we developed a software tool using Python programming language to faciliate the process of assembly line balancing for mixed model production.[1][2][3]

2 Project Management

In order to obtain a proper structure to fulfil the project requirements, we developed the known project management tools studied in the previous semesters. Those tools helped us to dissect different tasks and distribute them.

2.1 Magic Triangle

After gaining some theoretical knowledge of the course in the first weeks, we started developing our project on the 27th of October, 2023. In the beginning, we had to choose a project by the given list or come up with our own idea. We decided to take that into account and develop a tool which should help fellow students to gain more knowledge in the topic around the assembly line production. For that, we had resources of 90 workload hours per person which should help us achieve the desired goals. Regarding the sources

[1] cf. Friedli et al., 2010.
[2] cf. Rane et al., 2017.
[3] cf. Yusuf et al., 2020.

we were able to use, there were no boundaries which allows us to gain the theoretical knowledge of assembly line balancing through literature of scientific papers as well as materials from previous semesters where we talked about the assembly line production in detail. In the development of the Python tool for the practical part, we will primarily conduct internet research, as it proved to be a more time-efficient approach for addressing our specific issue. Additionally, we set biweekly meetings with our stakeholder which should help us developing the tool by talking about what features should be applied and how to make it as user-friendly as possible for students. Within the group, we are gonna implement those by meeting each other once a week and also think about how else to improve the tool.

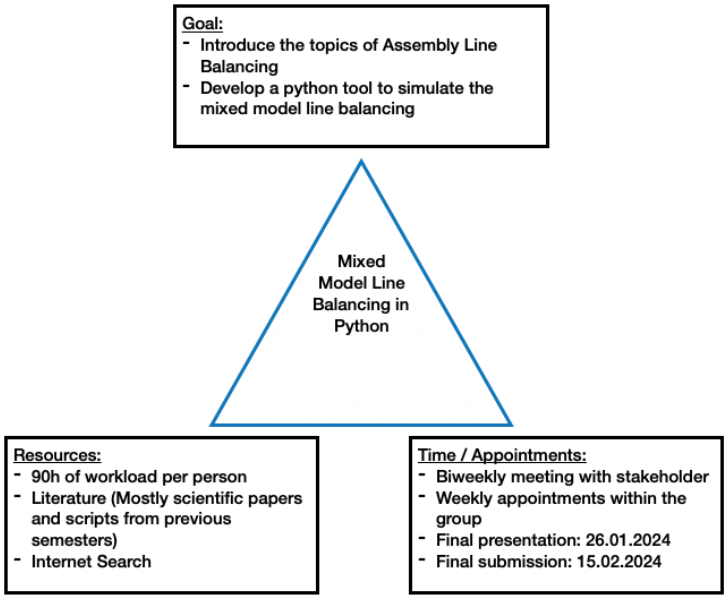


Figure 1: Magic Triangle (27.10.2023)

After finishing and refining our tool, we drew the magic triangle once again in comparison to the initiative one. As to be seen the triangle with the red border filled with blue, demonstrates our changes. For that, the most change that took place belongs to the category of resources. Especially by taking a look at our workload hours we invested more time than was initially planned. By an extension 20h per person, it allowed us to develop the tool as desired. This was necessary for the reason that we never learned how to build a tool with a graphical user interface. We allocated the additional time for gaining the Know-How which also lead to the change of our weekly appointments to two-day-meetings. The goal as well as the dates of submission did not vary. For that

reason, the modified triangle mostly shifted to the left since it had the biggest impact of our project process.

