INTERNATIONAL JOURNAL OF INDUSTRIALIZED CONSTRUCTION
2020, VOL. 1, No. 1
https://doi.org/10.29173/ijic215

Planning semi-automated precast concrete production using genetic algorithm

Chen CHEN[1], Thomas PHANG[2], Robert L. K. TIONG[3]

[1] *Research Fellow, School of Civil and Environmental Engineering, Nanyang Technological University*
[2] *PhD Candidate, School of Civil and Environmental Engineering, Nanyang Technological University*
[3] *Associate Professor, School of Civil and Environmental Engineering, Nanyang Technological University*

ABSTRACT

Although fully automated production systems have been developed and used in some industry leaders, most of the precast concrete factories have yet to be developed to that stage. Semi-automated production lines are still popularly used. As production productivity can be maximally improved within the physical constraints by applying a sound production plan, this paper tends to propose a production planning method for the semi-automated precast concrete production line using genetic algorithm (GA). The production planning problem is formulated into a flexible job shop scheduling problem (FJSSP) model and solved using an integrated approach. Thanks to the development of new technologies such as building information modeling (BIM) platform and radio frequency identification (RFID), implementation of a just-in-time (JIT) schedule in the semi-automated precast concrete production line becomes practicable on the grounds of risk mitigation and enhanced demand forecast capability. In this regard, the optimization objectives are minimum makespan, station idle time, and earliness and tardiness penalty. An example was applied to validate the integrated GA approach. The experimental results show that the developed GA approach is a useful and effective method for solving the problem that it can return high-quality solutions. This paper thus contributes to the body of knowledge new precast concrete production planning method for practical usage.

Submitted January 2020
Revised May 2020
Accepted June 2020
Published July 2020

Corresponding Author
Chen Chen
chenchen@ntu.edu.sg

School of Civil and Environmental Engineering,
Nanyang Technological University
50 Nanyang Avenue
Singapore 639798

DOI http://doi.org/10.29173/ijic215
Pages 48-63
ISSN 2563-5034

KEYWORDS

Semi-automated production, Genetic algorithm, Just-in-time, Precast concrete, Scheduling

Introduction

Precast concrete construction is the manufacturing of concrete products in a pre-made and reusable mold and then, after being cured in a controlled environment, transporting, positioning and assembling them into a structure with minimal additional site work. Precast concrete construction has attracted worldwide concerns because of its significant role in the realization of sustainable development goals and its market is boosted by the rapid pace of industrialization and urbanization. There are many incentives and schemes available from governments to push for precast concrete construction. Take Singapore as an illustration, the Housing and Development Board (HDB) has been actively adopting precast concrete technology since the 1980s. Today, the precast concrete implementation level is maintained at about 70% for each HDB project [1]. The government directives accelerate the rate of precast concrete adoption and eventually speed up the move of modernization, mechanization, and industrialization in the precast concrete industry [2].

As the construction industry is shifting to manufacturing and mass production, implementing the best manufacturing practices such as lean manufacturing (LM) will help to improve productivity and customer satisfaction. Originated from the Japanese Toyota Production System (TPS), LM facilitates the production of diverse products in flexible volumes [3]. It is one approach that suits for any company, regardless of firm size, business sector, organizational culture, and geographic location. LM is a philosophy grounded on five basic principles: value, value streams, flow, pull and perfection [4]. Under the umbrella of lean philosophy and principles, there are a wide variety of lean tools, for example: value stream mapping (VSM), Kanban/pull, just-in-time (JIT), total production maintenance (TPM), 5S (sort, straighten, shine, standardize and self-discipline), cellular manufacturing, Jidoka (automation) and continuous improvement. In short, LM can be described as a systematic method for the elimination of waste within a manufacturing process [5]. Seven potential sources of wastes including over-production, waiting, transportation, over-processing, inventory, motion, and defects, are often mentioned that need due consideration during lean implementation [6,7]. Some researchers also add excess human resources as the eighth waste [8].

Precast concrete companies are mostly small and medium-sized enterprises (SMEs) [9]. While LM tools such as VSM, Kanban/pull, and TPM are mostly utilized, literature reviews from past journals show that JIT production is rarely implemented in SMEs [10]. Certainly, JIT is never easy for larger companies to implement, too. JIT refers to the production of goods to meet customer demand exactly, in time, quality and quantity. Many people may confuse about JIT and LM. JIT aims at creating the ideal balance of inventory and workflow to avoid excess and shortages, while LM is implemented to improve quality and reduce waste. Because JIT and LM share most of the same characteristics, goals, and philosophy, although strictly speaking JIT is one LM tool, a lot of people deem JIT as another term of LM. Besides, the terms JIT, TPS, pull production and Kanban are often used interchangeably in the literature, too [11]. JIT flow depends on production leveling. Compared to larger companies, SMEs have disadvantages of lacking resources and skills. In addition, the lack of scale and bargaining power hinder the smooth operations of their production systems under the effects of market dynamism.

Despite many challenges, Dowlatshahi and Taham [12] argued that JIT is still applicable to SMEs. The emergence of various cloud-based collaborative systems, which make real-time information sharing amongst all stakeholders in the supply chain a reality, is gradually mitigating the traditional barriers of SMEs to implement JIT. Particularly, in the construction sector, the building information modeling (BIM) collaboration cloud platforms enable more effective and efficient communications for different parties on the design and construction phases [13]. In this context, this paper aims to perform JIT optimization in precast concrete companies for precast concrete production.

Similar attempts have been observed in previous studies. Chan and Hu [14] developed a flow shop sequencing model using a genetic algorithm (GA) for production organization in precast factories. The objective is a weighted function based on the maximum makespan and the penalties for earliness and tardiness. The latter part represents the JIT criterion. Ko and Wang [15] improved the model by taking the limited buffer size between stations into consideration. However, these previous models assume a fully automated production line. In reality, few precast concrete companies have developed to that level. The majority of them combine manual and automated operations, thus achieving a semi-automated status. Compared to fully automated production lines, semi-automated ones are more flexible, whereby there are more decision variables that need to be determined. Therefore, the models designed for fully automated production lines cannot be readily applicable to the semi-automated ones. To fill in this gap, the study attempts to develop a production scheduling model, particularly to address the nature of a semi-automated precast concrete production line. Similar to the previous ones, the optimization model will contain a JIT criterion.

The remaining of this paper is organized as follows: after this introduction, Section 2 describes the ways to improve the practices of a precast concrete factory and a typical semi-automated precast concrete production line and its

main problems. Section 3 develops the problem model and Section 4 gives the solving algorithm, which is a simulation-based GA approach. An experimental case study is given in Section 5. And finally, the paper is concluded in Section 6.

The lean approach in precast concrete production

The ways to get lean

The practices of a precast concrete factory are usually divided into two phases: the design phase and the fabrication phase. Before fabrication, shop drawings of individual precast concrete elements are required to be prepared and detailed. The shop drawings are usually produced in-house and submitted to architects/owners for approval. There may be several rounds of revisions and resubmissions. The procedure continues until the drawings get approved. Any discrepancies among the architectural, structural and production requirements must be resolved before sending these drawings for fabrication to minimize abortive works. In general, an overall reduction of lead time requires process improvements in both the design phase and the fabrication phase.

Nowadays, the use of BIM software can significantly simplify the design phase [16]. The BIM models are not purely geometric models or 3D representations as before. They are integral models that contain all information in terms of architecture, installations, and constructions. Every time a specific type of information is added into the 3D BIM model, an extra dimension is set and, for this reason, various dimensions have been generated, for example, 4D BIM (construction sequencing), 5D BIM (cost), 6D BIM (energy consumption), and 7D BIM (asset lifecycle). In the era of the cloud, the BIM platform provides a common data environment. By virtue of the cloud-based BIM technology, the creation of piece details is as simple as a button click event. File transfer can be done automatically according to the status of the information. And any change of information will auto-populate. As a result, the BIM-automated processes streamline the repetitive workflow and guarantee the quality of the results.

On the other hand, the fabrication productivity can be maximally improved within the physical constraints by applying a sound production plan. Production planning is a complex process that covers a wide variety of activities to manage a balance between customer demand and available capacity and resources. Scheduling is a critical step in production planning. It fixes the starting and completing date and time for each operation and the amount of work to do. A well-planned production always has a well-planned schedule. As precast concrete elements are usually big, bulky and heavy, and more often than not, require the use of cranes for hoisting, JIT delivery alleviates the space constraints for storage and the traffic congestion on the construction site. Hence, to create a JIT precast concrete production schedule seems a very attractive goal for the industry.

Nevertheless, the success of a JIT implementation relies on many factors such as external economic environments, global and logistical issues, behavioral constraints, intractable accounting systems, small supplier difficulties and so on [17], and moreover, leads to a major organizational change, the companies that want to adopt JIT must consider a trade-off between costs and savings. JIT companies usually bear more inventory risks. Empirically, they cannot tolerate fluctuations of more than 10% in the schedule. Hence, the precast concrete companies are unable to achieve the JIT goals by increasing efficiency on their part alone if their clients and contractors are not equally effective. Surveys and interviews showed that there was a time that the precast concrete companies were reluctant to adopt JIT as they lacked confidence in various factors such as site readiness and unreliable logistics [18].

