Smart Production Flow Algorithm Project

VitrA ARTEMA

Sabancı Üniversitesi — FACULTY OF ENGINEERING AND NATURAL SCIENCES

Student(s)

- Senanur Açıl
- Fulya Ayhan
- Valeria Casagrande

Faculty Member(s)

- Tonguç Ünlüyurt

Company Advisor(s)

- Muhammed Burak Göller
- Can Özbalkan

ABSTRACT

The project "Smart Production Flow Algorithm" is conducted for a company named Vitra, which is engaged in the manufacturing of products for home, office and public space. The company is mostly specialized in furniture manufacturing such as bathroom products and product materials which are known as raw materials. In the company workers decide in which order they will fill the cars with completed parts of the products by using their experiences and instincts. The completed cars which include the semi-finished products are directed to assembly. The goal of this project is making these processes automatized by using algorithms instead of the experiences of workers.

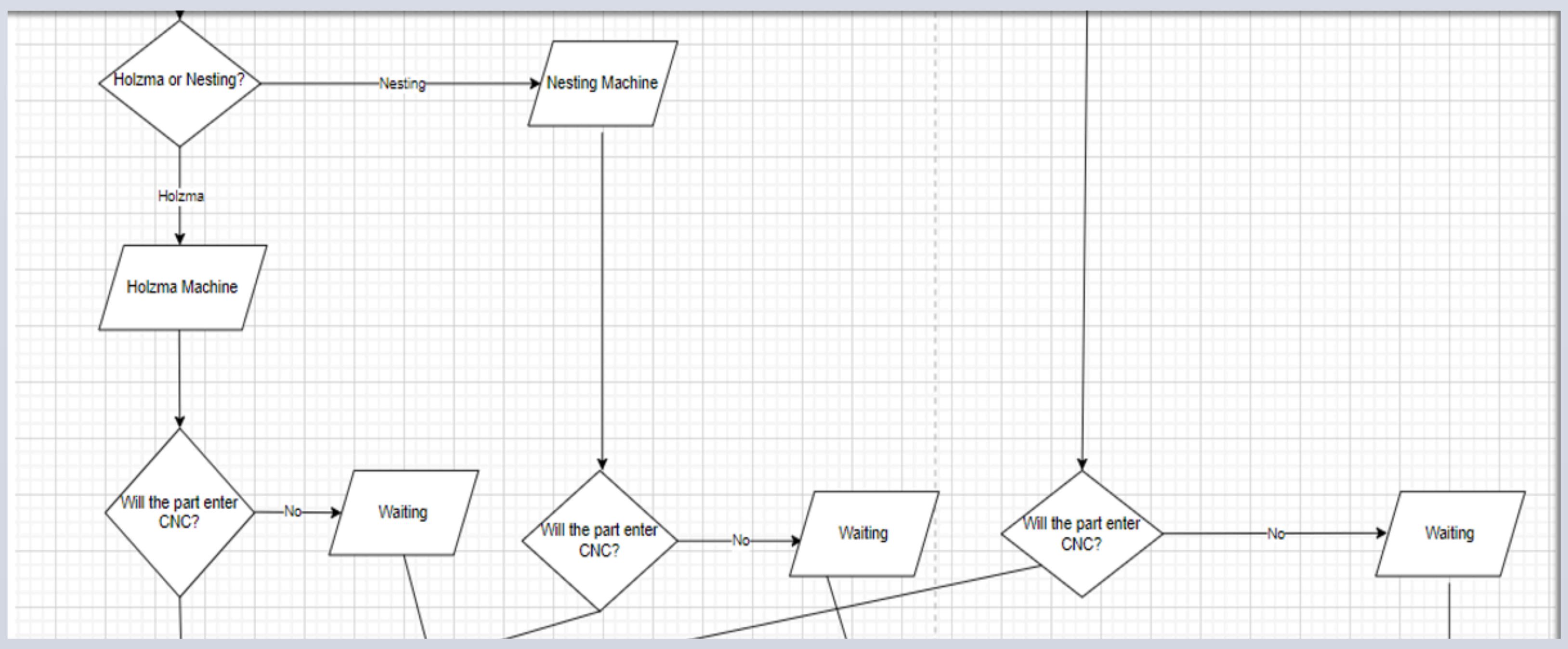
- The reason for trying to make the process automated is mostly to prevent the time-consuming activity of deciding the loading order of the cars and increasing the efficiency of the working processes
- Transition from the current system to the automatized system was conducted by developing a software-based algorithm on Python. While conducting the project, the first step was to analyze the data provided by the firm and understand the steps of the process. Consecutively, an algorithm which will automate the process was developed.

OBJECTIVES

In the project, the problem was addressed by several methodologies which were implemented on a Python algorithm. Due to the time constraint of shifts, the Knapsack algorithm was implemented in order to operate the machines for the maximum time in a certain shift interval.

- The Thermopress machine was the most used machine in terms of operation time. Therefore, the Knapsack algorithm was built on to optimize the Thermopress machine.
- Another implemented methodology was the parts that merely entered the Holzma and Nesting machines were thrown to the end of the row in the code. Our aim was to ensure that each machine works to the maximum, to prevent the parts that will enter the Nesting or Holzma machines taking the time of other products. Lastly, the parts which take the least amount of processing time are prioritized among other parts.
- The most important finding is the automated order of parts according to shifts implemented in Python code. Therefore, the desired result, which was an automated parts order system, was implemented in Python.

