Automatic Scheduling for AGV-based Digital Manufacturing Platforms (DMPs) using Digital Twins

Submitted by

Zhou Yuanyuan & Shi Xuan

Sichuan University & Soochow University

Supervisor:

ASSOC. PROF. VEERAVALLI, BHARADWAJ

Examiner：

DR ARTHUR TAY

The final year project work was carried out under the 3+1+1 Educational Framework at The National University of Singapore (Suzhou) Research Institute

December 2022

ABSTRACT

As technology redefines the manufacturing industry, it has become essential to use digital manufacturing platforms（DMP） to maximize operational efficiency and accelerate Industry 4.0. The introduction of AGV has further improved the efficiency of the intelligent logistics system. In practical production, scheduling is an essential part of optimizing the whole chain.

For product processing production, job shop scheduling and flow shop scheduling are mainly used, and the system performance is evaluated through indicators such as blockage rate and machine occupancy rate; the concept of machine learning is introduced on the basis of scheduling, and the digital twin model is used to realize the prediction of the scheduling system and the allocation of resources. The final result is an efficient intelligent manufacturing system.

Index Terms: job shop scheduling; flow shop scheduling; machine learning; digital twin

LIST OF FIGURES

CONTENTS

1. INTRODUCTION

1.1 AGV-systems

AGVs were conceived in 1954 by Barrett Electronics Corporation, Illinois, a manufacturing company in the United States. Since then AGVs became widespread all over the globe in every logistics environment where goods are transported. Currently, the market of AGVs is growing fast and is very dynamic. the future growth of AGV systems is caused by the emergence of flexible manufacturing systems, the rising demand for customized AGVs and the adoption of industrial automation by SMEs. The growth rate of AGVs in this forecast period will be the highest in Europe, especially in e-commerce and material handling. The current AGV-systems are well known and widely implemented in manufacturing, medicine, and logistics.[1]

As technology redefines the manufacturing industry, it has become essential to use digital manufacturing platforms to maximize operational efficiency and accelerate Industry 4.0. Currently, the rise of digital platforms for manufacturing is a reality as they play a key role in supporting collaborative manufacturing, service, analysis, and forecasting processes in business networks. Moreover, they provide flexibility to enterprises by fast and simple orchestration of services and applications.[2]

1.2 Digital Twins

A digital twin is a digital representation of an intended or actual real-world physical product, system, or process (a physical twin) that serves as the effectively indistinguishable digital counterpart of it for practical purposes, such as simulation, integration, testing, monitoring, and maintenance.

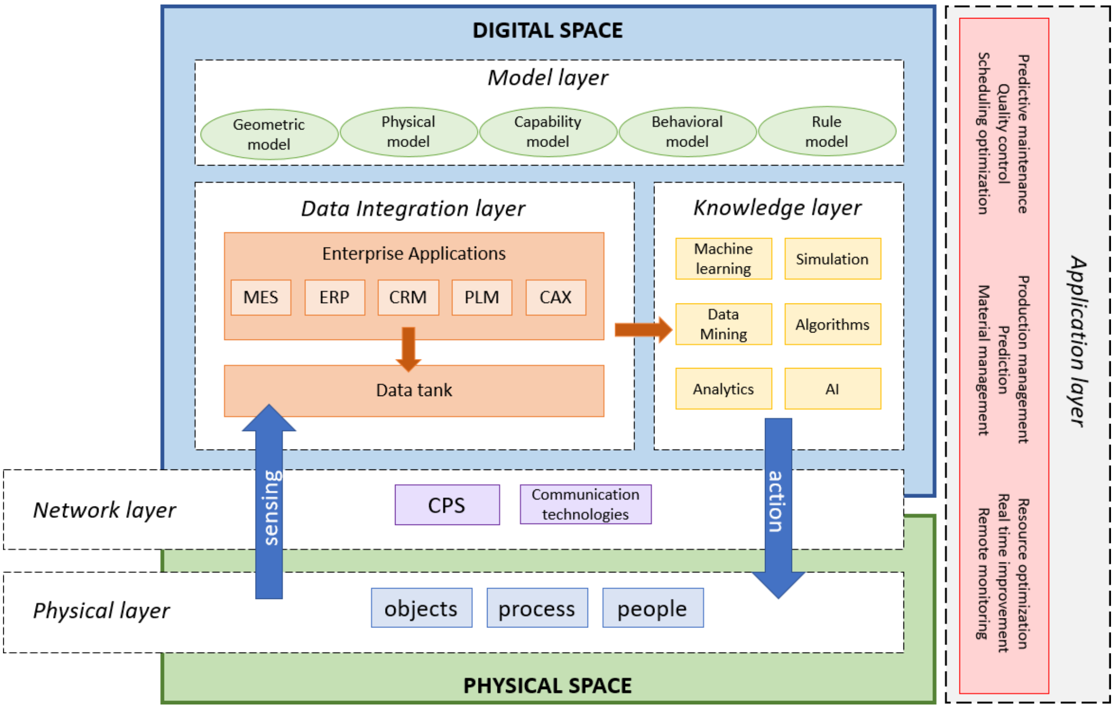


Fig 1 Digital Twin Model

The digital twin has been intended from its initial introduction to be the underlying premise for Product Lifecycle Management[3] and exists throughout the entire lifecycle, create, build, operate/support, and dispose, of the physical entity it represents. Since information is granular, the digital twin representation is determined by the value based use cases it is created to implement. The digital twin can and does often exist before there is a physical entity. The use of a digital twin in the create phase allows the intended entity's entire lifecycle to be modeled and simulated.[4] A digital twin of an existing entity can, but must not necessarily, be used in real time and regularly synchronized with the corresponding physical system.

1.3 Simulation Platform

SimPy is a process-based discrete-event simulation framework based on standard Python. SimPy is released as open source software under the MIT License. The first version was released in December 2002.

Its event dispatcher is based on Python's generators and can also be used for asynchronous networking or to implement multi-agent systems. Simulations can be performed “as fast as possible”, in real time or by manually stepping through the events.

Processes in SimPy are simple Python generator functions and are used to model active components like customers, vehicles or agents. SimPy also provides various types of shared resources to model limited capacity congestion points (like servers, checkout counters and tunnels). From version 3.1, it will also provide monitoring capabilities to aid in gathering statistics about resources and processes.

2. BACKGROUND

2.1 Related Works

(a)Yilin Fang et al. [4] proposed a job shop scheduling method based on digital twin, which can realize virtual reality interaction, real time mapping and symbiotic evolution. They also presented a Dynamic Interactive Scheduling Strategy to help the workshop establish a corresponding emergency response mechanism. This mechanism can obtain the dynamic disturbance occurring in physical space, and generate a new scheduling to adapt to the latest production environment through analysis of the specific types of dynamic events.

(b) Angelo Corallo et al. [4] proposed a comprehensive framework that integrates all the main components of both physical and digital space and describes their relationships, called the “hexadimensional shop floor digital twin” (HexaSFDT). The framework can help model the concept of a real scenario by reducing its complexity, allow the definition of data flows from the physical to the virtual spaces and vice-versa, and help stimulate the identification of all the elements in a unique reference framework.

(c) Cunbo Zhuang et al. [4] proposed a framework of smart production management and control approach for complex product assembly shop-floors based on digital twin. The framework can help realize(1) real-time acquisition, organization, and management of the physical assembly shop-floor data, (2) construction of the assembly shop-floor digital twin, (3) digital twin and big data-driven prediction of the assembly shop-floor, and (4) digital twin-based assembly shop-floor production management and control service.