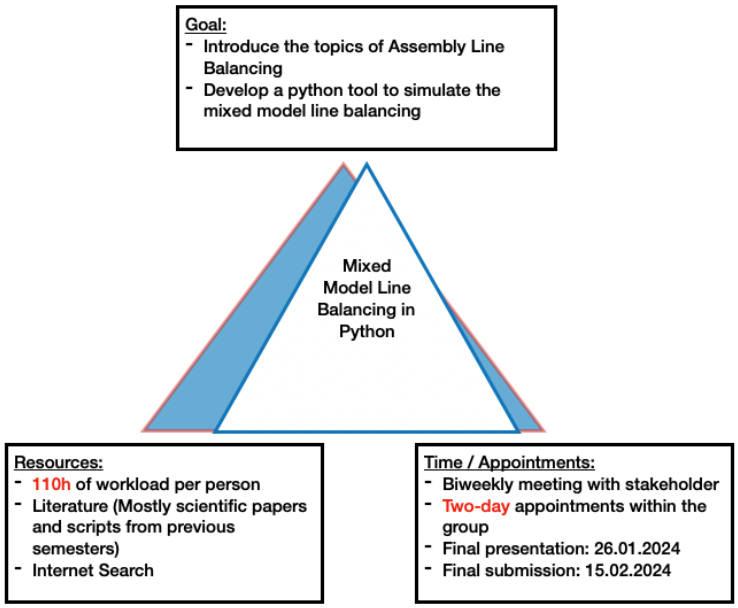


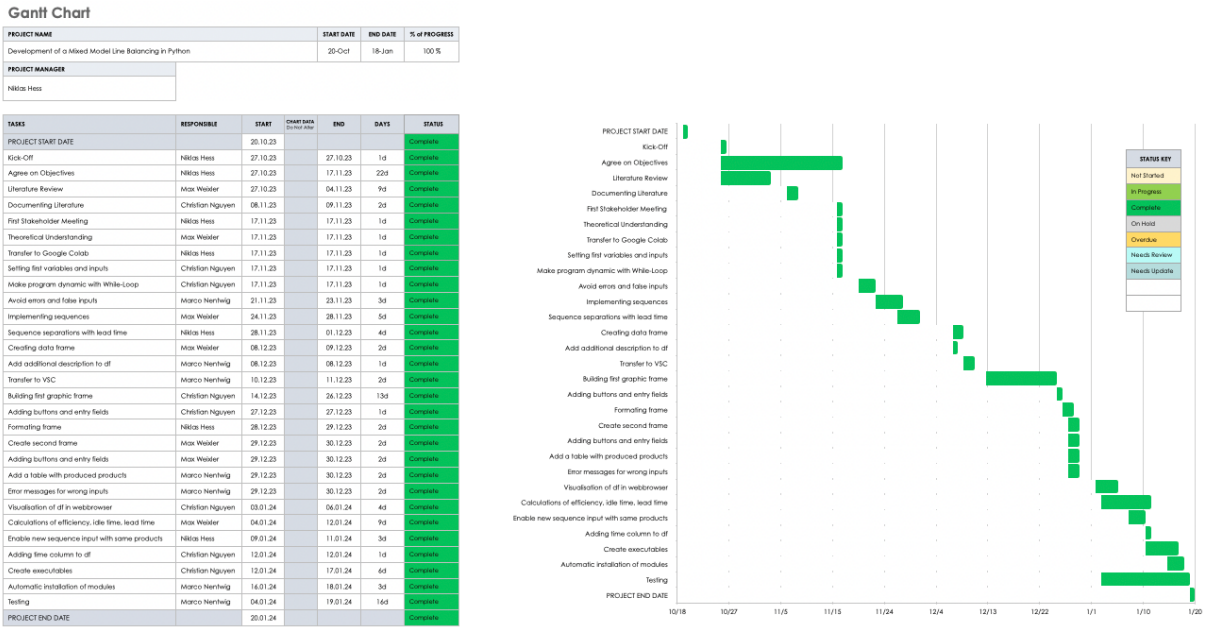
Figure 2: Magic Triangle (24.01.2024)

2.2 Gantt-Chart

Another project management tool we applied was the Gantt-Chart. Here, the stations of our project process can be seen in much more detail then in the magic triangle. The length of the bars in the diagram are linked to the duration of how long each station took. As to be seen, the creation and definition of our objectives had a big impact on our project. The reason for that was the lack of clarity within the group and the vague definition of what we want to achieve with the tool. By talking with our stakeholder Prof. Dr.-Ing. Nicole Stricker at the first meeting, we clarified the objectives and refined what we want to implement within the tool. Another big factor regarding the length is the implementation of our already developed code to Visual Studio Code (VSC). This had to be done in order to achieve the graphical user interface (GUI) so students do not necessarily see our code but can execute our tool by simply executing an application. Not only building the first frame and brainstorming how the GUI should look like took us some time but also doing researches on how to actually perform a GUI in Python since we haven't done that so far.

