Multi-objective Metaheuristic Algorithms for Precast Production Scheduling Problems

Lehuang Zong

A Thesis Submitted in Fulfillment of the Requirements for the Degree of Master of Engineering in Industrial and Systems Engineering
Prince of Songkla University
2019

Thesis Title Multi-objective Metaheuristic Algorithms for Precast Production Scheduling Problems

Author Mr. Lehuang Zong

Major Program Industrial and Systems Engineering

Major Advisor	**Examining Committee:**
....	....Chairperson
(Asst. Prof. Dr. Wanatchapong Kongkaew)	(Assoc. Prof. Dr. Nikorn Sirivongpaisal)
	....Committee
	(Asst. Prof. Dr. Sorawit Yaovasuwanchai)
	....Committee
	(Asst. Prof. Dr.Wannarat Suntiamorntut)
	……..Committee
	(Asst. Prof. Dr. Wanatchapong Kongkaew)

The Graduate School, Prince of Songkla University, has approved this thesis as fulfillment of the requirements for the Master of Engineering Degree in Industrial and Systems Engineering.

……………………………………
(Assoc. Prof. Dr. Teerapol Srichana)
Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

…………………………………….Signature
(Asst. Prof. Dr. Wanatchapong Kongkaew)
Major Advisor

……..………………....………….Signature
(Mr. Lehuang Zong)
Candidate

I hereby certify that this work has not already been accepted in substance for any degree, and is not being concurrently submitted in candidature for any degree.

……..…………………….Signature

(Mr. Lehuang Zong)

Candidate

Thesis Title Multi-objective Metaheuristic Algorithms for Precast Production Scheduling Problems

Author Mr. Lehuang Zong

Major Program Industrial and Systems Engineering

Academic Year 2018

ABSTRACT

Researchers have developed multi-objective precast production scheduling models (MOPPSM) to describe practical constraints and objectives encountered in precast manufacturing. To adapt to realistic customer orders, this study improved MOPPSM by considering lot delivery of precast components (MOPPSM-LD). Furthermore, we firstly proposed two competitive metaheuristics called multi-objective variable neighbourhood search (MOVNS) and non-dominated sorting genetic algorithm II (NSGA-II) to optimise both MOPPSM and MOPPSM-LD. The performance of the two algorithms, measured by spread and distance metrics, were compared with a benchmark algorithm called multi-objective genetic local search (MOGLS). The experimental results showed that MOVNS and NSGA-II can successfully solve both MOPPSM and MOPPSM-LD problems. And the MOVNS outperformed NSGA-II and MOGLS while the NSGA-II was capable of searching as good as the MOGLS.

Keywords: precast production scheduling, variable neighbourhood search, NSGA-II, multi-objective optimisation, distance metrics

ACKNOWLEDGEMENTS

I hereby would like to extend my sincere gratitude and indebtedness to my supervisor, Asst. Prof. Dr. Wanatchapong Kongkaew, for his careful and liberal guidance, patience, exact insights and stimulating suggestions in supervising my research throughout my master study at Prince of Songkla University (PSU). During these past 2 years he motivated, supported and believed in me. He also showed utmost solicitude for my studies, works and even my life. I have benefited greatly from his vast technical expertise and insight. Without his precise and appropriate instructions at every step during my research, I could not accomplish my postgraduate study smoothly.

I would also like to thank two external professors, Prof. Deming Lei who is from School of Automation, Wuhan University of Technology, in China for providing valuable insights and suggestions. Also, I would like to thank all the committee members; Asst. Prof. Dr. Sorawit Yaovasuwanchai from King Mongkut's University of Technology Thonburi, Assoc. Prof. Dr.Nikorn Sirivongpaisal, and Asst. Prof. Dr.Wannarat Suntiamorntut, from the Faculty of Engineering, PSU, for their invaluable advice, contributions and suggestions towards the improvement of my research thesis.

Meanwhile, I would like to appreciate Assoc. Prof. Dr.Sakesun Suthummanon and Asst. Prof. Dr. Supapan Chaiprapat of the Faculty of Engineering, PSU, for their continuous encouragement, patience, and help during the course of my study, especially when I almost lost my courage and zeal to continue. Also, great thanks to all the lecturers, staffs and fellow students in the Department of Industrial and Systems Engineering, PSU, for their invaluable advices, generous help and sincere encouragement.

I would like to extend my appreciation to the Higher Education Research Promotion, the Thailand's Education Hub for Southern Region of ASEAN Countries (TEH-AC) Project Office of the Higher Education Commission and Prince of Songkla University for funding this project, giving me an opportunity to participate in this exciting journey of research works. Finally, I would like to express my heartfelt gratitude to my family in China for their heir continuous and endless support.

