Digital Twin-Based Job Shop Scheduling Algorithm for Poultry Feed Industrial Factory

Nehal Tarek
Information Systems Department,
Faculty of Computers and Information,
Menoufia University
Shebin El Kom, Egypt
nehaltarek@ci.menofia.edu.eg

Nancy A. El-Hefnawy
Information Systems Department,
Faculty of Computers and Informatics,
Tanta University
Tanta, Egypt
nancyabbas_1@ics.tanta.edu.eg

Hatem Abdel-Kader
Information Systems Department,
Faculty of Computers and Information,
Menoufia University
Shebin El Kom, Egypt
hatem6803@yahoo.com

Amira Abdelatey
Information Systems Department,
Faculty of Computers and Information,
Menoufia University
Shebin El Kom, Egypt
amira.abdelatey@gmail.com

***Abstract* — Digital Twin (DT) is a rising technology concept for representing the physical-world assets in digitalized form. Real time events can be mapped using DT by monitoring the physical part of an entity and employing both artificial intelligence algorithms and big data analytic tools to create DT models that represent the virtual part. In smart manufacturing, job shop scheduling is affecting the production efficiency. To ensure the maximum utilization of the available resources, a robust job shop scheduling technique is proposed for a poultry feed industrial factory. This has been achieved via two phases. Phase one, a flexible digital twin model is built to simulate the actual physical system of the production lines in the adopted factory. The main objectives of this phase are modeling, monitoring, and graphing the production lines in the factory. In phase two, an efficient job shop scheduling algorithm is presented to schedule and reschedule the jobs among the healthy production lines. The optimization technique used in this phase is a genetic algorithm with enhanced chromosome representation to ensure that all the populations' individuals generated are feasible solutions which reduce the time required to achieve convergence.**

Keywords— job shop scheduling problem, smart manufacturing, digital twin, knowledge graph.

I. INTRODUCTION

The digital twin concept is the process of creating a digital model which is the same as a physical model. This process is modelled to monitor the physical entity by giving a virtualize simulated model, collecting data from the physical part to use it in optimization techniques as illustrated in [1]. The first visualization of a DT model was demonstrated [2]. The authors in [3] defined the digital twin as the integration of three primary components: physical part, digital part, and connection for the data flow between physical and digital parts.

The physical part or physical twin is an asset which can be a building, enterprise, system, component, process, car, human; etc. The virtual part is a digital replica for this asset which models all the components and processes of the physical twin. All the data collected from the physical part is analyzed using data analytic tools to build a simulated model that can reflect the behavior of the physical system, and then further used to both predict and detect failure. The data connection transfers data from the physical part to the digital one and brings back the information excluded to the physical part. This information includes the simulated model, the prediction of equipment failure, the detection of uncertainty events triggered in the system. The mission of data processing in DT logically using machine learning and deep learning techniques. It is a vital step to help simulating and optimizing the system. The technology of DT is widely used in almost all the fields of life including agriculture, electronic, smart manufacturing, renewable energy, education, healthcare, transport, etc. Considering the industrial domain, which is the concerned domain in this paper, creation of the DT models in manufacturing is a smart process, and it is an open area for research.

Recently, with industry five revolution, the concept of big data became popular with the tremendous amount of data that could be collected and needed to be well organized using the big data analytic tools. The Internet of Things (IoT) technology is the essential tool that can facilitate the process of collecting data in the real world. The process of digitalization using IoT spreads to multiple domains such as smart factories, healthcare, environments, and education [4]. Authors in [5] illustrate that the simulated DT models integrate both industry and academia and that helps in well visualizing the operation cycle. The data collected from the physical twin requires accuracy as it is used to create the DT models. To ensure the efficiency of the DT models, the data flow between the physical twin and the digital twin must be sufficient in all phases of the product life cycle [6]. To perform maintenance, all the DT models must be accessed. The analysis of data provided to retrieve information improves the system performance [7]. Using the DT technology in various domains of industry has been turned into a scalable challenge [8].

Job-shop scheduling problem (JSSP) is considered as decisive NP-Hard problem in manufacturing. The goal is to solve this problem by assigning several jobs to multiple machines in the predetermined time slots following the pre known time processing to optimize the given fitness functions, e.g., minimize the completion time of the production cycle, reduce energy consumption, eliminate tardiness, and maximize the utilization rate of machines, taking a list of constraints into account.

JSSP can be divided into a few types based on a series of assumptions that have been made, these assumptions are the key to characterizing the JSSP model. Some of these types are explained as follow:

(1) Classical JSSP: the most basic problem is the classical JSSP, in which there are many assumptions are involved and most of these assumptions are to make the JSSP problem easier.

