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: 83493

Module: Smart Factory

Semester: 7

Professor: Prof. Dr.-Ing. Nicole Stricker

Submission Date: February 1, 2024

Contents

List of Figures

Acronyms

CSF	**C**ritical **S**uccess **F**actor
EFQM	**E**uropean **F**oundation for **Q**uality **M**anagement
FTE	**F**ull **T**ime **E**quivalent
ISO	**I**nternational **O**rganization for **S**tandardization
KPI	**K**ey **P**erformance **I**ndicator
OEE	**O**verall **E**quipment **E**ffectiveness
PI	**P**erformance **I**ndicator
RI	**R**esult **I**ndicator
VPN	**V**irtual **P**rivate **N**etwork VSC

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 (Friedli et al., 2010). 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 (Yusuf et al., 2020). 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.

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 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.

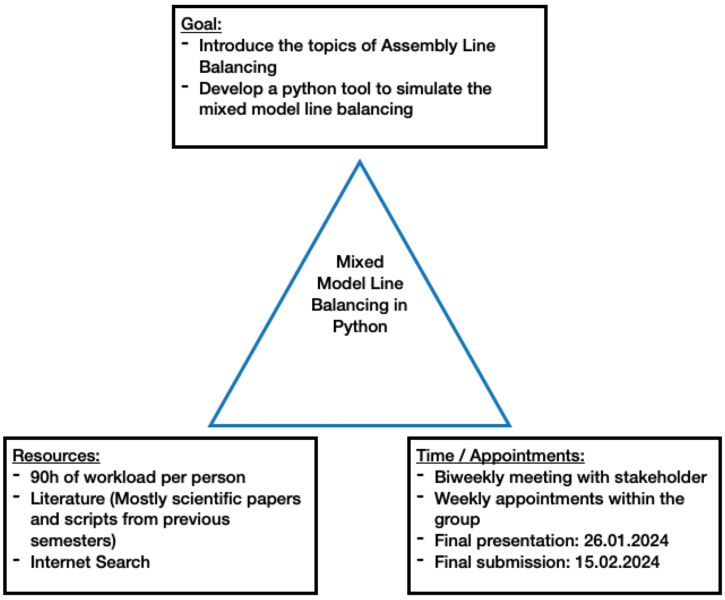

Goal:
- Introduce the topics of Assembly Line Balancing
- Develop a python tool to simulate the mixed model line balancing
Mixed Model Line Balancing in Python
Resources:
- 90h of workload per person
- Literature (Mostly scientific papers and scripts from previous semesters)
- Internet Search
Time / Appointments:
- Biweekly meeting with stakeholder
- Weekly appointments within the group
- Final presentation: 26.01.2024
- Final submission: 15.02.2024

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.