Lehuang Zong

TABLE OF CONTENTS

Page

ABSTRACT V

ACKNOWLEDGEMENTS VI

TABLE OF CONTENTS VIII

LIST OF TABLES X

LIST OF FIGURES XII

CHAPTER 1 1

1.1 Motivation 1

1.2 Research Objectives 4

1.3 Research Benefits 4

1.4 Scopes of the Research 4

1.5 Overview of the Research 6

CHAPTER 2 7

2.1 Planning and scheduling for precast production 7

2.2 Summary 11

CHAPTER 3 14

3.1 Introduction to precast production process 14

3.2 Flow shop sequencing problem (FSSP) 16

3.3 Multi-objective precast production scheduling model (MOPPSM) 18

3.4 Multi-objective precast production scheduling model with lot delivery (MOPPSM-LD) 25

CHAPTER 4 30

4.1 The proposed MOVNS for the MOPPSM and MOPPSM-LD 31

4.2 The proposed NSGA-II for the MOPPSM and MOPPSM-LD 37

4.3 The comparative algorithm MOGLS for the MOPPSM and MOPPSM-LD42

CHAPTER 5 46

5.1 Problem cases 46

5.1.1 MOPPSM: case 1-10 47

5.1.2 MOPPSM-LD: case 11-15 50

5.2 Performance Measures 57

5.3 Parameters settings 60

5.4 Computational Results and Comparison 62

5.4.1 Results and Comparison for solving MOPPSM 62

5.4.2 Results and Comparison for solving MOPPSM-LD 67

CHAPTER 6 70

6.1 Conclusions 70

6.2 Suggestions and future work 71

REFERENCES 72

VITAE 86

LIST OF TABLES

Table **Page**

Table 2.1 Features of precast production scheduling formulations……….…………....9

Table 3.1 Comparison between MOPPSM and MOPPSM-LD……..............…………26

Table 4.1 A sample of PC order for MOPPSM …...………..…...……..…...……..….33

Table 4.2 A sample of PC order for MOPPSM-LD……………...……....………….…34

Table 5.1 Production data of PCs for 10 cases of MOPPSM…..…………….…...…48

Table 5.2 Problem cases with different size...………...……….........……..…...……49

Table 5.3 Operation time and E/T penalties for MOPPSM-LD....……...……...……51

Table 5.4 Customer orders in case 11: 10 orders and 20 PCs..……...……...……....52

Table 5.5 Customer orders in case 12: 10 orders and 40 PCs……..……………..…53

Table 5.6 Customer orders in case 13: 26 orders and 60 PCs……..…..................…54

Table 5.7 Customer orders in case 14: 26 orders and 80 PCs……..……..........……55

Table 5.8 Customer orders in case 15: 26 orders and 100 PCs…...……...….............56

Table 5.9 Parameter settings of all algorithms…........…...……......…...…...…...……61

Table 5.10 The maximum CPU times for terminating the tested algorithms in each case....………....………......…....………....………...……….........……....………61

Table 5.11 Comparison of the three algorithms using the SP and $D1_R$ indicators…………..……………....…..………....……….…………….....………..……..64

Table 5.12 Comparison of the three algorithms using the D_{min} and D_{max} indicators………......……………....…………...…………....……..……...……....……..65

Table 5.13 Comparison of the three algorithms in solving MOPPSM-LD using the SP and $D1_R$ indicators…...………......……………....…...……….………...………...68

Table 5.14 Comparison of the three algorithms in solving MOPPSM-LD using the D_{min} and D_{max} indicators……….....………......………......………......…………....………..……...68

LIST OF FIGURES

Figure **Page**

Figure 3.1 Precast concrete production process……………..…….………………………..15

Figure 3.2 The classical flow shop scheduling problem……………………………..18

Figure 3.3 Two situations of interruptible operations (*j*=1, 2, 5, 6)..………………..21

Figure 3.4 Two situations of uninterruptible operations (*j*=3)…………………..…..22

Figure 3.5 Three situations of uninterruptible operations (*j*=4)..........................….23

Figure 3.6 Operation processes of precast concrete production and delivery……....26

Figure 4.1 Flow chart of the proposed metaheuristics..….31

Figure 4.2 Chromosome representation of MOPPSM. …………..…………….……34

Figure 4.3 Chromosome representation of MOPPSM-LD……………………..……34

Figure 4.4 Neighbourhood structures……………………………....……………36

Figure 4.5 Genetic operators' schemas.…………………..……….……….…….…40

Figure 5.1 Performance comparison of the three algorithms in solving MOPPSM (10 cases) using the SP, D_{min}, $D1_R$ and D_{max} indicator.………………………….....66

Figure 5.2 Performance comparison of the three algorithms in solving MOPPSM-LD (case 11-15) using the SP, D_{min}, $D1_R$ and D_{max} indicator……………..………………………………………..……..…......69

CHAPTER 1

INTRODUCTION

1.1 Motivation

Construction industries have been facing problems of low quality, cost overruns, delay, environmental impacts as well as poor safety records because of the risk, uncertainties, labour complexity and dynamic change associated with the industry (Konczak and Paslawski 2015; Othman et al. 2017). To overcome these problems, an industrialised precast construction technique has been developing rapidly to supply building materials since the 1950's. Different from traditional construction method, precast construction requires precast components (PCs), such as beams, columns and girders, to be prefabricated in factories, transported and installed in site according to its erection schedule. Therefore, it has totally changed the whole building process and resulted in fast building of houses and reduction in cost irrespective of weather conditions to satisfy today's rising demand for modern buildings because of increasing population. Furthermore, like manufacturers in other industries, manufacturers in precast industry has steadily been driven toward large-scale production in order to satisfy the customer's higher and higher demand so as to improve their competitive edge in today's global markets for the last few decades. With the increasing economic pressure, therefore, manufacturing transformations have naturally required studies and the industry to pay more attention to innovative management.