(c) Ronald J. Mantel et al. [4] presented a method to realize the design and operational control of Automated Guided Vehicle (AGV) systems. The paper has solved three main problems when designing the systems: the track layout, the number of AGVs required and the operational transportation control.

(d) Francisco Fraile et al.[4] proposed an integrated reference model for digital manufacturing platforms. The platforms combine the Industrial Internet of Things (IoT) systems with other value-added services to suit different manufacturing processes.

2.2 Job Shop Scheduling and Flow Shop Scheduling

Scheduling means allocating the shared resources to competing activities over a period of time. The focus is on investigating machine scheduling problems. The jobs and machines represent the activities and the shared resources separately.

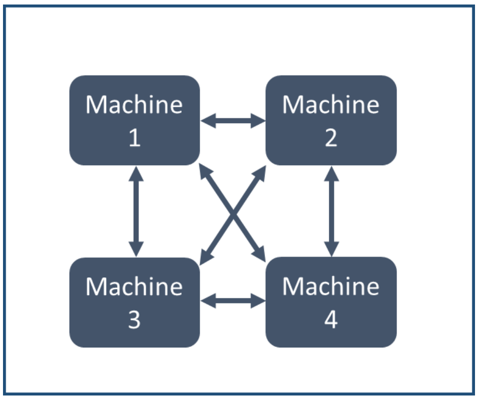


Fig 2 Jobshop Scheduling

The job shop scheduling problem (JSSP for short) deals with such a problem, that is, we determine the order or sequence for processing a group of jobs through multiple machines in the best way. More specifically, there is a set of jobs $J = \{J_1, \ldots, J_n\}$ which should be processed by a group of m machines. Every job from J has to go through a fixed sequence of machines to be processed. Job J_i consists of an ordered sequence of operations O_{1j}, O_{2j},..., $O_{k_j j}$. During a given time p_{ij}, a specific machine m_{ij} must process operation O_{ij}. Each machine can process only one operation and one job at most one at a time. The maximum completion time of all the jobs is called the length or makespan of the schedule. The problem is to schedule the jobs to minimize the time C_{max}.

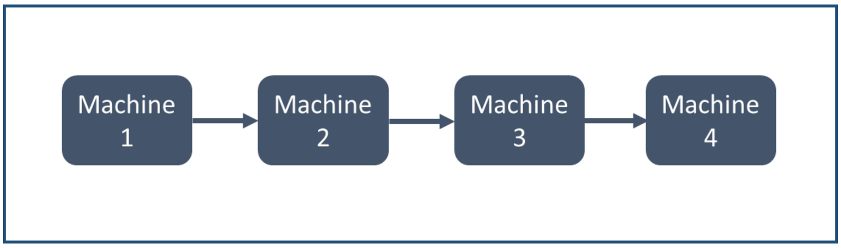


Fig 3 Flowshop Scheduling

Flow shop scheduling is a special case of job shop scheduling. Different from the job shop scheduling, when dealing with flow shop scheduling problem, operations within one job must be processed in a given order. This means the first operation gets executed on the first machine, and when the first operation has been finished, the second operation starts to be processed on the second machine, and so on until all the given operations have been done. The problem is also to schedule the jobs to minimize the makespan.

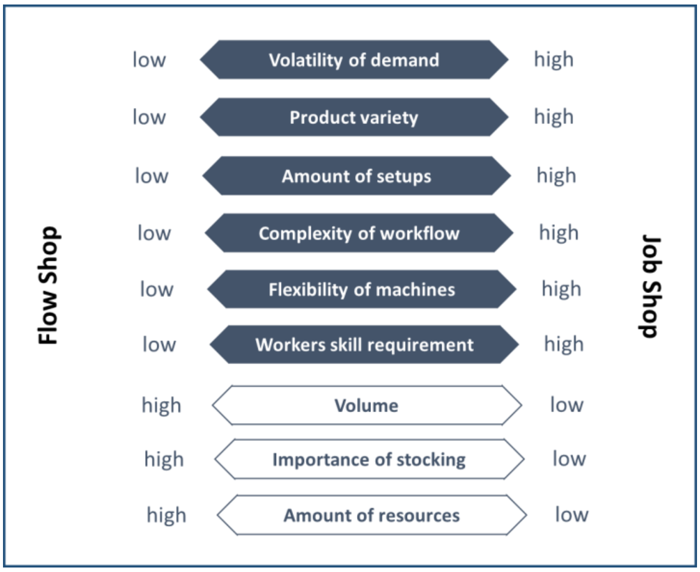


Fig 4 Comparation between Jobshop and Flowshop

In terms of a flow shop, this would be a manufacturing site with 100% standardization operated in an assembly production line. On the other hand, a job shop would be a business with 100% customization with a typical batch size of 1, which implies that every finished product is unique.

2.3 Digital Twin Model

The increasing production intricacy in a more demanding market is gathering momentum for the integration of the physical and digital world. At the same time, human's escalating practical needs for industrial products are challenging the digital model's capability to interact with the physical object. The digital twin was conceived in this context and has sparked a far-reaching industry revolution. In the beginning, digital twins were employed primarily in the military and aerospace. [4] Currently, the digital twin is in a period of rapid development, which has already progressed from theoretical research to pragmatic implementation and has been used in various fields

Digital Twin is now considered as one of the enabling technologies of Industry 4.0, which has played a significant role in in many different industries. Through using model, sensors, data and software, it can couple the physical system with the virtual representation to monitor and analyze data. Even if there are changes in its working environment such as weather and the input products, digital twin can quickly adapt to it and take these new data into account with the help of the support enabling technologies like machine learning. So in order to supervise the production system of the industry in real time, deal with the problems in time, and obtain the optimal production line, it is of great importance to apply the digital world and the digital twin technology.

3. DESIGN AND PERFORMANCE

3.1 Simulated Scenarios

In this semester, we completed the simulation of the following three scenarios, whose requirements are as follows.

3.1.1. Scenario A

There are three different types of products in one line, labelled 0, 1 and 2. The corresponding products are fed into the corresponding machines 0, 1 and 2 for processing, and upon completion the three types of products are output in the three lines.

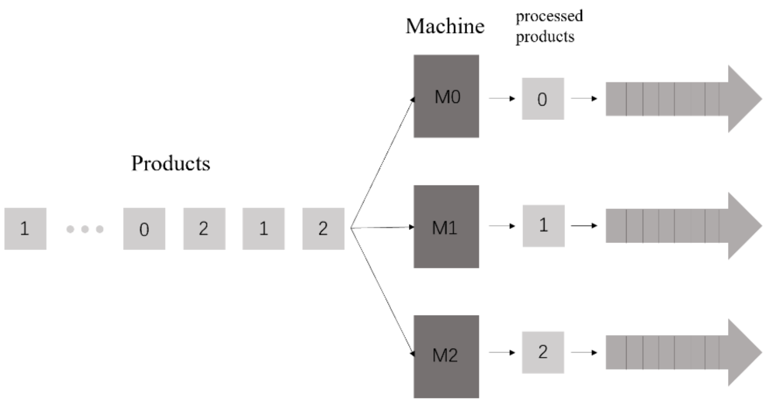


Fig 5 Schematic diagram of Scenario A

3.1.2. Scenario B

There are three different types of products in one line, labelled 0, 1 and 2. Each workpiece needs to go through several machines to complete the corresponding processing steps. e.g. product 0 needs machine 0 and machine 1 to complete the processing, product 1 needs machines 2, 3 and 4 to process, product 2 needs machines 5, 6, 7 and 8 to process. And three types of products are output in the three lines respectively.