One advantage was that we already studied the basics of Python in the previous semesters

through different projects and also this semester helped us gaining more knowledge of how to set up dataframes.



Gantt Chart

PROJECT NAME	START DATE	END DATE	% of PROGRESS
Development of a Mixed Model Line Balancing in Python	20-Oct	18-Jan	100 %

PROJECT MANAGER
Niklas Hess

TASKS	RESPONSIBLE	START	CHART DATA Do Not Alter	END	DAYS	STATUS
PROJECT START DATE		20.10.23				Complete
Kick-Off	Niklas Hess	27.10.23		27.10.23	1d	Complete
Agree on Objectives	Niklas Hess	27.10.23		17.11.23	22d	Complete
Literature Review	Max Weisler	27.10.23		04.11.23	9d	Complete
Documenting Literature	Christian Nguyen	08.11.23		09.11.23	2d	Complete
First Stakeholder Meeting	Niklas Hess	17.11.23		17.11.23	1d	Complete
Theoretical Understanding	Max Weisler	17.11.23		17.11.23	1d	Complete
Transfer to Google Colab	Niklas Hess	17.11.23		17.11.23	1d	Complete
Setting first variables and inputs	Christian Nguyen	17.11.23		17.11.23	1d	Complete
Make program dynamic with While-Loop	Christian Nguyen	17.11.23		17.11.23	1d	Complete
Avoid errors and false inputs	Marco Nentwig	21.11.23		23.11.23	3d	Complete
Implementing sequences	Max Weisler	24.11.23		28.11.23	5d	Complete
Sequence separations with lead time	Niklas Hess	28.11.23		01.12.23	4d	Complete
Creating data frame	Max Weisler	08.12.23		09.12.23	2d	Complete
Add additional description to df	Marco Nentwig	08.12.23		08.12.23	1d	Complete
Transfer to VSC	Marco Nentwig	10.12.23		11.12.23	2d	Complete
Building first graphic frame	Christian Nguyen	14.12.23		26.12.23	13d	Complete
Adding buttons and entry fields	Christian Nguyen	27.12.23		27.12.23	1d	Complete
Formating frame	Niklas Hess	28.12.23		29.12.23	2d	Complete
Create second frame	Max Weisler	29.12.23		30.12.23	2d	Complete
Adding buttons and entry fields	Max Weisler	29.12.23		30.12.23	2d	Complete
Add a table with produced products	Marco Nentwig	29.12.23		30.12.23	2d	Complete
Error messages for wrong inputs	Marco Nentwig	29.12.23		30.12.23	2d	Complete
Visualisation of df in webbrowser	Christian Nguyen	03.01.24		06.01.24	4d	Complete
Calculations of efficiency, idle time, lead time	Max Weisler	04.01.24		12.01.24	9d	Complete
Enable new sequence input with same products	Niklas Hess	09.01.24		11.01.24	3d	Complete
Adding time column to df	Christian Nguyen	12.01.24		12.01.24	1d	Complete
Create executables	Christian Nguyen	12.01.24		17.01.24	6d	Complete
Automatic installation of modules	Marco Nentwig	16.01.24		18.01.24	3d	Complete
Testing	Marco Nentwig	04.01.24		19.01.24	16d	Complete
PROJECT END DATE		20.01.24				Complete

