M. Tech Production Engineering Rishabh Nirala 2020

SCHEDULING WITH COST OPTIMIZATION USING PYTHON PROGRAMMING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF REQUIREMENTS

FORTHE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

PRODUCTION ENGINEERING

Submitted by

RISHABH NIRALA

2K18/PIE/25

Under the supervision of

DR.A.K MADAN

(Professor)

DEPARTMENT OF MECHANICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road,Delhi-110042

JULY,2020

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana road,Delhi-11042

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana road,Delhi-11042

CANDIDATE'S DECLARATION

I RISHABH NIRALA, ROLLNo-2K18/PIE/25 student of M.Tech(Production Engineering), hereby declare that the project dissertation titled "SCHEDULING WITH COST OPTIMIZATION USING PYTHON PROGRAMMING" which is submitted by me to the Department of Mechanical Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of the Master of technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition

Place: Delhi **(RISHABH NIRALA)**

Date: 10/10/2020

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana road,Delhi-11042

CERTIFICATE

This is to certify that the major project report entitled, **"SCHEDULING WITH COST OPTIMIZATION USING PYTHON PROGRAMMING**" submitted by **RISHABH NIRALA (2K18/PIE/25)** in unfinished accomplishment of the necessity for the grant of **Master of Technology** degree in **Production Engineering** from **Delhi Technological University**, Delhi is an trustworthy work supported out by him beneath my supervision & guidance. The matter personified in the project description has not been reconcile to any other University/Institute for the award of any Degree or Diploma to the best of our knowledge and belief.

SIGNATURE

Dr. A. K. MADAN (Professor)

Department of Mechanical Engineering

Delhi Technological University, Delhi

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana road,Delhi-11042

ACKNOWLEDGEMENTS

I wish to express my profound gratitude and indebtedness to **Dr. A. K. MADAN (Professor),** Department of Mechanical Engineering, Delhi Technological University, Delhi for introducing the present topic and for their inspiring guidance, constructive criticism and valuable suggestion throughout the project work. Last but not least, my sincere thanks to all our friends who have patiently extended all sorts of help for accomplishing this undertaking.

Lastly my most sincere gratitude is extended to my family, especially my mother, father and brother who have given their utmost support throughout my life.

RISHABH NIRALA (2K18/PIE/25)

Department of Production Engineering,

Delhi Technological University, Delhi

ABSTRACT

This project deals with modern methods which can be used for planning and scheduling of manufacturing systems. Creating of production plans, using different software products is discussed. Advanced planning and scheduling methods are used for planning and scheduling. The outputs as Gantt charts are presented. Different optimization methods are applied for schedule optimization. Production plan, for every machine is created, after applying the optimization. Scheduling by using Lekin software and by using Python Programming is described. Cost of Scheduling is also optimize using programming model. In this project, Jupyter notebook is used as Python IDE and some packages of libraries are used.

KEYWORDS: Modern methods, production planning, Gantt chart, Lekin software and Python Programming.

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

Sequencing and scheduling is a type of basic leadership that assumes a critical job in manufacturing and service industries. In the current aggressive condition compelling sequencing and scheduling has become a need for endurance in the commercial center. Organizations need to meet delivery dates who have been focused on clients, as inability to do so may bring about huge loss of altruism. They additionally need to plan exercises so as to utilize the assets accessible in a proficient way.

The fundamental issues related with scheduling of FLEXIBLE MANUFACTURING SYSTEM are machine stacking, part steering, apparatus arranging and portion, material dealing with gadget task, and directing just as assignment timing issues. It is a basic leadership process with the objective of advancing at least one goals. The assets and assignments in an association can take numerous structures. The assignment can be worked in a generation procedure. Each assignment may have a specific need level, a soonest conceivable beginning time, and a due date. The targets likewise can take numerous structures. One target might be the minimization of culmination time of the last task, and another might be the minimization of the quantity of errands finished after their individual due dates [3].

Scheduling started to be paid attention to in assembling toward the start of twentieth century, with crafted by Henry Gantt and different pioneers. Be that as it may, it took numerous years for the main booking distributions to show up in modern building and activity examine writing. A portion of the main production showed up in Naval Research Logistics Quarterly in the mid 1950s and contained outcomes by S.M Johnson and J.R Jackson. During the 1960s a critical measure of works was finished by powerful programming and whole number programming definitions of scheduling issues. After Richard Karp's well known paper on Complexity Theory, the exploration in the 1970's centered mostly around the unpredictability chain of importance of scheduling issues. During the 1980s a few distinct bearings were sought after in the scholarly world furthermore, industry with the expansion measure of consideration paid to stochastic scheduling issues. Likewise, as close to home PCs began to saturate manufacturing facilities, scheduling frameworks were being created for the age of usable schedule in practice. This framework structure and advancement was, and is, being finished by computer scientists, operations researchers, industrial engineers[8].

Before the finish of 1970's and right off the bat in 80s researchers began utilizing Artificial Intelligent (AI) as a way to adapt to vulnerability thinking underway scheduling. Inside these years significant measure of exertion has been coordinated towards the portrayal and control of dubious data. For most recent two decades the issue of vulnerability is a significant thought during any decision-making process and scheduling is no special case . Inside the scheduling space there is a huge level of vulnerability both from natural vulnerabilities (machine breakdown or surge requests) and scheduling vulnerabilities (repercussions of which are exponential and in this manner too expensive to even consider evaluating) which considered by recent researchers.

In this paper brief examination has been done on recent literatures on production scheduling using artificial intelligence (AI) as a reaction to scheduling vulnerabilities. Three primary AI strategies among later writing have been spread in following segments. Therefore, futher segment exhibits the use of FUZZY TECHNIQUE, application of NEURAL NETWORK in production scheduling and GENETIC ALGORITHM in this paper[9].

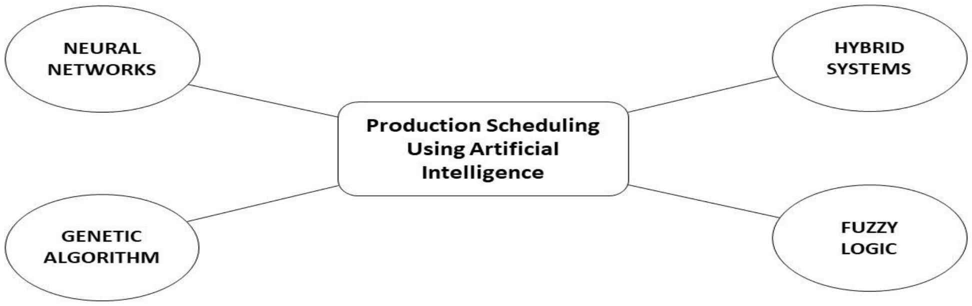


FIGURE : Segments of an Intelligent Production Scheduling

FIGURE 1[3]

2. TECHNIQUES USED IN PRODUCTION SCHEDULING

FUZZY LOGIC

The term FUZZY mean things which are not clear or dubious. In actuality, we may go over a circumstance where we can't choose whether the announcement is valid or bogus. Around then, FUZZY logic offers entirely important adaptability for thinking. We can likewise think about the vulnerabilities of any circumstance.

FUZZY logic calculation takes care of an issue in the wake of thinking about every single accessible datum. At that point it takes the most ideal choice for the given the information. The FUZZY logic strategy emulates the method for basic leadership in a human which think about every one of the potential outcomes between advanced qualities T and F[2].

In spite of the fact that, the idea of FUZZY logic had been examined since the 1920's. The term fluffy logic was first utilized with 1965 by Lotfi Zadeh a teacher of UC Berkeley in California. He saw that traditional PC logic was not fit for controlling information speaking to abstract or misty human thoughts.

Fluffy logic has been applied to different fields, from control hypothesis to AI. It was intended to enable the PC to decide the qualifications among information which is neither genuine nor bogus. Something like the procedure of human thinking. Like Little dull, Some brilliance, and so on [16].

some important characteristics of fuzzy logic:

- Flexible and easy to implement machine learning technique
- Helps you to mimic the logic of human thought
- Logic may have two values which represent two possible solutions
- Highly suitable method for uncertain or approximate reasoning
- Fuzzy logic views inference as a process of propagating elastic constraints

- Fuzzy logic allows you to build nonlinear functions of arbitrary complexity.
- Fuzzy logic should be built with the complete guidance of experts

when we should not use fuzzy logic :

- If you don't find it convenient to map an input space to an output space
- Fuzzy logic should not be used when you can use common sense
- Many controllers can do the fine job without the use of fuzzy logic

Fuzzy Logic Architecture

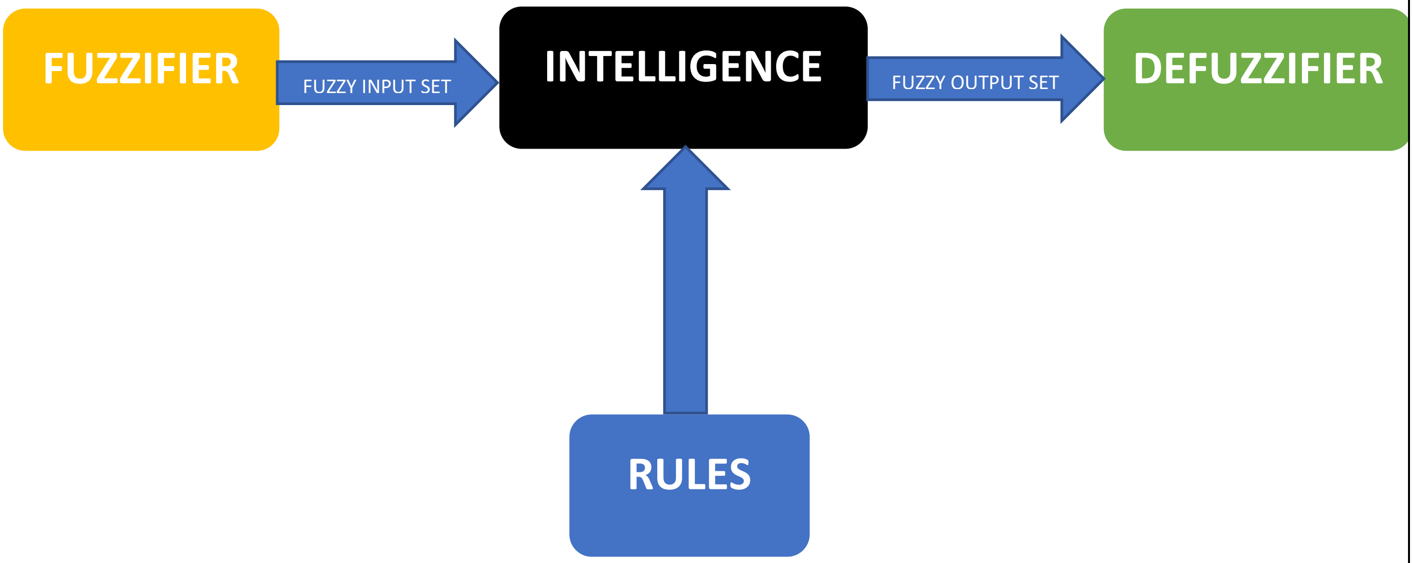


Figure 2. FUZZY LOGIC ARCHITECTURE[3]

Artificial Neural Networks

The designer of the first neurocomputer, Dr. Robert Hecht-Nielsen, characterizes a neural system as − "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs".

The possibility of ANNs depends on the conviction that working of human mind by causing the correct associations, to can be imitated utilizing silicon and wires as living neurons and dendrites[5].

The human mind is made out of 86 billion nerve cells called neurons. They are associated with other thousand cells by Axons. Boosts from outside condition or contributions from tactile organs are acknowledged by dendrites. These data sources make electric driving forces, which rapidly travel through the neural system. A neuron would then be able to send the message to other neuron to deal with the issue or doesn't send it forward.

ANNs are made out of different hubs, which emulate organic neurons of human mind. The neurons are associated by connections and they interface with one another. The hubs can take input information and perform basic activities on the information. The aftereffect of these activities is passed to different neurons. The yield at every hub is called its actuation or hub esteem[17].

Each connection is related with weight. ANNs are equipped for realizing, which happens by modifying weight esteems.

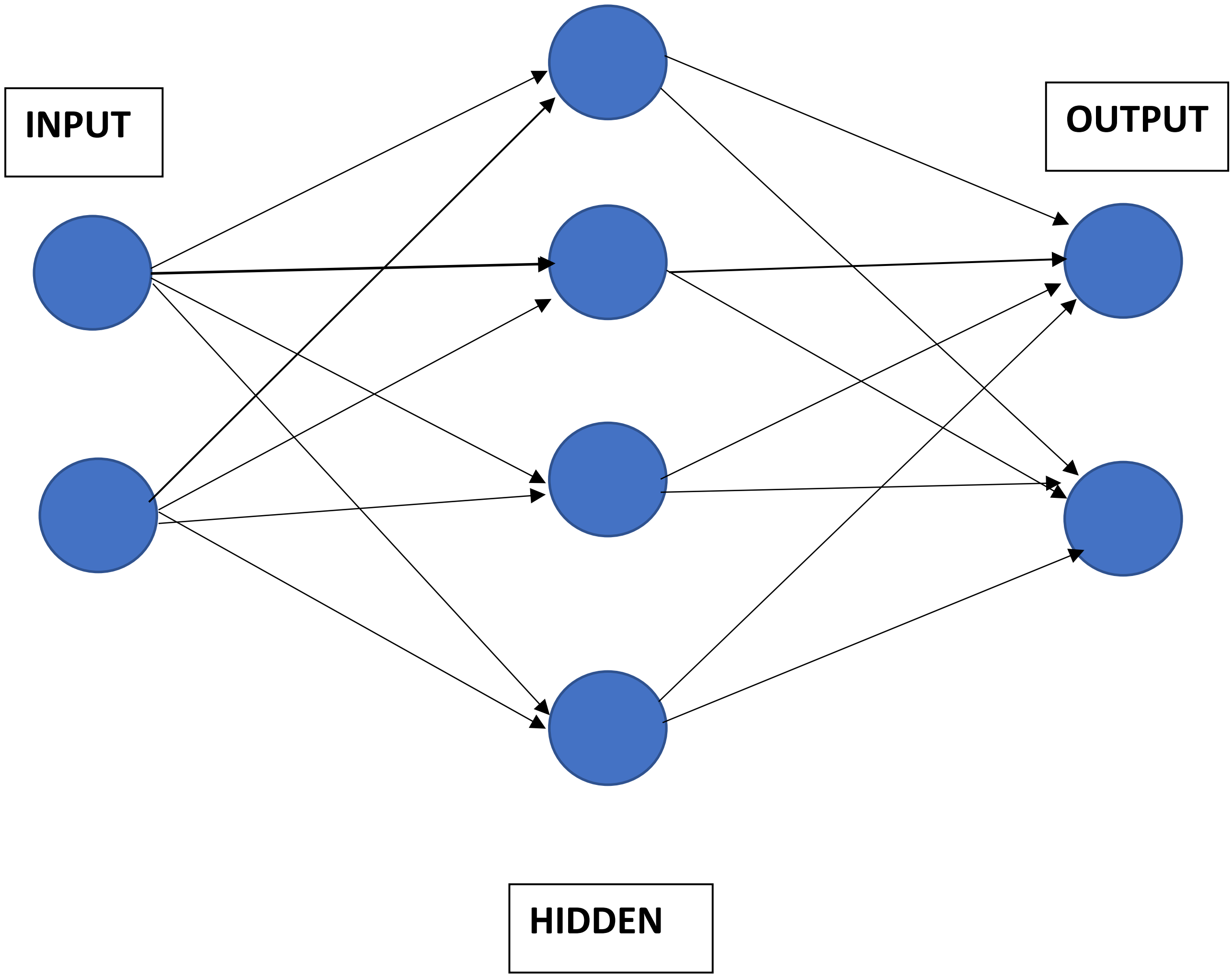


Figure 3. Simple Artificial Neural Network

There are two Artificial Neural Network topologies – **Feed Forward** and **Feed Back[6].**

Feed Forward:

The data stream is unidirectional. A unit sends data to other unit from which it doesn't get any data. There are no input circles. They are utilized in design age/acknowledgment/characterization. They have fixed sources of info and yields[3].