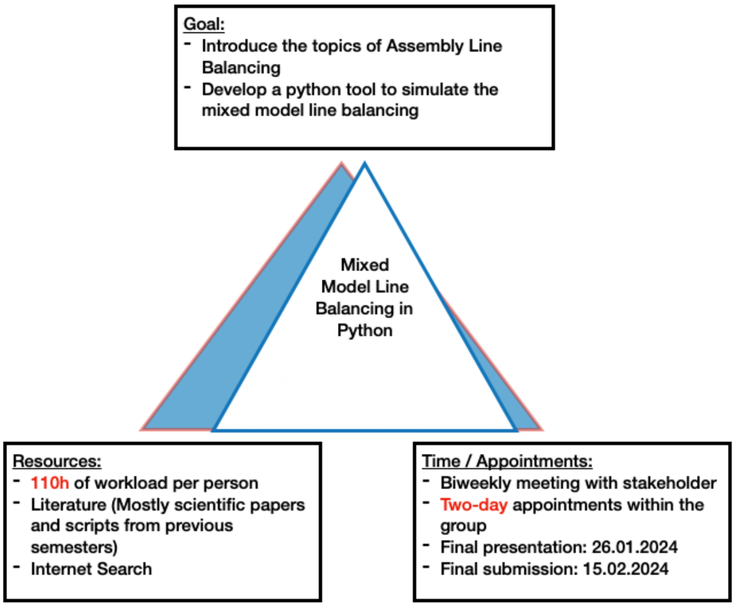
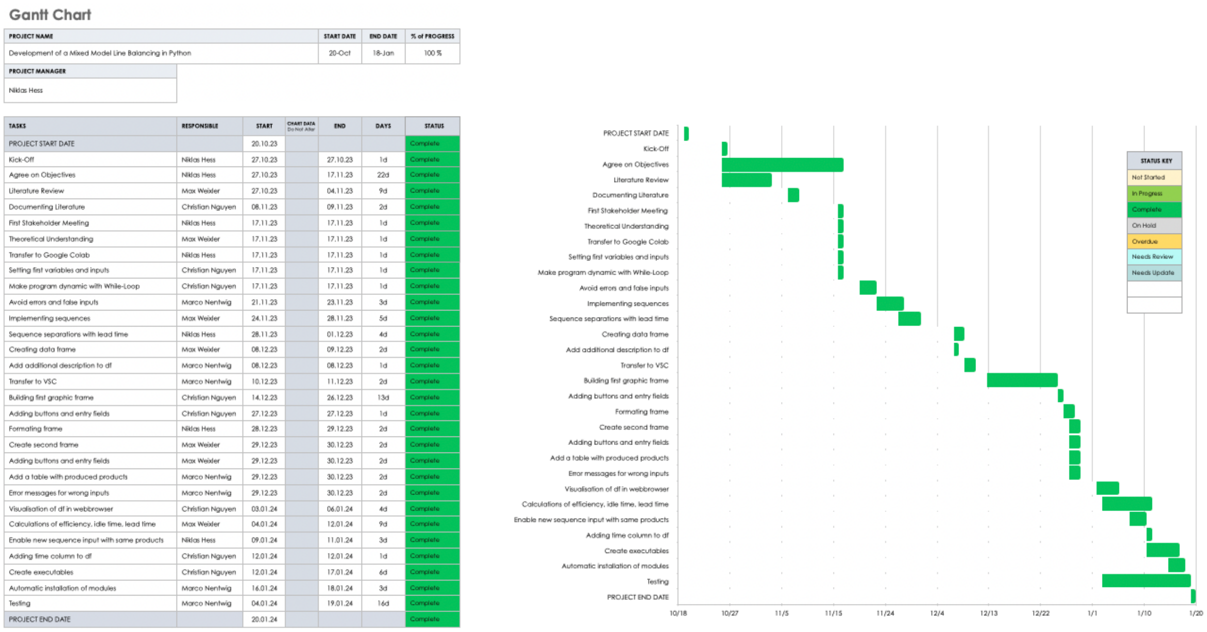

Goal:
- Introduce the topics of Assembly Line Balancing
- Develop a python tool to simulate the mixed model line balancing
Mixed Model Line Balancing in Python
Resources:
- 110h of workload per person
- Literature (Mostly scientific papers and scripts from previous semesters)
- Internet Search
Time / Appointments:
- Biweekly meeting with stakeholder
- Two-day appointments within the group
- Final presentation: 26.01.2024
- Final submission: 15.02.2024

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 and 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 Weixler	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 Weixler	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 Weixler	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 Weixler	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 Weixler	29.12.23		30.12.23	2d	Complete
Adding buttons and entry fields	Max Weixler	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 Weixler	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

PROJECT START DATE
Kick-Off
Agree on Objectives
Literature Review
Documenting Literature
First Stakeholder Meeting
Theoretical Understanding
Transfer to Google Colab
Setting first variables and inputs
Make program dynamic with While-Loop
Avoid errors and false inputs
Implementing sequences
Sequence separations with lead time
Creating data frame
Add additional description to df
Transfer to VSC
Building first graphic frame
Adding buttons and entry fields
Formating frame
Create second frame
Adding buttons and entry fields
Add a table with produced products
Error messages for wrong inputs
Visualisation of df in webbrowser
Calculations of efficiency, idle time, lead time
Enable new sequence input with same products
Adding time column to df
Create executables
Automatic installation of modules
Testing
PROJECT END DATE
10/18
10/27
11/5
11/15
11/24
12/4
12/13
12/22
1/1
1/10
1/20
STATUS KEY
Not Started
In Progress
Complete
On Hold
Overdue
Needs Review
Needs Update

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.

Statutory Declaration

Hereby we, the individuals listed below, declare that we have independently authored the present work, that we have not previously submitted it as an examination performance at any other university or in any other course of study, and that we have not used sources or aids other than those specified. All passages in the work that have been taken literally or paraphrased from publications or other external expressions are clearly identified as such.

Ort, Datum: Aalen, 1 February 2024

Unterschrift: ______________________

Unterschrift: ______________________

Unterschrift: ______________________