(2) Dynamic JSSP: all the involved attributes concerning the static JSSP are pre-defined and remain constant the whole scheduling process, that is not the case in the dynamic JSSP whereas the attributes of the problem can be changed during the scheduling problem such as the machine availability status or new jobs arriving.

(3) Flexible JSSP: the flexible JSSP can go deeper in the assumptions where the available machines for a specific operation can be part of the total machines not all of them, also the time processing of a certain operation can be a factor which depends on the assigned machine, the precedence constraint also is determined to follow a certain sequence of operations within the job.

Due to various types of the JSSP problem, optimization techniques using artificial intelligence algorithms should be used to optimize and solve the problems of JSSP to provide near to optimum feasible solutions. Genetic algorithm (GA) is defined as one of the most popular AI algorithms that is used to find a near to optimal solution for the flexible JSSP problem, and it is the algorithm used in this paper.

The remaining of the paper is organized as follow: Section II introduces the DT-based job shop scheduling system and its related work. Section III explains the proposed system for Poultry Feed Industrial Factory. Section IV presents the architecture of the proposed software for digital twin based simulated knowledge graph. The near-to-optimal distribution of the job shop scheduling problem is implemented in Section V defines. Finally, Section VI discusses the simulated digital twin and optimization algorithm for the adopted poultry feed factory. Then, Section VII concludes the results and presents the future work directions.

II. RELATED WORK

Research has been conducted to ensure solving the scheduling problems using the advantages of DT. The authors in [9] proposed a scheduling algorithm based on the DT technology. The goal of the research was to reduce total makespan time and reduce the scheduling deviation considering the fast change of circumstances in the industrial factories, and failure that could happen to a certain part and other uncertain events that cause asymmetry between the real progress of production and the scheduled plan.

In [10], the authors proposed a DT-enhanced methodology for dynamic scheduling in which they used a DT with five-dimension for the machines in the job shop scheme. By using DT, the proposed method presented three services: (1) availability prediction service, (2) disturbance detection service, and (3) performance evaluation service. The goal is to obtain both high efficiency (by minimizing the makespan) and high stability (by minimizing the changes of starting time of each operation). However, the algorithm will be useless if the DT models are not accurate, the simulated models must be updated continuously to ensure accuracy. Also, it is time consuming to build complete DT models for the manufacture and the cost is very high.

The authors in [11] presented a robust scheduling framework for a Flow Shop Scheduling Problem (FSSP). The framework is based on the GAs and discrete event simulation. The framework is a digital representation of the manufacturer. Although the proposed algorithm still needs improvement in the data transfer protocol to ensure detect the machine failure in a reasonable time manner.

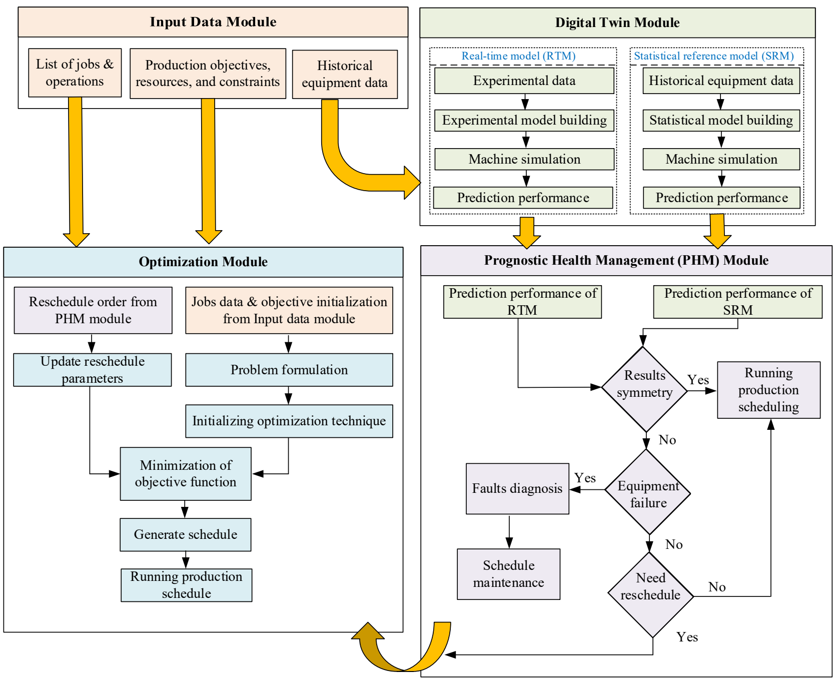


Figure 1 Proposed system architecture

In [12], digital twin data has been used to develop a proactive job shop scheduling algorithm in multi-job multi-machine system. In this work, makespan delay time was reduced by modifying the start-up time and also adjusting processing sequence of the operations that hasn't been processed yet. This algorithm is rule-based, so to ensure the effectiveness of the algorithm, more rules must be applied to detect all the uncertainty events that could happen during the production process.