It is clear that production scheduling has a dramatic impact on the success of precast fabrication, because it involves with making accurate decisions on when, which, and how many production tasks should be finished to meet their due dates of delivery as fast as possible. In other words, an efficient scheduling policy can contribute to a good performance of system and individuals while inappropriate production scheduling policies can lead to inefficient utilisation of resources as well as unnecessary delays. Unfortunately, in practice, the current precast production schedules are fairly arranged by experience-based estimation and subjective approaches of the scheduler, thereby frequently resulting in overstocking of PCs, low resource utilisation and delayed delivery in the precast industry (Ko and Wang 2011; Prata, Pitombeira-Neto, Sales 2015; Ko 2016; Wang, Hu, and Gong 2018). Therefore, it is necessary and urgent for precast industry to use more efficient and reliable scheduling techniques and improve its production efficiency for adapting to the increasingly fierce market competition. Since precast production has its own characteristics, some investigations have gradually concentrated on modelling methods. Several formal precast production scheduling models have been developed due to their convenience of considering more realistic constraints prevailing in the precast industry. These realistic constraints mainly include off-normal working time and non-pre-emptible fabrication operations (Chan and Hu 2001, 2002a), limited workers and cranes (Leu and Hwang 2002), buffer size between production stations (Ko and Wang 2010, 2011; Ko 2016), mould availability (Benjaoran, Dawood, and Hobbs 2005), different concrete formulas (Tharmmaphornphilas and Sareinpithak 2013), multiple production lines (Yang, Ma, and Wu 2016), interaction between the production and delivery (Wang and Hu 2017; Wang, Hu, and Gong 2018).

While the results given by these models in the tests seem promising, some limitations can be identified with these models. Firstly, most of aforementioned models were solved by genetic algorithm (GA) based optimisation algorithms. However, other competitive metaheuristic methods, such as variable neighbourhood search (VNS) algorithm and non-dominated sorting genetic algorithm II (NSGA-II), have been ignored by previous investigations. Secondly, all the scheduling models assumed that the start delivery time of all PCs are independent. It means that every PC can be delivered to its customer immediately after achieving the delivery strength without waiting for other PCs. In practice, however, delivery waiting time is always inevitable when at least two PCs are required to be delivered together (called as "lot delivery" in this paper) to the customer at the same time. That is, in order to be delivered, a PC completed earlier needs to wait for all the other uncompleted PCs with the same customer order and due date to be completed. Lot delivery is widespread in industries both because of the realistic requirement to the customer order and cost saving for transportation. If the waiting time of lot delivery is neglected, the calculation results of completion time would be less than that in the real case, and the early and late delivery of PCs will occur. Hence, the objectives of this study were to explore the abilities of different competitive metaheuristic methods in solving the precast production scheduling problems and to develop a more comprehensive model with consideration of lot delivery for efficient and effective production scheduling in precast plants.

1.2 Research Objectives

The specific objectives of this thesis are:

(1) To formulate an advanced scheduling decision-making tool for precast fabrication, in order to meet delivery dates with minimum completion time of production.

(2) To provide the decision-makers with a more realistic alternative set of optimum solutions for better decision making by adopting multi-objective algorithms to tackle the precast scheduling problems.

1.3 Research Benefits

The benefits of the research include:

(1) Find out the efficient optimisation method to make a rapid acquisition of scheduling process of precast fabrication and delivery.

(2) Providing a set of efficient schedules (Pareto-optimal solutions) to help the production scheduler choosing the most suitable schedule.

1.4 Scopes of the Research

The scopes of the present research are limited to the following:

(1) Production process: The precast scheduling focused on eight main operations: (1) assembly mould, (2) reinforcement setting, (3) concrete pouring, (4) concrete curing, (5) demoulding or mould stripping, (6) PC finishing, (7) PC storing, and (8) PC delivery. Raw material preparation such as mixing of concrete was not considered in our scheduling model.

(2) Mathematical model: Multi-objective precast production scheduling model (MOPPSM) generated by Ko and Wang (2011) was developed two type, i.e. MOPPSM with buffer size constraint and MOPPSM without consideration of buffer size. In this study, we focused on the latter. Furthermore, we extended MOPPSM to multi-objective precast production scheduling model with lot delivery (MOPPSM-LD) according to realistic customer orders. Both MOPPSM and MOPPSM-LD were proposed based on make-to-order precast production.

(3) Multiple objectives: Two conflicting objectives, i.e. the makespan and total penalty costs of earliness and tardiness (E/T) minimisation, were considered in MOPPSM and MOPPSM-LD. Although actual practice takes care of more than just single objective, little research has been done with the concern of multiple objectives in theory of shop scheduling problem (Behnamian 2016). In the area of scheduling theory, maximum completion time (i.e. so-called “makespan”) is commonly-used objective to estimate the performance of production scheduling model (Johnson 1954). Therefore, makespan was taken as first objective in our developed scheduling models. Moreover, comparing with minimisation of maximum completion time, manufacturers are more concerned with accurate decisions about which, and how many production tasks should be delivered to their customers either before or after the exact time are needed, i.e. philosophy of just-in-time (JIT). For this reason, in this research, total penalty costs of E/T was considered as the second objective in order to perfectly fit to JIT production control policy (Behnamian and Fatemi Ghomi 2014).

(4) Algorithms: MOVNS and NSGA-II algorithms, cited from Geiger (2004) and Deb et al. (2002), respectively, were applied and compared the

performance with MOGLS algorithm proposed by Ko and Wang (2011) in solving MOPPSM and MOPPSM-LD.