Now that BIM platforms allow for greater transparency and involvement of project partners, it becomes much easier to coordinate design, fabrication and site operations. Sacks *et al* [19] stated that BIM-enabled visualizations facilitate pull flow and deeper collaboration between teams on and off-site. Moreover, Jeong et al. [20] mentioned that the BIM-integrated simulation aid the reliable prediction of on-site performance, and reliability in production allows all project partners to manage their work with minimum buffers.

Radio frequency identification (RFID) is another emerging technology that RFID tags are now widely used to replace the traditional barcode system in precast concrete factories for managing and tracking precast concrete elements. RFID technology improves visibility from the point of manufacture, throughout the supply chain. It provides a greater degree of flexibility for system control and information dissemination [21]. The supporting role of RFID in the context of lean thinking has been extensively discussed by many researchers [22,23].

Generally, the benefits of BIM and RFID for SMEs are the same as those for larger companies. Therefore, implementation of a JIT schedule in semi-automated precast concrete production lines is practicable on the grounds of risk mitigation and enhanced demand forecast capability thanks to these new technologies [24].

A semi-automated production line

The products that are fabricated in a precast concrete factory fall into two main categories; namely precast

concrete elements and prefabricated modular units. Furthermore, precast concrete elements consist of two types: precast reinforced concrete elements and precast prestressed concrete elements. They include individual structural elements like walls, floors, and roofs, as well as columns, beams, and stairs, while prefabricated modular units refer to free-standing 3D modules which are completed with internal finishes, fixtures, and fittings, examples are prefabricated bathroom unit (PBUs) and prefabricated prefinished volumetric construction (PPVCs). The current study is focused on the fabrication of precast concrete elements, especially, the precast reinforced concrete elements.

Fabrication of precast reinforced concrete elements can use either stationary or mobile systems. When using the stationary system, the positions of the products are fixed from beginning to end. Workers have to carry the tools and materials to search for their assigned jobs in the plant from time to time. One team of workers may need to perform multiple tasks and multiple processes. On the contrary, in the mobile system, workers' positions are fixed, and the production process is divided into different stages, enabling workers to focus on specific tasks. Rather than staying at fixed positions, products move along the processes by means of automated guided vehicles (AGVs), or a rolling track, or a conveyor system. The flow becomes continuous if operating 24/7.

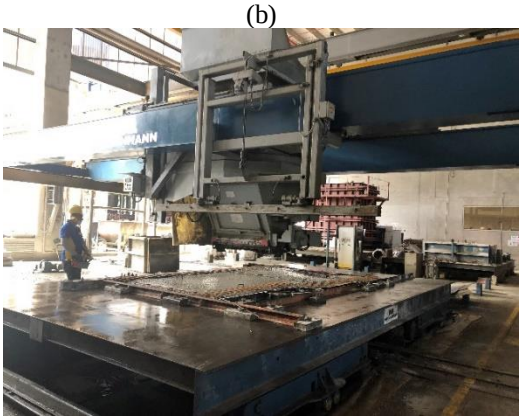
The stationary system is the entry into the precast concrete production, for mass production, the use of a mobile system is more common as the latter improves productivity. The mobile system is also called a pallet circulation system or a carousel plant, in which molds are placed on steel production pallets, moving from one station (which is a fixed space in the plant) to another, with different sequence activities being performed on each station. Operations on the stations can be done by either humans or robots. In the manual handling case, operations on the stations are allowed to change, thus increasing the flexibility. To manufacture products with different types, semi-automated mobile systems are most commonly used, especially in SMEs. A semi-automated production line combines the flexibility of manual processes and the reliability of automated systems.

Converting from a conventional stationary system to a process-oriented mobile system, process reengineering has to be performed on the traditional precast concrete production process [25]. The process flow for the manufacture of precast reinforced concrete elements is now divided into the following sequences: (1) setting mold; (2) placing reinforcement; (3) casting; (4) curing; (5) demolding; and (6) finishing. In a semi-automated plant, some of these processes are conducted with the workers' hands, for example, setting mold, placing reinforcement, demolding, and finishing, while the processes of curing and casting are usually fully automated, see Figure 1.

Figure 2 presents the layout of a typical semi-automated precast concrete plant. Manual operations are performed on fixed space, which are deemed as stations. In this mobile system, pallets are moved from one station to another by rolling tracks. Casting is made by an automatic casting system. Concrete curing is conducted in curing chambers, which are high-rise racks that have plenty of space for curing as well as storing for the work-in-progress (WIPs). There is a computer-controlled conveyor system, enabling fast storage and retrieval. The finished goods are temporarily placed in the storage yard and finally loaded onto trucks by means of a bridge crane.

(a)

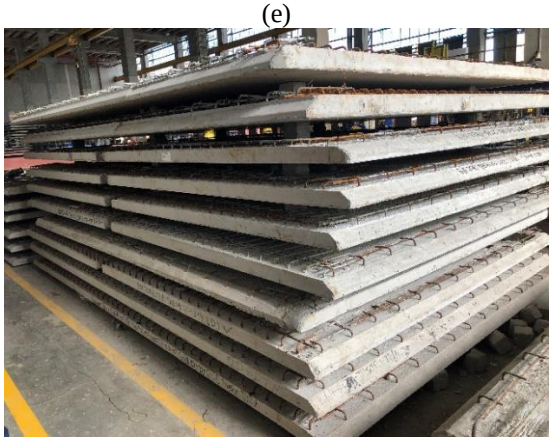
(b)



(c)

(d)

(e)



(f)

Figure 1. Site photos: (a) manually setting mold; (b) manually placing rebar; (c) automatic casting; (d) automatic curing; (e) manually demolding and finishing; (f) finished goods storage

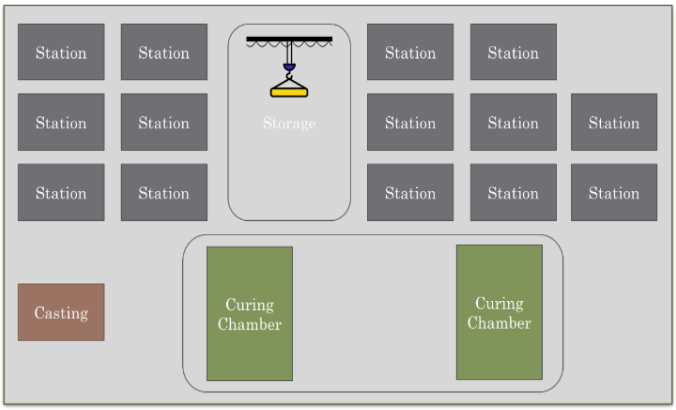


Figure 2. Plant layout of a semi-automated precast concrete production line

The main problems

From a site visit to a middle-sized precast concrete company in Singapore, the following major problems were identified in its semi-automated production line of precast reinforced concrete elements.

- ***Difficult to achieve an efficient schedule*** – If the products are heterogeneous with different requested delivery dates, it is difficult to design an efficient schedule. Some jobs may have to wait for the availability of the stations, whilst some stations have to wait idly for the arrival of new jobs.
- ***Resource conflict*** – Having just one work shift, it creates problems in coordinating access to common resources. For example, under a casting cycle of one day, there will be a crane demand surge in the morning for demolding hardened elements.
- ***High vertical stacks*** – The vertical stacking of finished goods leads to problems when stacking high. Firstly, when products in the middle are wanted, all the others above them must be removed first. Secondly, unstable stacks may become a safety hazard as they may topple and injure workers. Thirdly, quality issues may occur on the bottom products as they perhaps experience excessive stress.
- ***Excessive stock*** – On-time delivery is the most important metric to measure a precast concrete factory's performance. Usually, the factory will overproduce so that it can rapidly respond to demand variations. Hence, excessive stock is a pervasive problem in the precast concrete factory. The surplus consumes a lot of storage space and maybe forgotten resulted from poor inventory management.

The identified problems seem to be prevalent in the middle-sized precast concrete companies. A direct way to solve them fundamentally is to apply a sound production schedule with JIT objective that balances resources and

minimizes inventory level. This paper aims to provide such an optimal production plan to solve these problems.

Precast concrete production modeling

Model description

The precast concrete production scheduling problem is found to possess many of the characteristics of the traditional job shop scheduling problem (JSSP) [14]. JSSP regards production as a continuous flow. In the JSSP, multiple jobs are processed on several machines. Each job consists of a sequence of operations, which must be performed in a given order, and each operation must be processed on a specific machine. The JSSP model suits the automated precast concrete production. However, when a machine to perform which operation becomes uncertain as what usually happens in the semi-automated precast concrete production system, the JSSP model is not very applicable. Therefore, the problem must be re-visited.

The flexible job shop scheduling problem (FJSSP) is an extension of the classical JSSP which allows an operation to be processed by any station from a given set. The FJSSP consists of two sub-problems: the assignment problem and the scheduling problem. The assignment problem is to assign operations to each station such that a station is dedicated to a set of operations, while the scheduling problem is to select stations to process certain operations out of the operation set and determine the processing order of jobs on each station. The planning problem for this semi-automated case can be modeled as a FJSSP.