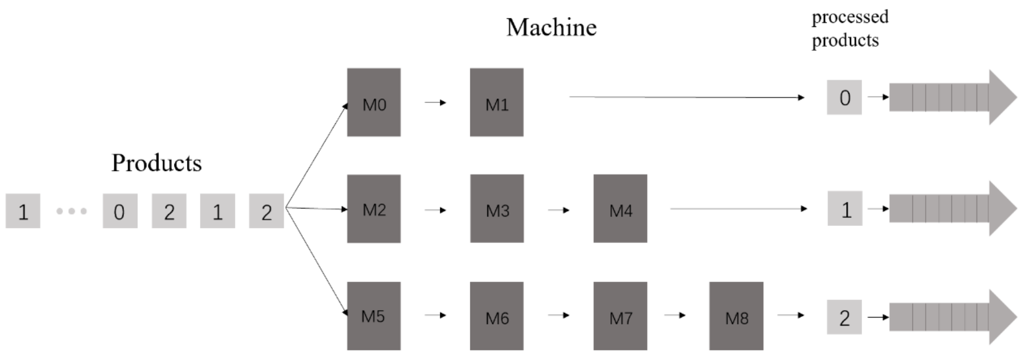


Fig 6 Schematic diagram of Scenario B

3.1.3. Scenario C

In this scenario there are several products, each with a number of processing steps, and the sequences of products are all randomly generated in terms of their labels, the labels and the number of machines performing the processing, so that some scheduling algorithms are required during the production process, which is the basis for the scheduling of digital manufacturing platforms.

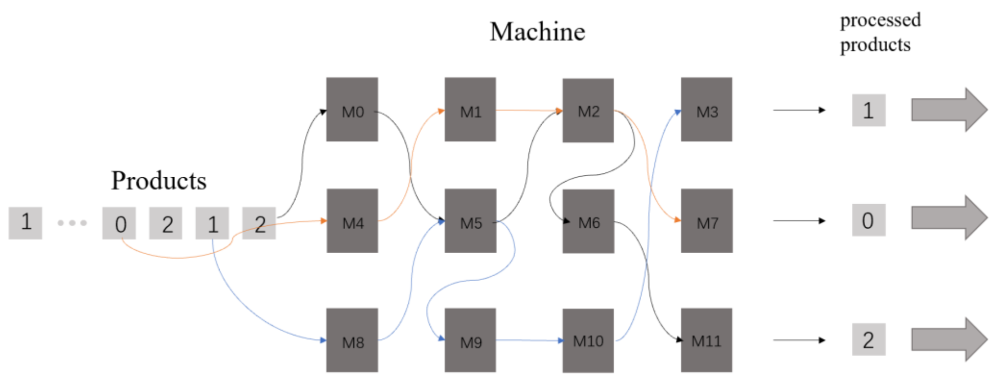


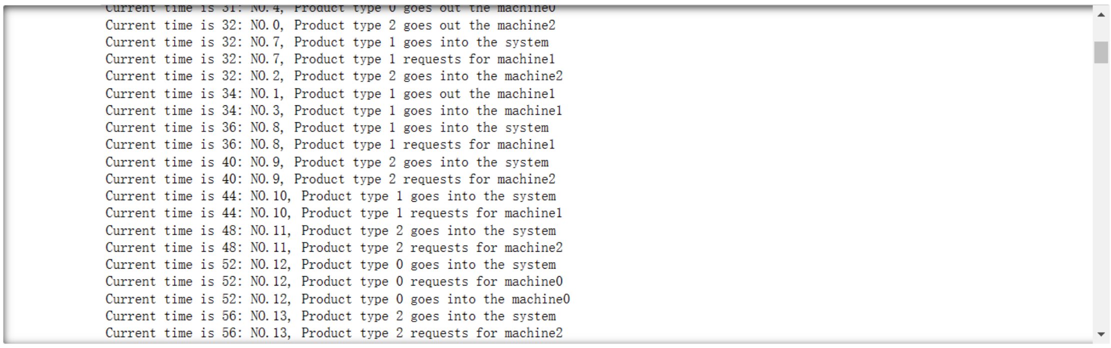
Fig 7 Schematic diagram of Scenario C

In practice, two implementations of this scenario are considered, with small batch production usually using job shop scheduling and large-scale production using flow shop scheduling.

3.2 Outcomes

3.2.1 Scenario A

The outcome of the simulation of scenario A is as follows.



```
Current time is 31: NO.4, Product type 0 goes out the machine0
Current time is 32: NO.0, Product type 2 goes out the machine2
Current time is 32: NO.7, Product type 1 goes into the system
Current time is 32: NO.7, Product type 1 requests for machine1
Current time is 32: NO.2, Product type 2 goes into the machine2
Current time is 34: NO.1, Product type 1 goes out the machine1
Current time is 34: NO.3, Product type 1 goes into the machine1
Current time is 36: NO.8, Product type 1 goes into the system
Current time is 36: NO.8, Product type 1 requests for machine1
Current time is 40: NO.9, Product type 2 goes into the system
Current time is 40: NO.9, Product type 2 requests for machine2
Current time is 44: NO.10, Product type 1 goes into the system
Current time is 44: NO.10, Product type 1 requests for machine1
Current time is 48: NO.11, Product type 2 goes into the system
Current time is 48: NO.11, Product type 2 requests for machine2
Current time is 52: NO.12, Product type 0 goes into the system
Current time is 52: NO.12, Product type 0 requests for machine0
Current time is 52: NO.12, Product type 0 goes into the machine0
Current time is 56: NO.13, Product type 2 goes into the system
Current time is 56: NO.13, Product type 2 requests for machine2
```

Fig 8 Outcome of Scenario A

As we can see, the classified products find the corresponding machines for processing according to their types. And if the machine a product request is processing another product, the product must wait until the machine is released.

3.2.2 Scenario B

The outcome of the simulation of scenario B is as follows.

```
Current time is 201: NO.4, Product type 1 goes out of the system
Current time is 216: NO.7, Product type 2 goes into the system
Current time is 216: NO.7, Product type 2 is added into shop 2
Current time is 236: NO.8, Product type 1 goes into the system
Current time is 236: NO.8, Product type 1 is added into shop 1
Current time is 236: NO.8, Product type 1 goes into the shop 1
Current time is 250: NO.5, Product type 2 goes out of the shop 2
Current time is 250: NO.5, Product type 2 goes out of the system
Current time is 250: NO.6, Product type 2 goes into the shop 2
Current time is 260: NO.9, Product type 2 goes into the system
Current time is 260: NO.9, Product type 2 is added into shop 2
Current time is 290: NO.10, Product type 1 goes into the system
Current time is 290: NO.10, Product type 1 is added into shop 1
Current time is 304: NO.8, Product type 1 goes out of the shop 1
Current time is 304: NO.8, Product type 1 goes out of the system
Current time is 304: NO.10, Product type 1 goes into the shop 1
Current time is 310: NO.11, Product type 2 goes into the system
Current time is 310: NO.11, Product type 2 is added into shop 2
Current time is 319: NO.6, Product type 2 goes out of the shop 2
Current time is 319: NO.6, Product type 2 goes out of the system
```

Fig 9 Outcome of Scenario B

As we can see, the classified products find the corresponding shops for processing according to their types, and each shop contains several machines respectively.