In [13], the authors made an attempt for minimizing the FJSSP makespan, using an improved hybrid GA called the VND-hGA, by integrating the BBPSO barebones particle swarm optimization and VND into one framework of the GA. Using benchmark instances, experiments were conducted to validate the proposed strategy. An evaluation was made for the performance of the VND-hGA.

III. Architecture of the Proposed System

Recently, the process of digital twinning became a vital research topic in smart manufacturing. Also, JSSP is an important research field a way earlier before DT. With the industry revolutions, the whole process of production became a smart manufacture, The DT technology is adopted to solve JSSP in smart manufacturing industry. The proposed system is architected into four modules as shown in Figure 1.

1. Input Data Module

There are three types of input data that will be collected in this module: historical equipment data, dataset for both jobs and operations associated with the jobs and this dataset also contains the available machines for each operation and the corresponding processing time, also the mathematical model is defined by determining production objectives, resources, and list of constraints.

2. Digital Twin Module

This module will be described in detail in section IV. Four layers have been developed to form four-tier architecture; this architecture was sufficient to represent the adopted factory. These layers are: Data, Physical, presentation, and Analytic layer.

3. Prognostic Health Management (PHM) Module

In this module, the results obtained from the digital twin module are compared to determine whether the physical twin is actually identical to the virtual or not. If they are the same, then the system is running normally and no changes needed to be done. If there is a difference between them, then moving to the next step which is checking equipment failure.

Checking if a failure occurred to one of the equipment in the physical system is a necessary step. If no failure existed, then a check for whether the system needs a reschedule or not is performed to decide whether to move to the reschedule module or no action need to be taken and continue on the initial schedule plan. If there is a failure with any of the equipment, a fault diagnosis is performed in order to figure the problem and report to the user that a maintenance needed to be scheduled. In this case, the machine with the breakdown will be excluded from the schedule plan and a rescheduling event will be triggered.

4. Optimization Module

In this module, the following steps are to be followed:

- Step 1: defining the problem formulation using the production objectives and list of both assumptions and constraints.
- Step 2: preparing the dataset (required jobs, number of operations, a list of available machines for every operation and time processing for each operation on different machines).
- Step 3: Initializing optimization algorithm to minimize the fitness function and to generate the schedule plan.
- Step 4: In case of reschedule needed, moving to step 2 to update the parameters.

The poultry feed industrial factory has three production lines, each line consists of seven machines which are (miller, mixer, adding liquid and nutrients, pelletizer, cooler, scale and sewing machine). Each machine's function is considered as an operation and the seven operations represent a single job in the adopted factory. The machines in the lines vary in power and average capacity. For example, the milling machine in the first line can produce 3-4 T/H while the machine in the second line can produce 8-12 T/H and the machine in the third line can produce 15-25 T/H. Assume loading the average capacity of each machine, the processing time can be calculated and determined so that in the phase of preparing data, each operation (for example, the milling operation) is assigned a processing time on the three milling machines through the three production lines. These seven operations are following the precedence constraint so that the pelletizer operation cannot be done before the mixing operation and so on, and that constraint is considered during running the genetic algorithm optimization technique.

To determine whether the system needs a reschedule or not, a time threshold is defined and allowed to be delayed in the makespan. Once the threshold is exceeded, the reschedule algorithm is triggered with the updated parameters.

IV. DESCRIPTION OF DIGITAL TWIN FRAMEWORK

A. Physical Twin

Each production line in the factory consists of seven machines doing several tasks beginning with raw materials passing all the required operations until the end which is the specified final product. Considering that there are three production lines, each line consists of seven different job machines. The seven machines are (miller, mixer, adding liquid and nutrients, pelletizer, cooler, scale and sewing machine) so totally there are 21 machines in the factory. The data for simulation was calculated based on the power rating and loading average capacity for every machine in the production lines. corresponding corrections have been made in this case.