The problem model is defined as follows: A set of jobs are processed on several different stations. The operation sequence of all the jobs is the same as shown in Figure 3. Stations are flexible to perform different operations, and jobs can revisit the same station for different operations. Limited central storage space is assumed. There are two sub-problems. One is to assign operations on each station, and another is to determine the processing order of jobs at each station.

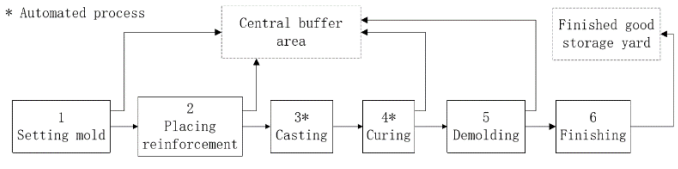


Figure 3. Operation sequence

Assumptions

There are several constraints for the job shop problem. For example, no operation for a job can be started until the previous operation for that job is completed, a machine can only work on one operation at a time, and an operation, once started, must run to completion. In the following, the main model assumptions are summarized.

1. Flexible stations can perform different operations, but each station executes one operation at a time.
2. One station only processes one job at a time and one job can be handled by at most one station at any given time.
3. Operations of a job must be performed in a specific order.
4. A distinction is made between working time, non-working time, and overtime.
5. Processing, set-up, loading, and unloading times are available and are deterministic.
6. The WIPs are stored in a central yard.
7. The operation once started, must run to completion.
8. The first-come-first-serve rule is applied. One job immediately starts its operation when station is available.
9. Such issues as machine failure or downtime, scraps, rework, material and worker unavailability are ignored and left as issues to be considered during real-time control.

Notations

Parameters:

O_i: the operation, 1,…, o

J_j: the job, 1,…,n

M_k: the station, 1,…,m

$P(O_i, J_j, M_k)$: the process time for performing operation O_i of job J_j on station M_k

$C(O_i, J_j, M_k)$: the completion time for performing operation O_i of job J_j on station M_k

$W(O_i, J_j, M_k)$: the wait time of job J_j that finishes operation O_i on station M_k for its next operation

$I(M_k)$: the accumulated time that station M_k has no job

T: the accumulated completion time

$ET_a(J_j)$: the earliest due date for job J_j

$ET_b(J_j)$: the latest due date for job J_j

$B(t)$: the available central buffer size at time t

$\chi(J_j, t)$: the available number of the mold of job J_j at time t

$\alpha(J_j)$: the earliness penalty coefficient

$\beta(J_j)$: the tardiness penalty coefficient

H_w: the working hours provided by a workday

H_N: the non-working hours in a working day

H_E: the allowable overtime in a workday

D; the working days

Decision variables:

$x(O_i, M_k)$: binary variable, 1 if operation i is assigned to station M_k; 0 otherwise

$y(M_k, r)$: integer, 1,…,n , implies the preference rank of a job on station M_k

The following equations are the same as those in the paper of Chan and Hu [14] which defined the problem as the traditional JSSP. Because in the JSSP the problem is deemed as a continuous flow, equation used to calculate the completion time should be:

$$\begin{aligned} C(O_i, J_j, M_k) &= max\{c_1, c_2\} + P(O_i, J_j, M_k) \\ c_1 &= C(O_i, J_{j-1}, M_k) + W(O_i, J_{j-1}, M_k) \\ c_2 &= C(O_{i-1}, J_j, M_l) \end{aligned} \quad (1)$$

where c_1 means the station available time; c_2 means the completion time of the previous operation of the job, M_l is the station that the job just visited. Waiting time on the station incurs when jobs cannot go to the central buffer area because the buffer area is full.

Because except for casting and curing operations which are uninterruptible, the other operations including setting mold, placing reinforcement, demolding, and finishing are interruptible which means they can be stopped if the job cannot be completed within the working hours and can be continued on the next working day. The interrupted operations are formulated in Equation (2):

$$C(O_i = 1,2,5,6, J_j, M_k) = \begin{cases} T & T < 24D + H_w \\ T + \varepsilon \cdot H_N & T \geq 24D + H_w \end{cases} \quad (2)$$

where $D = integer(T/24)$
and $\varepsilon = integer((T - 24D)/H_w)$.

Since concrete casting can be executed without a worker, the casting can be scheduled into overtime. But because casting is an uninterrupted activity, the job must be postponed to the next working day if it cannot be completed within the working hours plus the overtime. The completion time for casting is calculated using Equation (3):

$$C(O_i = 3, J_j, M_k) = \begin{cases} T & T \leq 24D + H_w + H_E \\ 24(D+1) + P(O_i = 3, J_j, M_k) & T > 24D + H_w + H_E \end{cases} \quad (3)$$

Curing is also an uninterrupted activity. Steam curing generally takes 12 to 16 hours without workers in the curing chamber. Thus, it is frequently executed overnight. The completion time can be deemed as the beginning of the next working day. The completion time for curing is formulated using Equation (4):

$$C(O_i = 4, J_j, M_k) = \begin{cases} T & T < 24D + H_w \\ 24(D+1) & T \geq 24D + H_w \end{cases} \quad (4)$$

The steel molds are critical resources for precast concrete production. The molds are assigned according to the job sequence. The new job can enter the production only when there are available station ($c_1 \leq t$) and mold ($\chi(J_j, t) > 0$), besides, the production line is not fully loaded ($B(t) > 0$).

The model
Objective function (Z_1) of the assignment problem is:

$$Minimize\ Z_1 = \omega_1 \cdot Z_2 + \omega_2 \cdot \sum_{M_k} \left\{ \sum_{O_i} \sum_{J_j} W(O_i, J_j, M_k) + I(M_k) \right\} \quad (5)$$

where $\omega_1 + \omega_2 = 1$; the second part refers to the station idle time.

Objective function (Z_2) of the scheduling problem is:

$$\begin{aligned} Minimize\ Z_2 = \omega_3 \cdot T + \omega_4 \cdot \sum_{J_j} \{ & \alpha(J_j) \cdot Max(0, ET_a(J_j) - \\ & T) + \beta(J_j) \cdot Max\left(0, T - ET_b(J_j)\right) \} \end{aligned} \quad (6)$$

where $\omega_3 + \omega_4 = 1$; the first part is the accumulated completion time and the second part is the earliness and tardiness (E/T) penalty for the JIT criterion.
The model subjects to the following constraints:

$$\sum_{M_k} x(O_i, M_k) \geq 1 \quad (7)$$

$$\sum_{O_i} x(O_i, M_k) \geq 1 \quad (8)$$

Equation (7) means that at least one station should perform the operation and Equation (8) means that each station should at least process one operation.

The solving approach

Generally, to solve the assignment and scheduling problems in FJSSP, there are two types of approaches: hierarchical and integrated [26]. In the former approach, the two problems are solved independently. In contrast, the integrated approach solves the two problems dependently. Chan et al. [27] developed such an integrated approach to solving the FJSSP under resource constraints. The objective function of the assignment problem in their work is a weighted function of the makespan which is the objective of the sequencing problem and the machine idle cost. They solved the assignment problem and the sequencing problem iteratively. Similarly, this study will also develop an integrated approach to solve the developed model.

Because the developed model cannot be solved by classical optimization approaches, a simulation-based GA approach is applied. GA is one of the popularly used meta-heuristics to solve scheduling problems [28]. GA is inspired by Charles Darwin's theory of natural evolution. The algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. Five phases are considered in a GA, they are initial population, fitness function, selection, crossover and mutation. The process begins with a set of individuals which is called a population. Each individual is a potential solution of the problem. An individual is characterized by a set of parameters (variables) known as genes. Genes are joined into a string to form a chromosome. The fitness function determines how fit an individual is. It gives a fitness score to each individual. The probability that an individual will be selected for reproduction is based on its fitness score.

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation. Offspring are created by crossover and mutation methods. Crossover is the most significant phase in a GA. Mutation occurs to maintain diversity within the population and prevent premature convergence.

The assignment problem and the scheduling problem are solved by separate GAs. If the generated chromosome for the assignment problem is feasible, we will continue to calculate its fitness by solving the scheduling problem. The fitness values are obtained by running a production simulation. The simulation module is one part of the GA as fitness function. It will return the makespan, the E/T penalty and the station idle time for each GA individual. The evolutionary process of the approach is represented in Figure 4. Details of each step are discussed below.