3.2.3 Scenario C

3.2.4 Input Sequence model

1) Machine sequence:

Randomly generates the number of machine types between (10, 20) and label all machines generated starting from 0.

```
The total types of machines:
 13
labels for each machine:
 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

Fig 10 Machine sequence

2) Product sequence:

Randomly generates the number of product types between (15, 25) and label all products generated starting from 0.

```
The total types of products:
 21
```

Fig 11 Product number

Randomly generates the number of steps required to produce each product, and the corresponding machine number for each step.

```
Numbers of machines for each type of product:
 [5, 4, 3, 5, 4, 5, 3, 5, 4, 4, 4, 4, 5, 4, 3, 5, 3, 4, 3, 5, 3]
```

Fig 12 Product's machine number

Using a two-dimensional array to record the length of the sequence, the class of machines through which each product passes.

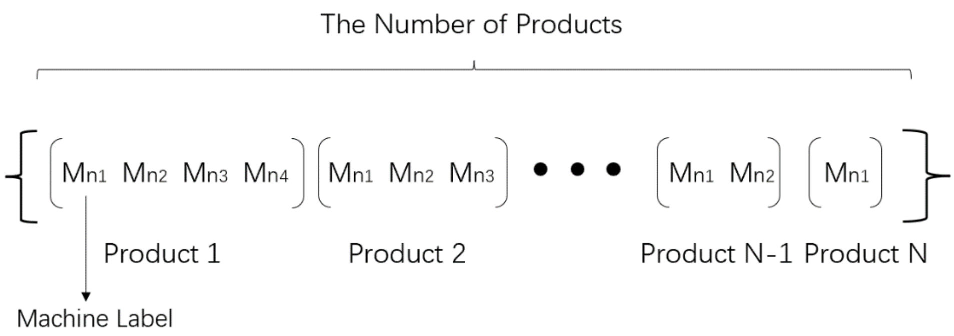


Fig 13 The structure of two-dimensional array

```
Producting Sequence:
[[3, 5, 1, 12, 8], [10, 8, 3, 11], [0, 6, 2], [3, 11, 10, 2, 4], [6,
9, 11, 3], [4, 6, 7, 10, 2], [6, 10, 5], [1, 12, 7, 5, 8], [8, 11, 0,
9], [2, 7, 9, 5], [11, 8, 10, 6], [11, 4, 6, 2], [2, 7, 8, 9, 11], [1
0, 9, 6, 0], [9, 2, 8], [9, 2, 7, 4, 0], [9, 0, 1], [6, 10, 8, 4],
[7, 10, 4], [8, 2, 9, 3, 12], [6, 5, 2]]
```

Fig 14 Product sequence

The figure above shows that the procedures of processing each product, eg. the first array shows the production process of product1, it needs to go through the machine3, machine4, machine1 and machine9 to finish the production processing.

3) Input sequence:

Set up five production lines and randomly generate the length and the type of the product sequence for each line.

```
The input production sequences for each line:
S[0]= [11, 18, 11, 8, 7, 2, 17, 11, 15, 20, 20, 5, 7, 3, 6, 4, 10, 11, 19, 7, 6, 10, 1, 3, 5, 4, 14, 6, 19, 2, 16, 20, 14, 12, 18, 4, 15, 1
7, 7, 14, 8, 18, 13, 15, 9, 13, 8, 17, 9, 9, 14, 19, 0, 11, 0, 18, 8, 12, 10, 8, 2, 9, 5, 6, 6, 19, 18, 1, 4, 8, 17, 6, 5, 1, 18, 20, 11, 2
0, 5, 10, 8, 3, 0, 16, 11, 16, 8, 6, 10, 5, 17, 15, 13, 7, 11, 15, 20, 19, 17, 4, 0, 8, 4, 11, 14, 18, 9, 7, 17, 18, 1, 15, 14, 0, 7, 17, 8,
12, 14, 9, 11, 14, 14, 0, 16, 11, 2, 10, 17, 19, 19, 1, 8, 11, 15, 11, 1, 11, 5, 7, 16]

S[1]= [9, 15, 17, 5, 2, 16, 10, 16, 4, 19, 17, 13, 8, 5, 9, 4, 18, 7, 8, 7, 19, 4, 8, 6, 19, 8, 14, 17, 19, 19, 3, 15, 2, 5, 5, 18, 5, 12,
6, 17, 12, 5, 16, 5, 20, 19, 7, 5, 4, 15, 18, 14, 0, 5, 1, 6, 7, 0, 17, 18, 15, 13, 20, 16, 6, 6, 6, 11, 17, 9, 4, 9, 8, 19, 4, 2, 12, 15, 1
5, 20, 7, 1, 19, 13, 4, 17, 18, 19, 10, 6, 10, 13, 1, 2, 6, 0, 12, 19, 18, 8, 11, 9, 19, 16, 4, 15, 12, 2, 9, 19, 1, 17, 9, 0, 8, 16]

S[2]= [10, 8, 5, 3, 8, 14, 3, 14, 15, 17, 12, 6, 1, 2, 17, 2, 1, 3, 13, 12, 15, 19, 13, 12, 2, 3, 18, 10, 10, 11, 3, 10, 3, 0, 12, 5, 10, 1
1, 10, 17, 18, 1, 2, 14, 14, 17, 12, 3, 14, 15, 19, 9, 0, 4, 2, 11, 19, 9, 17, 8, 15, 15, 10, 17, 4, 6, 13, 18, 6, 16, 17, 5, 6, 15, 2, 11,
0, 0, 4, 19, 19, 15, 16, 15, 9, 18, 14, 18, 10, 19, 16, 16, 19, 7, 19, 8, 0, 19, 20, 17, 11, 15, 16, 6, 14, 20, 2, 7, 5, 9, 10, 19, 12, 4,
0, 6, 10, 14, 17, 8, 14, 5, 17, 16, 5, 0, 9, 8, 18, 14, 12, 9, 15, 9, 8, 18, 20, 12, 1, 17, 2, 14, 2, 9, 11, 15]

S[3]= [19, 4, 13, 8, 4, 0, 0, 4, 12, 5, 18, 9, 0, 19, 6, 18, 17, 0, 7, 11, 13, 5, 7, 6, 9, 19, 16, 20, 9, 16, 18, 16, 11, 10, 12, 14, 9, 20,
7, 20, 5, 9, 9, 18, 15, 14, 10, 18, 20, 0, 9, 8, 2, 1, 6, 17, 3, 2, 2, 10, 8, 2, 18, 6, 5, 0, 15, 10, 3, 10, 9, 16, 12, 10, 5, 7, 0, 15, 17,
9, 11, 1, 6, 16, 18, 17, 6, 5, 12, 5, 9, 10, 13, 1, 20, 1, 5, 13, 9, 5, 19, 11, 12, 20, 0, 3, 5, 7, 6]