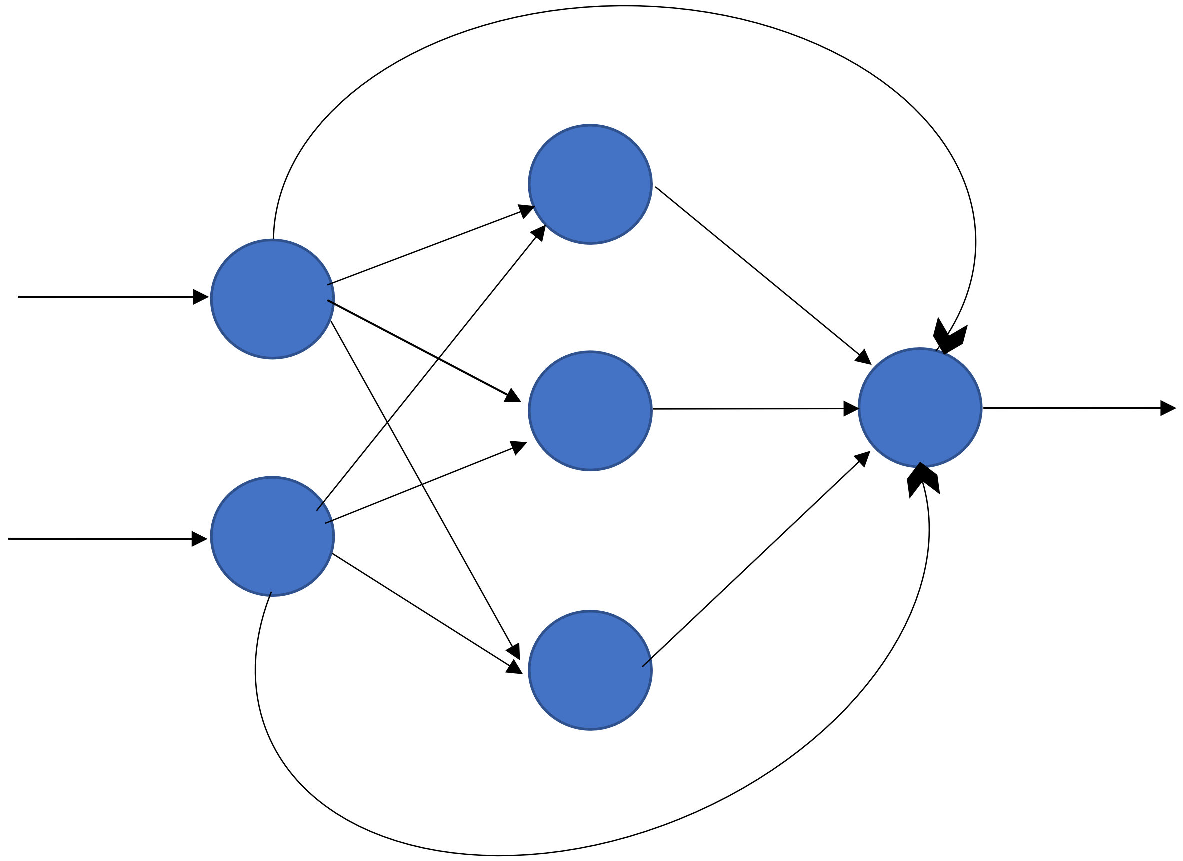
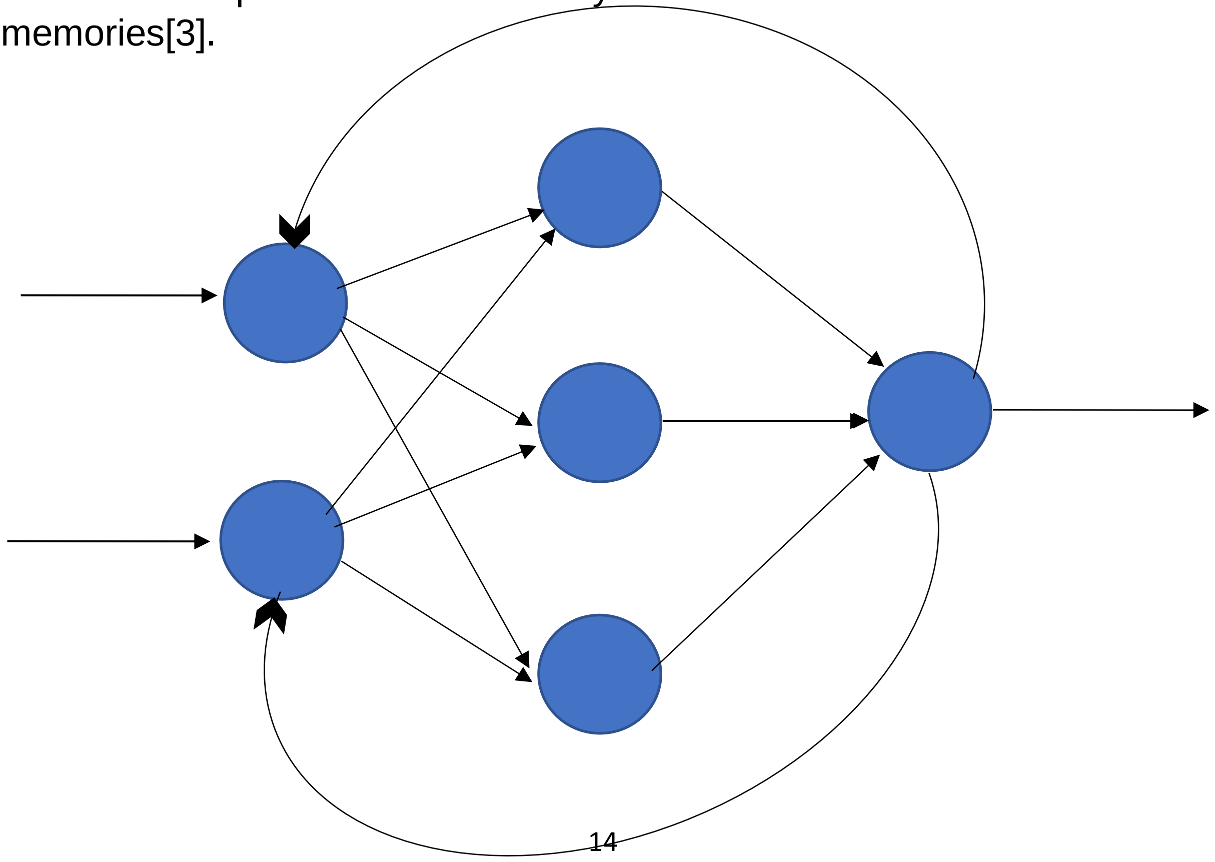


Figure 4. Feed Forward Simple Artificial Neural Network

Feed Back :

Feedback loops are allowed. They are used in content addressable memories[3].



Genetic Algorithms

Genetic Algorithms (GAs) are versatile heuristic search calculations that have a place with the bigger piece of developmental calculations. Hereditary calculations depend on the thoughts of regular determination and hereditary qualities. These are astute abuse of irregular pursuit gave verifiable information to coordinate the inquiry into the area of better execution in arrangement space. They are normally used to create excellent answers for enhancement issues and search issues[11].

Genetic Algorithm mimic the procedure of characteristic determination which implies those species who can adjust to changes in their condition can endure and recreate and go to people to come. In basic words, they stimulate "natural selection" among individual of back to back age for taking care of an issue. Every age comprise of a populace of people and every individual speaks to a point in search space and conceivable arrangement. Every individual is spoken to as a string of character/integer/float/bits. This string is undifferentiated from the Chromosome[11].

Basic Steps

The process of using genetic algorithms goes like this:

1. Creating an Initial population
2. Defining a Fitness function
3. Selecting the parents
4. Making a Crossover
5. Mutation

When to Use Genetic Algorithms

GAs are bad for a wide range of issues. They're best for issues where there is a reasonable method to assess wellness. In the event that your inquiry space isn't very much compelled or your assessment procedure is computationally costly, GAs may not discover arrangements in a normal measure of time. As far as I can tell, they're most useful when there is a better than average calculation set up, however the "handles" simply should be changed[14].

When does this process stop?

Our populace has a fixed size. As new components are shaped, old components with low wellness score are evacuated. At the point when the populace has merged, i.e., no new components are repeated which are fundamentally not the same as the past populace, at that point we may state that the hereditary calculation has given a lot of answers for our concern[19].

Pseudocode

```
• START
  — create the initial population

• Compute fitness

• REPEAT
  — Selection
  — Crossover
  — Mutation
  — Compute fitness

• UNTIL population has converged

• STOP
```

Production scheduling aims to maximize the efficiency of the operation and reduce costs.

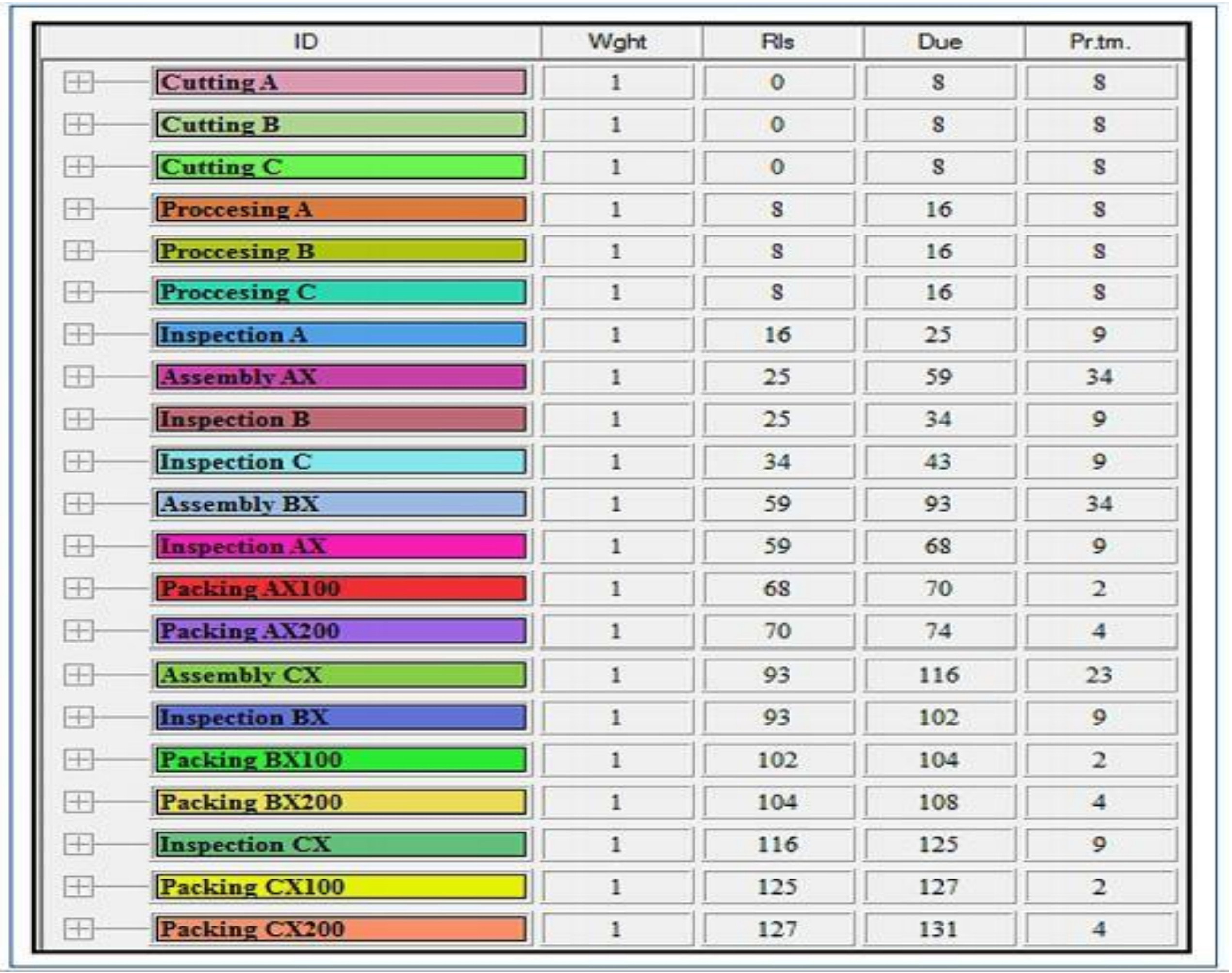
Manufacturing model configurations (models) are presented using different software products. Models include all processes from receiving of raw materials, manufacturing to purchasing. The purchases raw materials are 'Steel-M1' and 'Gear-X' and the produces products are AX100 AX200 BX100 BX200 CX100 and CX200. They are produced through cutting, additional processing, assembly, and in the end they are packed and sold[22].

The flow of processes for producing each product is shown below:

- Steel-M1 – Cutting – Processing – Inspection1 – Shaft-A;
- Steel-M1 – Cutting – Processing – Inspection1 – Shaft-B;
- Steel-M1 – Cutting – Processing – Inspection1 – Shaft-C;
- A, Gear-X – Assembly – Inspection3 – AX;
- B, Gear-X – Assembly – Inspection3 – BX;
- C, Gear-X – Assembly – Inspection3 – CX;
- AX – Packing – AX100;
- AX – Packing – AX200;
- BX – Packing – BX100;
- BX – Packing – BX200;
- CX – Packing – CX100;
- CX – Packing – CX200;

Production scheduling using 'Lekin':

LEKIN® [Pinedo 2012] is a scheduling system developed at the Stern School of Business, NYU. LEKIN® was designed as an educational tool with the main purpose of introducing the students to scheduling theory and its applications. Besides that, the system's extensibility allows (and encourages) to use it in algorithm development. The project has been directed by Professor Michael L. Pinedo and Associate Professor Xiuli Chao[22]. The framework defines the type of scheduling problem that is to be entered. For scheduling the simulation model of the order is picked a "Flexible Job Shop" environment. Once the framework has been selected, the number of the machines and jobs is to be specified. In the current project 6 workcenters are needed. Different jobs are added to the workcenters and their machines. After adding the jobs other settings like processing time, release date, repeatability of the work and the job route, in the machine environment, are also specified. Full list of every jobs and the total amount of time needed for manufacturing is shown in the Fig.1. The total production time is 131 hours. Once the relevant data of the problem has been entered, the scheduling process is started. The ‘Early Due Date(EDD)’ dispatching rule is set in order to generate a schedule. The generated gantt chart of the jobs is show in Fig.2[22].



ID	Wght	Rls	Due	Pr.tm.
Cutting A	1	0	8	8
Cutting B	1	0	8	8
Cutting C	1	0	8	8
Proccesing A	1	8	16	8
Proccesing B	1	8	16	8
Proccesing C	1	8	16	8
Inspection A	1	16	25	9
Assembly AX	1	25	59	34
Inspection B	1	25	34	9
Inspection C	1	34	43	9
Assembly BX	1	59	93	34
Inspection AX	1	59	68	9
Packing AX100	1	68	70	2
Packing AX200	1	70	74	4
Assembly CX	1	93	116	23
Inspection BX	1	93	102	9
Packing BX100	1	102	104	2
Packing BX200	1	104	108	4
Inspection CX	1	116	125	9
Packing CX100	1	125	127	2
Packing CX200	1	127	131	4

Figure 5. Full list of the jobs for producing the simulation model in ‘Lekin’

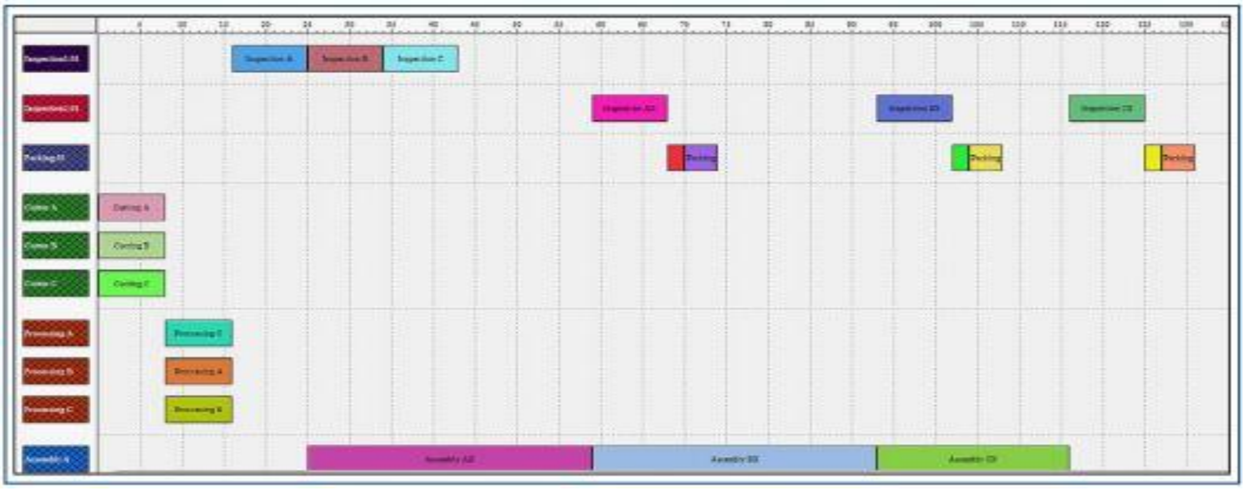
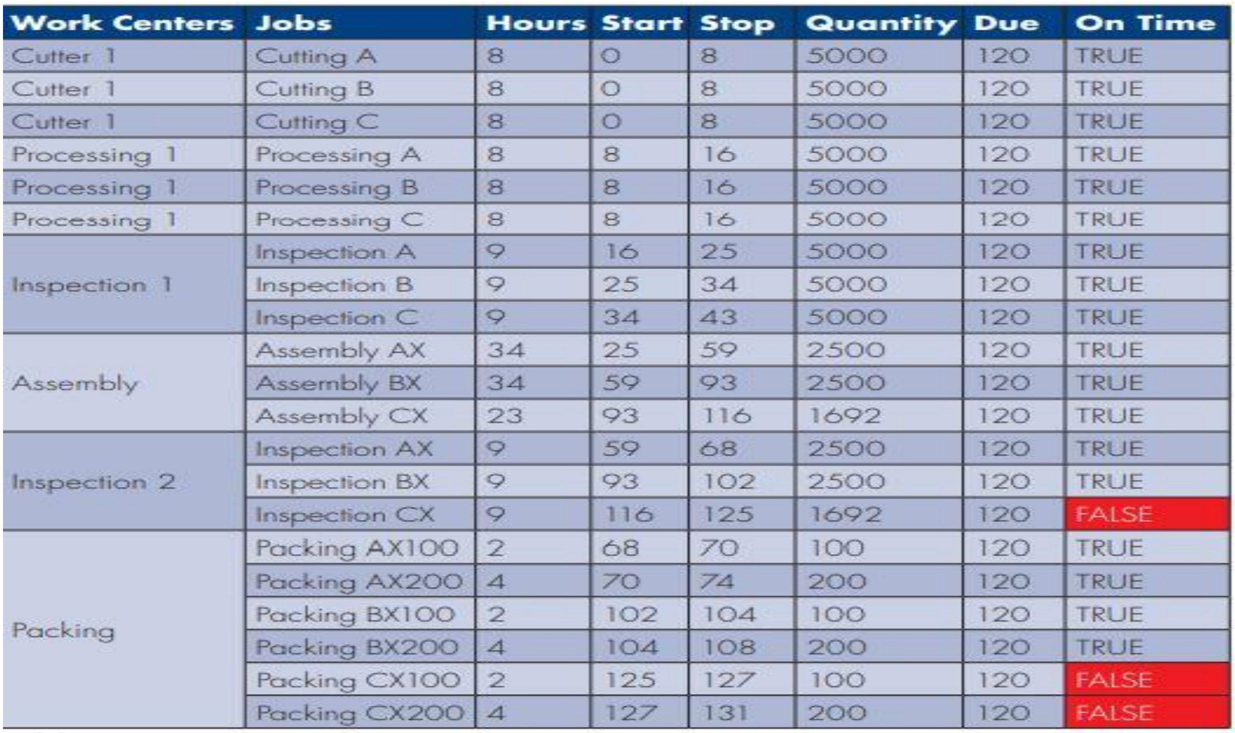


Figure 6. Gantt chart of the jobs in the environment of ‘Lekin’

The schedule can be examined in the ‘Sequence Window’, where several relevant data of the schedule are shown. Once one or more schedules are generated their performance are specified including the scheduling criteria and the running time to obtain different sequences.