Raw materials (Barley, Corn, and Soya) must be gone through a preparation phase that includes magnet for the metal impurities and filter for non-metal impurities. The first process is milling the raw materials in separate containers then moved to the mixing operation after dosing the accurate quantities of each raw material to balance the nutrients following the standards of this industry. Next, a pelletizer which is used to pelltize the product. Finally, packing the product and moving it to the warehouses to be distributed.

B. Virtual Twin

The virtual twin of the proposed system is visualized as a knowledge graph. This graph is used to integrate information from different sources [14]. It is a paradigm to replicate data. In [15] the authors extracted features by using knowledge graphs and that helped with the time series data (data collected by sensors and transferred in real time). Authors in [16] using a network have modelled the communication of DT parts by using knowledge graphs and visualized how to self-heal a fault. The proposed system on this paper is based on a previous work for the authors [17]. As mentioned earlier in section III, the proposed architecture has four layers which completely represent the factory and were shown in Figure 2.

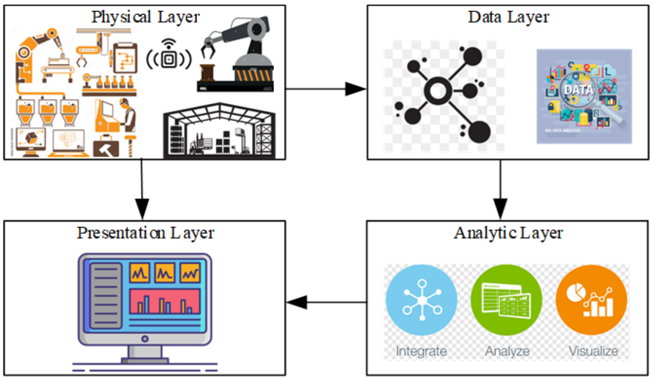


Figure 2 The four-tier architecture

All the assets in the factory are included in the physical layer, the sensors generate data stream, and this data can be transferred using radiofrequency identification (RFID) or WiFi.

The data layer involves a dynamic data. This data was generated by the assets and can be (measurements for current and temperature, streaming videos, images, etc....). This layer also contains DT models and relationships between possessions.

An instance for the factory DT is included in the analytic layer and it contains the simulated model to monitor the status of the production lines.

The presentation layer mainly consists of the dashboard to show all the information retrieved from the virtual part to the decision maker. It shows the visualized graph of the factory as stored in the Neo4j database. The dashboard uses REST API to link to the application.

It is a goal to minimize latency of all the four layers while operating to ensure the efficiency of the system.

C. Prototype and Deployment of Software

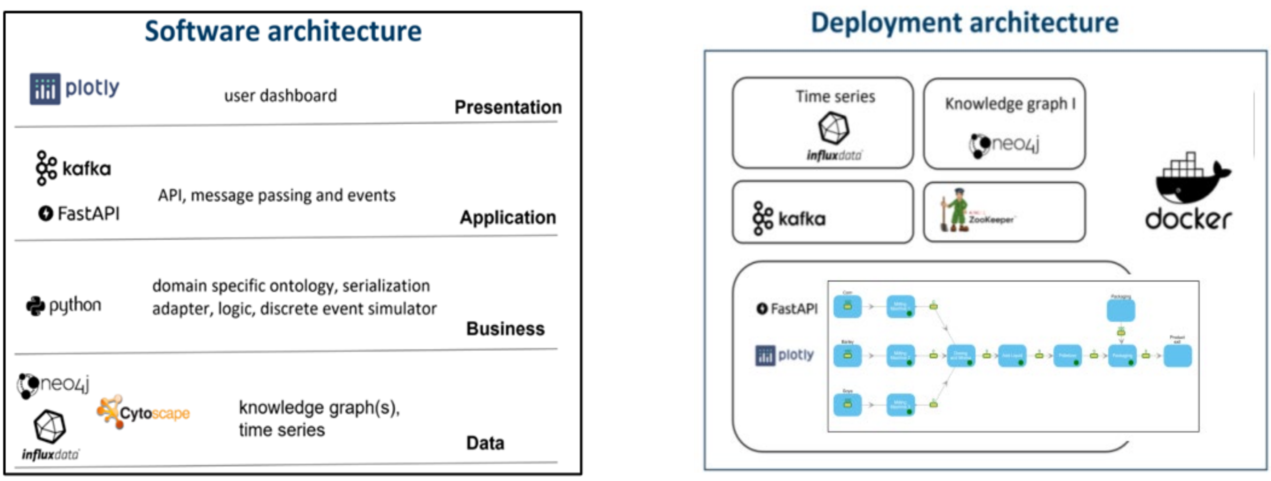
The implementation of the proposed framework has been gone through multiple phases, and various software tools have been used. As shown in Figure 3-a, the software structure is described. It starts with influx database which stores the sensor data and the Neo4j database which represents the factory as a knowledge graph. Python programming language was used for serialization and adapting, also the discrete event simulator was built using the python library simpy. Kafka was used to pass messages and finally there is a user dashboard to represent the system. Figure 3-b demonstrates the deployment phases up to the plot phase of the factory's DT model. The plotly dash was used for the user dashboard and docker was used to present the DT graph for the user.