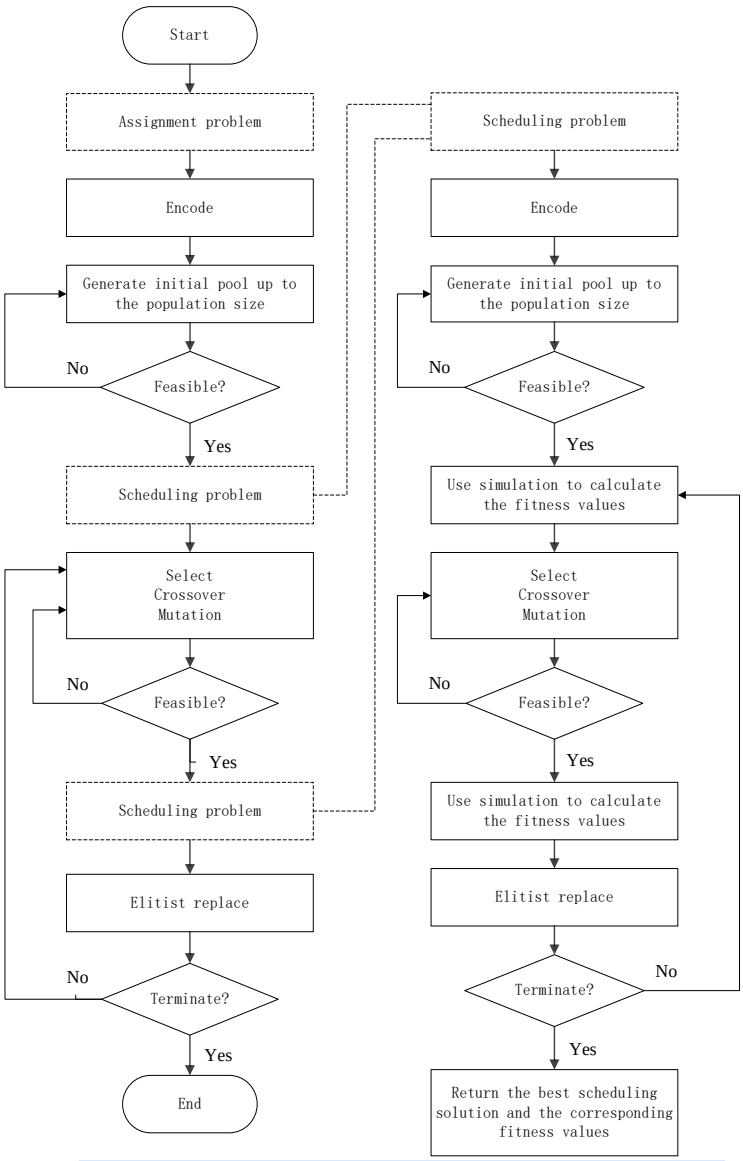


Figure 4. Process of the simulation-based genetic algorithm

Assignment problem

The assignment problem is the outside loop.

Encode

To successfully implement GA, solutions should be encoded into a chromosome structure. Solutions for the assignment problem can be defined as a $o \times m$ matrix:

$$X = \begin{pmatrix} x(1,1) & \cdots & x(1,m) \\ \vdots & x(O_i, M_k) & \vdots \\ x(o,1) & \cdots & x(o,m) \end{pmatrix}$$

where $x(O_i, M_k)$ is a binary gene with a value of either 0 or 1. If $x(O_i, M_k) = 1$, it means that station M_k can process operation O_i. In addition, $x(O_i, M_k)$ subjects to the constraints represented by Equations (7) and (8). If the chromosome is infeasible, it will be corrected by altering one gene randomly from 0 to 1 to meet the minimum requirement.

Initialize population

A set of initial solutions are generated randomly.

Select

Selection is made based on the fitness values. A chromosome that has a higher fitness value has a bigger chance of survival. A Roulette-wheel method is adopted for selection. Roulette-wheel selection is known as fitness proportionate selection. The probability of selection could be imagined to a Roulette wheel in a casino. Usually a proportion of the wheel is assigned to each of the possible selections. Then a random selection by rotating the Roulette wheel.

Crossover

The crossover operator produces the next generation by exchanging partial information from parents. Figure 5 shows the crossover schema. A random column cut point is generated. Then, a portion of the parents is exchanged according to this cut point.

$$\text{Parent A} \begin{pmatrix} V_1 & V_2 & V_3 \\ V_4 & V_5 & V_6 \\ V_7 & V_8 & V_9 \end{pmatrix} \quad \text{Parent B} \begin{pmatrix} W_1 & W_2 & W_3 \\ W_4 & W_5 & W_6 \\ W_7 & W_8 & W_9 \end{pmatrix} \Rightarrow \quad \text{Child A} \begin{pmatrix} W_1 & W_2 & V_3 \\ W_4 & W_5 & V_6 \\ W_7 & W_8 & V_9 \end{pmatrix} \quad \text{Child B} \begin{pmatrix} V_1 & V_2 & W_3 \\ V_4 & V_5 & W_6 \\ V_7 & V_8 & W_9 \end{pmatrix}$$

Figure 5. Crossover schema for station assignment

Mutate

The mutation operator alters the values of genes in the chromosome generated from the crossover operator. Figure 6 shows the mutation schema. A number

$r_1 (1 \le r_1 \le 6)$ is randomly generated. Then, the following procedure is repeated for r_1 times: randomly select a gene and alter its value from 0 to 1, or vice versa.

$$\text{Parent} \begin{pmatrix} V_1 & V_2 & V_3 \\ V_4 & V_5 & V_6 \\ V_7 & V_8 & V_9 \end{pmatrix} \Rightarrow \text{Child} \begin{pmatrix} V_1 & 1-V_2 & V_3 \\ 1-V_4 & V_5 & V_6 \\ V_7 & V_8 & V_9 \end{pmatrix}$$

Figure 6. Mutation schema for station assignment

Elitist replace

Elitist replacement involves copying a small portion of the fittest candidates, unchanged, into the next generation, while the remaining portion is randomly generated.

Scheduling problem

The scheduling problem is embedded inside the assignment problem, solving which gives the fitness values for both problems. In the GA for the scheduling problem, the initial population creation, Roulette-wheel selection, and the elitist replacement are the same as those in the GA for the assignment problem. Below discusses their difference.

Encode

The scheduling problem is encoded by a preference list-based representation [29]. It can be defined as a $m \times n$ matrix:

$$Y = \begin{pmatrix} y(1,1) & \cdots & y(1,n) \\ \vdots & y(M_k, r) & \vdots \\ y(m,1) & \cdots & y(m,n) \end{pmatrix}$$

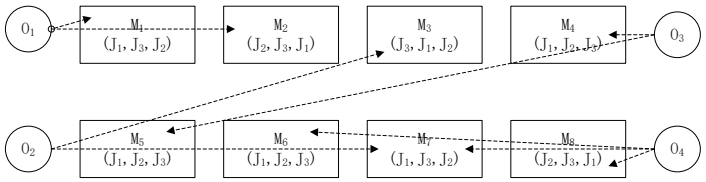
where each matrix row corresponds to a station. M_k denotes the station index. r refers to a sequence location within the preference list. It increments from 1 to n along the column axis. $y(M_k, r)$ is a job index. For example, let $y(M_k, r_1) = J_a$ and $y(M_k, r_2) = J_b$. If $r_1 < r_2$, it means that J_a will be preferred than J_b on station M_k. Rather than strictly controlling the job processing orders, the representation scheme allows available jobs with higher priorities in the list to be processed first.

Simulation

Figure 7 gives an example with eight stations, four operations, and three jobs, to show how to deduce an actual schedule from the given chromosomes. From the preference lists for the stations, one can deduce the schedulable preferential operations according to the given operation precedence constraints. The core simulation principle is as follows: When a station is free, it will request a job. Those executable jobs will compete for the station. But the one with the highest rank in the preference list of that station will win. For example, in the case, for stations M_1 and M_2, the first winners are jobs J_1 and J_2, respectively. However, because the start of the first operation also depends on mold availability and production capacity, if J_1 is waiting for its mold, the next preferential job will be J_3 according to the preference list, and so on.

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$y = \begin{pmatrix} J_1 & J_3 & J_2 \\ J_2 & J_3 & J_1 \\ J_3 & J_1 & J_2 \\ J_1 & J_2 & J_3 \\ J_1 & J_2 & J_3 \\ J_1 & J_2 & J_3 \\ J_1 & J_3 & J_2 \\ J_2 & J_3 & J_1 \end{pmatrix}$$



Station	Job schedule				
M_1	J_1	J_3			
M_2	J_2				
M_3		J_2	J_3		
M_4			J_1	J_3	
M_5			J_2		
M_6				J_1	
M_7		J_1			J_3
M_8				J_2	

Figure 7. An example with eight stations, four operations, and three jobs

Evaluation functions

Equations (5) and (6), which are weighted sums of two criteria, are used as fitness functions. But the two criteria in the fitness function may not be of the same magnitude in some cases, perhaps the contribution of one criterion is dominated by another. In this regard, it is suggested to normalize them in order to handle the two criteria equally [30]. However, the scalar weight factors are likely to be problem-dependent and it is not easy to choose their proper values. The following linear equation is used to normalize each criterion in the fitness functions.

$$f_m = \begin{cases} 1 & m > m_b \\ \frac{m - m_a}{m_b - m_a} & m_a < m \le m_b \\ 0 & m \le m_a \end{cases} \quad (9)$$

$$f_t = \begin{cases} 1 & t > t_b \\ \frac{t-t_a}{t_b-t_a} & t_a < t \le t_b \\ 0 & t \le t_a \end{cases} \quad (10)$$

where m and t denote the values of the two criteria, respectively; m_a and t_a denote the minimum values obtained by using a *single* objective GA search; and m_b and t_b denote the maximum values from the single-objective GA. Therefore, the fitness function can be represented as Equation (11)

$$f = \omega_m \cdot f_m + \omega_t \cdot f_t \quad (11)$$

where ω_m is a random real number over the closed interval [0, 1]; and $\omega_t = 1 - \omega_m$.