Production planning and scheduling using ‘Spreadsheets’:

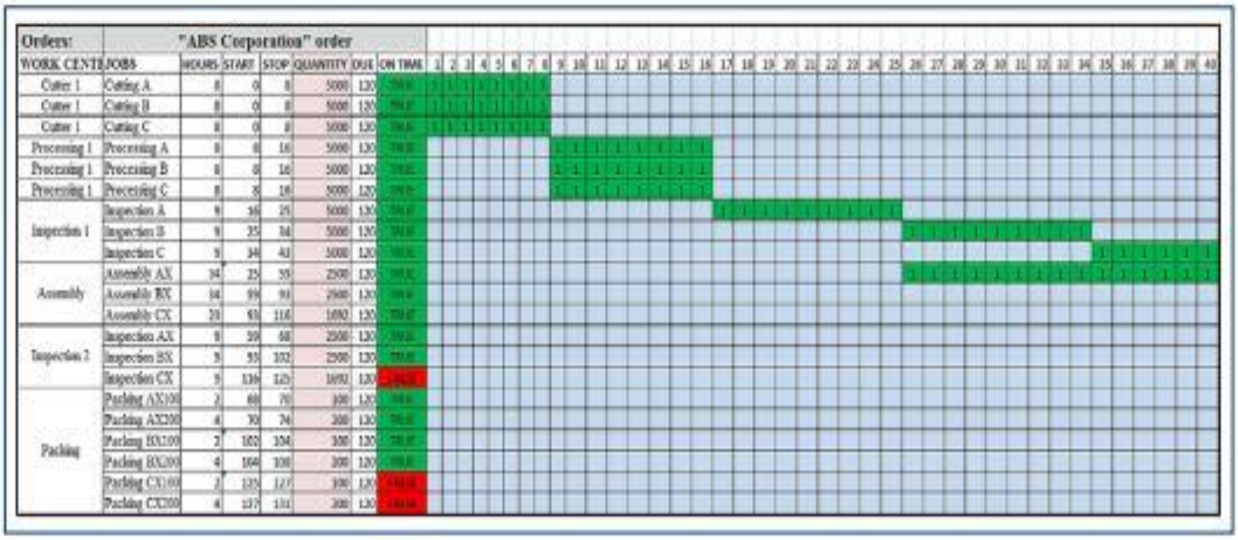
Scheduling production order using spreadsheets can be done by using the software product ‘Excel’ or similar. Spreadsheets have earned themselves a bad reputation amongst software purists, because they can, and often are, used in an unstructured way. Building a scheduling system requires a structured and disciplined approach. The approach used here is to create lists in the form of databases, with a heading at the top of each column, and with universal formulas that can be copied and pasted down a column, and work on every row. If all the calculations are done in a structured database, then reports, with sub-totals and charts, can easily be created with a ‘PivotTable’ [Rice 2014]. The beginning of the developing the spreadsheets should start, with creating the columns for the workcenters or machines and the jobs that are going to be done in every center[22]. Knowing the time set for the jobs to be done and using ‘Excel’ a true and false column is created, showing which job will be done on time and which will not. The final date added to the spreadsheets is shown in ‘tab. 1’.



Work Centers	Jobs	Hours	Start	Stop	Quantity	Due	On Time
Cutter 1	Cutting A	8	0	8	5000	120	TRUE
Cutter 1	Cutting B	8	0	8	5000	120	TRUE
Cutter 1	Cutting C	8	0	8	5000	120	TRUE
Processing 1	Processing A	8	8	16	5000	120	TRUE
Processing 1	Processing B	8	8	16	5000	120	TRUE
Processing 1	Processing C	8	8	16	5000	120	TRUE
Inspection 1	Inspection A	9	16	25	5000	120	TRUE
	Inspection B	9	25	34	5000	120	TRUE
	Inspection C	9	34	43	5000	120	TRUE
Assembly	Assembly AX	34	25	59	2500	120	TRUE
	Assembly BX	34	59	93	2500	120	TRUE
	Assembly CX	23	93	116	1692	120	TRUE
Inspection 2	Inspection AX	9	59	68	2500	120	TRUE
	Inspection BX	9	93	102	2500	120	TRUE
	Inspection CX	9	116	125	1692	120	FALSE
Packing	Packing AX100	2	68	70	100	120	TRUE
	Packing AX200	4	70	74	200	120	TRUE
	Packing BX100	2	102	104	100	120	TRUE
	Packing BX200	4	104	108	200	120	TRUE
	Packing CX100	2	125	127	100	120	FALSE
	Packing CX200	4	127	131	200	120	FALSE

Figure 7. Applied date for simulating the production model using spreadsheets

After calculating which job will be done on time, it is necessary to generate a Gantt chart in order to visualize the result. “Excel” spreadsheets can be used to generate such chart. Part of the generated Gantt chart is show in ‘fig.3’.



Orders:	"ABS Corporation" order						
WORK CENTE	JOBS	HOURS	START	STOP	QUANTITY	DUE	ON TIME
Cutter 1	Cutting A	8	0	8	5000	120	TRUE
Cutter 1	Cutting B	8	0	8	5000	120	TRUE
Cutter 1	Cutting C	8	0	8	5000	120	TRUE
Processing 1	Processing A	8	8	16	5000	120	TRUE
Processing 1	Processing B	8	8	16	5000	120	TRUE
Processing 1	Processing C	8	8	16	5000	120	TRUE
Inspection 1	Inspection A	9	16	25	5000	120	TRUE
	Inspection B	9	25	34	5000	120	TRUE
	Inspection C	9	34	43	5000	120	TRUE
Assembly	Assembly AX	34	25	59	2500	120	TRUE
	Assembly BX	34	59	93	2500	120	TRUE
	Assembly CX	23	93	116	1690	120	TRUE
Inspection 2	Inspection AX	9	59	68	2500	120	TRUE
	Inspection BX	9	93	102	2500	120	TRUE
	Inspection CX	9	116	125	1690	120	FALSE
Packing	Packing AX100	2	68	70	100	120	TRUE
	Packing AX200	4	70	74	200	120	TRUE
	Packing BX100	2	102	104	100	120	TRUE
	Packing BX200	4	104	108	200	120	TRUE
	Packing CX100	2	125	127	100	120	FALSE
	Packing CX200	4	127	131	200	120	FALSE

Figure 8. Generating a Gantt chart using “Excel” spreadsheets

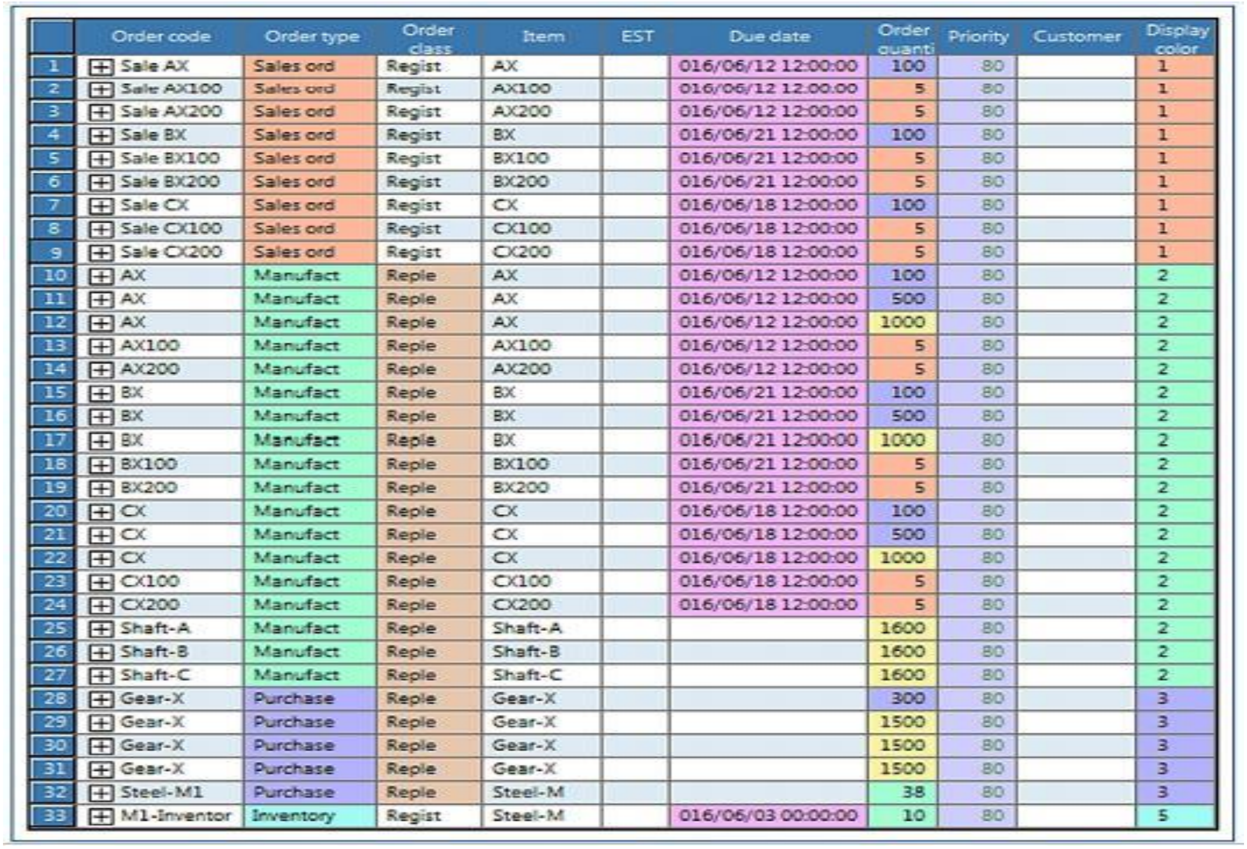
Every square is equal to one hour. The jobs that are going to be late are presented in different color. After visualizing the result, the manufacturer can easily decide what changes needs to be made[22].

Production planning and scheduling using ‘Asprova’:

At the beginning of planning and scheduling the production with ‘Asprova®’ (Asprova Corporation) the manufacturer have to pick a type of production module. The software product’ Asprova’ is divided into different modules. The objectives vary by each module. The module that is going to be used to schedule the production is ‘Advance Planning and Scheduling (APS)’. Data that Asprova requires for scheduling can be input directly, imported from an external system or text files, or copied and pasted from ‘Microsoft Excel’. The ‘Integrated Master Editor’ is a table for maintaining master data. It is especially convenient for creating and editing master data because it combines master settings and related information in a single table. The ‘Asprova APS’ master data is created using the ‘Integrated Master Editor’. Basically inputting data into the ‘Integrate Master’ is like inputting data into the ‘Excel’ spreadsheets (fig.4). Asprova has the possibility to create and maintain production orders. The full list of the orders is shown in ‘fig.4’ In the “Order” table manufacturer adds settings about every information regarding the production of the items, such are sales or resource purchases, due dates, quantites, ect. Using the order table provided in

‘Asprova’, scheduling and managing of the productions is a lot more easy and clear.

After all the needed data is entered the finale result can be displayed in different Gantt charts. On ‘fig.5’ is presented the ‘Resource Gantt Chart’ showing all the needed information about the flow of loads and which load is not going to be done on time[22].



	Order code	Order type	Order class	Item	EST	Due date	Order quanti	Priority	Customer	Display color
1	⊞ Sale AX	Sales ord	Regist	AX		016/06/12 12:00:00	100	80		1
2	⊞ Sale AX100	Sales ord	Regist	AX100		016/06/12 12:00:00	5	80		1
3	⊞ Sale AX200	Sales ord	Regist	AX200		016/06/12 12:00:00	5	80		1
4	⊞ Sale BX	Sales ord	Regist	BX		016/06/21 12:00:00	100	80		1
5	⊞ Sale BX100	Sales ord	Regist	BX100		016/06/21 12:00:00	5	80		1
6	⊞ Sale BX200	Sales ord	Regist	BX200		016/06/21 12:00:00	5	80		1
7	⊞ Sale CX	Sales ord	Regist	CX		016/06/18 12:00:00	100	80		1
8	⊞ Sale CX100	Sales ord	Regist	CX100		016/06/18 12:00:00	5	80		1
9	⊞ Sale CX200	Sales ord	Regist	CX200		016/06/18 12:00:00	5	80		1
10	⊞ AX	Manufact	Reple	AX		016/06/12 12:00:00	100	80		2
11	⊞ AX	Manufact	Reple	AX		016/06/12 12:00:00	500	80		2
12	⊞ AX	Manufact	Reple	AX		016/06/12 12:00:00	1000	80		2
13	⊞ AX100	Manufact	Reple	AX100		016/06/12 12:00:00	5	80		2
14	⊞ AX200	Manufact	Reple	AX200		016/06/12 12:00:00	5	80		2
15	⊞ BX	Manufact	Reple	BX		016/06/21 12:00:00	100	80		2
16	⊞ BX	Manufact	Reple	BX		016/06/21 12:00:00	500	80		2
17	⊞ BX	Manufact	Reple	BX		016/06/21 12:00:00	1000	80		2
18	⊞ BX100	Manufact	Reple	BX100		016/06/21 12:00:00	5	80		2
19	⊞ BX200	Manufact	Reple	BX200		016/06/21 12:00:00	5	80		2
20	⊞ CX	Manufact	Reple	CX		016/06/18 12:00:00	100	80		2
21	⊞ CX	Manufact	Reple	CX		016/06/18 12:00:00	500	80		2
22	⊞ CX	Manufact	Reple	CX		016/06/18 12:00:00	1000	80		2
23	⊞ CX100	Manufact	Reple	CX100		016/06/18 12:00:00	5	80		2
24	⊞ CX200	Manufact	Reple	CX200		016/06/18 12:00:00	5	80		2
25	⊞ Shaft-A	Manufact	Reple	Shaft-A			1600	80		2
26	⊞ Shaft-B	Manufact	Reple	Shaft-B			1600	80		2
27	⊞ Shaft-C	Manufact	Reple	Shaft-C			1600	80		2
28	⊞ Gear-X	Purchase	Reple	Gear-X			300	80		3
29	⊞ Gear-X	Purchase	Reple	Gear-X			1500	80		3
30	⊞ Gear-X	Purchase	Reple	Gear-X			1500	80		3
31	⊞ Gear-X	Purchase	Reple	Gear-X			1500	80		3
32	⊞ Steel-M1	Purchase	Reple	Steel-M			38	80		3
33	⊞ M1-Inventor	Inventory	Regist	Steel-M		016/06/03 00:00:00	10	80		5

Figure 9. Order table

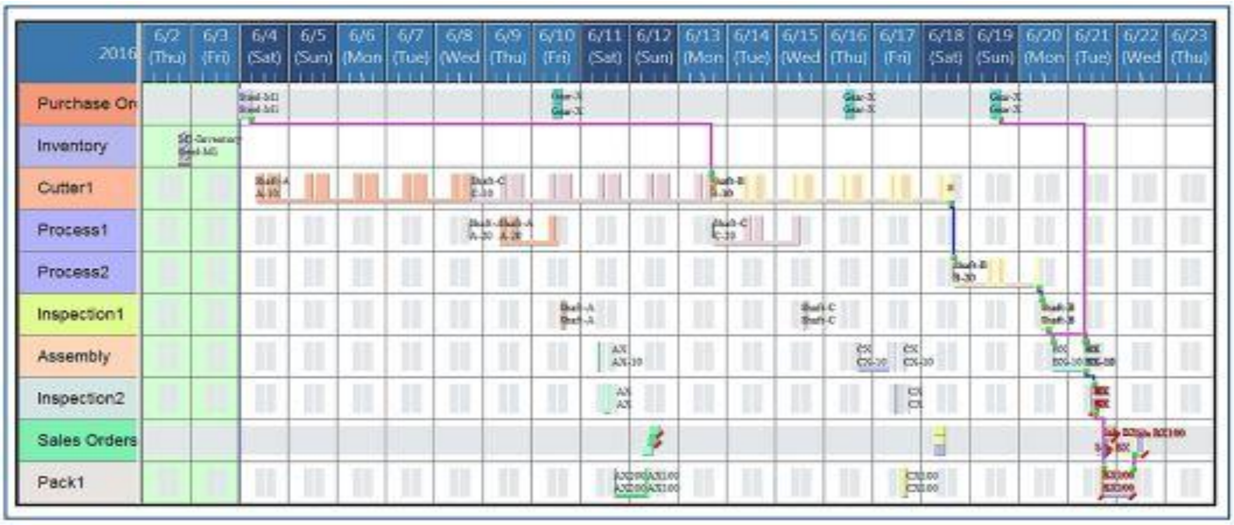
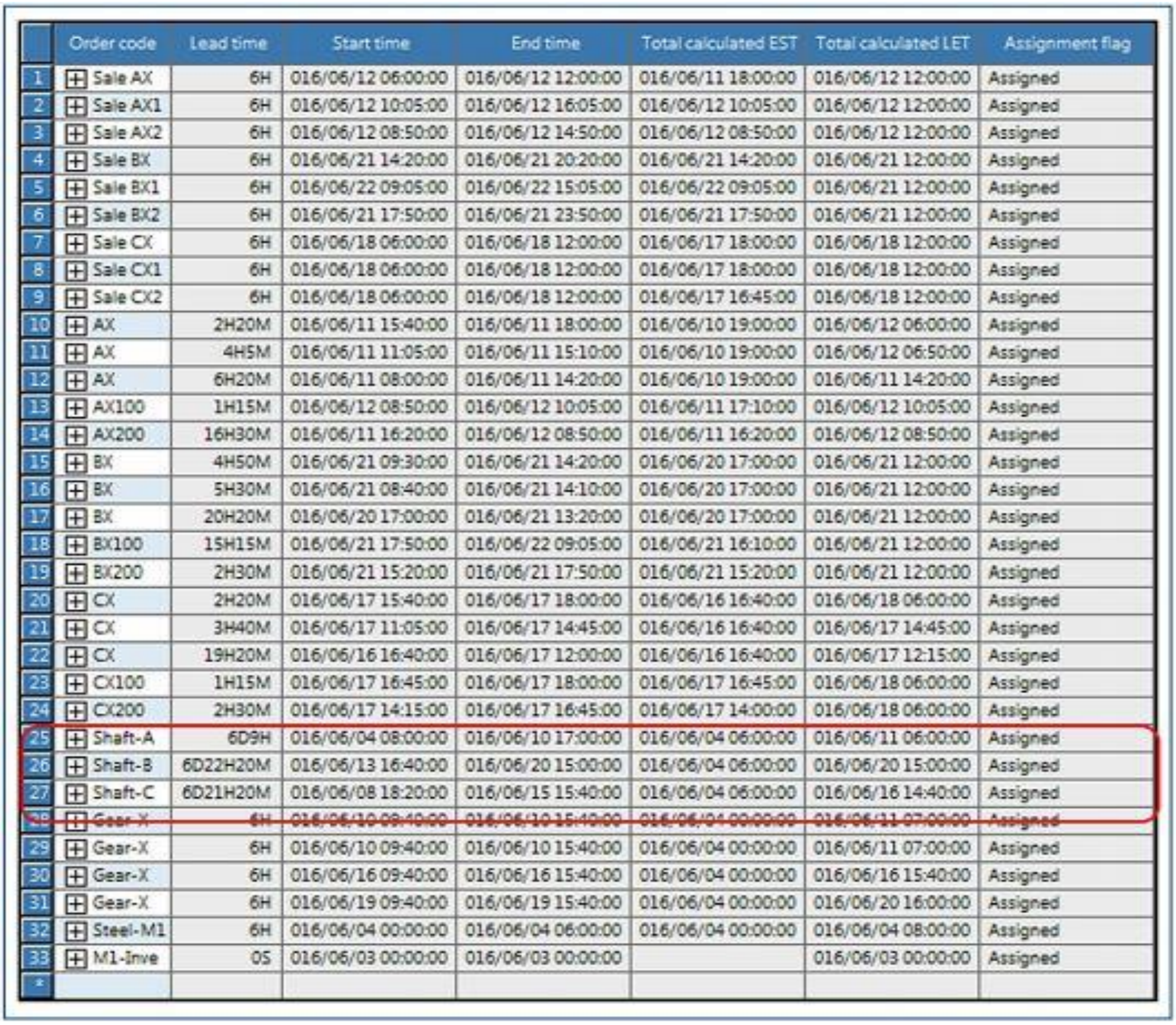