V. JOB SHOP SCHEDULING PROBLEM

A. Problem formulation

The proposed (JSSP) algorithm is considered a flexible algorithm and it can be summarized as follows: Considering N jobs can be listed as $J = \{J_1, J_2, \ldots, J_N\}$ and M machines can be listed as $M = \{M_1, M_2, \ldots, M_M\}$. For each job, there is a pre-known sequence of operations. Each single operation can be processed on one machine which can be selected out of a list of pre-determined machines, The FJSSP is defined as assigning a sequence of the operations to the available machines satisfying the described constraints. The difference between the flexible job shop scheduling problem and the classical one is that each operation, of the later one, has a set of machines to be processed on and cannot be processed on other machines in the system [18].

In this paper, several hypotheses are considered and can be summarized as follows:



(a) Software architecture.

(b) Deployment architecture.

Figure 3 Implementation of framework software

1) At time 0, all machines are available.
2) At time 0, all jobs are released.
3) Only one operation can be processed on a machine at a time.
4) Operations are to be processed without interruption by one machine selected from the set of available machines.
5) Time for processing each operation varies on different machines.
6) For each job, the sequence of operations is predefined and remains unchanged.
7) The machines setup time is ignored, and the transfer time from one machine to another is negligible.

B. Optimization Algorithm

To solve the FJSSP, the GA described in [19] is justified and used in the proposed job shop algorithm. Genetic algorithm is one of the most popular artificial intelligence-based optimization algorithms with the advantage of good convergence of solution sets and reducing computational complexity. Keep tracking of minimized fitness function which is in this case minimizing the makespan required to finish all the given jobs. And thus, the accuracy of the results is improved. In this technique the individuals are evenly extended through the entire domain to ensure the diversity of the population.

To illustrate the chromosome representation in a simplified form, consider the JSSP instance shown in Table 1. It consists of two jobs: the first job has three operations, and the second job has two operations. Each operation can be processed on certain machines which are determined in the alternate machine set. Also, each operation has a processing time for each machine in the alternate machine set.

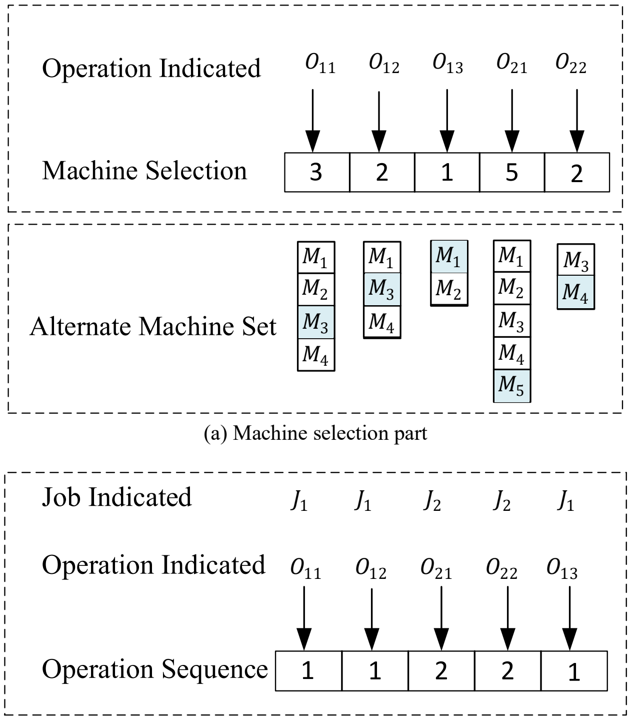
TABLE 1: PROCESSING TIMETABLE OF AN INSTANCE

Job	Operation	M_1	M_2	M_3	M_4	M_5
J_1	O_{11}	2	3	6	4	–
	O_{12}	5	–	3	2	–
	O_{13}	7	8	–	–	–
J_2	O_{21}	4	2	1	7	6
	O_{21}	–	–	4	3	–

The used chromosome can be divided into two sequenced parts: machine selection part and operation sequence part. The machine selection part has the length L which is the number of total operations in all the jobs. This part contains an index referring to the assigned machine from the alternate machine set. The operation sequence part which also has the length L contains the proposed order for jobs and operations.