Crossover

Two parents are selected, and a single point crossover is applied between them. As Figure 8 shows, the crossover point is placed between job J_1 and job J_2. Then, the child inherits the sequences of job J_1 from Parent A. The remaining blanks in the child are filled by referring to the unselected jobs (i.e., job J_2 and job J_3) in Parent B.

Parent A $\begin{pmatrix} \boldsymbol{J_1} & J_2 & J_3 \\ \boldsymbol{J_1} & J_3 & J_2 \\ J_3 & J_2 & \boldsymbol{J_1} \end{pmatrix}$

Parent B $\begin{pmatrix} J_3 & J_1 & J_2 \\ J_1 & J_2 & J_3 \\ J_2 & J_1 & J_3 \end{pmatrix}$ ⇨ Child $\begin{pmatrix} J_1 & J_3 & J_2 \\ J_1 & J_2 & J_3 \\ J_2 & J_3 & J_1 \end{pmatrix}$

Figure 8. Crossover schema for job sequence

Mutate

Figure 9 shows the mutation schema. The sequences of two randomly selected jobs in one row are interchanged.

Parent $\begin{pmatrix} J_1 & \boldsymbol{J_2} & \boldsymbol{J_3} \\ J_1 & J_3 & J_2 \\ J_3 & J_2 & J_1 \end{pmatrix}$ ⇨ Child $\begin{pmatrix} J_1 & J_3 & J_2 \\ J_1 & J_3 & J_2 \\ J_3 & J_2 & J_1 \end{pmatrix}$

Figure 9. Mutation schema for job sequence

Experimental case study

The proposed algorithm was coded in C# language and tested on a laptop computer. In this section, a test problem will be employed to examine the performance of the proposed algorithm.

Table 1 presents the production data of the test problem which comprises 10 precast concrete element types and 3 mold types. The data was adapted from Benjaoran et al. [31], in which O1 (setting mold), O2 (placing reinforcement), O5 (demolding), and O6 (finishing) are manual operations and performed on flexible stations, while O3 (casting) and O4 (curing) are automated operations and performed by casting machines and curing chambers, respectively. Besides, the normal working hours per day is 8 hours, and the maximum overtime for casting per workday is 4 hours. This sample data was also used by other researchers to test their developed algorithms, such as Ko and Wang [15].

Because O_4 (curing) automatically starts without workers after O_3 (casting). O_3 and O_4 can be deemed as one group. Besides, the stations for O_3 and O_4 are not flexible for the other operations. As a result, their corresponding gene values in the chromosome of the assignment problem are fixed to be "1". Assuming a six-station scenario where there are four flexible stations that can perform any operations and two nonflexible stations that are reserved for O_3 and O_4, the chromosome structure of the assignment problem will be:

$$X = \begin{pmatrix} x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ x & x & x & x & 0 & 0 \\ x & x & x & x & 0 & 0 \end{pmatrix} \begin{matrix} O_1 \\ O_2 \\ O_3, O_4 \\ O_5 \\ O_6 \end{matrix}$$

in which x denotes a binary value.

The related GA parameter settings are as follows: The crossover rate is 0.95, the mutation rate is 0.2, and the elitist ratio is 0.25. These parameters were determined according to the results from previous studies [32-34] through a trial and error approach. Solution quality and computational time are usually positively correlated. However, the computational cost associated with an integrated and iterative approach can be very huge. In this regard, the compromised goal becomes to generate solutions in a relatively reasonable computational time. One can decide the choice of the appropriate population size and generations also using a trial and error approach.

Table 2 compares the solutions under different population size and iteration times. One can see that a smaller population over shorter generations gives the same makespan results as using a bigger population over longer generations. Even for the E/T penalty and the station idle time, the difference is not significant. Hence, we recommend using a smaller population and multiple isolated shorter runs. We pick up the individual who receives the best fitness value as the final solution for the problem. The use of multiple smaller runs is also recommended by the results of many empirical studies as the savings on execution time seems marginal when compared to a single longer run [35]. Finally, evolution over 100 generations with population size 20 was executed and repeated 10 times in the following

experiments considering the balance between solution quality and computational effort.

Using a single-objective GA (either the makespan, or the E/T penalty, or the station idle time), the possible minimum and maximum values of the three objectives (i.e., the makespan, the E/T penalty and the station idle time) can be obtained, as shown in Table 3, which will be used to normalize the fitness values in the multi-objective GA. Table 4 provides the optimal results using the multi-objective GA. The corresponding weights were randomly generated. Playing with the weights one can generate different Pareto optimal solutions as required.

Figure 10 shows different schedules with and without resource constraints. On the one hand, Figure 10(a) and Figure 10(c) show the cases with sufficient mold supply, which means molds are always available whenever they are wanted. Therefore, new jobs can immediately start their first operations when the relevant stations become empty. On the other hand, Figure 10(b) shows the case without sufficient mold supply. In this regard, the start of new jobs is constrained by not only the availability of the stations but also the availability of the molds. In Figure 10(a) and Figure 10(b), there are four flexible stations and two nonflexible stations. In Figure 10(c) there are three flexible stations and three nonflexible stations. The flexible stations can perform O1 (setting mold), O2 (placing reinforcement), O5 (demolding) or O6 (finishing) while the nonflexible stations can only perform O3 & O4 (casting and curing). Storage capacity is unlimited. The 10 jobs circulate around different stations according to their operation sequences, respectively. The obtained minimum makespan in Figure 10(a) is 121.6 hours, in Figure 10(b) is also 121.6 hours, in Figure 10(c) is 96.8 hours.

The main finding by comparing the cases of different flexible stations with and without enough mold supply is that a small supply of flexible stations and molds do not necessarily lead to longer makespan, sometimes, even gives better results as Figure 10(a) and Figure 10(c) demonstrate. The reason is due to the assumption applied in the model that one job immediately starts operation when station is available. As a result, more new jobs may be invited when there are more flexible stations which can perform the first operations of the jobs. The increased number of jobs that are concurrently in the system is likely to cause more waiting wastes. One can see that there are more jobs in the storage in Figure 10(a) since the nonflexible stations become bottlenecks when there are more jobs concurrently in the system and less nonflexible stations provided than in Figure 10(c). Therefore, it is suggested to run different combinations in order to find the best one.

We further take a closer look to the schedule presented in Figure 10(b). Its corresponding optimal solution are given as below:

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} O_1 \\ O_2 \\ O_{3,4} \\ O_5 \\ O_6 \end{matrix}$$

$$Y = \begin{pmatrix} 8 & 1 & 4 & 6 & 2 & 3 & 5 & 9 & 7 & 10 \\ 1 & 5 & 4 & 2 & 9 & 3 & 10 & 7 & 6 & 8 \\ 6 & 10 & 4 & 1 & 2 & 5 & 9 & 3 & 7 & 8 \\ 4 & 7 & 2 & 8 & 6 & 1 & 5 & 3 & 9 & 10 \\ 8 & 2 & 3 & 1 & 10 & 4 & 7 & 5 & 9 & 6 \\ 4 & 2 & 6 & 9 & 1 & 5 & 10 & 3 & 7 & 8 \end{pmatrix} \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix}$$

Despite of the same makespan as Figure 10(a), the detailed schedules are different. Mold availability is the same important as station availability. In Figure 10(b), the new job will not start until its mold is available when one job that uses the same mold has finished all operations and left the production system. For example, according to the job preference list of Station 3, Job 4 and Job 1 can start earlier than Job 2. However, because Job 4 and Job 1 need to use Mold A, they must wait for it until Job 6 has finished using it. Consequently, Job 2 which uses Mold B is awarded the priority.