Figure 10. ‘Resource Gantt Chart’ of the production mode

ANALYZING OUTPUTS FROM THE ASPROVA SIMULATION MODEL OF PRODUCTION SYSTEM:

After setting which of the items is going to be for sale, in the ‘Order table’ of the program, Asprova automatically generates manufacturing and purchasing orders. Asprova assigns end times for the manufacturing orders also. The program generates an assignment list for the orders. On ‘fig.6’ is shown the full list of the orders of the simulation model. In such way the manufacturer can check the starting time and the finishing time for every order. On the assignment list is also shown the time needed, with these manufacturing settings, for every order to be done. As shown on the figure, the longest job is for manufacturing ‘Shaft-B’ and it requires 6 days 22 hours and 20 minutes. The last order, that is going to be executed, is for the order „Sale BX1“ and it is going to be done on 2016/06/22 at 15:05:00 o’clock[22].



	Order code	Lead time	Start time	End time	Total calculated EST	Total calculated LET	Assignment flag
1	⊞ Sale AX	6H	016/06/12 06:00:00	016/06/12 12:00:00	016/06/11 18:00:00	016/06/12 12:00:00	Assigned
2	⊞ Sale AX1	6H	016/06/12 10:05:00	016/06/12 16:05:00	016/06/12 10:05:00	016/06/12 12:00:00	Assigned
3	⊞ Sale AX2	6H	016/06/12 08:50:00	016/06/12 14:50:00	016/06/12 08:50:00	016/06/12 12:00:00	Assigned
4	⊞ Sale BX	6H	016/06/21 14:20:00	016/06/21 20:20:00	016/06/21 14:20:00	016/06/21 12:00:00	Assigned
5	⊞ Sale BX1	6H	016/06/22 09:05:00	016/06/22 15:05:00	016/06/22 09:05:00	016/06/21 12:00:00	Assigned
6	⊞ Sale BX2	6H	016/06/21 17:50:00	016/06/21 23:50:00	016/06/21 17:50:00	016/06/21 12:00:00	Assigned
7	⊞ Sale CX	6H	016/06/18 06:00:00	016/06/18 12:00:00	016/06/17 18:00:00	016/06/18 12:00:00	Assigned
8	⊞ Sale CX1	6H	016/06/18 06:00:00	016/06/18 12:00:00	016/06/17 18:00:00	016/06/18 12:00:00	Assigned
9	⊞ Sale CX2	6H	016/06/18 06:00:00	016/06/18 12:00:00	016/06/17 16:45:00	016/06/18 12:00:00	Assigned
10	⊞ AX	2H20M	016/06/11 15:40:00	016/06/11 18:00:00	016/06/10 19:00:00	016/06/12 06:00:00	Assigned
11	⊞ AX	4H5M	016/06/11 11:05:00	016/06/11 15:10:00	016/06/10 19:00:00	016/06/12 06:50:00	Assigned
12	⊞ AX	6H20M	016/06/11 08:00:00	016/06/11 14:20:00	016/06/10 19:00:00	016/06/11 14:20:00	Assigned
13	⊞ AX100	1H15M	016/06/12 08:50:00	016/06/12 10:05:00	016/06/11 17:10:00	016/06/12 10:05:00	Assigned
14	⊞ AX200	16H30M	016/06/11 16:20:00	016/06/12 08:50:00	016/06/11 16:20:00	016/06/12 08:50:00	Assigned
15	⊞ BX	4H50M	016/06/21 09:30:00	016/06/21 14:20:00	016/06/20 17:00:00	016/06/21 12:00:00	Assigned
16	⊞ BX	5H30M	016/06/21 08:40:00	016/06/21 14:10:00	016/06/20 17:00:00	016/06/21 12:00:00	Assigned
17	⊞ BX	20H20M	016/06/20 17:00:00	016/06/21 13:20:00	016/06/20 17:00:00	016/06/21 12:00:00	Assigned
18	⊞ BX100	15H15M	016/06/21 17:50:00	016/06/22 09:05:00	016/06/21 16:10:00	016/06/21 12:00:00	Assigned
19	⊞ BX200	2H30M	016/06/21 15:20:00	016/06/21 17:50:00	016/06/21 15:20:00	016/06/21 12:00:00	Assigned
20	⊞ CX	2H20M	016/06/17 15:40:00	016/06/17 18:00:00	016/06/16 16:40:00	016/06/18 06:00:00	Assigned
21	⊞ CX	3H40M	016/06/17 11:05:00	016/06/17 14:45:00	016/06/16 16:40:00	016/06/17 14:45:00	Assigned
22	⊞ CX	19H20M	016/06/16 16:40:00	016/06/17 12:00:00	016/06/16 16:40:00	016/06/17 12:15:00	Assigned
23	⊞ CX100	1H15M	016/06/17 16:45:00	016/06/17 18:00:00	016/06/17 16:45:00	016/06/18 06:00:00	Assigned
24	⊞ CX200	2H30M	016/06/17 14:15:00	016/06/17 16:45:00	016/06/17 14:00:00	016/06/18 06:00:00	Assigned
25	⊞ Shaft-A	6D9H	016/06/04 08:00:00	016/06/10 17:00:00	016/06/04 06:00:00	016/06/11 06:00:00	Assigned
26	⊞ Shaft-B	6D22H20M	016/06/13 16:40:00	016/06/20 15:00:00	016/06/04 06:00:00	016/06/20 15:00:00	Assigned
27	⊞ Shaft-C	6D21H20M	016/06/08 18:20:00	016/06/15 15:40:00	016/06/04 06:00:00	016/06/16 14:40:00	Assigned
28	⊞ Gear-X	6H	016/06/10 09:40:00	016/06/10 15:40:00	016/06/04 00:00:00	016/06/11 07:00:00	Assigned
29	⊞ Gear-X	6H	016/06/10 09:40:00	016/06/10 15:40:00	016/06/04 00:00:00	016/06/11 07:00:00	Assigned
30	⊞ Gear-X	6H	016/06/16 09:40:00	016/06/16 15:40:00	016/06/04 00:00:00	016/06/16 15:40:00	Assigned
31	⊞ Gear-X	6H	016/06/19 09:40:00	016/06/19 15:40:00	016/06/04 00:00:00	016/06/20 16:00:00	Assigned
32	⊞ Steel-M1	6H	016/06/04 00:00:00	016/06/04 06:00:00	016/06/04 00:00:00	016/06/04 08:00:00	Assigned
33	⊞ M1-Inve	0S	016/06/03 00:00:00	016/06/03 00:00:00		016/06/03 00:00:00	Assigned
*							

Figure 11. The assignment list with the time needed for the orders to be done

Either the engineer or the manufacturer, using Asprova, is able to monitor the status, of every order, at any given time. Using the generated Gantt chart (fig.5), production managers receive visualized manufacturing model. That way, the final results are easily understood. The final result for selling the items of 'AX' series is shown in 'fig.7'. As shown on the figure the orders that are not going to be done on the time are shown with color red. In this case the selling of items 'AX100' and 'AX200' is going to be delayed. The result does not satisfy the demand[22].

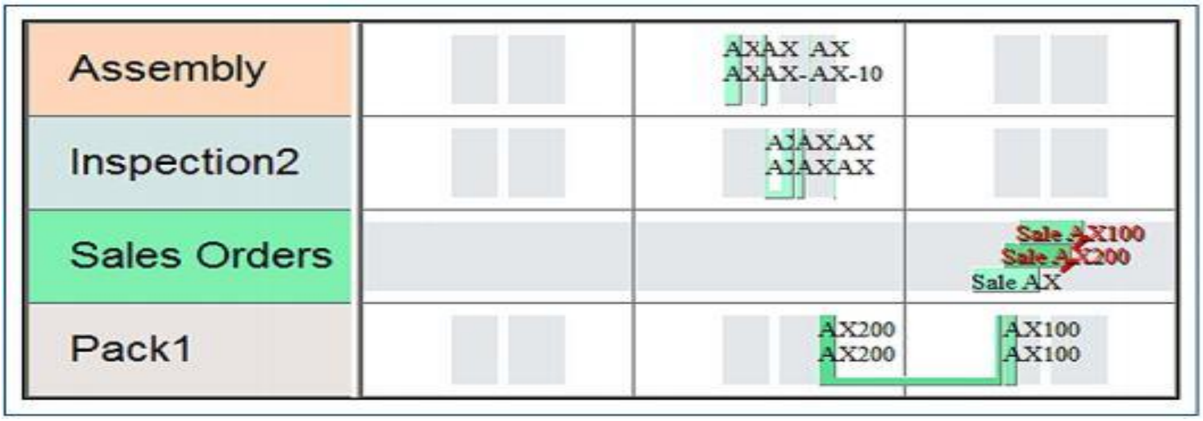


Figure 12. Gantt chart for selling the constructions with Shaft-A

The final result for selling the items of 'CX' series is shown in 'fig.8'. In this case the orders are going to be done on time. The result shows that the production schedule satisfies the demand.

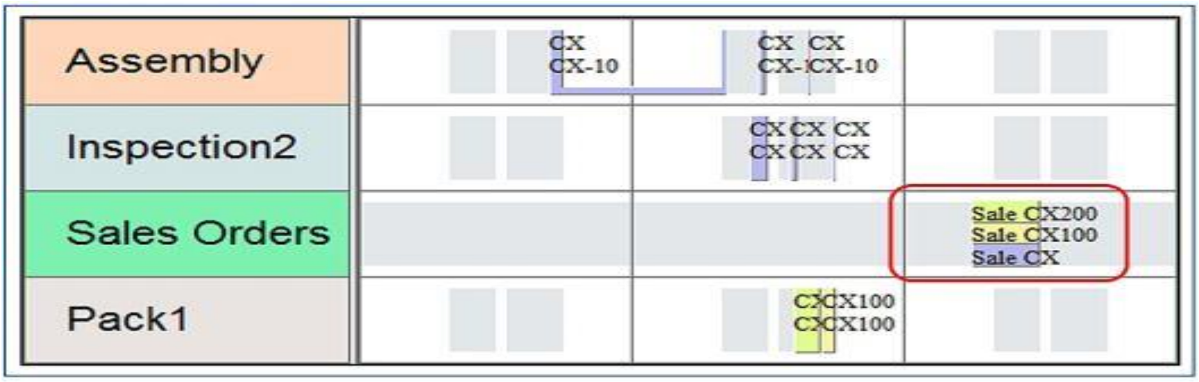
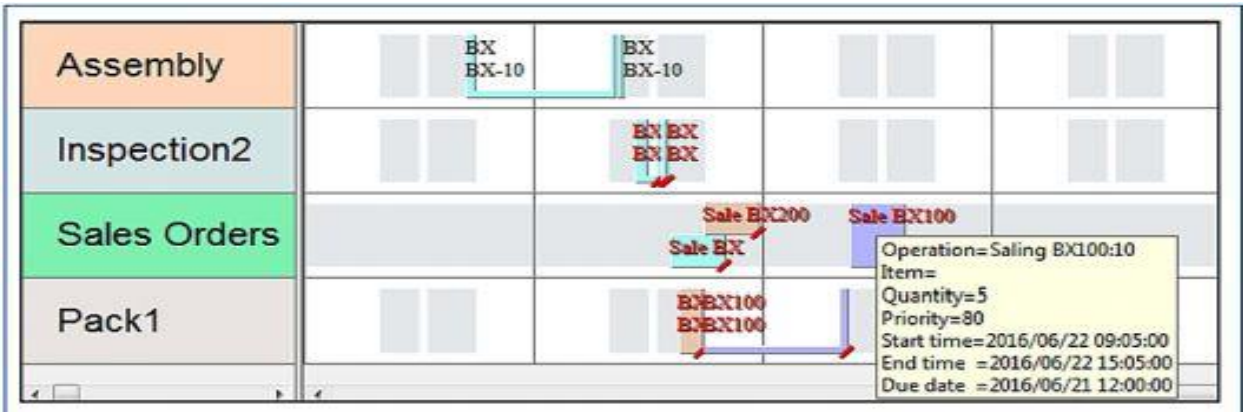


Figure 13. Gantt chart for selling the constructions with Shaft-C

The final result for selling the items of 'BX' series is shown in "fig. 9". In this case the orders are also going to be delayed. The result does not satisfy the demand. On the figure is also shown, that in certain time, the whole production starts to delay. Thanks to the visual result, provided from 'Asprova', manufacturer can trace back along the production line to see in which moment the production is slowed, and to take measure. The graph shows that the production starts to delay at the moment of the inspection of the assembled ‘BX’ items (fig. 9)[22].

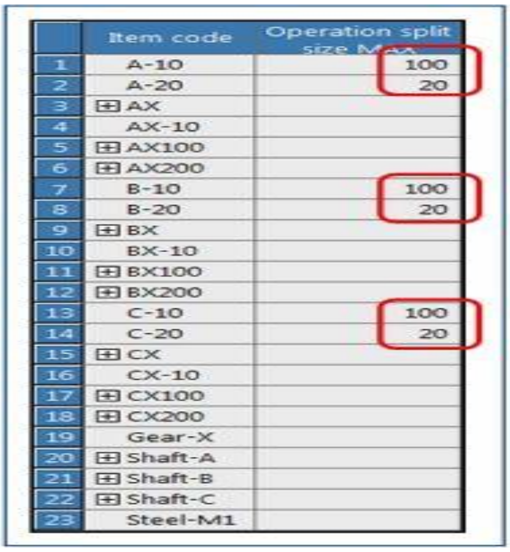


Operation=Saling BX100:10
Item=
Quantity=5
Priority=80
Start time=2016/06/22 09:05:00
End time =2016/06/22 15:05:00
Due date =2016/06/21 12:00:00

Figure 14. Gantt chart for selling the constructions with Shaft-B

USING ‘ASPROVA’ FOR OPTIMIZING THE PRODUCTION SCHEDULES:

The final results that the simulation from ‘Asprova’ shows is not always the optimum, since some orders are not going to be done on time. That is why it is necessary, after the final model of the production is ready, to perform additional optimization for finding the best possible solution. The optimization of the production model is based on ‘Local Search’ algorithms. The objective of the optimization is to fit the requested due dates for every one order. The function that is going to be used, from the ‘Asprova' program, is operation split size (fig.10). What the function does is a splitting the operation, if the operation quantity is larger than a specific value. Like ‘Local Search’ algorithms, a number of tests will be done, until the wanted result is achieved. The new data is added into the operation split size column of the ‘Item table’ [fig.10].



	Item code	Operation split size Max
1	A-10	100
2	A-20	20
3	⊞ AX	
4	AX-10	
5	⊞ AX100	
6	⊞ AX200	
7	B-10	100
8	B-20	20
9	⊞ BX	
10	BX-10	
11	⊞ BX100	
12	⊞ BX200	
13	C-10	100
14	C-20	20
15	⊞ CX	
16	CX-10	
17	⊞ CX100	
18	⊞ CX200	
19	Gear-X	
20	⊞ Shaft-A	
21	⊞ Shaft-B	
22	⊞ Shaft-C	
23	Steel-M1	

Figure 15. Using operation split size function in ‘Asprova’

The parameters “Operation split size MAX” shown on ‘fig.10’ presents the final quantity that is going to be used for optimizing the production model. This quantity is achieved after several numbers of simulations runs using different input data[22]. The function was applied for larger processes (manufacturing Shaft-A, B and C). Optimized Gantt chart is shown on ‘fig.11’.