PROJECT DETAILS



According to data from the company, the shifts were determined by the machine that required the most processing time. The total usage time of the Holzma machine, which is the time required if all parts in that machine are to be produced, is calculated in the algorithm. Then, the total usage time of other machines which are CNC, Edge Banding, Nesting, Thermopress, Weeke200, and Biesse Brema are also calculated in the algorithm. The Thermopress was the most used machine in terms of working time process.Therefore, the methodology was implemented for optimizing the Thermopress machine. All values are converted to integers by rounding as it is difficult to operate with float values. Then, to test the values in Knapsack function, the values are converted to the regular list.

PROJECT DETAILS II

Urunkod	HOLZMA	NESTING	CNC	KENAR BANTLAMA	THERMOPRESS	Weeke200	Biesse
MVKLAETJ65K	3864	510	72	321,0003056	2816,47409	1890	
MVKLADAYN65R	536	170	0	168	1421,930054	810	
MVS2LV85	0	1275	300	1250,014814	2481,46059	0	
MVEKODAYNA60	0	1006,277988	217,4796979	0	0	1350	
MVYS5K60LVD	0	1360	320	5079,975802	15097,3811	14400	
MVQDRDAYNA80	10824	4488	1143,972922	0	0	0	
MVQDRLV80	0	340	79,99717153	0	0	0	
MVKLAETJ80C	4648	852,3554217	119	748,9982973	5271,701187	1575	
MVSENLV100C2	2060	1190	240	749,9834983	5408,007553	5400	
MVKLAETJ65C	4074	730,5903614	96	639,0077571	4435,168106	1890	
MVMIAB35	0	360	480	0	0	0	
MVKLADAYN100	3939	1768	117,0185185	1092,032004	11501,99241	7020	
MVS2LV85C	0	85	0	0	0	0	
MVCENLV60	0	168,2215738	39,99897119	23,026	206,88125	540	
MVFLWBD45	2002	476	79,94926792	1553,987573	2631,590982	2205	
MVCENB35	0	541,4245847	48	96	1808,291971	1080	
MVQDRLV100	0	2465	579,9934589	0	0	0	
MVSTPDDAYNA80	5000	4760	1240	880	4192,931727	0	
MVQDRLV65	0	1020	239,9985075	0	0	0	
MVKLAUSTDLP	97	68	0	42	526,5927177	90	
MVSENLV100	3502	1122	204	1785	10263,41633	11475	
MVYS5K50LV	0	136	32	482,0172911	1591,308528	1440	
MVSENBDSEP	656	396,0202788	63,3300435	895,9804413	3886,679369	5040	
MVFLWLV100C1	1644	816	200,884007	2538,037289	3681,155586	3780	

I. In the first step of the Knapsack algorithm, the thermopress was tested. Since the Thermopress is the most time-consuming machine (108629 seconds), it is the best one for trying the algorithm with that machine. Knapsack problem was implemented for each of the machines.

II. As the machines Holzma and Nesting are first-run machines, they are examined first. To keep all machines running as much as possible, the parts that need to go into only Holzma and Nesting are scheduled at the end of production.

III. The parts which are produced in Holzma and Nesting machines are sorted from smallest to largest by using the "sort_value( )" function according to their production times. The reason behind this was, when the parts were done with the Holzma and Nesting, then the parts entered the consecutive machines.

IV. The null values were converted to zero values because the summation of processing times of parts are calculated mathematically in some parts of the code.

CONSTRAINTS

- Constraint about machine order is considered to assign the parts to the machines in that order. As Holzma and Nesting have the same function in terms of taking the same roles in the cutting process, we paid attention to assigning the parts to either Holzma or Nesting and checked the Excel data file accordingly. The order of the machines are Holzma or Nesting, CNC, Edge banding, Thermopress, Biesse Brema and Weeke 200.
- Each machine works 3*400 minutes every day with 3 shifts.
- All machines have to run at maximum level.
- All parts of a product must be in the same shift.
- The outsourced plates must be considered while constructing the Python code.
- The developed algorithms should be suitable for all possible types of data to make the system sustainable.
  - In the project, there are no economic, environmental and health constraints.

CONCLUSIONS

- The tasks were successfully finalized according to processing constraints, the company supervisor's requests and main objective which was maximizing the machine's running. Company is going to revise the algorithm which we created in our project without using the algorithm directly. Moreover, Smart Production Flow Algorithm Project is a unique solution for special requests and constraints therefore the project is not a development of any previous method or project.
- The result that our algorithm gives is an excel table containing all the products that will be produced in a certain day, the machines needed for each product and their respective times in those machines.

REFERENCES

- *Activity selection problem: Greedy algo-1*. GeeksforGeeks. (2022, January 12). Retrieved

April6,2022,https://www.geeksforgeeks.org/activity-selection-problem-greedy-algo-1/?ref=lbp

- *Greedy algorithms introduction - javatpoint*. www.javatpoint.com. (n.d.). Retrieved April

6, 2022, from https://www.javatpoint.com/greedy-algorithms

- *Navone, E. C. (2022, February 21). Python code examples – sample script coding tutorial forbeginners.freeCodeCamp.org.Retrieved,May25,2022,from,https://www.freecodecamp.org/n ews/python-code-examples-sample-script-coding-tutorial-for-beginners/*
- *The python standard library*¶. The Python Standard Library - Python 3.10.4 documentation. (n.d.). Retrieved May 25, 2022, from https://docs.python.org/3/library/ https://app.diagrams.net/