Table 1. Production data

Job ID	Mold	Processing time (hours)						Due dates (hours)	Penalty rate per hour	
		O_1	O_2	$O_3{}^1$	$O_4{}^1$	O_5	O_6		Earliness	Tardiness
1	A	2	1.6	2.4	12	2.5	1	112	2	10
2	B	3.4	4	4	12	2.4	5	112	2	10
3	A	0.8	1	1.2	12	0.8	0	112	1	10
4	A	0.6	0.8	1	12	0.6	2	112	1	10
5	C	3	3.6	2.4	12	2.4	3	208	2	10
6	A	3	3.2	3	12	3	1.6	128	2	10
7	C	1.3	0.9	2.4	12	1.9	1.8	144	2	10
8	B	1.7	1.4	1.1	12	0.9	0.7	144	2	20
9	A	2.2	1.8	1.2	12	2.3	0.7	144	1	20
10	C	1.6	3.2	2.3	12	2.1	2.7	240	1	20

Table 2. Results of using different population sizes over different iteration times

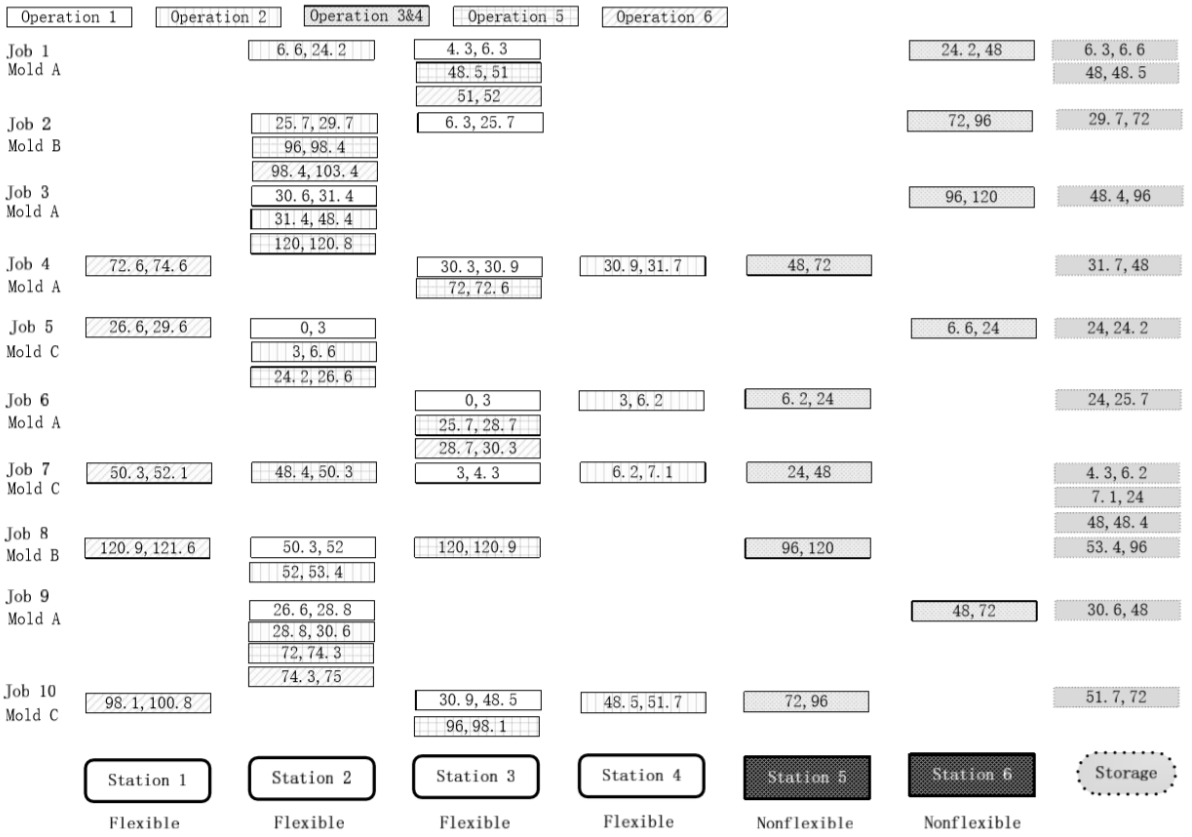
Population size	Generation	Makespan (hours)	CPU time (seconds)	E/T penalty (units)	CPU time (seconds)	Station idle time (hours)	CPU time (seconds)
20	20	121.6	99	534.6	98	3.2	99
20	50	121.6	621	467.1	624	3.2	639
20	100	121.6	2478	397.5	2494	3.2	2547
50	50	121.6	4141	413.6	4107	3.2	4152
50	100	121.6	16222	397.5	16441	3.2	16131
100	50	121.6	18106	397.5	18062	3.2	18207
100	100	121.6	74397	397.5	74409	3.2	74458

Table 3. Results of 10 attempts using single-objective GA

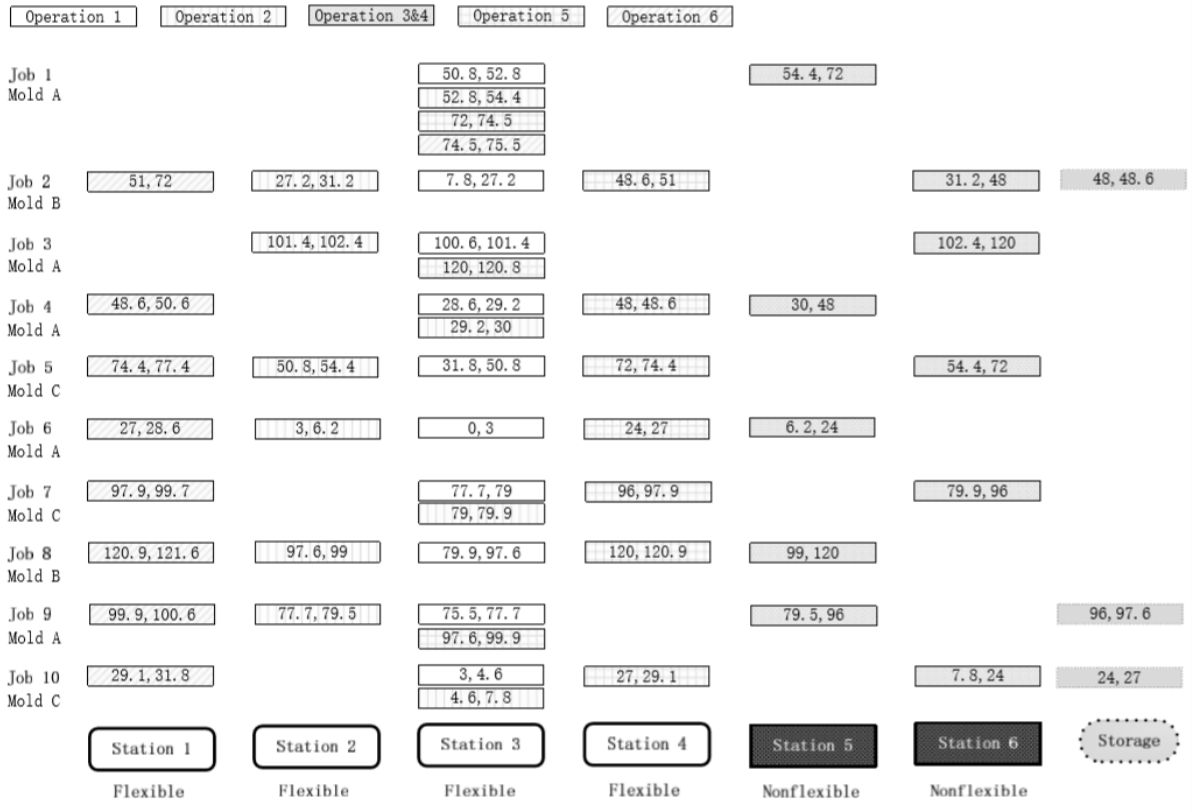
Attempt	Makespan (hours)				E/T penalty (units)				Station idle time (hours)			
	Min	CPU time (seconds)	Max	CPU time (seconds)	Min	CPU time (seconds)	Max	CPU time (seconds)	Min	CPU time (seconds)	Max	CPU time (seconds)
1st	121.6	2502	246.3	2489	397.5	2521	7776.7	2520	3.2	2512	742.7	2503
2nd	121.6	2545	246.3	2540	397.5	2498	7813.3	2518	3.2	2528	759.3	2526
3rd	121.6	2490	246.3	2525	397.5	2501	7813.3	2503	3.2	2517	759.3	2506
4th	121.6	2511	245.4	2499	397.5	2534	7776.7	2529	3.2	2524	759.3	2534
5th	121.6	2513	246.3	2517	397.5	2499	7776.7	2542	3.2	2531	759.3	2522
6th	121.6	2503	246.3	2524	397.5	2547	7776.7	2534	3.2	2523	759.3	2517
7th	121.6	2489	245.4	2533	397.5	2519	7777.4	2527	3.2	2511	759.3	2533
8th	121.6	2508	245.4	2528	397.5	2488	7776.7	2536	3.2	2541	745.1	2529
9th	121.6	2522	246.3	2496	397.5	2533	7813.3	2528	3.2	2533	745.1	2492
10th	121.6	2536	245.4	2512	397.5	2526	7813.3	2535	3.2	2528	758.8	2530
Best	121.6		246.3		397.5		7813.3		3.2		759.3	