(5) Input data: The experimental input data for 10 cases of MOPPSM and 5 cases of MOPPSM-LD were generated based on standard cited data from Chan and Hu (2001), Wang and Hu (2017) and Benjaoran et al. (2005). These standard data have been also frequently cited by the subsequent works (Chan and Hu 2002a; Ko and Wang 2011, 2016; Yang, Ma, and Wu 2016).

1.5 Overview of the Research

This thesis is divided into six chapters, of which the first has been covered. Chapter 2 present an exhaustive theory and literature review with respect to the production scheduling modelling techniques. Chapter 3 discusses on the mathematical model of the MOPPSM based on FSSM and the proposed MOPPSM-LD. Chapter 4 describes the proposed MOVNS and NSGA-II algorithms for solving the MOPPSM and MOPPSM-LD. Chapter 5 demonstrates the experiments and computational results of the proposed approaches. Some conclusions and recommendations are given in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

2.1 Planning and scheduling for precast production

In order to eliminate the unexpected consequence of manually arranging production schedule, researchers have begun developing the modelling and computational techniques. These techniques adopted for dealing with precast production planning and scheduling have nearly gone through two stages according to the year of publications, as can be seen in Table 2.1.

In the first stage, simulation techniques emerged in early years, mainly from 1993 to 2000. In the work of Dawood and Neale (Dawood and Neale 1993; Dawood 1993, 1995, 1996) as well as Vern and Gunal (1998), simulation technique was considered as an effective tool to create a computer-based capacity-planning model for analysing the difficulties of scheduling and to help production managers to explore alternative options so that better planning decisions can be made (Dawood and Neale 1993; Dawood 1993, 1995, 1996; Vern and Gunal 1998). Their proposed models were treated as factory simulators to automate the production planning process by applying the scheduling rules which were generated base on the factory attributes before the actual manufacturing begins. However, these models ignored many practical constraints encountered in the precast industry. Moreover, most of developed planning models only treated time minimisation or cost as a main objective while schedulers more consider about the problems like which, how many, and when PCs should be fabricated to satisfy their transportation due dates. This may result in factory fabrication below the optimal plant capacity, overstocked inventory, unnecessary idle waiting time, and/or missed delivery dates. Therefore, more formal production scheduling models with many practical constraints and objectives prevailing in the precast industry should be developed (Chan and Hu 2000, 2001, 2002a). In second stage, mainly from 2001 till now, many formal scheduling models which can absorb more practical constraints have been developed for precast production. The models have mainly been developed on the constrained-based precast scheduling model (CPSM) (Chan and Hu 2002b), the parallel machine scheduling model (PMSM) (Tharmmaphornphilas and Sareinpithak 2013), and the flow shop scheduling model (FSSM) (Chan and Hu 2001, 2002a; Leu and Hwang 2001, 2002; Benjaoran, Dawood, and Hobbs 2005; Ko and Wang 2010, 2011; Yang, Ma, and Wu 2016). Besides, these models were mainly solved by computational intelligence (CI) methods and some of the works compared the optimisation performance of proposed CI with that of heuristic rules.

Table 2.1 Features of precast production scheduling formulations.

Authors	Modeling technique			Computational techniques											
						Heuristics									
	CPSM	PMSM	FSSM	GA	Simulation	Palmer	Gupta	CDS	RA	EDD	ASAP	CBE	MIP	SPT	LST
Dawood 1993					√										
Dawood 1995					√										
Dawood 1996					√										
Dawood & Neale 1993					√										
Vern and Gunal 1998					√										
Al-Bazi & Dawood 2008					√										
Konczak & Paslawski 2015					√										
Chan and Hu 2000			√	√		√	√	√	√	√					
Chan & Hu 2001			√	√		√	√	√	√	√					
Chan and Hu 2002a			√	√		√	√	√	√	√					
Leu and Hwang 2001			√	√											
Leu and Hwang 2002			√	√											
Benjaoran etc. 2005			√	√											
Ko & Wang 2010			√	√											
Ko & Wang 2011			√	√		√	√	√	√	√					
Yang etc. 2016			√	√						√				√	√
Wang & Hu 2017			√	√											
Wang etc. 2018			√	√	√										
Chan and Hu 2002b	√									√	√	√			
Tharmmaphornphilas & Sareinpithak 2013		√											√		

Specifically, pioneering researchers, Chan and Hu (2000, 2001, 2002a) and also Leu and Hwang (2001, 2002), proposed the FSSM for the typical precast concrete manufacturing processes by using the genetic algorithm (GA) approach as a solution. Chan and Hu (2000, 2001, 2002a) made their precast flow shop model very realistic to separate the work day into working and non-working hours and to classify the production activities as two types: pre-emptive and non-pre-emptive. In their research, GA was adopted as the optimisation to achieve the combination objectives of minimising makespan and total tardiness penalty incurred from delayed deliveries of precast products and compared to some classical heuristics, such as Palmer's heuristic (Palmer 1965), Campbell Dudek Smith (CDS) heuristic (Campbell, Dudek, and Smith 1970), Gupta's heuristic (Gupta 1971), earliest due date (EDD) rule, and rapid access (RA) heuristic (Dannenbring 1977). The consideration further improved the viability of applying computational methods in solving precast production scheduling problems. Leu and Hwang (2001, 2002) studied the influence of constraints in different sharing factory resources including equipment and working crew on the overall production makespan. Genetic algorithm was also applied in their proposed FSSM model to generate the optimal or near-optimal production schedules which can consider resource utilisation, mixed production and minimum makespan.