To illustrate the chromosome representation The two parts of the chromosome are illustrated in Figure 4 (a) and (b) respectively.

In Figure 4-a, the array contains five genes representing an index referring to the alternate machine set. In the second job, the first operation is assigned to M5, and the second operation is assigned to M4. In Figure 4-b, the array contains also five genes representing the proposed sequence for all the given operations. In this case, the first operation of the first job is processed first followed by the second operation, then the first operation of the second job is processed followed by the second operation of the same job. Finally, the third operation of the first job is processed. The value in the gene refers to the number of job considering the precedence rule of the operations.



(a) Machine selection part

(b) Operation sequence part

Figure 4 Chromosome representation

Figure 5 shows the basic procedure of the genetic algorithm. The ending condition of the algorithm is to perform 100 iterations, so the checking of the iteration number is performed before each iteration. First, an initial population of size N is generated. The first generation population of size N was obtained through the two basic operations of global selection and local selection. The chromosome representation described earlier has ensured that all the populations generated are feasible solutions, this helps in reducing the computational time and achieving convergence faster. Second, an evaluation for the population will be performed to calculate the minimum makespan of each population in the initial solution. After that, a parent selection will be done to choose the parents and put them in the mating pool to run the basic genetic algorithm operations: both crossover and mutation to generate a new child population. Then the evaluation process will be done again. These steps will be repeated through all the iterations until the final iteration. Then concluding the results with the best makespan obtained.

VI. RESULTS

A. Results of Dashboard

Combining all the software tools used in this system, a DT visualization can be obtained. Starting with influx database to store data gathered in real time from sensors, this data can be used later to detect anomalies in the system. Neo4j database is also used to store the knowledge graph of the factory. Zookeeper and Kafka are used to handle the process of passing messages in Influx database. Also, the python library FastAPI was used to code API. The simpy Python library has been used to trigger discrete event simulation. And finally, docker was used to deploy the system and the dashboard is updated continuously. Figure 6 depicts the graph of the factory with a predetermined product amount of 100 units.

B. Optimization Results

First, an evaluation for the genetic algorithm used in this paper is performed by comparing the results with those published by Mastrolilli and Gambardella in [20] and GENACE in [21]. The proposed algorithm is firstly tested with Brandimarte's data set (BR data) [22], ten problems are included in the data set that have various number of jobs, operations, and machines. Three problems out of the ten are selected to be tested: MK1, MK2 and MK4 and the results are shown in Table 2.

TABLE 2: RESULTS OF BR DATA

Problem	Size $n \times m$	T_o	M&G	GENACE	Proposed GA
Mk1	10×6	55	40	41	42
Mk2	10×6	58	26	29	29
Mk4	15×8	90	60	67	67

The first column in the table indicates the problem number while the second column determines the number of jobs and number of machines, the third column indicates the total number of operations in the problem. The table shows that the genetic algorithm used in this work has recorded as good results as GENACE algorithm in two problems and a very similar result in the third one.

To use the scheduling strategy on the adopted factory, a case study is given. It is assumed that there are ten jobs that need to be done, each job consists of 7 sequenced operations. The operations within a single job must follow the precedence constraint so that no operation can be started unless the previous operation is already done. Three machines are available for each operation to be processed on through the three production lines of the poultry feed factory. The processing time for these three machines is fixed and pre-defined.

Figure 7 shows an evaluation of the designed fitness function. It depicts the evolution of the makespan through 100 iterations. It should be considered that the use of the enhanced GA randomly impacts the convergence of results.

It can be noticed that the convergence started a little after the twentieth iteration with minimum makespan of 77 units of time where the unit of time is equal to five minutes. As a result, relating to the designed fitness function. It can be concluded that this situation can better suit the practicality of scheduling in smart manufacturing applications.