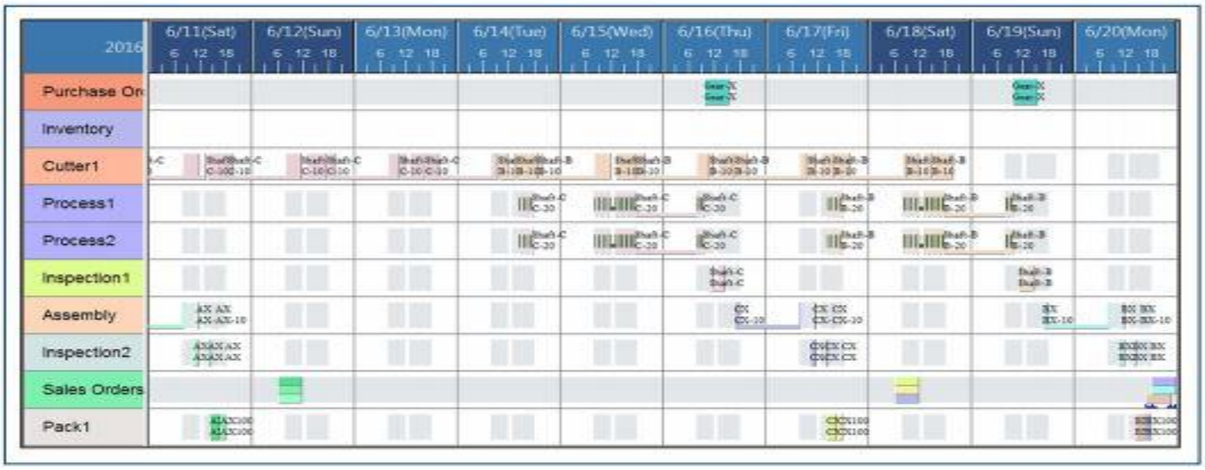
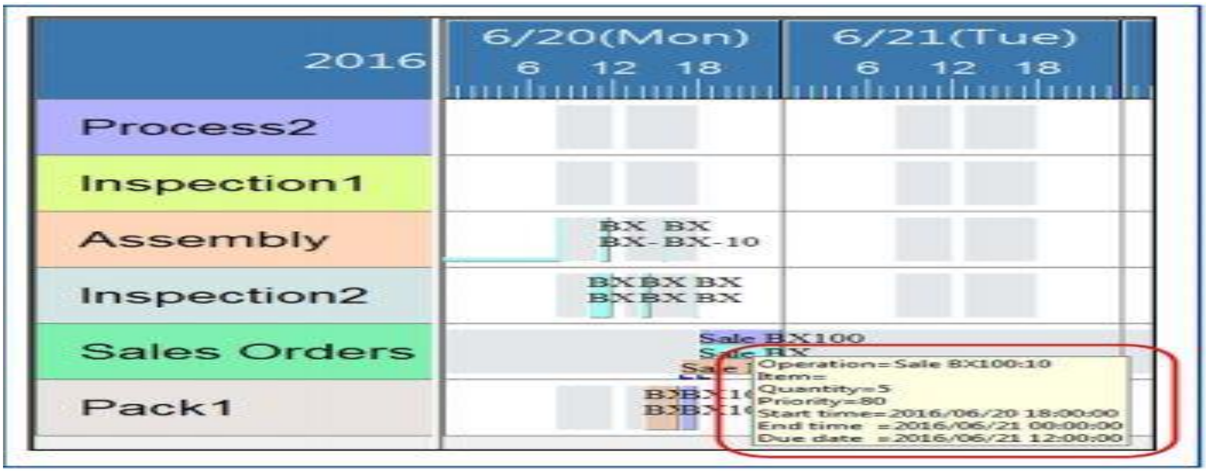


Figure 16. Resource Gantt chart after optimization

Applying the ‘Local Search’ algorithm method and using the “Asprova” function “Operation split size”, led to optimizing the production model and setting every order to be done on time. The end date for the last order (BX100) is until 2016/06/21 00:00 o’clock (fig.12). Thanks to this optimization the production schedule fit perfectly to the due date plan. The schedules are generated for every workcenter or process, according to the manufacturing plan, for the whole production period[22].



Operation=Sale BX100:10
Item=
Quantity=5
Priority=80
Start time=2016/06/20 18:00:00
End time =2016/06/21 00:00:00
Due date =2016/06/21 12:00:00

Figure 17. End time of the final order in the simulation model of production

3. LITERATURE REVIEW

Serial Number	Name of Author's	Paper Title	Name of Journal	Research Work	Result	Conclusion
1)	"Nehzati" , "Taravatsadat" , "Napsiah Ismail"	'Application of Artificial Intelligent in Production Scheduling: a critical evaluation and comparison of key approaches'	'International Conference on Industrial Engineering and Operations Management'	Investigation of different instruments and method like Fuzzy Logic, Neural, and Genetic Algorithm	Audit of Fuzzy Logic, Neural , Genetic Algorithm on Production Scheduling.	A present example in amassing plants is to move towards extraordinarily versatile creation structures that can respond quickly to demand changes and to the treatment of a collection of things.
2)	"Karl Kempf", "Bruce Russell", "Sanjiv Sidhu", and "Stu Barrett"	'AI-Based Schedulers in Manufacturing Practice: Report of a Panel Discussion'	'AI Magazine Volume 11 Number 5 (1990)'	A Viewpoint on Scheduling utilizing AI from Carnegie Group Incorporated, Texas Instruments, INTEL Corporation, Intellection.	Using AI in Scheduling of Production And Assembling Process lead to better Productivity and Efficient Working.	Computer based intelligence strategies that are available today can and ought to be applied to the arrangement of assembling planning issues.
3)	"B. Rodriguez-Somoza", "R. Gahin" and "E. A. Puente"	'PRODUCTION SCHEDULING USING AI TECHNIQUES'	'IFAC Information Control Problems in Manufacturing Technology'	Audit of Artificial Intelligence and applied it to give adaptability in planning	The program was created in a HP 520 with HP-UX Operating System and utilizing C language. What's more, on AT good PC with DOS 3.3 and utilizing Turbo C of Borland.	the CPU Time for each progression, it increments directly with the interest of pieces.

Serial Number	Name of Author	Paper Title	Name of Journal	Research Work	Result	Conclusion
4)	"Sumit Das" , "Aritra Dey" , "Akash Pal" , "Nabamita Roy"	'Applications of Artificial Intelligence in Machine Learning: Review and Prospect'	'International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 9, April 2015'	A concise audit and future possibility of the immense uses of AI has been made.	The utilizations of AI are in this manner endless regardless it stays a functioning field of research with enormous advancement choices and a promising future.	challenge is to create rise mechanized solution at basic condition utilizing AI idea, which can limit the blunder in analysis.
5)	"Jahanzaib Shabbir", "Tarique Anwer"	'Artificial Intelligence and its Role in Near Future'	'JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015'	the proposed research expected towards investigating on how the human insight varies from the man-made brainpower	Simulated intelligence puts together its activity with respect to getting to colossal measures of data, preparing it, dissecting it and, as per its activity calculations, executing assignments to take care of specific issues.	Because of the new figuring designs of the cloud, this innovation turns out to be progressively moderate for any association.
6)	"FABRIZIO SEBASTIANI"	'Machine Learning in Automated Text Categorization'	'Consiglio Nazionale delle Ricerche, Italy'	talk about in detail issues relating to three unique issues, to be specific, record portrayal, classifier development, and classifier assessment.	more research in mechanized contentbased ordering for mixed media reports is required.	TC is a decent benchmark for checking whether a given learning method can scale up to considerable sizes.

Serial Number	Name of Author	Paper Title	Name of Journal	Research Work	Result	Conclusion
7)	“Ayon Dey”	‘Machine Learning Algorithms: A Review’	‘International Journal of Computer Science and Information Technologies , Vol. 7 (3) , 2016, 1174-1179’	A short survey on different AI calculations have been examined	Investigation of Algorithm Method which are Supervised Learning, Unsupervised Learning, Semi - Supervised Learning, Reinforcement Learning, Multitask Learning, Ensemble Learning, Neural Network Learning, Instance-Based Learning.	Today every single individual is utilizing AI purposely or unconsciously. From getting a prescribed item in internet shopping to refreshing photographs in long range informal communication locales. This paper gives a prologue to a large portion of the famous AI calculations.
8)	“Kajaree Das” , “Rabi Narayan Behera”	‘A Survey on Machine Learning : Concept , Algorithms and Applications’	‘International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 2, February 2017’	This paper centres around clarifying the idea and advancement of Machine Learning, a portion of the prominent Machine Learning calculations and attempt to think about three most well known calculations dependent on some essential ideas.	Some dataset was utilized and execution of every calculation as far as preparing time, expectation time and precision of forecast have been recorded and thought about.	Being totally information driven and being able to look at a lot of information in littler interims of time, ML calculations has an edge over manual or direct programming. Additionally they are regularly progressively precise and not inclined to human predisposition.
9)	“Iqbal Muhammad” and “Zhu Yan”	‘SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY’	‘SUPERVISED MACHINE LEARNING APPROACHES : A SURVEY DOI: 10.21917/ijsc.2015.0133’	In this investigation, we will concentrate on the techniques which are being utilized for regulated learning.	Directed AI techniques are being applied in various domains.Due to extent of this paper, it is extremely hard to examine the quality and shortcomings of every calculation of ML.	The exhibition of SVM and Neural Networks is better when managing multidimensions and persistent highlights. While rationale based frameworks will in general perform better when managing discrete/clear cut highlights.

Serial Number	Name of Author	Paper Title	Name of Journal	Research Work	Result	Conclusion
10)	“Sreelekha Panda”	‘Impact of AI in Manufacturing Industries’	‘International Research Journal of Engineering and Technology Volume: 05 Issue: 11 | Nov 2018’	study on impacts difficulties of AI in assembling ventures.	Modern AI can be implanted to existing items or administrations to make them increasingly compelling, solid, more secure, and last more.	From the plan procedure and generation floor, to the inventory network and organization, AI is bound to change the manner in which we fabricate items and procedure materials until the end of time.
11)	“Thorsten Wuesta” , “Daniel Weimerb”, “Christopher Irgensc” and “Klaus-Dieter Thobend”	‘Machine learning in manufacturing: advantages, challenges, and applications’	‘Production & Manufacturing Research: An Open Access Journal, 2016 VOL. 4, NO. 1, 23–45’	this paper contributes in showing a review of accessible AI methods and organizing this fairly confused zone.	The structure is recognizing unaided AI, RL, and directed AI as a potential method to assemble the accessible calculations and applications	it very well may be said with certainty, ML is as of now a useful asset for some applications inside (wise) producing frameworks and shrewd assembling and its significance will increment further later on.
12)	“D T Pham” and “A A Afify”	‘Machine-learning techniques and their applications in manufacturing’	‘Manufacturing Engineering Centre, Cardiff University, Cardiff, UK’	This paper assesses a few AI strategies and analyzes applications in which they have been effectively conveyed.	The greater part of the current AI techniques for producing different models can improve fundamentally on the exactness of single models, however to the detriment of conceivability	Unique consideration is given to inductive realizing, which is among the most experienced of the AI approaches at present accessible. Momentum patterns and ongoing improvements in AI inquire about are additionally examined

Serial Number	Name of Author	Paper Title	Name of Journal	Research Work	Result	Conclusion
13)	“Irnran Ali Chaudhry”, “Muiz uddin Sharni” and “Abid Ali Khan”	‘MANUFACTURING APPLICAT IONS OF ARTIFICIAL INTELLIGENCE’	‘1. eng. & app l. sc i. Vol. 23 No .2 July - Dec. 2004 ISSN 102 3-862’	This paper centers around seven AI instruments, which are being utilized to take care of different issues in assembling field: information ba sed/master frameworks, fluffy logi c, neural system s, hereditary calculations and requirement fulfillment.	This paper has investigated five such devices, to be specific, information based/expcrt frameworks, fluffy rationale, neura l systems, hereditary calculations and imperative fulfillment.	numerous new assembling applications will be created and that, for requesting undertakings extraordinary er use will be made of half and half apparatuses joining the stren gth of two of more devices looked into.
14)	“Bin He”	‘Artificial Intelligent for Intelligent Manufacturing and Robotics’	‘Robotics & Automation Engineering Journal (RAEJ)’	the proposed research intended towards role of artificial intelligence in robotics and intelligent manufacturing.	the intelligent manufacturing is the deep fusion and integration of information technology and intelligent technology and equipment manufacturing process technology.	Manufacturing industry is a pillar of national development, and the equipment manufacturing is the lifeblood of national industrial development.
15)	“Farid Meziane”, “Sunil Vadera”, “Khiary Kobbacy” and “Nathan Proudlove”	‘Intelligent Systems in Manufacturing: Current Developments and Future Prospects’	‘Integrated Manufacturing Systems (currently published as: Journal of Manufacturing Technology Management)’ **ISSN:** 09576061	future patterns in utilizing AI for assembling frameworks	audit of man-made brainpower methods utilized in assembling frameworks has been done effectively.	Astute Manufacturing Systems, Artificial Intelligence, Neural Networks, Genetic Algorithms, Knowledge-based Systems are important part for executing AI in assembling.

Serial Number	Name of Author	Paper Title	Name of Journal	Research Work	Result	Conclusion
16)	“Annina Simon” , “Mahima Singh Deo” , “S. Venkatesan “, ”D.R. Ramesh Babu”	‘An Overview of Machine Learning and its Applications’	‘International Journal of Electrical Sciences & Engineering (IJESE) Volume 1, Issue 1; 2015 pp. 22-24’	this examination paper is about ongoing advances in innovation, explicitly profound learning a region of AI which discovers applications in enormous information investigation and man-made brainpower.	Profound learning systems have been reprimanded on the grounds that there is no chance to get of speaking to causal connections, (for example, among illnesses and their manifestations), and the calculations neglect to secure dynamic thoughts like "kin" or "indistinguishable from."	Profound Learning is just a little advance towards building machines which have human-like knowledge. Further headways must be made so as to accomplish our definitive objective. Associations like Google, Facebook, Microsoft and Baidu (a Chinese internet searcher) are becoming tied up with this innovation and investigating different roads accessible.
17)	“Wayne J. Davis” , “Albert T. Jones”	‘Artificial Intelligence Techniques in Real-Time Production Scheduling’	‘NISTIR 88-3891 Issued February 1989’	This paper tends to the ongoing creation booking issue as an extraordinary instance of an a lot bigger class of continuous basic leadership/control issues.	This paper has assessed five such apparatuses, specifically, information based/expcrt frameworks, fluffy rationale, neura l systems, hereditary calculations and requirement fulfillment.	An increasingly summed up calculation for ongoing basic leadership to develop and that extra issue zones will be characterized for which AI will give an advantageous arrangement approach.
18)	“Jay Lee” ,”Jaskaran Singh” , “Moslem Azamfar”	‘Industrial Artificial Intelligence’	‘Department of Mechanical Engineering , University of Cincinnati, Cincinnati, USA’	the key empowering influences for this transformative innovation alongside their huge favorable circumstances are talked about.	this examination clarifies "Beacon Factories" as a rising status applying to the top producers that have actualized Industrial AI in their assembling environment and increased noteworthy money related advantages.	this examination will fill in as a rule and guide for specialists and businesses towards this present reality usage of Industrial AI.

Serial Number	Name of Author	Paper Title	Name of Journal	Research Work	Result	Conclusion
19)	"Chuda Basnet" , "Joe H. Mize"	'Scheduling and Control of Flexible Manufacturing Systems: A Critical Review'	'Department of Management Systems University of Waikato ,Hamilton, New Zealand and School of Industrial Engineering and Management Oklahoma State University, Stillwater, U.S.A.'	An audit of writing concerning the activities part of FMS. Articles underlining numerous methodological points of view are fundamentally checked on.	Incredible steps have been made in the booking and control writing of FMS. There is currently a full grown writing utilizing diverse methodological methodologies.	Future work should be done in researching the utilization of the techniques in the down to earth field, in making the control frameworks more easy to use, and in growing increasingly thorough control frameworks.
20)	"Shirin Sohrabi"	'AI Planning for Enterprise: Putting Theory Into Practice'	'Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)'	Number of AI Planning applications for Enterprise and talk about various difficulties in applying AI Planning in that setting.	Effectively explored the connection between AI Planning and a few applications and e key research difficulties and abridge the advancement made to date in tending to them.	The last challenge is to address everything else that has to do with building up an AI Planning-based start to finish framework. This incorporates collaboration with the clients, change of information into perceptions, programmed interpretation of the space information into an AI Planning issue, post-handling the plans, and representation of the arrangement.
21)	"Mark S. Fox"	'Industrial Applications of Artificial Intelligence'	'Intelligent Systems Laboratory, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213, USA'	The survey centers around where, at each point in the item life cycle, there are issues to be comprehended, where AI is at present being applied, and where it might be applied later on.	Effectively survey present and future utilizations of Artificial Intelligence (AI) and Knowledge-Based frameworks to assembling.	Frameworks that improve our critical thinking by settling on better choices all the more rapidly will be made. Frameworks that incorporate more information about the manufacturing plant floor and, subsequently, settle on better choices will be made.

4. SCHEDULING USING PYTHON PROGRAMMING

In this project, I am going to design schedule for the upcoming week of workshop. Here, I took some consideration that is, each day has two shifts of 8 hours, there are 10 employees(4 are managers and 6 are workers). Each employee need to work for 40 hours per week in a total of 5 shifts. It is mandatory to work atleast 3 shifts but not more than 7 shifts. For every employee who worked more than 40 hours are given overtime cost which is 1.5 times of regular cost per hour. There is a mandatory condition that each shift require at least 1 manager. I am also going to find the optimal cost of schedule.

OBJECTIVE FUNCTION :-

$$Min\left(\sum_{w} RegCost_{w} * RegShifts_{w} + \sum_{w} OTCost_{w} * OTShifts_{w}\right)$$

So, lets begin...,

4.1 CODE FOR MODEL

All codes are written in jupyter notebook. The result mention below is totally jupyter notebook oriented.