S[4]= [1, 2, 0, 13, 14, 12, 1, 16, 17, 20, 18, 8, 1, 9, 8, 6, 4, 13, 13, 2, 2, 3, 10, 20, 19, 16, 14, 2, 17, 5, 18, 2, 0, 19, 11, 5, 1, 8, 1
8, 8, 17, 16, 18, 18, 4, 3, 6, 14, 9, 19, 11, 16, 12, 12, 4, 16, 18, 17, 12, 20, 8, 6, 8, 15, 20, 1, 1, 12, 18, 7, 9, 8, 9, 6, 15, 7, 7, 3,
11, 11, 14, 10, 10, 11, 4, 16, 20, 11, 20, 18, 0, 4, 18, 10, 1, 12, 0, 15, 16, 8, 17, 1, 12, 10, 3, 16, 2, 8, 0, 18, 9, 7, 8, 2, 19, 2, 18,
15, 3, 0, 15, 1, 14]
```

Fig 15 Input sequence

3.2.5 Parameter Assignment

1) Processing Time for Each Machine:

   Random generation of processing times for each machine between (10, 30), which means the n^{th} machine spend on one product.

   [17, 24, 26, 17, 21, 12, 13, 23, 28, 29, 27, 29, 20]

   Fig 16 Machine’s processing time

2) Block Time:

   It takes a certain amount of time for the machine to process each product, so when the machine is occupied as the part enters the line, blocking occurs. We use the blocking parameter to keep track of the current machine blocking situation.

block1: track the total amount of products that have been blocked by other products by a certain time

block2: track how many products are blocked by other products at a certain time

3) Current Machine Status:

When the machine is processing a product, then the status of the machine is occupied. This is essential when determining the operational status of the product.

m_occupied: The number of machines that are occupied

m_vacant: The number of machines that are vacant

3.2.6 Simpy Simulation

1) simulation environment:

Create a simpy environment, then allocate relevant resources for the use, here set the number of each type as resources.

```
[<simpy.resources.resource.Resource at 0x28c9a2592e0>,
 <simpy.resources.resource.Resource at 0x28c9a259ac0>,
 <simpy.resources.resource.Resource at 0x28c9a2598e0>,
 <simpy.resources.resource.Resource at 0x28c9a2591f0>,
 <simpy.resources.resource.Resource at 0x28c9a259f70>,
 <simpy.resources.resource.Resource at 0x28c9a259a30>,
 <simpy.resources.resource.Resource at 0x28c9a259d00>,
 <simpy.resources.resource.Resource at 0x28c9a259610>,
 <simpy.resources.resource.Resource at 0x28c9a259310>,
 <simpy.resources.resource.Resource at 0x28c9a259850>,
 <simpy.resources.resource.Resource at 0x28c9a259d90>,
 <simpy.resources.resource.Resource at 0x28c9a259a90>,
 <simpy.resources.resource.Resource at 0x28c9a259d60>]
```

Fig 17 Resource allocation

2) Implementing functions:

The first function (input_sequence( )) is the action to inject the arrival product into the system.

The mutual function (system_process( )) is to receive these input sequences, inject the arriving products into the waiting sequence and feeds them to the appropriate machine for processing when the machine is idle.

The current time slice of the simulation space can be obtained during the run with env.now().

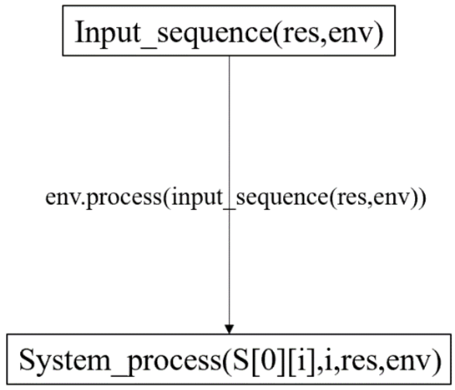


Fig 18 Interaction of functional functions in the simpy environment

3.3 Implement on Scheduling & Evaluation

3.3.1 Job shop Scheduling

Statement: All simulations below are based on data from pipeline S[0]

1) Job shop Core Strategy

A job is characterized by its route, its processing requirements, and its priority. In a job shop the mix of products is a key issue in deciding how and when to schedule jobs. Jobs may not be completed based on their arrival pattern in order to minimize costly machine set-ups and change-overs. Work may also be scheduled based on processing time, from shortest to longest.

A product is processed by several machines and in the machine data stored for the product, the core technology is to prioritize the machines according to the length of its waiting list.

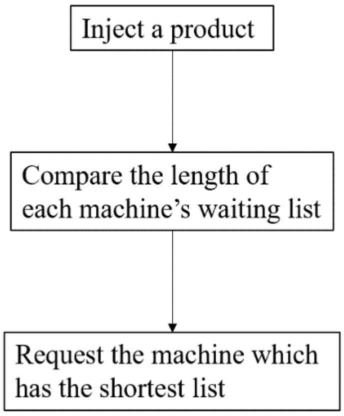


Fig 19 The Core Strategy in Job shop Scheduling

The following diagram provides a concrete illustration of the arrangement of machines running in job shop scheduling

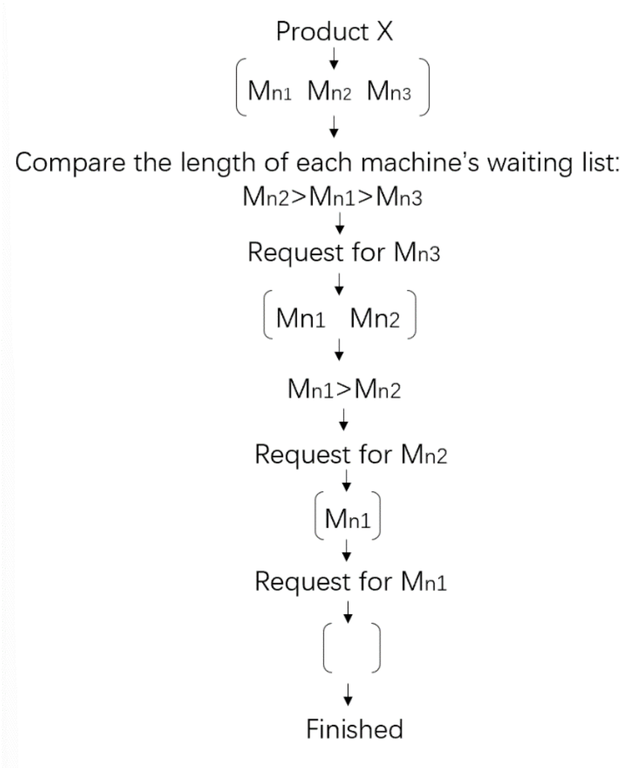


Fig 20 Machine comparative operation mechanism

In the real simulation, when a type 11 product goes in to the system at a certain moment, to complete the processing of this product it requires to go through machine11, machine4, machine 6 and machine2.

a. At this point machine2 has the shortest waiting sequence;

b. After the machining of machine 2 is completed, machine 4 has the shortest waiting sequence of the three remaining machines.

c. After the machining of machine 4 is completed, machine 11 has the shortest waiting sequence of the three remaining machines.

d. The remaining machine 6 carries out the final processing of the product.

This is a simulation of the actual situation.

X [M1, M2, M3, M4] Y: X represents the "Minimum waiting sequence machines", Y represents the number of steps remaining. e.g. 2 [11, 4, 6] 3: The processing of the product by machine2 is carried out/completed and needs to be completed by machine11, machine4 and machine6, where existing 3 machines need to be passed

```
Current time is 557: NO.76, Product type 11 goes into the system
[11, 4, 6, 2] 4
Current time is 557: NO.76, Product type 11 requests for machine2
Current time is 557: NO.76, Product type 11 goes into the machine2