Table 4. Optimal results using multi-objective GA

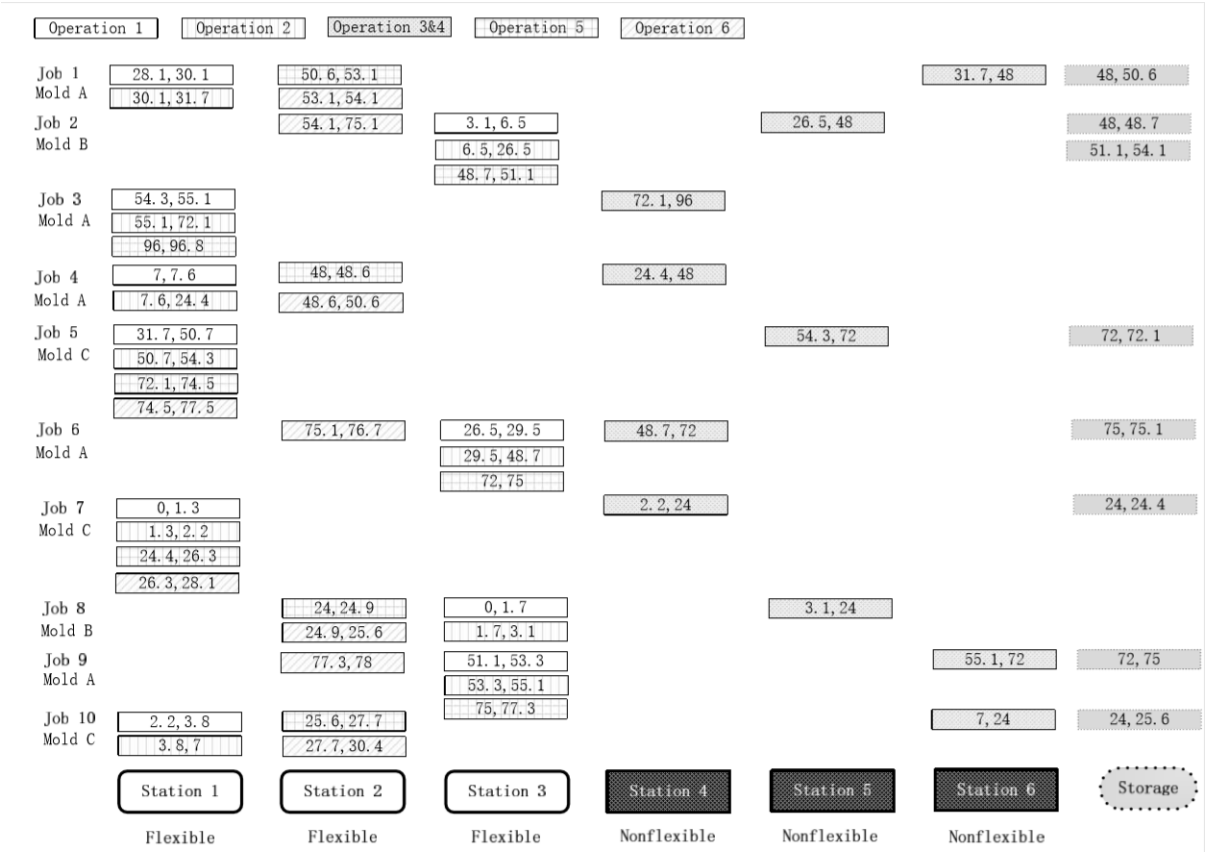
Index	ω_1	ω_2	ω_3	ω_4	Makespan (hours)	E/T penalty (units)	Station idle time (hours)	Assignment objective (units)	Scheduling objective (units)
1	0.01	0.99	0.77	0.23	126.1	831.2	7.7	0.006305	0.041238
2	0.14	0.86	0.09	0.91	127.9	827.3	3.7	0.008589	0.057288
3	0.17	0.83	0.33	0.67	126.2	837.2	6.2	0.012116	0.051899
4	0.23	0.77	0.86	0.14	125.4	897.2	7	0.012067	0.035641
5	0.36	0.64	0.46	0.54	126.4	790.1	6.2	0.019205	0.046295
6	0.46	0.54	0.24	0.76	127.8	773.6	7	0.025933	0.050477
7	0.57	0.43	0.02	0.98	148.8	677	3.2	0.023540	0.041298
8	0.58	0.42	0.29	0.71	127	769	6.7	0.029857	0.048126
9	0.63	0.37	0.22	0.78	127.5	762.2	6.5	0.032339	0.048768
10	0.68	0.32	0.98	0.02	125.4	890.1	8.5	0.023454	0.031192
11	0.71	0.29	0.55	0.45	126.2	795.1	8.2	0.033453	0.044416
12	0.87	0.13	0.28	0.72	126.6	771.4	8.8	0.042313	0.047529
13	0.89	0.11	0.12	0,98	125.6	852	62.2	0.060011	0.057783
14	0.90	0.10	0.73	0.27	121.7	915	31.2	0.021187	0.019427
15	0.99	0.01	0.71	0.29	121.7	906	67.2	0.021096	0.020455


Operation 1
Operation 2
Operation 3&4
Operation 5
Operation 6
Job 1 Mold A
6. 6, 24. 2
4. 3, 6. 3
48. 5, 51
51, 52
24. 2, 48
6. 3, 6. 6
48, 48. 5
Job 2 Mold B
25. 7, 29. 7
96, 98. 4
98. 4, 103. 4
6. 3, 25. 7
72, 96
29. 7, 72
Job 3 Mold A
30. 6, 31. 4
31. 4, 48. 4
120, 120. 8
96, 120
48. 4, 96
Job 4 Mold A
72. 6, 74. 6
30. 3, 30. 9
72, 72. 6
30. 9, 31. 7
48, 72
31. 7, 48
Job 5 Mold C
26. 6, 29. 6
0, 3
3, 6. 6
24. 2, 26. 6
6. 6, 24
24, 24. 2
Job 6 Mold A
0, 3
25. 7, 28. 7
28. 7, 30. 3
3, 6. 2
6. 2, 24
24, 25. 7
Job 7 Mold C
50. 3, 52. 1
48. 4, 50. 3
3, 4. 3
6. 2, 7. 1
24, 48
4. 3, 6. 2
7. 1, 24
48, 48. 4
Job 8 Mold B
120. 9, 121. 6
50. 3, 52
52, 53. 4
120, 120. 9
96, 120
53. 4, 96
Job 9 Mold A
26. 6, 28. 8
28. 8, 30. 6
72, 74. 3
74. 3, 75
48, 72
30. 6, 48
Job 10 Mold C
98. 1, 100. 8
30. 9, 48. 5
96, 98. 1
48. 5, 51. 7
72, 96
51. 7, 72
Station 1
Station 2
Station 3
Station 4
Station 5
Station 6
Storage
Flexible
Flexible
Flexible
Flexible
Nonflexible
Nonflexible

(a)


Operation 1
Operation 2
Operation 3&4
Operation 5
Operation 6
Job 1 Mold A
50. 8, 52. 8
52. 8, 54. 4
72, 74. 5
74. 5, 75. 5
54. 4, 72
Job 2 Mold B
51, 72
27. 2, 31. 2
7. 8, 27. 2
48. 6, 51
31. 2, 48
48, 48. 6
Job 3 Mold A
101. 4, 102. 4
100. 6, 101. 4
120, 120. 8
102. 4, 120
Job 4 Mold A
48. 6, 50. 6
28. 6, 29. 2
29. 2, 30
48, 48. 6
30, 48
Job 5 Mold C
74. 4, 77. 4
50. 8, 54. 4
31. 8, 50. 8
72, 74. 4
54. 4, 72
Job 6 Mold A
27, 28. 6
3, 6. 2
0, 3
24, 27
6. 2, 24
Job 7 Mold C
97. 9, 99. 7
77. 7, 79
79, 79. 9
96, 97. 9
79. 9, 96
Job 8 Mold B
120. 9, 121. 6
97. 6, 99
79. 9, 97. 6
120, 120. 9
99, 120
Job 9 Mold A
99. 9, 100. 6
77. 7, 79. 5
75. 5, 77. 7
97. 6, 99. 9
79. 5, 96
96, 97. 6
Job 10 Mold C
29. 1, 31. 8
3, 4. 6
4. 6, 7. 8
27, 29. 1
7. 8, 24
24, 27
Station 1
Station 2
Station 3
Station 4
Station 5
Station 6
Storage
Flexible
Flexible
Flexible
Flexible
Nonflexible
Nonflexible

(b)



(c)

Figure 10. Minimum makespan schedule: (a) four flexible stations; (b) one mold for each mold type; (c) three flexible stations

Conclusions

This study established a practical production model based on the flexible job shop scheduling model for the semi-automated production status in current precast concrete factories. The model consists of two sub-problems: the assignment problem which assigns operations to each station such that a station is dedicated to a set of operations; and the scheduling problem which selects stations to process certain operations out of the set and determines the processing order of jobs on each station. The assignment problem and the scheduling problem are solved using an integrated approach. A multi-objective GA is applied to search for the best assignment plan with minimum makespan and station idle time. For every assignment plan, another multi-objective GA is applied to search for the best scheduling plan with minimum makespan and earliness and tardiness penalty.

Although a fully automated production line is the ideal state, most precast concrete factories have yet to be developed to that stage. From many factory tours, it is found that semi-automated production lines are popularly used. The semi-automated production line combines human labor and automation. Compared to the fully automated one, it has more flexibility that one station can perform multiple operations, whereby in addition to a schedule, an operation assignment plan is also required. Therefore, the production planning problem is formulated into an FJSSP model, an extension of the classical JSSP model.

An example was applied to validate the integrated GA approach. The experimental results show that the developed GA approach is a useful and effective method for solving the problem that it can return high-quality solutions. Usually, the computational cost associated with an integrated and iterative approach is very huge. So, for further study, additional experiments on real situations with different input data sizes should be conducted in order to arrive at a more definite conclusion on the efficacy of the developed GA approach in solving the FJSSP for semi-automated precast concrete production planning. Furthermore, we wish to develop an active scheduling system which can propose actions to users automatically when changes are required based on an agent-based system.