Import required packages

```
import pandas as pd
import numpy as np
from gurobipy import *
```

Create a list of workers and shifts

```
shiftList =
['Monday1','Monday2','Tuesday1','Tuesday2','Wednesday1','Wednesday2','Thurs
day1','Thursday2','Friday1','Friday2','Saturday1','Saturday2','Sunday1','Su
nday2']

workerList =
['EE01','EE02','EE03','EE04','EE05','EE06','EE07','EE08','EE09','EE10']
```

Define shift requirements

```
shiftReq = [3,2,4,4,5,4,5,4,2,4,5,4,3,5]

shiftRequirements = {s : shiftReq[i] for i,s in enumerate(shiftList) }
```

Clarify worker availability

```
# Assume everyone is available
availability = pd.DataFrame(np.ones((len(workerList), len(shiftList))),
index=workerList, columns=shiftList)

# For illustration, assume following people are unavailable: EE02 on
Tuesday1, EE05 on Saturday2, EE08 on Thursday1
availability.at['EE02','Tuesday1'] = 0
availability.at['EE05','Saturday2'] = 0
availability.at['EE08','Thursday1'] = 0

# Create dictionary of final worker availability
avail = {(w,s) : availability.loc[w,s] for w in workerList for s in
shiftList}
```

Specify who is a manager and who isn't

```
mgmtList = ['EE01','EE03','EE05','EE07']

nonmgmtList = [x for x in workerList if x not in mgmtList]
```

Define total shift cost per worker

```
# Cost of a regular shift
regCost = [200,100,225,110,190,105,210,120,95,100]

# Cost of overtime shift
OTCost = [1.5*x for x in regCost]

# Create dictionaries with costs
regularCost  = { w : regCost[i] for i,w in enumerate(workerList) }

overtimeCost  = { w : OTCost[i] for i,w in enumerate(workerList) }
```

Input assumptions

```
# Range of shifts that every workers is required to stay between
minShifts = 3
maxShifts = 7

# Number of shifts to trigger overtime
OTTrigger = 5

model = Model('Workers Scheduling')
Using license file C:\Users\risha\gurobi.lic
Academic license - for non-commercial use only


# ub ensures that workers are only staffed when they are available
x = model.addVars(workerList, shiftList, ub=avail, vtype=GRB.BINARY,
name='x')

regHours = model.addVars(workerList, name='regHrs')
overtimeHours = model.addVars(workerList, name='overtimeHrs')
overtimeTrigger = model.addVars(workerList, name = "OT_Trigger", vtype =
GRB.BINARY)
```

```
# Ensure proper number of workers are scheduled

shiftReq = model.addConstrs((
    (x.sum('*',s) == shiftRequirements[s] for s in shiftList)
), name='shiftRequirement')
```

Differentiate between regular time and overtime

Decompose total shifts for each worker into regular shifts and OT shifts

```
regOT1 = model.addConstrs((regHours[w] + overtimeHours[w] == x.sum(w,'*')
for w in workerList))
```

Ensure that regular shifts are accounted for first for each nurse before counting OT shifts

```
regOT2 = model.addConstrs((regHours[w] <= OTTrigger for w in workerList))
```

Only allow the OT trigger to come on when regular shift count is greater than regular shift limit

```
regOT3 = model.addConstrs((regHours[w] / OTTrigger >= overtimeTrigger[w]
for w in workerList))
```

Ensure each worker stays within min and max shift bounds

```
minShiftsConstr = model.addConstrs(((
    x.sum(w,'*') >= minShifts for w in workerList)
), name='minShifts')

maxShiftsConstr = model.addConstrs(((
    x.sum(w,'*') <= maxShifts for w in workerList)
), name='maxShifts')
```

Ensure every shift has at least one manager

```
for s in shiftList:
    model.addConstr((quicksum(x.sum(m,s) for m in mgmtList) >= 1),
name='mgmtStaffing'+str(s))
```

```
# Minimize total cost, accounting for pay difference between regular time
and overtime

model.ModelSense = GRB.MINIMIZE

Cost = 0
Cost += (quicksum(regularCost[w]*regHours[w] +
overtimeCost[w]*overtimeHours[w] for w in workerList))

model.setObjective(Cost)
```