Current time is 583: NO.76, Product type 11 goes out of machine2
2 [11, 4, 6, 2]
2 [11, 4, 6] 3
[11, 4, 6] 3
Current time is 583: NO.76, Product type 11 requests for machine4

Current time is 583: NO.76, Product type 11 goes into the machine4

Current time is 604: NO.76, Product type 11 goes out of machine4
4 [11, 4, 6]
4 [11, 6] 2
[11, 6] 2
Current time is 604: NO.76, Product type 11 requests for machine11
Current time is 604: NO.76, Product type 11 goes into the machine11
```

```
Current time is 633: NO.76, Product type 11 goes out of machine11
11 [11, 6]
11 [6] 1
[6] 1
Current time is 633: NO.76, Product type 11 requests for machine6

Current time is 639: NO.76, Product type 11 goes into the machine6

Current time is 652: NO.76, Product type 11 goes out of machine6
6 [6]
6 [] 0
```

Fig 21 Real simulation example

2) Cumulative Machine Utilization (CMU)

To evaluate the performance of this scheduling system, cumulative machine utilization (CMU) can be used from a machine perspective.

$$Response\ Time = Time\ for\ the\ product\ to\ leave\ the\ machine - Product\ request\ time\ for\ this\ macine$$

$$CMU = \frac{Response\ Time}{Machine's\ Actual\ Processing\ Time}$$

The average Response Time for each machine:

[42.96666666666667, 28.894736842105264, 279.1515151515151, 34.357142857142854, 134.875, 16.393939393939394, 56.61290322580645, 122.58974358974359, 316.0769230769231, 322.03508771929825, 240.6122448979592, 273.265306122449, 27.047619047619047]

Each machine's CMU is:

[2.5274509803921568, 1.2039473684210527, 10.736596736596736, 2.0210084033613445, 6.4226190476190474, 1.3661616161616161, 4.354838709677419, 5.329988851727982 5, 11.288461538461538, 11.10465819721718, 8.91156462585034, 9.422941590429275, 1.3523809523809525]

From the results of the calculation we can see that the value of CMU varies going to a large interval. If a machine is used frequently it can lead to a long waiting interval for that machine, which reduces the efficiency of the whole scheduling system. This means that the smaller the value of the CMU (always greater than 1), the more efficiently the machine is used.

3) Back-up Machine

In order to solve the problem of reducing the overall efficiency of the scheduling system due to different usage frequency, we add some back-up machines to improve the overall efficiency of the system and reduce the CMU value of each type of machine according to the usage time and frequency of different types of machines.

Rules for adding back-up machines:

a. Count the number of times each machine is used in the S[0] queue.
b. Obtain the total running time of each machine, i.e. the frequency of machine use multiplied by the running time of the machine.
c. Get the shortest runtime of all machines, divide the total runtime of each machine by the shortest runtime and round up to the nearest integer to the number of additional spare machines.

Machine used times: [148, 85, 303, 140, 194, 148, 263, 195, 282, 275, 236, 201, 92]

Total processing time for each type of machine: [2516, 2040, 7878, 2380, 4074, 1776, 3419, 4485, 7896, 7975, 6372, 5829, 1840]

Minimum time: 1776

Numbers of machines in each type: [1, 1, 4, 1, 2, 1, 2, 3, 4, 4, 4, 3, 1]

Fig 22 Back-up machine number(job shop)

After adding the back-up machines, we calculate the Response Time and CMU again.

The average Response Time for each machine:

[30.966666666666665, 34.63157894736842, 27.318181818181817, 24.821428571428573, 25.083333333333332, 15.121212121212121, 14.983870967741936, 23.76923076923077, 29.76923076923077, 32.70175438596491, 27.653061224489797, 33.93877551020408, 27.285714285714285]

Each type of machine's CMU is:

[1.8215686274509804, 1.4429824561403508, 1.0506993006993006, 1.4600840336134455, 1.1944444444444444, 1.2601010101010102, 1.152605459057072, 1.0334448160535117, 1.0631868131868132, 1.1276467029643074, 1.0241874527588815, 1.1703026038001407, 1.3642857142857143]

We compare the efficiency of the original machine with the efficiency after adding the back-up machine (CMU is the comparison value).

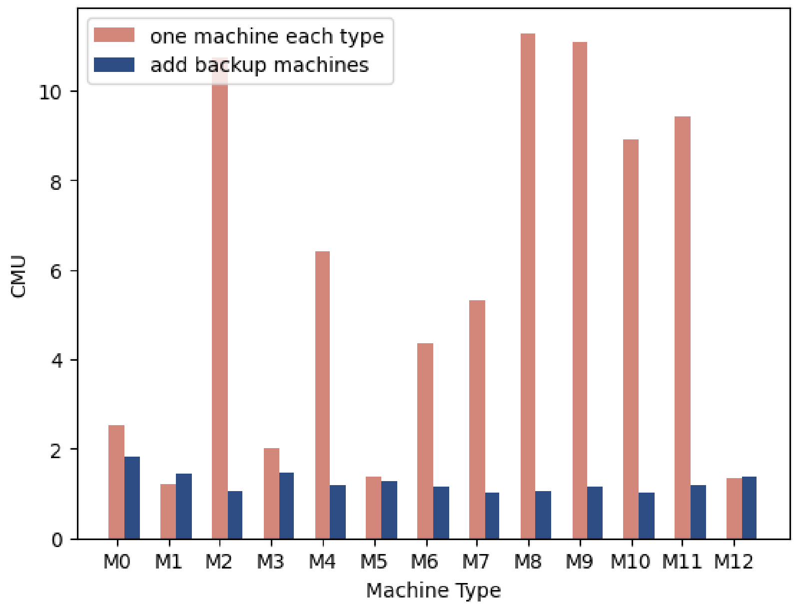


Fig 23 Comparison of machine utilization rates (job shop)

From the bar chart we can see that the average CMU values have been optimized to a large extent and that the introduction of the back machine has had a huge effect on the efficiency of the system.

4) Occupation Rate

Machine efficiency can also be evaluated by the machine's occupation rate, which we discretize in time, using time slices to record the number of machines that are occupied by products.

$$Occupied\ percentage = \frac{The\ number\ of\ machines\ that\ are\ occupied}{Total\ number\ of\ machines}$$

The number of occupied machines and machine occupation ratios are plotted on a scatter plot as shown below.

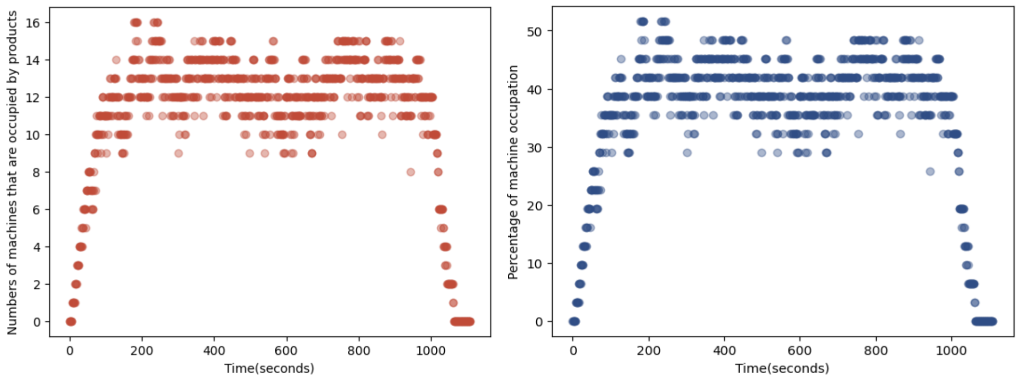


Fig 24 Occupation Rate of machines (job shop)

5) Blocking Rate

The performance of the system can also be assessed by products by tracking the number of blocked products. We use two variables to track product blocking, block1 to record all product blocking throughout the process and block2 to record the number of product blocking at a given time.

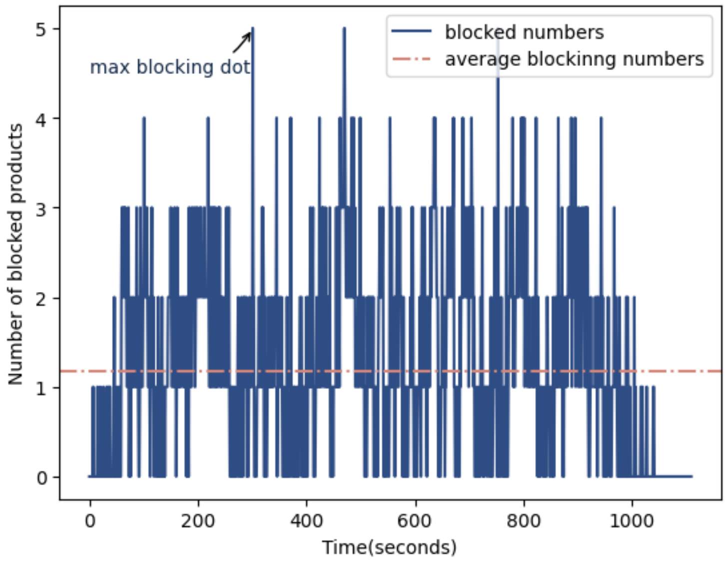


Fig 25 Blocked products (job shop)

The following data on the entire blocking process was obtained by recording and calculating.