In contrary to previous studies that optimise the proposed problem with traditional single-objective GA, Benjaoran et al. (2005) and Ko & Wang (2010, 2011) developed multi-objective GA optimisation methods. Benjaoran et al. (2005) developed a bespoke precast flow shop scheduling model (BP-FSSM) which considered multiple objectives including minimisation of makespan, late delivery penalty, and machine idle time. The sensitivity analysis was used to test the model parameters. Ko & Wang (2010, 2011) proposed a multi-objective precast production scheduling model (MOPPSM). In MOPPSM, buffer sizes between stations were considered and filled the research gap in area of precast production scheduling in which ignored buffer size by previous research. A metaheuristic algorithm namely multi-objective genetic local search (MOGLS) was developed in the research for searching optimum solutions with minimisation of makespan and tardiness penalties of their developed MOPPSM, and MOGLS outperforms two comparative multi-objective methods: constant weight GA and vector evaluated GA, in solving MOPPSM.

Later, Yang et al. (2016) proposed a FSSM for precast production of multiple production lines and developed a corresponding optimisation approach to facilitate optimised scheduling by using GA. Multiple objectives including makespan, penalty cost of E/T, avoiding frequent type change of PCs during production were considered in this model. Previous works mentioned above ignored storing and transportation processes when solving the make-to-order precast production scheduling problem even though it is one of main bottlenecks limiting the productivity of an entire construction project (Liu, Zhang, and Li, 2014). Hence, differ from previous works, Wang and Hu (2017) improved the traditional precast production scheduling model by integrating the mould manufacturing, PC storing, and PC delivery processes in the perspective of whole supply chain of PC manufacturing.

2.2 Summary

Based on the review of previous literatures, some conclusions and issues that this study attempts to solve can be summarised:

Firstly, in previous researches, several formal scheduling models, such as CPSM (Chan and Hu 2002b), PMSM (Tharmmaphornphilas and Sareinpithak 2013), and MOPPSM (Ko and Wang 2010, 2011), have been developed for make-to-order precast production. In the models, different realistic constraints, such as off-normal working time (Chan and Hu 2001, 2002a), limited workers and cranes (Leu and Hwang 2002), buffer size between production stations (Ko and Wang 2010, 2011; Ko 2016) and so on, prevailing in the precast industry were considered. However, aforementioned research studies developed make-to-order precast production scheduling models based on the assumption that every PC can be delivered to its customer without waiting for any other PC. In practice, however, this situation only happens when for each different due date only one PC is ordered and expected to be delivered. Normally, delivery waiting time is inevitable when at least two PCs are required to be delivered together to the customer at the same time. That is, lot delivery is required since a customer order, generally involves not only one PC but multiple types and multiple numbers of PCs. If the waiting time of lot delivery is neglected, the calculation results of completion time would be less than that in the real case and the early and late delivery of PCs will occur. To improve, therefore, this research modified MOPPSM by considering the so-called lot delivery constraint and adopted a multi-objective precast production scheduling model with lot delivery (MOPPSM-LD). In MOPPSM-LD, two conflicting objectives, namely the minimisation of the makespan and the total penalty costs of E/T, are considered.

Secondly, GA based optimisation algorithms were the most frequently used computational techniques to solve the precast production scheduling problem in aforementioned research studies. The GA based optimisation algorithms, as one of well-known metaheuristics, has been proven to provide better solutions than the heuristics and dispatching rules to solve the precast production scheduling problem (Chan and Hu 2001, 2002a, 2002b; Ko and Wang 2011; Yang, Ma, and Wu 2016). However, none of the previous research compared GA based optimisation with other competitive metaheuristic methods, such as multi-objective variable neighbourhood search (MOVNS) algorithm and non-dominated sorting genetic algorithm II (NSGA-II), in solving precast production scheduling problems. Actually, VNS has been proposed and applied successfully to solve various production scheduling problems since 2001 (Adibi, Zandieh, and Amiri 2010; Hansen and Mladenovíc 2001; Lei and Guo 2011, 2014; Lei 2015, 2017). Similarly, the NSGA-II first introduced by Deb et al. (2002) is one of the most proficient evolutionary algorithms used for solving multi-objective optimisation problems. Therefore, further research is needed to explore the abilities of different competitive metaheuristic methods, i.e. MOVNS and NSGA-II, in solving the problems. Both MOPPSM and MOPPSM-LD cases were developed to test the performance of the two proposed algorithms. Moreover, according to Ko and Wang (2011), MOGLS was proven to be an efficient method that achieved successfully searching for optimum production schedules for MOPPSM and outperformed seven methods, including the Palmer, RA, CDS and Gupta heuristics, the EDD rule, the vector evaluated genetic algorithm and the constant weight genetic algorithm. Therefore, in this work, this algorithm was considered as the comparative algorithm in solving MOPPSM and MOPPSM-LD due to its competitive performance.