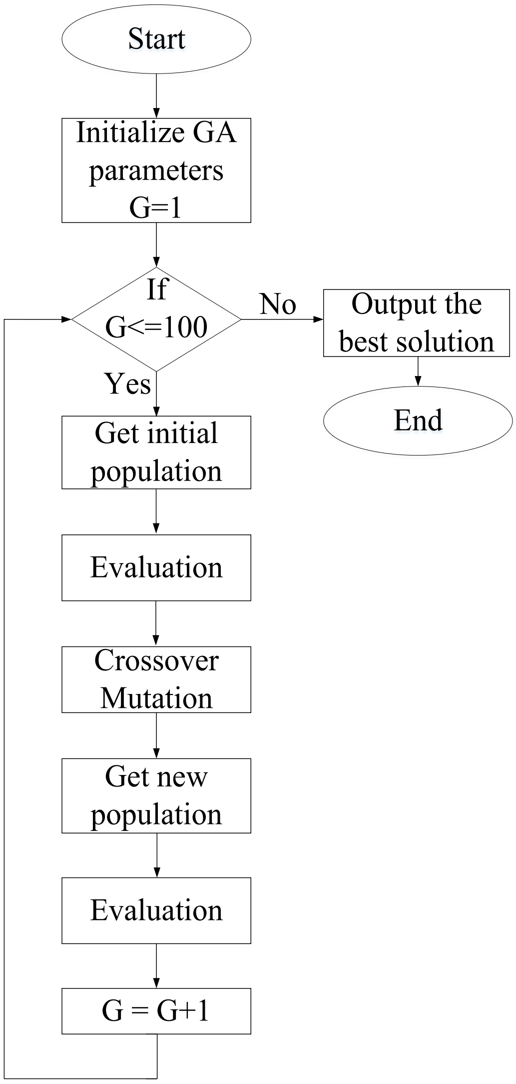


Figure 5 Flowchart of the genetic algorithm

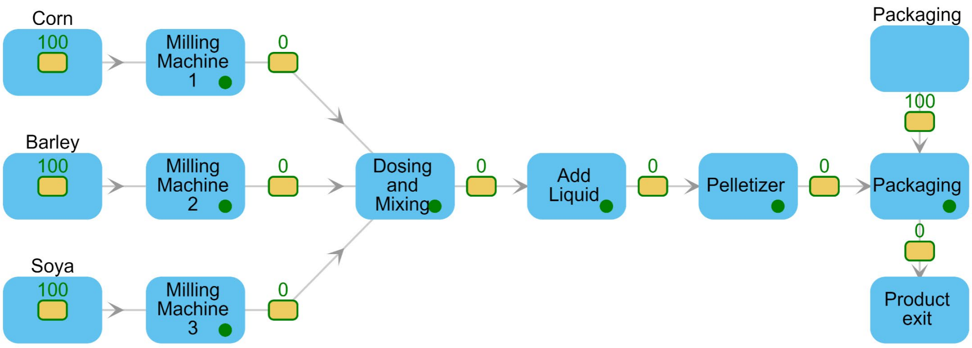


Figure 6 Dashboard for visualizing the digital twin framework

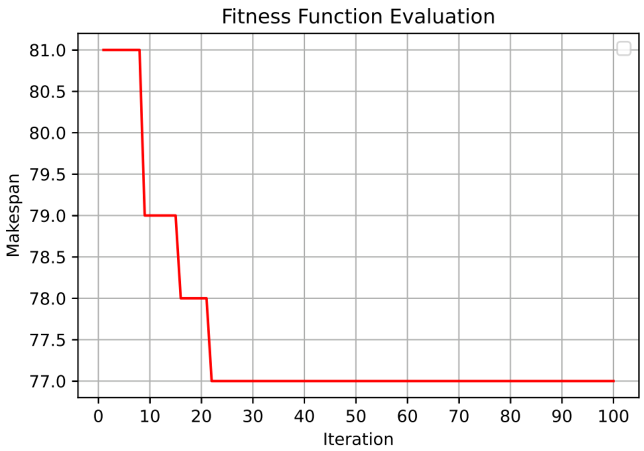


Figure 7: Fitness Function Evaluation

VII. Conclusion And Future Directions

The proposed research has concentrated on designing a DT-based job shop scheduling algorithm to optimize the redistribution working operation of the machines of a poultry feed industrial factory. A near optimal makespan has been obtained using the proposal justified GA. In This study, a four-layer scheme has been successfully employed through a flexible framework of the DT to represent the factory behaviour. Using this flexible framework, a job shop scheduling algorithm for the adopted factory has been developed using the genetic algorithm optimization technique. The algorithm was tested and evaluated. From the last results in this stage, detecting anomalies in the proposed system is the future direction of using the designed flexible DT framework and then reschedule with the updated parameters.

References

[1] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat and S. Bakiras, "The Role of AI, Machine Learning, and Big Data in Digital Twinning: A Systematic Literature Review, Challenges, and Opportunities," in IEEE Access, vol. 9, pp. 32030-32052, 2021, doi: 10.1109/ACCESS. 2021. 3060863.