```
The end time of the whole process is 1110
The max block number is 5, happens at time 301.
The average blocking number is 1.181081081081081, the average blocking rate is 0.008436293436293436
```

Fig 26 Performance of job shop

3.3.2 Flow shop Scheduling

Statement: All simulations below are based on data from pipeline S[0]

1) Flow shop Core Strategy:

    A flow shop is only characterized by its route and its processing requirements. A product must follow the given order of the machines in which it needs to go through in sequence. It cannot skip any machine in its sequence. And If a product has to pass through a fixed machine which is now processing other products, no matter how long the waiting sequence of the machine is, the product must wait until all the previous products have been processed.

    For example, product18, type19 needs to go through [M8, M2, M9, M3, M12] in sequence. But we can see that when product18 requests the machine3(there is only one machine3) at 226, machine3 is being used by product22(start from 225 and the processing time of machine3 is 11), so product18 must wait until product22 is finished.

```
Product18 Type19 requests machine3 at 226

Product22 Type1 enters machine3 at 225
```

Fig 27 The time product18 request machine3

When product22 has already been processed by machine3, product18 can finally enter machine3 at 242 to be processed.

```
Product22 Type1 leaves machine3 at 242

Product18 Type19 enters machine3 at 242
```

Fig 28 The time product18 enters machine3

2) Cumulative Machine Utilization (CMU):

When the scheduling is flow shop scheduling, the CMU is as follows.

The average response time for each machine:

[19.1, 31.31578947368421, 271.1969696969697, 19.142857142857142, 30.229166666666668, 14.424242424242424, 22.870967741935484, 37.794871794871796, 248.35384615384615, 204.12280701754386, 191.6734693877551, 80.59183673469387, 20.142857142857142]

Each machine's CMU is:

[1.1235294117647059, 1.3048245614035088, 10.430652680652681, 1.1260504201680672, 1.439484126984127, 1.202020202020202, 1.7593052109181142, 1.6432552954292086, 8.86978021978022, 7.038717483363581, 7.09901738473167, 2.7790288529204785, 1.0071428571428571]

The value of CMU also varies going to a large interval just like the job shop scheduling. So we also need to add the back-up machines to reduce the value of CMU.

3) Back-up Machine

Calculate the number of back-up machines according to the previous steps.

```
Machine used times: [148, 85, 303, 140, 194, 148, 263, 195, 282, 275, 236, 201, 92]
Total processing time for each type of machine: [2516, 2040, 7878, 2380, 4074, 1776, 3419, 4485, 7896, 7975, 6372, 5829, 1840]
Minimum time: 1776
Numbers of machines in each type: [1, 1, 4, 1, 2, 1, 2, 3, 4, 4, 4, 3, 1]
```

Fig 29 Back-up machine number (flow shop)

After adding the back-up machines, calculate the Response Time and CMU again.

The average Response Time for each machine:

[22.333333333333332, 29.42105263157895, 26.015151515151516, 23.428571428571427, 22.604166666666668, 15.0, 14.161290322580646, 23.0, 28.723076923076924, 29.0, 27.0, 29.0, 22.047619047619047]

Each machine's CMU is:

[1.3137254901960784, 1.2258771929824561, 1.0005827505827507, 1.3781512605042017, 1.0763888888888888, 1.25, 1.0893300248138957, 1.0, 1.025824175824176, 1.0, 1.0, 1.0, 1.1023809523809525]

Compare the efficiency of the original machine with the efficiency after adding the back-up machines again of the flow shop scheduling.

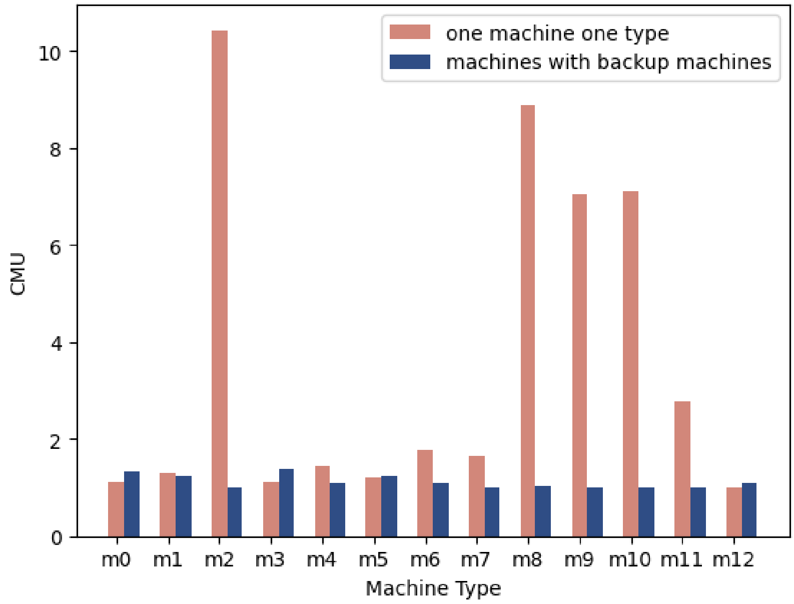


Fig 30 Comparison of machine utilization rates (flow shop)

The average CMU values have also been optimized to a large extent.

4) Occupation Rate

The number of occupied machines and machine occupation ratios of flow shop are shown below.

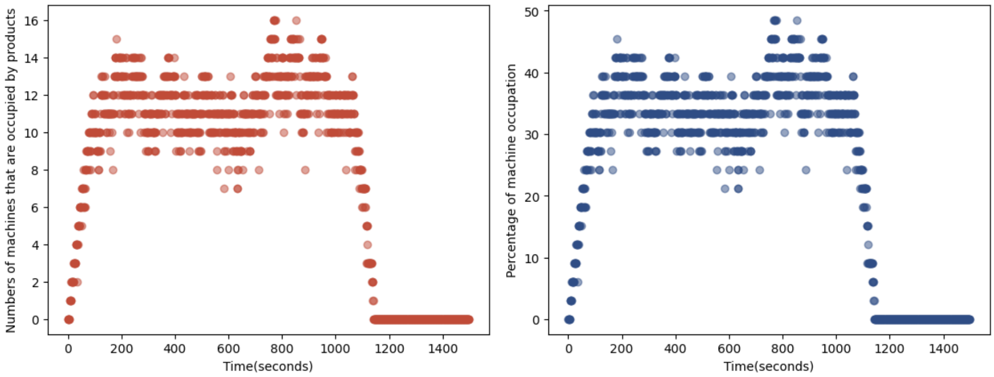


Fig 31 Occupation Rate of machines (flow shop)

5) Blocking rate

The number of blocked products during the entire simulation time of flow shop scheduling is show as follows.

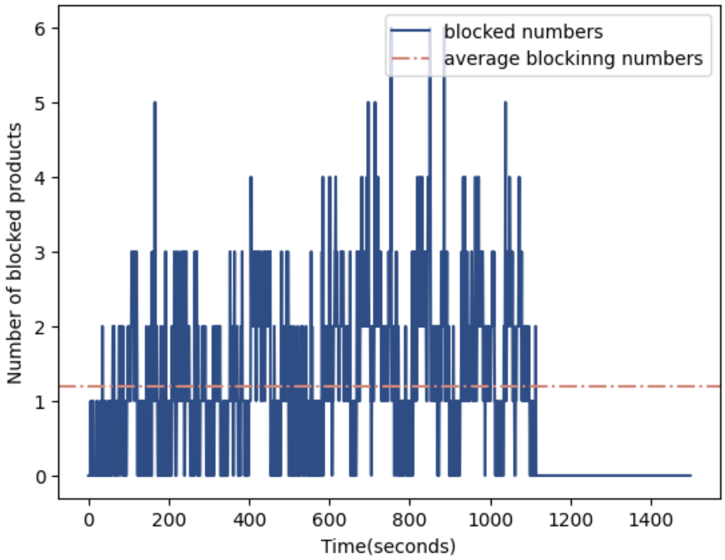


Fig 32 Blocked Products (flow shop)

The following data on the entire blocking process was also obtained by recording and calculating.