4.2 Running the optimization

```
model.write("Optimized_Scheduling.lp")
file = open("Optimized_Scheduling.lp", 'r')
print(file.read())
file.close()
\ Model Workers Scheduling
\ LP format - for model browsing. Use MPS format to capture full model
detail.
Minimize
  200 regHrs[EE01] + 100 regHrs[EE02] + 225 regHrs[EE03] + 110 regHrs[EE04]
   + 190 regHrs[EE05] + 105 regHrs[EE06] + 210 regHrs[EE07]
   + 120 regHrs[EE08] + 95 regHrs[EE09] + 100 regHrs[EE10]
   + 300 overtimeHrs[EE01] + 150 overtimeHrs[EE02]
   + 337.5 overtimeHrs[EE03] + 165 overtimeHrs[EE04]
   + 285 overtimeHrs[EE05] + 157.5 overtimeHrs[EE06]
   + 315 overtimeHrs[EE07] + 180 overtimeHrs[EE08]
   + 142.5 overtimeHrs[EE09] + 150 overtimeHrs[EE10]
Subject To
 shiftRequirement[Monday1]: x[EE01,Monday1] + x[EE02,Monday1]
   + x[EE03,Monday1] + x[EE04,Monday1] + x[EE05,Monday1] + x[EE06,Monday1]
   + x[EE07,Monday1] + x[EE08,Monday1] + x[EE09,Monday1] + x[EE10,Monday1]
   = 3
 shiftRequirement[Monday2]: x[EE01,Monday2] + x[EE02,Monday2]
   + x[EE03,Monday2] + x[EE04,Monday2] + x[EE05,Monday2] + x[EE06,Monday2]
   + x[EE07,Monday2] + x[EE08,Monday2] + x[EE09,Monday2] + x[EE10,Monday2]
   = 2
```

```
shiftRequirement[Tuesday1]: x[EE01,Tuesday1] + x[EE02,Tuesday1]
  + x[EE03,Tuesday1] + x[EE04,Tuesday1] + x[EE05,Tuesday1]
  + x[EE06,Tuesday1] + x[EE07,Tuesday1] + x[EE08,Tuesday1]
  + x[EE09,Tuesday1] + x[EE10,Tuesday1] = 4
shiftRequirement[Tuesday2]: x[EE01,Tuesday2] + x[EE02,Tuesday2]
  + x[EE03,Tuesday2] + x[EE04,Tuesday2] + x[EE05,Tuesday2]
  + x[EE06,Tuesday2] + x[EE07,Tuesday2] + x[EE08,Tuesday2]
  + x[EE09,Tuesday2] + x[EE10,Tuesday2] = 4
shiftRequirement[Wednesday1]: x[EE01,Wednesday1] + x[EE02,Wednesday1]
  + x[EE03,Wednesday1] + x[EE04,Wednesday1] + x[EE05,Wednesday1]
  + x[EE06,Wednesday1] + x[EE07,Wednesday1] + x[EE08,Wednesday1]
  + x[EE09,Wednesday1] + x[EE10,Wednesday1] = 5
shiftRequirement[Wednesday2]: x[EE01,Wednesday2] + x[EE02,Wednesday2]
  + x[EE03,Wednesday2] + x[EE04,Wednesday2] + x[EE05,Wednesday2]
  + x[EE06,Wednesday2] + x[EE07,Wednesday2] + x[EE08,Wednesday2]
  + x[EE09,Wednesday2] + x[EE10,Wednesday2] = 4
shiftRequirement[Thursday1]: x[EE01,Thursday1] + x[EE02,Thursday1]
  + x[EE03,Thursday1] + x[EE04,Thursday1] + x[EE05,Thursday1]
  + x[EE06,Thursday1] + x[EE07,Thursday1] + x[EE08,Thursday1]
  + x[EE09,Thursday1] + x[EE10,Thursday1] = 5
shiftRequirement[Thursday2]: x[EE01,Thursday2] + x[EE02,Thursday2]
  + x[EE03,Thursday2] + x[EE04,Thursday2] + x[EE05,Thursday2]
  + x[EE06,Thursday2] + x[EE07,Thursday2] + x[EE08,Thursday2]
  + x[EE09,Thursday2] + x[EE10,Thursday2] = 4
shiftRequirement[Friday1]: x[EE01,Friday1] + x[EE02,Friday1]
  + x[EE03,Friday1] + x[EE04,Friday1] + x[EE05,Friday1] + x[EE06,Friday1]
  + x[EE07,Friday1] + x[EE08,Friday1] + x[EE09,Friday1] + x[EE10,Friday1]
  = 2
shiftRequirement[Friday2]: x[EE01,Friday2] + x[EE02,Friday2]
  + x[EE03,Friday2] + x[EE04,Friday2] + x[EE05,Friday2] + x[EE06,Friday2]
  + x[EE07,Friday2] + x[EE08,Friday2] + x[EE09,Friday2] + x[EE10,Friday2]
  = 4
shiftRequirement[Saturday1]: x[EE01,Saturday1] + x[EE02,Saturday1]
  + x[EE03,Saturday1] + x[EE04,Saturday1] + x[EE05,Saturday1]
  + x[EE06,Saturday1] + x[EE07,Saturday1] + x[EE08,Saturday1]
  + x[EE09,Saturday1] + x[EE10,Saturday1] = 5
shiftRequirement[Saturday2]: x[EE01,Saturday2] + x[EE02,Saturday2]
  + x[EE03,Saturday2] + x[EE04,Saturday2] + x[EE05,Saturday2]
  + x[EE06,Saturday2] + x[EE07,Saturday2] + x[EE08,Saturday2]
  + x[EE09,Saturday2] + x[EE10,Saturday2] = 4
shiftRequirement[Sunday1]: x[EE01,Sunday1] + x[EE02,Sunday1]
  + x[EE03,Sunday1] + x[EE04,Sunday1] + x[EE05,Sunday1] + x[EE06,Sunday1]
  + x[EE07,Sunday1] + x[EE08,Sunday1] + x[EE09,Sunday1] + x[EE10,Sunday1]
  = 3
shiftRequirement[Sunday2]: x[EE01,Sunday2] + x[EE02,Sunday2]
  + x[EE03,Sunday2] + x[EE04,Sunday2] + x[EE05,Sunday2] + x[EE06,Sunday2]
  + x[EE07,Sunday2] + x[EE08,Sunday2] + x[EE09,Sunday2] + x[EE10,Sunday2]
  = 5
R14: - x[EE01,Monday1] - x[EE01,Monday2] - x[EE01,Tuesday1]
  - x[EE01,Tuesday2] - x[EE01,Wednesday1] - x[EE01,Wednesday2]
  - x[EE01,Thursday1] - x[EE01,Thursday2] - x[EE01,Friday1]
  - x[EE01,Friday2] - x[EE01,Saturday1] - x[EE01,Saturday2]
  - x[EE01,Sunday1] - x[EE01,Sunday2] + regHrs[EE01] + overtimeHrs[EE01]
  = 0
R15: - x[EE02,Monday1] - x[EE02,Monday2] - x[EE02,Tuesday1]
  - x[EE02,Tuesday2] - x[EE02,Wednesday1] - x[EE02,Wednesday2]
  - x[EE02,Thursday1] - x[EE02,Thursday2] - x[EE02,Friday1]
  - x[EE02,Friday2] - x[EE02,Saturday1] - x[EE02,Saturday2]
  - x[EE02,Sunday1] - x[EE02,Sunday2] + regHrs[EE02] + overtimeHrs[EE02]
  = 0
R16: - x[EE03,Monday1] - x[EE03,Monday2] - x[EE03,Tuesday1]
  - x[EE03,Tuesday2] - x[EE03,Wednesday1] - x[EE03,Wednesday2]
  - x[EE03,Thursday1] - x[EE03,Thursday2] - x[EE03,Friday1]
  - x[EE03,Friday2] - x[EE03,Saturday1] - x[EE03,Saturday2]
  - x[EE03,Sunday1] - x[EE03,Sunday2] + regHrs[EE03] + overtimeHrs[EE03]
  = 0
R17: - x[EE04,Monday1] - x[EE04,Monday2] - x[EE04,Tuesday1]
  - x[EE04,Tuesday2] - x[EE04,Wednesday1] - x[EE04,Wednesday2]
  - x[EE04,Thursday1] - x[EE04,Thursday2] - x[EE04,Friday1]
  - x[EE04,Friday2] - x[EE04,Saturday1] - x[EE04,Saturday2]
  - x[EE04,Sunday1] - x[EE04,Sunday2] + regHrs[EE04] + overtimeHrs[EE04]
  = 0
R18: - x[EE05,Monday1] - x[EE05,Monday2] - x[EE05,Tuesday1]
  - x[EE05,Tuesday2] - x[EE05,Wednesday1] - x[EE05,Wednesday2]
  - x[EE05,Thursday1] - x[EE05,Thursday2] - x[EE05,Friday1]
  - x[EE05,Friday2] - x[EE05,Saturday1] - x[EE05,Saturday2]
  - x[EE05,Sunday1] - x[EE05,Sunday2] + regHrs[EE05] + overtimeHrs[EE05]
  = 0
R19: - x[EE06,Monday1] - x[EE06,Monday2] - x[EE06,Tuesday1]
  - x[EE06,Tuesday2] - x[EE06,Wednesday1] - x[EE06,Wednesday2]
  - x[EE06,Thursday1] - x[EE06,Thursday2] - x[EE06,Friday1]
  - x[EE06,Friday2] - x[EE06,Saturday1] - x[EE06,Saturday2]
  - x[EE06,Sunday1] - x[EE06,Sunday2] + regHrs[EE06] + overtimeHrs[EE06]
  = 0
R20: - x[EE07,Monday1] - x[EE07,Monday2] - x[EE07,Tuesday1]
  - x[EE07,Tuesday2] - x[EE07,Wednesday1] - x[EE07,Wednesday2]
  - x[EE07,Thursday1] - x[EE07,Thursday2] - x[EE07,Friday1]
  - x[EE07,Friday2] - x[EE07,Saturday1] - x[EE07,Saturday2]
  - x[EE07,Sunday1] - x[EE07,Sunday2] + regHrs[EE07] + overtimeHrs[EE07]
  = 0
R21: - x[EE08,Monday1] - x[EE08,Monday2] - x[EE08,Tuesday1]
  - x[EE08,Tuesday2] - x[EE08,Wednesday1] - x[EE08,Wednesday2]
  - x[EE08,Thursday1] - x[EE08,Thursday2] - x[EE08,Friday1]
  - x[EE08,Friday2] - x[EE08,Saturday1] - x[EE08,Saturday2]
  - x[EE08,Sunday1] - x[EE08,Sunday2] + regHrs[EE08] + overtimeHrs[EE08]
  = 0
R22: - x[EE09,Monday1] - x[EE09,Monday2] - x[EE09,Tuesday1]
  - x[EE09,Tuesday2] - x[EE09,Wednesday1] - x[EE09,Wednesday2]
  - x[EE09,Thursday1] - x[EE09,Thursday2] - x[EE09,Friday1]
  - x[EE09,Friday2] - x[EE09,Saturday1] - x[EE09,Saturday2]
  - x[EE09,Sunday1] - x[EE09,Sunday2] + regHrs[EE09] + overtimeHrs[EE09]
  = 0
R23: - x[EE10,Monday1] - x[EE10,Monday2] - x[EE10,Tuesday1]
  - x[EE10,Tuesday2] - x[EE10,Wednesday1] - x[EE10,Wednesday2]
  - x[EE10,Thursday1] - x[EE10,Thursday2] - x[EE10,Friday1]
  - x[EE10,Friday2] - x[EE10,Saturday1] - x[EE10,Saturday2]
  - x[EE10,Sunday1] - x[EE10,Sunday2] + regHrs[EE10] + overtimeHrs[EE10]
  = 0
R24: regHrs[EE01] <= 5
R25: regHrs[EE02] <= 5
R26: regHrs[EE03] <= 5
R27: regHrs[EE04] <= 5
R28: regHrs[EE05] <= 5
R29: regHrs[EE06] <= 5
R30: regHrs[EE07] <= 5
R31: regHrs[EE08] <= 5
R32: regHrs[EE09] <= 5
R33: regHrs[EE10] <= 5
R34: 0.2 regHrs[EE01] - OT_Trigger[EE01] >= 0
R35: 0.2 regHrs[EE02] - OT_Trigger[EE02] >= 0
R36: 0.2 regHrs[EE03] - OT_Trigger[EE03] >= 0
R37: 0.2 regHrs[EE04] - OT_Trigger[EE04] >= 0
R38: 0.2 regHrs[EE05] - OT_Trigger[EE05] >= 0
R39: 0.2 regHrs[EE06] - OT_Trigger[EE06] >= 0
R40: 0.2 regHrs[EE07] - OT_Trigger[EE07] >= 0
R41: 0.2 regHrs[EE08] - OT_Trigger[EE08] >= 0
R42: 0.2 regHrs[EE09] - OT_Trigger[EE09] >= 0
R43: 0.2 regHrs[EE10] - OT_Trigger[EE10] >= 0
minShifts[EE01]: x[EE01,Monday1] + x[EE01,Monday2] + x[EE01,Tuesday1]
  + x[EE01,Tuesday2] + x[EE01,Wednesday1] + x[EE01,Wednesday2]
  + x[EE01,Thursday1] + x[EE01,Thursday2] + x[EE01,Friday1]
  + x[EE01,Friday2] + x[EE01,Saturday1] + x[EE01,Saturday2]
  + x[EE01,Sunday1] + x[EE01,Sunday2] >= 3
minShifts[EE02]: x[EE02,Monday1] + x[EE02,Monday2] + x[EE02,Tuesday1]
  + x[EE02,Tuesday2] + x[EE02,Wednesday1] + x[EE02,Wednesday2]
  + x[EE02,Thursday1] + x[EE02,Thursday2] + x[EE02,Friday1]
  + x[EE02,Friday2] + x[EE02,Saturday1] + x[EE02,Saturday2]
  + x[EE02,Sunday1] + x[EE02,Sunday2] >= 3
minShifts[EE03]: x[EE03,Monday1] + x[EE03,Monday2] + x[EE03,Tuesday1]
  + x[EE03,Tuesday2] + x[EE03,Wednesday1] + x[EE03,Wednesday2]
  + x[EE03,Thursday1] + x[EE03,Thursday2] + x[EE03,Friday1]
  + x[EE03,Friday2] + x[EE03,Saturday1] + x[EE03,Saturday2]
  + x[EE03,Sunday1] + x[EE03,Sunday2] >= 3
minShifts[EE04]: x[EE04,Monday1] + x[EE04,Monday2] + x[EE04,Tuesday1]
  + x[EE04,Tuesday2] + x[EE04,Wednesday1] + x[EE04,Wednesday2]
  + x[EE04,Thursday1] + x[EE04,Thursday2] + x[EE04,Friday1]
  + x[EE04,Friday2] + x[EE04,Saturday1] + x[EE04,Saturday2]
  + x[EE04,Sunday1] + x[EE04,Sunday2] >= 3
minShifts[EE05]: x[EE05,Monday1] + x[EE05,Monday2] + x[EE05,Tuesday1]
  + x[EE05,Tuesday2] + x[EE05,Wednesday1] + x[EE05,Wednesday2]
  + x[EE05,Thursday1] + x[EE05,Thursday2] + x[EE05,Friday1]
  + x[EE05,Friday2] + x[EE05,Saturday1] + x[EE05,Saturday2]
  + x[EE05,Sunday1] + x[EE05,Sunday2] >= 3
minShifts[EE06]: x[EE06,Monday1] + x[EE06,Monday2] + x[EE06,Tuesday1]
  + x[EE06,Tuesday2] + x[EE06,Wednesday1] + x[EE06,Wednesday2]
  + x[EE06,Thursday1] + x[EE06,Thursday2] + x[EE06,Friday1]
  + x[EE06,Friday2] + x[EE06,Saturday1] + x[EE06,Saturday2]
  + x[EE06,Sunday1] + x[EE06,Sunday2] >= 3
minShifts[EE07]: x[EE07,Monday1] + x[EE07,Monday2] + x[EE07,Tuesday1]
  + x[EE07,Tuesday2] + x[EE07,Wednesday1] + x[EE07,Wednesday2]
  + x[EE07,Thursday1] + x[EE07,Thursday2] + x[EE07,Friday1]
  + x[EE07,Friday2] + x[EE07,Saturday1] + x[EE07,Saturday2]
  + x[EE07,Sunday1] + x[EE07,Sunday2] >= 3
minShifts[EE08]: x[EE08,Monday1] + x[EE08,Monday2] + x[EE08,Tuesday1]
  + x[EE08,Tuesday2] + x[EE08,Wednesday1] + x[EE08,Wednesday2]
  + x[EE08,Thursday1] + x[EE08,Thursday2] + x[EE08,Friday1]
  + x[EE08,Friday2] + x[EE08,Saturday1] + x[EE08,Saturday2]
  + x[EE08,Sunday1] + x[EE08,Sunday2] >= 3
minShifts[EE09]: x[EE09,Monday1] + x[EE09,Monday2] + x[EE09,Tuesday1]
  + x[EE09,Tuesday2] + x[EE09,Wednesday1] + x[EE09,Wednesday2]
  + x[EE09,Thursday1] + x[EE09,Thursday2] + x[EE09,Friday1]
  + x[EE09,Friday2] + x[EE09,Saturday1] + x[EE09,Saturday2]
  + x[EE09,Sunday1] + x[EE09,Sunday2] >= 3
minShifts[EE10]: x[EE10,Monday1] + x[EE10,Monday2] + x[EE10,Tuesday1]
  + x[EE10,Tuesday2] + x[EE10,Wednesday1] + x[EE10,Wednesday2]
  + x[EE10,Thursday1] + x[EE10,Thursday2] + x[EE10,Friday1]
  + x[EE10,Friday2] + x[EE10,Saturday1] + x[EE10,Saturday2]
  + x[EE10,Sunday1] + x[EE10,Sunday2] >= 3
maxShifts[EE01]: x[EE01,Monday1] + x[EE01,Monday2] + x[EE01,Tuesday1]
  + x[EE01,Tuesday2] + x[EE01,Wednesday1] + x[EE01,Wednesday2]
  + x[EE01,Thursday1] + x[EE01,Thursday2] + x[EE01,Friday1]
  + x[EE01,Friday2] + x[EE01,Saturday1] + x[EE01,Saturday2]
  + x[EE01,Sunday1] + x[EE01,Sunday2] <= 7
maxShifts[EE02]: x[EE02,Monday1] + x[EE02,Monday2] + x[EE02,Tuesday1]
  + x[EE02,Tuesday2] + x[EE02,Wednesday1] + x[EE02,Wednesday2]
  + x[EE02,Thursday1] + x[EE02,Thursday2] + x[EE02,Friday1]
  + x[EE02,Friday2] + x[EE02,Saturday1] + x[EE02,Saturday2]
  + x[EE02,Sunday1] + x[EE02,Sunday2] <= 7
```

```
maxShifts[EE03]: x[EE03,Monday1] + x[EE03,Monday2] + x[EE03,Tuesday1]
  + x[EE03,Tuesday2] + x[EE03,Wednesday1] + x[EE03,Wednesday2]
  + x[EE03,Thursday1] + x[EE03,Thursday2] + x[EE03,Friday1]
  + x[EE03,Friday2] + x[EE03,Saturday1] + x[EE03,Saturday2]
  + x[EE03,Sunday1] + x[EE03,Sunday2] <= 7
maxShifts[EE04]: x[EE04,Monday1] + x[EE04,Monday2] + x[EE04,Tuesday1]
  + x[EE04,Tuesday2] + x[EE04,Wednesday1] + x[EE04,Wednesday2]
  + x[EE04,Thursday1] + x[EE04,Thursday2] + x[EE04,Friday1]
  + x[EE04,Friday2] + x[EE04,Saturday1] + x[EE04,Saturday2]
  + x[EE04,Sunday1] + x[EE04,Sunday2] <= 7
maxShifts[EE05]: x[EE05,Monday1] + x[EE05,Monday2] + x[EE05,Tuesday1]
  + x[EE05,Tuesday2] + x[EE05,Wednesday1] + x[EE05,Wednesday2]
  + x[EE05,Thursday1] + x[EE05,Thursday2] + x[EE05,Friday1]
  + x[EE05,Friday2] + x[EE05,Saturday1] + x[EE05,Saturday2]
  + x[EE05,Sunday1] + x[EE05,Sunday2] <= 7
maxShifts[EE06]: x[EE06,Monday1] + x[EE06,Monday2] + x[EE06,Tuesday1]
  + x[EE06,Tuesday2] + x[EE06,Wednesday1] + x[EE06,Wednesday2]
  + x[EE06,Thursday1] + x[EE06,Thursday2] + x[EE06,Friday1]
  + x[EE06,Friday2] + x[EE06,Saturday1] + x[EE06,Saturday2]
  + x[EE06,Sunday1] + x[EE06,Sunday2] <= 7
maxShifts[EE07]: x[EE07,Monday1] + x[EE07,Monday2] + x[EE07,Tuesday1]
  + x[EE07,Tuesday2] + x[EE07,Wednesday1] + x[EE07,Wednesday2]
  + x[EE07,Thursday1] + x[EE07,Thursday2] + x[EE07,Friday1]
  + x[EE07,Friday2] + x[EE07,Saturday1] + x[EE07,Saturday2]
  + x[EE07,Sunday1] + x[EE07,Sunday2] <= 7
maxShifts[EE08]: x[EE08,Monday1] + x[EE08,Monday2] + x[EE08,Tuesday1]
  + x[EE08,Tuesday2] + x[EE08,Wednesday1] + x[EE08,Wednesday2]
  + x[EE08,Thursday1] + x[EE08,Thursday2] + x[EE08,Friday1]
  + x[EE08,Friday2] + x[EE08,Saturday1] + x[EE08,Saturday2]
  + x[EE08,Sunday1] + x[EE08,Sunday2] <= 7
maxShifts[EE09]: x[EE09,Monday1] + x[EE09,Monday2] + x[EE09,Tuesday1]
  + x[EE09,Tuesday2] + x[EE09,Wednesday1] + x[EE09,Wednesday2]
  + x[EE09,Thursday1] + x[EE09,Thursday2] + x[EE09,Friday1]
  + x[EE09,Friday2] + x[EE09,Saturday1] + x[EE09,Saturday2]
  + x[EE09,Sunday1] + x[EE09,Sunday2] <= 7
maxShifts[EE10]: x[EE10,Monday1] + x[EE10,Monday2] + x[EE10,Tuesday1]
  + x[EE10,Tuesday2] + x[EE10,Wednesday1] + x[EE10,Wednesday2]
  + x[EE10,Thursday1] + x[EE10,Thursday2] + x[EE10,Friday1]
  + x[EE10,Friday2] + x[EE10,Saturday1] + x[EE10,Saturday2]
  + x[EE10,Sunday1] + x[EE10,Sunday2] <= 7
mgmtStaffingMonday1: x[EE01,Monday1] + x[EE03,Monday1] + x[EE05,Monday1]
  + x[EE07,Monday1] >= 1
mgmtStaffingMonday2: x[EE01,Monday2] + x[EE03,Monday2] + x[EE05,Monday2]
  + x[EE07,Monday2] >= 1
mgmtStaffingTuesday1: x[EE01,Tuesday1] + x[EE03,Tuesday1]
  + x[EE05,Tuesday1] + x[EE07,Tuesday1] >= 1
mgmtStaffingTuesday2: x[EE01,Tuesday2] + x[EE03,Tuesday2]
```

```
   + x[EE05,Tuesday2] + x[EE07,Tuesday2] >= 1
 mgmtStaffingWednesday1: x[EE01,Wednesday1] + x[EE03,Wednesday1]
   + x[EE05,Wednesday1] + x[EE07,Wednesday1] >= 1
 mgmtStaffingWednesday2: x[EE01,Wednesday2] + x[EE03,Wednesday2]
   + x[EE05,Wednesday2] + x[EE07,Wednesday2] >= 1
 mgmtStaffingThursday1: x[EE01,Thursday1] + x[EE03,Thursday1]
   + x[EE05,Thursday1] + x[EE07,Thursday1] >= 1
 mgmtStaffingThursday2: x[EE01,Thursday2] + x[EE03,Thursday2]
   + x[EE05,Thursday2] + x[EE07,Thursday2] >= 1
 mgmtStaffingFriday1: x[EE01,Friday1] + x[EE03,Friday1] + x[EE05,Friday1]
   + x[EE07,Friday1] >= 1
 mgmtStaffingFriday2: x[EE01,Friday2] + x[EE03,Friday2] + x[EE05,Friday2]
   + x[EE07,Friday2] >= 1
 mgmtStaffingSaturday1: x[EE01,Saturday1] + x[EE03,Saturday1]
   + x[EE05,Saturday1] + x[EE07,Saturday1] >= 1
 mgmtStaffingSaturday2: x[EE01,Saturday2] + x[EE03,Saturday2]
   + x[EE05,Saturday2] + x[EE07,Saturday2] >= 1
 mgmtStaffingSunday1: x[EE01,Sunday1] + x[EE03,Sunday1] + x[EE05,Sunday1]
   + x[EE07,Sunday1] >= 1
 mgmtStaffingSunday2: x[EE01,Sunday2] + x[EE03,Sunday2] + x[EE05,Sunday2]
   + x[EE07,Sunday2] >= 1
Bounds
 x[EE02,Tuesday1] = 0
 x[EE05,Saturday2] = 0
 x[EE08,Thursday1] = 0
Binaries
 x[EE01,Monday1] x[EE01,Monday2] x[EE01,Tuesday1] x[EE01,Tuesday2]
 x[EE01,Wednesday1] x[EE01,Wednesday2] x[EE01,Thursday1] x[EE01,Thursday2]
 x[EE01,Friday1] x[EE01,Friday2] x[EE01,Saturday1] x[EE01,Saturday2]
 x[EE01,Sunday1] x[EE01,Sunday2] x[EE02,Monday1] x[EE02,Monday2]
 x[EE02,Tuesday1] x[EE02,Tuesday2] x[EE02,Wednesday1] x[EE02,Wednesday2]
 x[EE02,Thursday1] x[EE02,Thursday2] x[EE02,Friday1] x[EE02,Friday2]
 x[EE02,Saturday1] x[EE02,Saturday2] x[EE02,Sunday1] x[EE02,Sunday2]
 x[EE03,Monday1] x[EE03,Monday2] x[EE03,Tuesday1] x[EE03,Tuesday2]
 x[EE03,Wednesday1] x[EE03,Wednesday2] x[EE03,Thursday1] x[EE03,Thursday2]
 x[EE03,Friday1] x[EE03,Friday2] x[EE03,Saturday1] x[EE03,Saturday2]
 x[EE03,Sunday1] x[EE03,Sunday2] x[EE04,Monday1] x[EE04,Monday2]
 x[EE04,Tuesday1] x[EE04,Tuesday2] x[EE04,Wednesday1] x[EE04,Wednesday2]
 x[EE04,Thursday1] x[EE04,Thursday2] x[EE04,Friday1] x[EE04,Friday2]
 x[EE04,Saturday1] x[EE04,Saturday2] x[EE04,Sunday1] x[EE04,Sunday2]
 x[EE05,Monday1] x[EE05,Monday2] x[EE05,Tuesday1] x[EE05,Tuesday2]
 x[EE05,Wednesday1] x[EE05,Wednesday2] x[EE05,Thursday1] x[EE05,Thursday2]
 x[EE05,Friday1] x[EE05,Friday2] x[EE05,Saturday1] x[EE05,Saturday2]
 x[EE05,Sunday1] x[EE05,Sunday2] x[EE06,Monday1] x[EE06,Monday2]
 x[EE06,Tuesday1] x[EE06,Tuesday2] x[EE06,Wednesday1] x[EE06,Wednesday2]
 x[EE06,Thursday1] x[EE06,Thursday2] x[EE06,Friday1] x[EE06,Friday2]
 x[EE06,Saturday1] x[EE06,Saturday2] x[EE06,Sunday1] x[EE06,Sunday2]
 x[EE07,Monday1] x[EE07,Monday2] x[EE07,Tuesday1] x[EE07,Tuesday2]
 x[EE07,Wednesday1] x[EE07,Wednesday2] x[EE07,Thursday1] x[EE07,Thursday2]
 x[EE07,Friday1] x[EE07,Friday2] x[EE07,Saturday1] x[EE07,Saturday2]
 x[EE07,Sunday1] x[EE07,Sunday2] x[EE08,Monday1] x[EE08,Monday2]
 x[EE08,Tuesday1] x[EE08,Tuesday2] x[EE08,Wednesday1] x[EE08,Wednesday2]
 x[EE08,Thursday1] x[EE08,Thursday2] x[EE08,Friday1] x[EE08,Friday2]
 x[EE08,Saturday1] x[EE08,Saturday2] x[EE08,Sunday1] x[EE08,Sunday2]
 x[EE09,Monday1] x[EE09,Monday2] x[EE09,Tuesday1] x[EE09,Tuesday2]
 x[EE09,Wednesday1] x[EE09,Wednesday2] x[EE09,Thursday1] x[EE09,Thursday2]
 x[EE09,Friday1] x[EE09,Friday2] x[EE09,Saturday1] x[EE09,Saturday2]
 x[EE09,Sunday1] x[EE09,Sunday2] x[EE10,Monday1] x[EE10,Monday2]
 x[EE10,Tuesday1] x[EE10,Tuesday2] x[EE10,Wednesday1] x[EE10,Wednesday2]
 x[EE10,Thursday1] x[EE10,Thursday2] x[EE10,Friday1] x[EE10,Friday2]
 x[EE10,Saturday1] x[EE10,Saturday2] x[EE10,Sunday1] x[EE10,Sunday2]
 OT_Trigger[EE01] OT_Trigger[EE02] OT_Trigger[EE03] OT_Trigger[EE04]
 OT_Trigger[EE05] OT_Trigger[EE06] OT_Trigger[EE07] OT_Trigger[EE08]
 OT_Trigger[EE09] OT_Trigger[EE10]
End
```

4.3 Solving the optimization

```
model.optimize()
Gurobi Optimizer version 9.0.2 build v9.0.2rc0 (win64)
Optimize a model with 78 rows, 170 columns and 666 nonzeros
Model fingerprint: 0x06cfc34c
Variable types: 20 continuous, 150 integer (150 binary)

Coefficient statistics:
  Matrix range     [2e-01, 1e+00]
  Objective range  [1e+02, 3e+02]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 7e+00]
Found heuristic solution: objective 12660.000000
Presolve removed 20 rows and 13 columns
Presolve time: 0.12s
Presolved: 58 rows, 157 columns, 623 nonzeros
Variable types: 20 continuous, 137 integer (137 binary)
Root relaxation: objective 7.535000e+03, 106 iterations, 0.06 seconds
    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node
Time
*    0     0               0    7535.0000000 7535.00000  0.00%     -    0s
Explored 0 nodes (106 simplex iterations) in 0.54 seconds
Thread count was 2 (of 2 available processors)
Solution count 2: 7535 12660
Optimal solution found (tolerance 1.00e-04)
Best objective 7.535000000000e+03, best bound 7.535000000000e+03, gap
0.0000%
```

5. RESULT

All codes are written in jupyter notebook. The result mention below is totally jupyter notebook oriented.