[2] M. W. Grieves, ''Virtually intelligent product systems: Digital and physical twins,'' Complex Syst. Eng., Theory Pract., pp. 175–200, 2019.

[3] M. Grieves, ''Digital twin: Manufacturing excellence through virtual factory replication,'' White Paper, 2014, pp. 1–7, vol. 1.

[4] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," IEEE Access, vol 8, pp. 108952-108971, 2020.

[5] F. Tao, H. Zhang, A. Liu and A. Y. C. Nee, "Digital Twin in Industry: State-of-the-Art," in IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2405-2415, April 2019, doi: 10.1109/TII.2018.2873186.

[6] A. Kusiak, "Smart manufacturing must embrace big data," Nature, vol. 544, no. 7648, pp. 23–25, 2017.

[7] Q. L. Qi. and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," IEEE Access, vol. 6, pp. 3585–3593, Jan. 2018, doi: 10.1109/ACCESS.2018.2793265.

[8] F. Tao, M. Zhang, J. F. Cheng, and Q. L. Qi, "Digital twin workshop: A new paradigm for future workshop," Comput. Integr. Manuf. Syst., vol. 23, no. 1, pp. 1–9, Jan. 2017, doi: 10.13196/j.cims.2017.01.01.

[9] Yilin Fang, Chao Peng, Ping Lou, Zude Zhou, Jianmin Hu, and Junwei Yan, "Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing", IEEE Transactions on Industrial Informatics, Vol. 15, No. 12, December 2019.

[10] Meng Zhang, Fei Tao, A.Y.C. Nee, "Digital Twin Enhanced Dynamic Job-Shop Scheduling", Journal of Manufacturing Systems 58, pp. 146–156, 2021.

[11] Elisa Negri, Vibhor Pandhare, Laura Cattaneo, Jaskaran Singh, Marco Macchi, Jay Lee, "Field-synchronized Digital Twin framework for production scheduling with uncertainty", Journal of Intelligent Manufacturing Vol. 32, pp.1207–1228, 2021.

[12] Fuqiang Zhang, Junyan Bai, Dongyu Yang, and Qiang Wang, "Digital twin data-driven proactive job-shop scheduling strategy towards asymmetric manufacturing execution decision", Scientific Reports, 2022.

[13] Zhifeng Liu , Junlong Wang , Caixia Zhang , Hongyan Chu , Guozhi Ding , Lu Zhang "A hybrid genetic-particle swarm algorithm based on multilevel neighbourhood structure for flexible job shop scheduling problem", Computers and Operations Research, 2021.

[14] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. d. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier et al., "Knowledge graphs," Synthesis Lectures on Data, Semantics, and Knowledge, vol. 12, no. 2, pp. 1–257, 2021.

[15] A. Banerjee, R. Dalal, S. Mittal, and K. P. Joshi, "Generating digital twin models using knowledge graphs for industrial production lines," in Proceedings of the 2017 ACM on Web Science Conference, ser. WebSci'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 425–430.

[16] Y. Zhu, D. Chen, C. Zhou, L. Lu, and X. Duan, "A knowledge graph based construction method for Digital Twin Network," in 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI). Beijing, China: IEEE, Jul. 2021, pp. 362–365.

[17] N. Tarek, N. A. El-Hefnawy, H. Abdel-Kader and A. Abdelatey, "A Flexible Digital Twin Framework for Poultry Feed Industrial Factory," 2022 32nd International Conference on Computer Theory and Applications (ICCTA), Alexandria, Egypt, 2022, pp. 253-257, doi: 10.1109/ICCTA58027.2022.10206352.

[18] Ho, N. B., Tay, J. C., Edmund, M., & Lai, K. (2007). An effective architecture for learning and evolving flexible job shop schedules. European Journal of Operational Research, 179, 316–333.

[19] Ho, N. B., Tay, J. C., Edmund, M., & Lai, K. (2007). An effective architecture for learning and evolving flexible job shop schedules. European Journal of Operational Research, 179, 316–333.

[20] Mastrolilli, M. and Gambardella, L.M. (2000), Effective neighbourhood functions for the flexible job shop problem. J. Sched., 3: 3-20.

[21] N. B. Ho and J. C. Tay, "GENACE: an efficient cultural algorithm for solving the flexible job-shop problem," Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), Portland, OR, USA, 2004, pp. 1759-1766 Vol.2, doi: 10.1109/CEC.2004.1331108.

[22] Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. Ann Oper Res 41, 157–183 (1993).