CHAPTER 3

MODELING OF PRECAST PRODCUTION SCHEDULING

Multi-objective precast production scheduling model (MOPPSM) was proposed based on traditional flow shop scheduling problem (FSSP) since it possesses many of characteristics of the precast production under the specialised method (Chan and Hu 2001, 2002a; Ko and Wang 2011). In this chapter, precast production process is firstly introduced in Section 3.1. Then, Section 3.2 describes the traditional FSSP. Later, MOPPSM and multi-objective precast production scheduling model with lot delivery (MOPPSM-LD) are presented in Section 3.3 and Section 3.4, respectively.

3.1 Introduction to precast production process

Warszawski and Ishai (1982) divided the precast production systems into two basic types, namely the stationary production system and travelling production system. With the stationary production system, all the basic production operations are performed at fixed locations which a comprehensive workforce is involved. In the traveling system, moulds are moved among different workstations in which different operations are processed by the different workforce with specialised tools and work methods. In the selected schedule of precast production, a certain amount of resources, such as cranes, manual labours and steel moulds, are assigned within a specific operation time to fabricate different PCs included precast slabs, beams, columns, stairs, girders, walls, etc. Precast production under the specialised method usually breaks the total production process into six operations in sequence, as depicted in Figure 3.1. Each of the operations is explained in detail as following (Ko and Wang 2010, 2011):

(1) Mould assembly: cleaning, oiling, installing and fastening the mould.

(2) Reinforcement setting: placing all of reinforcement cage, supporting cables (rebar), fixtures, conduits and other embedded parts into the PCs.

(3) Concrete pouring: casting, compacting, and levelling of concrete mix.

(4) Concrete curing: cream curing to accelerate the chemical-solidifying process or curing in natural air.

(5) Demoulding: stripping the side frame and removing the PCs out of mould.

(6) PC finishing: checking and repairing the PCs, placing the PCs in the stockyard, and cleaning the production line.

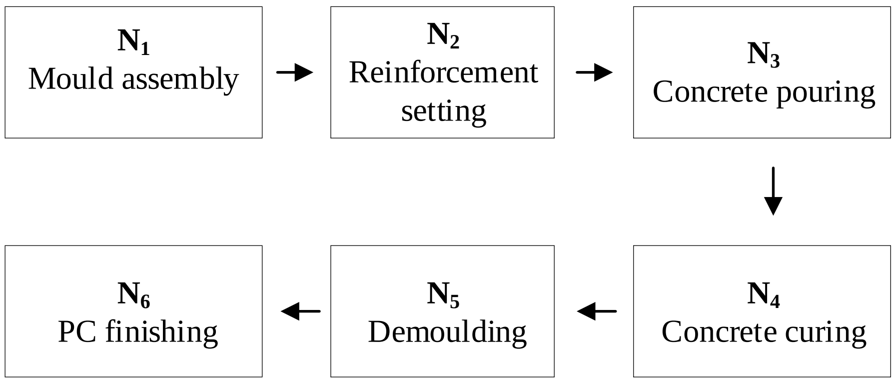

Figure 3.1 Precast concrete production process.

Materials preparation process including precast concrete mix, rebar cage manufacturing and tile processing are not included among the six main production operations, since they could be handled outside the precast fabrication area either in the precast plant or outsourced to another precast plant. The mould is assembled once it is prepared for the rebar cage and tools and this step provides a specific dimension. Following this, reinforcements are installed and the embedded parts are placed in positon for the sake of connecting and fixing with other PCs or with the structure when the PCs are erected. Once the embedded parts are in their position, premixed concrete transported from mixing area is then cast into the mould and curing for 12-16 hours (Ko and Wang 2011) until the PCs is hardened through natural process or cream curing process for accelerating the chemical-solidifying speed; then the steel moulds can be stripped and kept for reusing. The last fabrication step is PC finishing. The possible minor defects including peel-offs, uneven surfaces, and scratches are repaired in the PC finishing step. Then, moulds are removed and PCs are repaired and stored in the inventory area for natural curing, before being delivered to the construction site and installed in the final structure.

3.2 Flow shop sequencing problem (FSSP)

The traditional FSSP is shown in Figure 3.2. Each of n jobs in FSSP consists of m operations and each operation is executed on a specific machine. The underlying assumptions for this problem include the following:

(1) Each job has to be operated on all machines in the same sequence $1,2\ldots m$;

(2) Each machine operates only one job for each time;

(3) Each job is operated on one machine for each time;

(4) The job is uninterruptible;

(5) The set-up time of all operations are sequence-independent and are considered in the operation times;

(6) The operation orders of all jobs are same on each machine, and this is common determined sequence for all jobs.

The frequently used equation to computed the completion time is shown in Equations (3.1)-(3.4).

$$C_{11} = t_{11} \qquad i = 1, \quad j = 1 \tag{3.1}$$

$$C_{1j} = C_{1(j-1)} + C_{1j} \qquad j = 2, 3, \cdots, m \tag{3.2}$$

$$C_{i1} = C_{(i-1)1} + t_{i1} \qquad i = 2, 3, \cdots, n \tag{3.3}$$

$$C_{ij} = \max\left(C_{(i-1)j}, C_{i(j-1)}\right) + t_{ij} \qquad i = 2, 3, \cdots, n \;\; j = 2, 3, \cdots, m \tag{3.4}$$

Where C_{ij} denotes the completion time for i^{th} job in j^{th} machine and t_{ij} is operation time for that job $(t_{ij} \geq 0)$.