Data Availability Statement

Some or all data, models, or code generated or used during the study are available from the corresponding author by request.

Acknowledgement

This research was funded by grant number M4062549 from the JTC, Singapore. The authors are grateful to the founders of Robin Village Development Pte Ltd, Singapore, Mr. and Mrs. Seah, for their support of the research that resulted in the findings reported in this paper.

References

[1] Prefabrication Technology (2018). https://www.hdb.gov.sg/cs/infoweb/about-us/research-and-innovation/construction-productivity/prefabrication-technology

[2] Kim, S. E., & Zuhairi, A. H. (2017). *Modernisation, Mechanisation and Industrialisation of Concrete Structures* (1ed.). Wiley-Blackwell. https://doi.org/10.1002/9781118876503

[3] Zhang, L., Narkhede, B. E., & Chaple, A. P. (2017). "Evaluating lean manufacturing barriers: an interpretive process." *Journal of Manufacturing Technology Management*, 28(8), 1086-1114. https://doi.org/10.1108/JMTM-04-2017-0071

[4] Oliveira, R. I. d., Sousa, S. O., & Campos, F. C. d. (2019). "Lean manufacturing implementation: bibliometric analysis 2007-2018." *International Journal of Advanced Manufacturing Technology*, 101, 979-988. https://doi.org/10.1007/s00170-018-2965-y

[5] Shah, R., & Ward, P. T. (2003*). "*Lean manufacturing: context, practice bundles, and performance." *Journal of Operations Management*, 21(2), 129-149. https://doi.org/10.1016/S0272-6963(02)00108-0

[6] Monden, Y. (2011). *Toyota production system: an integrated approach to just-in-time* (4ed.). CRC Press, Taylor & Francis Group, Boca Raton, FL, USA.

[7] Liker, J. K. (2004). *The Toyota way: 14 management principles from the world's greatest manufacturer* (1ed.). McGraw-Hill Education, USA.

[8] Bucourt, M. d., Busse, R., Güttler, F., Wintzer, C., Collettini, F., Kloeters, C., et al. (2011). "Lean manufacturing and Toyota Production System terminology applied to the procurement of vascular stents in interventional radiology." *Insights Imaging*, 2(4), 415-423. https://doi.org/10.1007/s13244-011-0097-0

[9] Phang, T. C. H., Chen, C., & Tiong, R. L. K. (2019). "New model for identifying critical success factors influencing BIM adoption from precast concrete manufacturers' view." *Journal of Construction Engineering and Management*, 146(4). https://doi.org/10.1061/(ASCE)CO.1943-7862.0001773

[10] Yadv, V., Jain, R., Mittal, M. L., Panwar, A., & Lyons, A. C. (2019). "The propagation of lean thinking in SMEs." *Production planning & control*, 30(10-12), 854-865. https://doi.org/10.1080/09537287.2019.1582094

[11] Dubey, R., & Singh, T. (2014). "Understanding complex relationship among JIT, lean behavior, TQM and their antecedents using interpretive structural modelling and fuzzy MICMAC analysis." *The TQM Journal*, 27(1), 42-62. https://doi.org/10.1108/TQM-09-2013-0108

[12] Dowlatshahi, S., & Taham, F. (2009). "The development of a conceptual framework for Just-In-Time implementation in SMEs." *Production Planning and Control*, 20(7), 611-621. https://doi.org/10.1080/09537280903034305

[13] Wong, J., Wang, X., Li, H., Chan, G., & Li, H. (2014). "A review of cloud-based BIM technology in the construction sector." *Journal of Information Technology in Construction*, 19, 281-291. http://www.itcon.org/2014/16

[14] Chan, W. T., & Hu, H. (2002). "Production scheduling for precast plants using a flow shop sequencing model." *Journal of Computing in Civil Engineering*, 16(3), 165-174. https://doi.org/10.1061/(ASCE)0887-3801(2002)16:3(165)

[15] Ko, C.-H., & Wang, S.-F. (2010). "GA-based decision support systems for precast production planning." *Automation in Construction*, 19, 907-916. https://doi.org/10.1016/j.autcon.2010.06.004

[16] Nath, T., Attarzadeh, M., Tiong, R. L. K., Chidambaram, C., & Zhao, Y. (2015). "Productivity improvement of precast shop drawings generation through BIM-based process re-engineering." *Automation in Construction*, 54, 54-68. https://doi.org/10.1016/j.autcon.2015.03.014

[17] Polito, T., & Watson, K. (2006). "Just-in-time under fire: the five major constraints upon JIT practices." *The Journal of American Academy of Business*, 9(1), 8-13.

[18] Low, S. P., & Choong, J. C. (2001). "Just-in-time management of precast concrete components." *Journal of Construction Engineering and Management*, 127(6), 494-501. https://doi.org/10.1061/(ASCE)0733-9364(2001)127:6(494)

[19] Sacks, R., Radosavljevic, M., & Barak, R. (2010). "Requirements for building information modeling based lean production management systems for construction." *Automation in Construction*, 19(5), 641-655. https://doi.org/10.1016/j.autcon.2010.02.010

[20] Jeong, W. S., Chang, S., Son, J. W., & Yi, J.-S. (2016). "BIM-integrated construction operation

simulation for just-in-time production management." *Sustainability*, 8(11), 1-25. https://doi.org/10.3390/su8111106

[21] Samuel Y. L. Yin, H. P. T., J. C. Wang, S. C. Tsai (2009). "Developing a precast production management system using RFID technology." *Automation in Construction*, 18(5), 677-691. https://doi.org/10.1016/j.autcon.2009.02.004

[22] Powell, D., & Skjelstad, L. (2012). "RFID for the extended lean enterprise." *International Journal of Lean Six Sigma*, 3(3), 172-186. https://doi.org/10.1108/20401461211282691

[23] Ghelichi, A., & Abdelgawad, A. (2014). "A study on RFID-based Kanban system in inventory management." *Proceedings, IEEE International Conference on Industrial Engineering and Engineering Management, IEEM*, Selangor Darul Ehsan, Malaysia, Dec. 9-12, 2014, pp. 1357-1361. https://doi.org/10.1109/IEEM.2014.7058860

[24] Li, C., Zhengdao, Zhong, R. Y., Xue, F., Xu, G., Chen, K., Huang, G. G., et al. (2017). "Integrating RFID and BIM technologies for mitigating risks and improving schedule performance of prefabricated house construction." *Journal of Cleaner Production*, 165, 1048-1062. https://doi.org/10.1016/j.jclepro.2017.07.156

[25] Chen, J.-H., Yang, L.-R., & Tai, H.-W. (2016). "Process reengineering and improvement for building precast production." *Automation in Construction*, 68, 248-258. https://doi.org/10.1016/j.autcon.2016.05.015

[26] Brucker, P., & Schlie, R. (1990). "Job-shop scheduling with multi-purpose machines." *Computing*, 45, 369-375. https://doi.org/10.1007/BF02238804

[27] Chan, F. T. S., Wong, T. C., & Chan, L. Y. (2006). "Flexible job-shop scheduling problem under resource constraints." *International Journal of Production Research*, 44(11), 2071-2089. https://doi.org/10.1080/00207540500386012

[28] Lancaster, J. & Ozbayrak, M. (2007). "Evolutionary algorithms applied to project scheduling problems - a survey of the state-of-the-art." *International Journal of Production Research*, 45(2), 425-450. https://doi.org/10.1080/00207540600800326

[29] Cheng, R., Mitsuo, G., & Tsujimura, Y. (1996). "A tutorial survey of job-shop scheduling problems using genetic algorithms - I. representation." *Computers and Industrial Engineering*, 30(4), 983-997. https://doi.org/10.1016/0360-8352(96)00047-2

[30] Ishibuchi, H., & Murata, T. (1998). "A multi-objective genetic local search algorithm and its application to flowshop scheduling." *IEEE Transaction on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(3), 392-403. https://doi.org/10.1109/5326.704576

[31] Benjaoran, V., Nashwan, D., & Hobbs, B. (2005). "Flowshop scheduling model for bespoke precast concrete production planning." *Construction Management and Economics*, 23(1), 93-105. https://doi.org/10.1080/0144619042000287732

[32] Grefenstette, J. J. (1986). "Optimization of control parameters for genetic algorithms." *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), 122-128. https://doi.org/10.1109/TSMC.1986.289288

[33] Schaffer, J. D., Caruana, R., Eshelman, L. J., & Das, R. (1989). "A study of control parameters affecting online performance of genetic algorithms for function optimization." *Proceedings, 3rd International Conference on Genetic Algorithms*, George Mason University, Fairfax, Virginia, USA, Jun. 1989, pp. 51-60.

[34] Mitchell M. (1996). *An introduction to genetic algorithms*. MIT Press: Cambridge. ISBN: 9780262280013.

[35] Goldberg, D. E. (1989). "Sizing populations for serial and parallel genetic algorithms." *Proceedings, 3rd International Conference on Genetic Algorithms*, George Mason University, Fairfax, Virginia, USA, Jun. 1989, pp. 70-79.