```
The end time of the whole process is 1158
The max block number is 6, happens at time 754.
The average blocking number is 1.2124352331606219, the average blocking rate is 0.008660251665433013
```

Fig 33 Performance of flow shop

6) Laxity factor

In order to improve the efficiency of the flow shop scheduling, we introduce the concept of the laxity factor.

When we add the laxity factor, we actually set a threshold for every machine. When the number of products waiting for one machine exceeds the set threshold, the products that request for this machine later will skip it directly and go to the next machine according to the original procedure. And when the products have already passed all the other machines in the procedure, they will come back to the skipped machine and wait until they are processed.

The laxity factor helps the flow shop scheduling reduce the waiting time of the products, thus improving the overall efficiency.

When add the laxity factor, the outcome is as follows.

```
The end time of the whole process is 1152
The max block number is 6, happens at time 754.
The average blocking number is 1.2073665691192865, the average blocking rate is 0.008624046922280618
```

Fig 34 Performance of flow shop(with laxity factor)

We can see that the entire simulation time, the average blocking number and the average blocking rate all decrease a little compared with that before adding the laxity factor. So it can be confirmed that the laxity factor can indeed improve the flow shop scheduling efficiency to a certain extent.

4 CONCLUSION

The simulation of CA1 started with a system understanding of AGV and DMP, extensive literature reading and discrete time simulation using the simpy package under python.

In this phase we focused on discrete-time simulation for scheduling problem. In the simpy simulation environment, we simulated three different scenarios of the product production process, ranging from simple sequential arrangement to more complex out-of-order scheduling problems.

From scenario C, we simulated the system using job shop scheduling and flow shop scheduling for product production. The core ideas of the two scheduling methods are somewhat different and give us a deeper understanding of the differences pursued by small and large batch production.

Finally, the performance of the scheduling system was evaluated using Cumulative Machine Utilization, Machine Occupancy and Product Blockage, and a laxity factor was introduced into flowshop scheduling to improve the efficiency of high volume production.

5 FUTURE WORK

In CA1 we focused on scheduling, in the next phase we need to optimize the scheduling using machine learning and digital twin models.

Machine Learning in the Prediction

We expect to introduce some machine learning algorithms in CA2 to make predictions about how scheduling will work. For example, the system will be able to predict the scheduling length, scheduling time and blocking probability in order to improve the performance of the system by putting in backup machines in advance.

Online Scheduling

Online scheduling is more of a simulation of a virtual space where the prediction of the product processing scheduling system is carried out on the server side, which is also the concept of the virtual space in the digital twin.

Digital-Twin Scheduling

The architecture of Digital-Twin-based job shop scheduling consists of two parts: physical space and virtual space. The two parts communicate with each other through CPS units. In the virtual space, the scheduling data can be obtained from the monitored resource in the physical space, such as equipment, workers, task information, etc. The scheduling strategies can be obtained and simulated by the scheduling models and algorithms with the resource data obtained. The final verified scheduling plan is fed back to physical space for execution. In the physical space, the plan is decomposed into machine execution, operator distribution and material transportation etc. **错误!未找到引用源。**

The digital twin model is used to ultimately achieve optimal scheduling results for AGVs in digital manufacturing platforms.

6 REFERENCE

[1] De Ryck, Matthias, Mark Versteyhe, and Frederik Debrouwere. "Automated guided vehicle systems, state-of-the-art control algorithms and techniques." Journal of Manufacturing Systems 54 (2020): 152-173.

[2] Industrie 4.0 Working Group. Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0; Forschungsunion: Berlin, Germany, 2013.

[3] Grieves, Michael (October 31, 2002). "Completing the Cycle: Using PLM Information in the Sales and Service Functions". Retrieved November 19, 2022.

[4] Grieves, Michael (October 5, 2015). "Can the digital twin transform manufacturing". World Economic Forum Emerging Technologies. Retrieved November 19, 2022.

[5] Y. Fang, C. Peng, P. Lou, Z. Zhou, J. Hu and J. Yan, "Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing," in IEEE Transactions on Industrial Informatics, vol. 15, no. 12, pp. 6425-6435, Dec. 2019.

[6] Corallo, A.; Del Vecchio, V.; Lezzi, M.; Morciano, P. Shop Floor Digital Twin in Smart Manufacturing: A Systematic Literature Review. *Sustainability* 2021, *13*, 12987.

[7] Zhuang, C., Liu, J. & Xiong, H. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. Int J Adv Manuf Technol 96, 1149–1163 (2018).

[8] Mantel, R., & Landeweerd, H. R. A. (1995). Design and operational control of an AGV system. *International journal of production economics*, *41*(1-3), 257-266.

[9] Fraile F, Sanchis R, Poler R, et al. Reference models for digital manufacturing platforms[J]. Applied Sciences, 2019, 9(20): 4433.

[10] Fei Tao, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, Ping Ji. “Digital twin modeling”, Journal of Manufacturing Systems 64 (2022): 372-389.