The makespan is calculated as follows:

$$\text{Makespan} = C_{\max} = C_{nm} \tag{3.5}$$

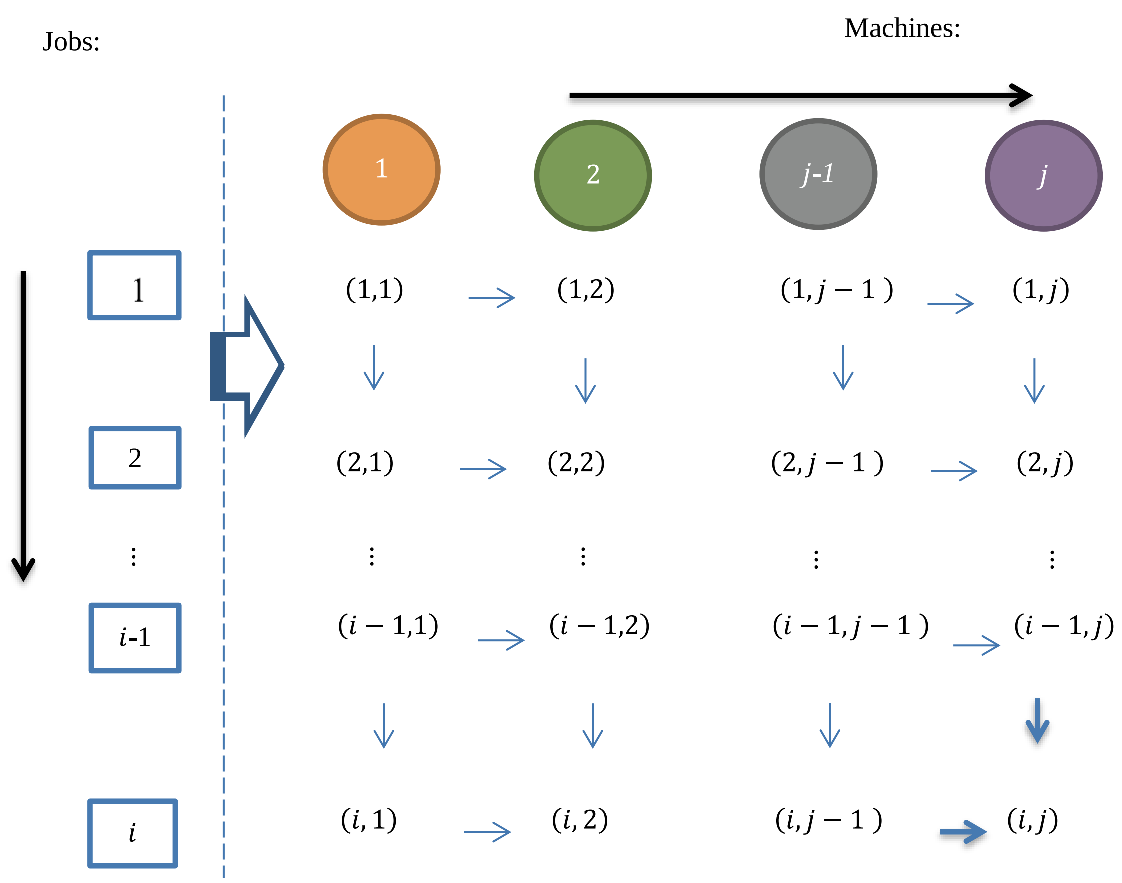

Figure 3.2 The classical flow shop scheduling problem.

3.3 Multi-objective precast production scheduling model (MOPPSM)

Multi-objective precast production scheduling model (MOPPSM) for make-to-order precast production has been proposed by Ko and Wang (2011) based on traditional FSSP. There are some similarities and differences between MOPPSM and traditional FSSP. The traditional FSSP indicates that each job has to be operated on several machines in the same sequence. The critical or meaningful function of "machine" is its capability to achieve specific operation. Relatively, in precast fabrication, every PC needs to be processed in 6 distinct workstations (with different specific resources, such as labours, cranes, tools) to respectively achieve 6 different operations, i.e. mould assembly, reinforcement setting, concrete pouring, concrete curing, demoulding, and PC finishing. Therefore, "jobs" in our precast flow shop scheduling model correspond to the PCs (denoted by n) to be produced in the various mould, these 6 distinct workstations can be defined as 6 "machines" (denoted by $m = 6$). With respect to the traditional FSSP, however, it cannot be applied for the precast production scheduling problem directly because some practical constraints encountered in the industry are disregarded in traditional FSSP. Firstly, there is no difference between normal working time and off-normal working time in the traditional FSSP, but interruptions inevitably happen in precast plants when workers punch out after working time which is normally 8 hours in a day. Furthermore, labours could be paid to work overtime if it is necessary, but overtime hours are limited. Secondly, all operations in traditional FSSP are uninterruptible. It means that an operation once started cannot be interrupted until its completion, while the operations in precast plants can be divided into interruptible operations (mould assembly, reinforcement setting, demoulding and PC finishing) and uninterruptible operations (concrete pouring, concrete curing). Thirdly, the common objective considered in traditional PPSP is makespan while in MOPPSM at least two practical objectives, e.g. makesnpan and penalty costs, have to be optimised synthetically.