Figure 3: Gantt-Chart

3 Results

After gaining all the needed knowledge to perform our task, the Python tool to simulate a Mixed-Model-Line-Balancing (MMLB) regarding an assembly line production was created. The given requirements to fulfill the project were that we have a fixed number of eight stations in total. We divided the program into three categories. The first one was the beginning frame in which students and other users can enter the products they want to produce. A list of products is given and a manually entered product to produce won't be accepted. According to the products, users should also enter the cycle time which states how long the production time of the product will be, respectively. A double entry of a product to be produced twice will also return an error message by the program which leads to the possibility to produce the maximum of five products. After submitting all the desired products to be produced, the next category will appear. A second frame with a table containing all the previously entered products is to be seen contributing to a better overview. Additionally, there are three other entry fields for which the users can enter the lead time for the stations, the sequence on which order the products should be produced and the amount of how often the sequence should be repeated. The lead time is due to

convenience equal for each station and when parts in the sequence exceed the lead time, the sequence will separate the sequence at this point and move the remaining products to be produced to the next row. This will continue until the total sequence left does not exceed the lead time anymore. After entering these datas, the user may finish the program and the default web browser will open itself with a table. This table shows the simulation of an assembly line production using a MMLB. All the products entered into the tool will run through each column which are demonstrating the stations. The first part of the sequence, until the lead time exceeds, gets things started in column or station one. After the lead time is reached, the first part of the sequence will be shifted to the next station and the second part of the sequence goes into station one. This will continue until the sequence multiplied by the entered repetition will be attained and all sequence ran through each station. Additionally to the table, a quick overview on how many parts of each products has been produced alongside with the total produced time. Moreover, the efficiency, idle time, and lead time are computed and displayed, enabling students to gain a deeper comprehension of the assembly line production. They can experiment with the inputs to potentially enhance the assembly line's efficiency.

4 Conclusion

Our task was to develop a Python tool to enhance the understanding of fellow students in terms of an assembly line production. For that, the MMLB was applied and we should reconstruct a simulation of the production line so students can interact with the tool, change values and see the impact of those changes. The initial phase was challenging because it had been some time since we last engaged with topics related to assembly line production and MMLB. We needed several days to revisit the theoretical foundations of these subjects before we could begin the actual development of the program. The development of a graphical user interface (GUI) and the creation of an executable for cross-platform compatibility also proved to be time-intensive tasks. Our lack of experience in this area necessitated additional self-study.

5 Lessons Learned

This project let us learn that achieving proper balance in assembly line production is not only an advantage but essential for a functional and organised production process. Moreover, it helps to minimise mistakes and errors, leading to an overall improvement in production efficiency. We also improved our Python skills by adding a user-friendly GUI. However, unclear goals caused us to spend extra time redoing work. Effective scheduling proved to be just as important, helping us to manage our time and resources better.

Literature

Friedli, T., Basu, P. K., Gronauer, T., and Werani, J. (2010). *The pathway to operational excellence in the pharmaceutical industry: overcoming the internal inertia*. ECV, Editio-Cantor-Verlag.

Rane, A. B., Sunnapwar, V. K., and Khot, S. (Jan. 2017). "Cost models for improved vehicle assembly line performance". In: *International Journal of Simulation and Process Modelling* 12.2, p. 111. DOI: 10.1504/ijspm.2017.083527. URL: https://doi.org/10.1504/ijspm.2017.083527.

Yusuf, M. A., Damayanti, D. D., and Astuti, M. D. (2020). "Mixed-Model Assembly Line Balancing in The Process of Assembling Trimming Area to Minimize Workstation Using RPW-MVM Method". In: *SHS Web of Conferences*. Vol. 86. EDP Sciences, p. 01028.

Declaration of Autonomy

We herewith declare that we have completed the present report independently, without making use of other than the specified literature and aids. All parts that were taken from published and non-published texts either verbally or in substance are clearly marked as such. This report has not been presented to any examination office in the same form.

Location, Date: Aalen, 12 February 2024

Signature: _____

Signature: _____

Signature: _____

Signature: _____