Outputs

Total cost of the proposed schedule

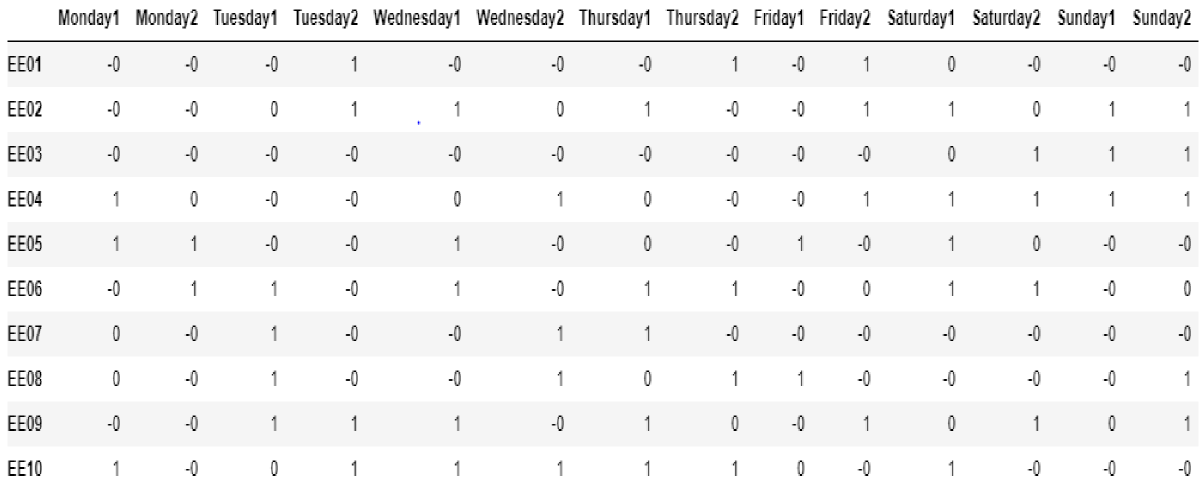
```
print('Total cost = $' + str(model.ObjVal))
Total cost = $7535.0
```

Dashboard of the schedule

```
sol = pd.DataFrame(data={'Solution':model.X}, index=model.VarName)
sol = sol.iloc[0:len(x)]

dashboard = pd.DataFrame(index = workerList, columns = shiftList)
for w in workerList:
    for s in shiftList:
        dashboard.at[w,s] = sol.loc['x['+w+','+s+']',][0]

dashboard
```



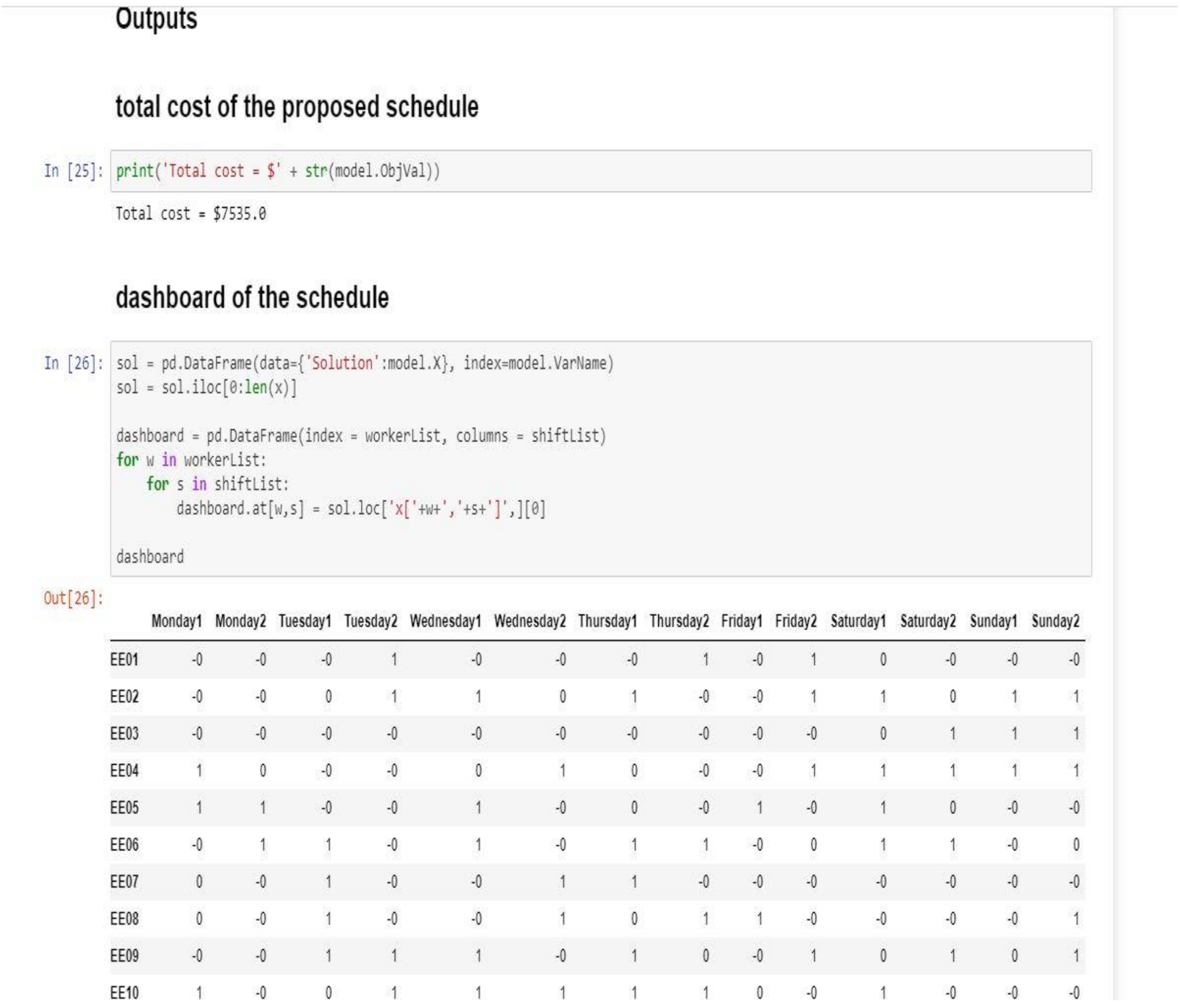
	Monday1	Monday2	Tuesday1	Tuesday2	Wednesday1	Wednesday2	Thursday1	Thursday2	Friday1	Friday2	Saturday1	Saturday2	Sunday1	Sunday2
EE01	-0	-0	-0	1	-0	-0	-0	1	-0	1	0	-0	-0	-0
EE02	-0	-0	0	1	1	0	1	-0	-0	1	1	0	1	1
EE03	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	0	1	1	1
EE04	1	0	-0	-0	0	1	0	-0	-0	1	1	1	1	1
EE05	1	1	-0	-0	1	-0	0	-0	1	-0	1	0	-0	-0
EE06	-0	1	1	-0	1	-0	1	1	-0	0	1	1	-0	0
EE07	0	-0	1	-0	-0	1	1	-0	-0	-0	-0	-0	-0	-0
EE08	0	-0	1	-0	-0	1	0	1	1	-0	-0	-0	-0	1
EE09	-0	-0	1	1	1	-0	1	0	-0	1	0	1	0	1
EE10	1	-0	0	1	1	1	1	1	0	-0	1	-0	-0	-0

Figure 18 - Output of the dashboard

An alternative view of the dashboard

```
shiftAssignments = {}
for s in shiftList:
    shiftAssignments.update({s: list(dashboard[dashboard[s] ==
1].loc[:,].index)})
shiftAssignments
```

```
{'Monday1': ['EE04', 'EE05', 'EE10'],
 'Monday2': ['EE05', 'EE06'],
 'Tuesday1': ['EE06', 'EE07', 'EE08', 'EE09'],
 'Tuesday2': ['EE01', 'EE02', 'EE09', 'EE10'],
 'Wednesday1': ['EE02', 'EE05', 'EE06', 'EE09', 'EE10'],
 'Wednesday2': ['EE04', 'EE07', 'EE08', 'EE10'],
 'Thursday1': ['EE02', 'EE06', 'EE07', 'EE09', 'EE10'],
 'Thursday2': ['EE01', 'EE06', 'EE08', 'EE10'],
 'Friday1': ['EE05', 'EE08'],
 'Friday2': ['EE01', 'EE02', 'EE04', 'EE09'],
 'Saturday1': ['EE02', 'EE04', 'EE05', 'EE06', 'EE10'],
 'Saturday2': ['EE03', 'EE04', 'EE06', 'EE09'],
 'Sunday1': ['EE02', 'EE03', 'EE04'],
 'Sunday2': ['EE02', 'EE03', 'EE04', 'EE08', 'EE09']}
```



Outputs

total cost of the proposed schedule

```
In [25]: print('Total cost = $' + str(model.ObjVal))
```

```
Total cost = $7535.0
```

dashboard of the schedule

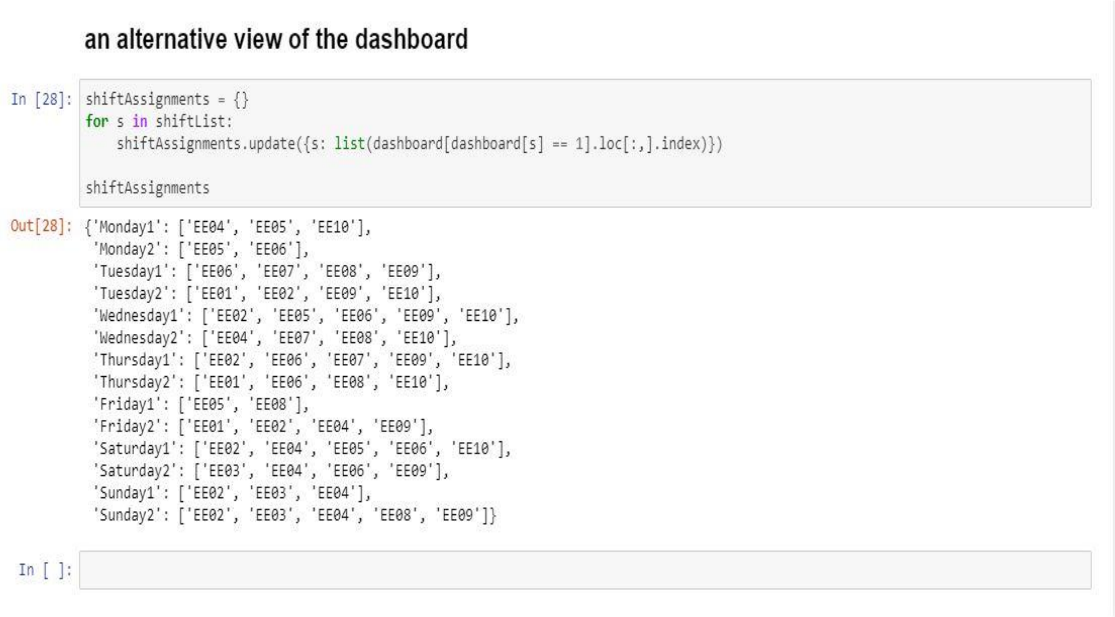
```
In [26]: sol = pd.DataFrame(data={'Solution':model.X}, index=model.VarName)
         sol = sol.iloc[0:len(x)]

         dashboard = pd.DataFrame(index = workerList, columns = shiftList)
         for w in workerList:
             for s in shiftList:
                 dashboard.at[w,s] = sol.loc['x['+w+','+s+']',][0]

         dashboard
```

Out[26]:

	Monday1	Monday2	Tuesday1	Tuesday2	Wednesday1	Wednesday2	Thursday1	Thursday2	Friday1	Friday2	Saturday1	Saturday2	Sunday1	Sunday2
EE01	-0	-0	-0	1	-0	-0	-0	1	-0	1	0	-0	-0	-0
EE02	-0	-0	0	1	1	0	1	-0	-0	1	1	0	1	1
EE03	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	0	1	1	1
EE04	1	0	-0	-0	0	1	0	-0	-0	1	1	1	1	1
EE05	1	1	-0	-0	1	-0	0	-0	1	-0	1	0	-0	-0
EE06	-0	1	1	-0	1	-0	1	1	-0	0	1	1	-0	0
EE07	0	-0	1	-0	-0	1	1	-0	-0	-0	-0	-0	-0	-0
EE08	0	-0	1	-0	-0	1	0	1	1	-0	-0	-0	-0	1
EE09	-0	-0	1	1	1	-0	1	0	-0	1	0	1	0	1
EE10	1	-0	0	1	1	1	1	1	0	-0	1	-0	-0	-0



an alternative view of the dashboard

```
In [28]: shiftAssignments = {}
         for s in shiftList:
             shiftAssignments.update({s: list(dashboard[dashboard[s] == 1].loc[:,].index)})

         shiftAssignments
```

```
Out[28]: {'Monday1': ['EE04', 'EE05', 'EE10'],
          'Monday2': ['EE05', 'EE06'],
          'Tuesday1': ['EE06', 'EE07', 'EE08', 'EE09'],
          'Tuesday2': ['EE01', 'EE02', 'EE09', 'EE10'],
          'Wednesday1': ['EE02', 'EE05', 'EE06', 'EE09', 'EE10'],
          'Wednesday2': ['EE04', 'EE07', 'EE08', 'EE10'],
          'Thursday1': ['EE02', 'EE06', 'EE07', 'EE09', 'EE10'],
          'Thursday2': ['EE01', 'EE06', 'EE08', 'EE10'],
          'Friday1': ['EE05', 'EE08'],
          'Friday2': ['EE01', 'EE02', 'EE04', 'EE09'],
          'Saturday1': ['EE02', 'EE04', 'EE05', 'EE06', 'EE10'],
          'Saturday2': ['EE03', 'EE04', 'EE06', 'EE09'],
          'Sunday1': ['EE02', 'EE03', 'EE04'],
          'Sunday2': ['EE02', 'EE03', 'EE04', 'EE08', 'EE09']}
```

```
In [ ]:
```

6. **CONCLUSION**

On the market there are a lot of different software program products that can be used for planning and scheduling purpose. Such products are:

- EnterpriseIIQ Scheduling;
- Synchro PRO;
- Resourse Manager DB;
- SMARTer Manager;
- Orchestrate Scheduling;
- Planvisage SCM;

Presented modern management methods for planning and scheduling of production processes are reliable, easy and allow quickly dealing with certain problems. Scheduling of the company's orders or Time-Table can also be design using simple Python Language. In this paper, I worked on Python Programming using Jupyter Notebook and able to design a working schedule for the workshop and also optimise the cost of schedule using gurobi package by considering both regular cost and overtime cost. The program runs smoothly and i got the desired result. Throughout this project I used some scheduling software and the same work is done by simple Python programming. Since, depending on the type of workload scheduling needs changes. Some of the method which are used by schedulers are :-

- Excel – cheap way to manage a production but hard to use and needs knowledge about the formulas;

- Lekin Software – easy to use but useful only for one order and has a lack of optimization algorithms;

- Python Programming – a lot of good functions and reliable product. The product has the possibility to generate different optimization variations.

The use of "Advanced planning and scheduling" management systems and proper selection of software, significantly reducing the time required for production planning and scheduling, and enables the prediction of potential production problems and their quickly solving. But for smaller scale production, we can use Python Programming to design schedule and other optimisation for efficient result.

5. REFERENCE:

1. “Taravatsadat Nehzati” , “Napsiah Ismail”;” Application of Artificial Intelligent in Production Scheduling: a critical evaluation and comparison of key approaches”; “International Conference on Industrial Engineering and Operations Management”.
2. “Karl Kempf”, “Bruce Russell”, “Sanjiv Sidhu”, and “Stu Barrett” ; “AI-Based Schedulers in Manufacturing Practice: Report of a Panel Discussion” ; “AI Magazine Volume 11 Number 5 (1990)”.
3. “B. Rodriguez-Somoza”, “R. Gahin” and “E. A. Puente” ; “PRODUCTION SCHEDULING USING AI TECHNIQUES” ; “IFAC Information Control Problems in Manufacturing Technology”.
4. “Sumit Das” , “Aritra Dey” , “Akash Pal” , “Nabamita Roy” ; “Applications of Artificial Intelligence in Machine Learning: Review and Prospect” ; “International Journal of Computer Applications (0975 – 8887) Volume 115 – No. 9, April 2015” .
5. “Jahanzaib Shabbir”, “Tarique Anwer”; “Artificial Intelligence and its Role in Near Future” ; “JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015”.
6. “FABRIZIO SEBASTIANI” ; “Machine Learning in Automated Text Categorization” ; “Consiglio Nazionale delle Ricerche, Italy” .
7. “Ayon Dey”; “Machine Learning Algorithms: A Review” ; “International Journal of Computer Science and Information Technologies, Vol. 7 (3) , 2016, 1174-1179” .
8. “Kajaree Das” , “Rabi Narayan Behera” ; “A Survey on Machine Learning: Concept, Algorithms and Applications” ; “International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 2, February 2017”.
9. “Iqbal Muhammad” and “Zhu Yan”; “SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY” ; “SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY DOI: 10.21917/ijsc.2015.0133” .
10. “Sreelekha Panda” ; “Impact of AI in Manufacturing Industries” ; “International Research Journal of Engineering and Technology” .
11. “Thorsten Wuesta” , “Daniel Weimerb”, “Christopher Irgensc” and “Klaus-Dieter Thobend” ; “Machine learning in manufacturing:

advantages, challenges, and applications" ; "Production & Manufacturing Research: An Open Access Journal, 2016 VOL. 4, NO. 1, 23–45" .

12. "D T Pham" and "A A Afify" ; "Machine-learning techniques and their applications in manufacturing" ; "Manufacturing Engineering Centre, Cardiff University, Cardiff, UK" .
13. "Imran Ali Chaudhry"," Muiz uddin Sharni" and "Abid Ali Khan" ; "MANUFACTURING APPLICATIONS OF ARTIFICIAL INTELLIGENCE" ; "1. eng. & app l. sc i. Vol. 23 No .2 July - Dec. 2004 ISSN 102 3-862".
14. "Bin He" ; "Artificial Intelligent for Intelligent Manufacturing and Robotics" : "Robotics & Automation Engineering journal (RAEJ)" .
15. "Farid Meziane" , "Sunil Vadera", "Khiary Kobbacy" and "Nathan Proudlove" ; "Intelligent Systems in Manufacturing: Current Developments and Future Prospects" ; "Integrated Manufacturing Systems (currently published as: Journal of Manufacturing Technology Management)" .
16. "Annina Simon" , "Mahima Singh Deo" , "S. Venkatesan ", "D.R. Ramesh Babu" ; "An Overview of Machine Learning and its Applications" ; "International Journal of Electrical Sciences & Engineering (IJESE) Volume 1, Issue 1; 2015 pp. 22-24" .
17. "Wayne J. Davis" , "Albert T. Jones" ; "Artificial Intelligence Techniques in Real-Time Production Scheduling" ; "NISTIR 88-3891 Issued February 1989" .
18. "Jay Lee", "Jaskaran Singh" , "Moslem Azamfar" ; "Industrial Artificial Intelligence" ; "Department of Mechanical Engineering, University of Cincinnati, Cincinnati, USA" .
19. "Chuda Basnet", "Joe H. Mize" ; "Scheduling and Control of Flexible Manufacturing Systems: A Critical Review" ; "Department of Management Systems University of Waikato ,Hamilton, New Zealand and School of Industrial Engineering and Management Oklahoma State University, Stillwater, U.S.A." .
20. "Shirin Sohrabi"; "AI Planning for Enterprise: Putting Theory Into Practice"; "Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)" .

21. “Mark S. Fox” ; “Industrial Applications of Artificial Intelligence” ; “Intelligent Systems Laboratory, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213, USA” .
22. “IVANA SIMEONOVOVA” , “ALEKSANDAR GYUROV”, “SIMEON SIMEONOV” ; “MODERN METHODS FOR MANUFACTURING PLANNING AND SCHEDULING” ; “DOI: 10.17973/MMSJ.2015_10_201519” .