To apply PPSP in MOPPSM, therefore, the assumptions for modelling PPSP could be modified and summarised as follows:

(1) Each PC has to be operated on all workstations in the same sequence $1,2\ldots m$;

(2) Each workstation operates only one PC for each time;

(3) Each PC is operated on one workstation for each time;

(4) Normal working time is 8 hours per day and off-working time is 16 hours per day. The allowable labour overtime is limited within 4 hours per day;

(5) There are both interruptible operations (mould assembly, reinforcement setting, demoulding and PC finishing) and uninterruptible operations (concrete pouring, concrete curing);

(6) More than two PCs can be curing simultaneously since concrete curing is a parallel operation which requires almost no labours and tools;

(7) The set-up times for all the operations are sequence-independent and are considered in the operation times;

(8) The operation orders of all the PCs are the same on each workstation, and this is common determined sequence for all PCs.

For interruptible operations (mould assembly, reinforcement setting, demoulding and PC finishing), they can be interrupted and continued to execute unfinished part of the interrupted operation next day if they cannot be finished within normal working time, which would cause inevitable interruption time T_N (off-normal working time). Figure 3.3 describes the two situations where an interruptible operation can be or cannot be finished within normal working time. According to Ko and Wang (2011), the accumulated completion time of interruptible operations could be calculated as:

$$C_{ij} = \begin{cases} T & \text{if } T < 24D + T_w \\ T + T_N & \text{if } T \geq 24D + T_w \end{cases}, \quad j = 1,2,5,6 \tag{3.6}$$

where j represents the interruptible workstations ($j=1,2,5,6$); T_w represents the working time in a work day; T denotes the accumulated completion time computed in Equation (3.7); D represents the working days represented when using a round up equation Equation (3.8):

$$T = \max(c_{(i\text{-}1)j}, c_{i(j\text{-}1)}) + t_{ij}\,, \quad i = 2,3,\ldots,n;\; j = 2,3,4,5,6 \tag{3.7}$$

$$D = \text{integer}\left(\frac{T}{24}\right) \tag{3.8}$$

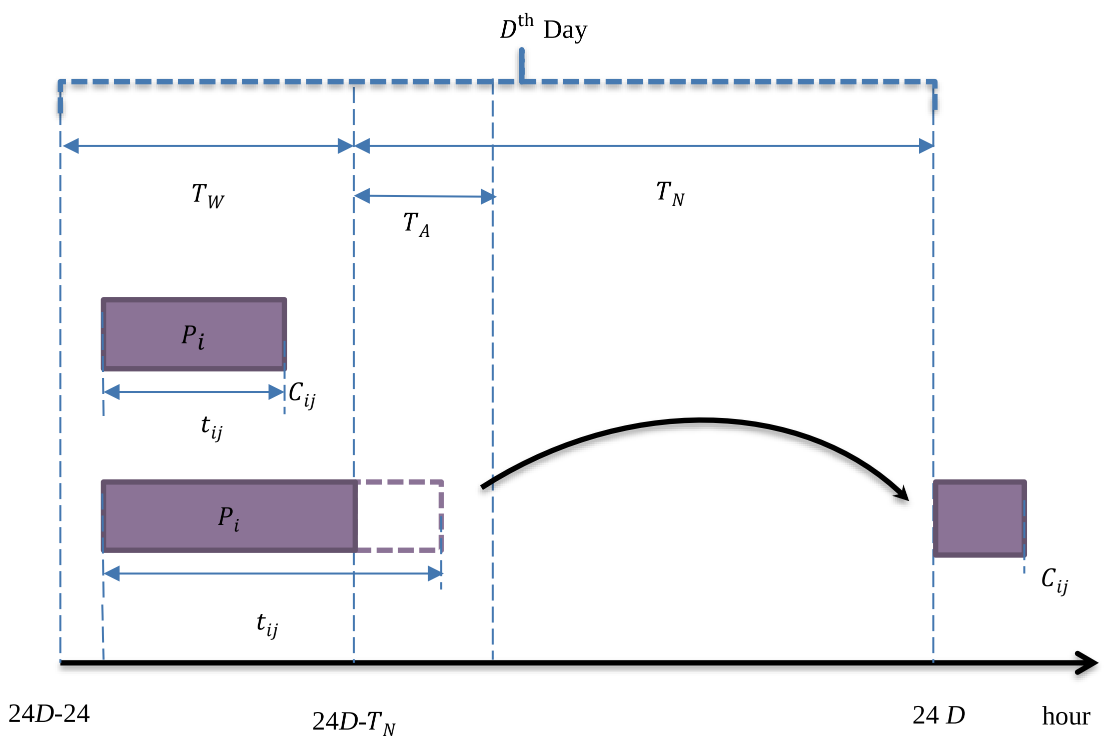

Figure 3.3 Two situations of interruptible operations (j=1, 2, 5, 6).

For the uninterruptible operations, there are two operations: called concrete pouring ($j = 3$) and concrete curing ($j = 4$). Concrete pouring would have to be postponed to the next working day if it could not be finished within the normal working time or allowable overtime. Figure 3.4 illustrates the two situations of concrete pouring where the process can or cannot be finished within allowable overtime. The completion time of concrete pouring could be calculated as:

$$C_{ij} = \begin{cases} T & \text{if } T < 24D + T_W + T_A \\ 24(D+1) + t_{ij} & \text{if } T \geq 24D + T_W + T_A \end{cases}, \quad j = 3 \tag{3.9}$$

Where T_A denotes the allowable overtime which is assumed to limited to 4 hours in one work day.

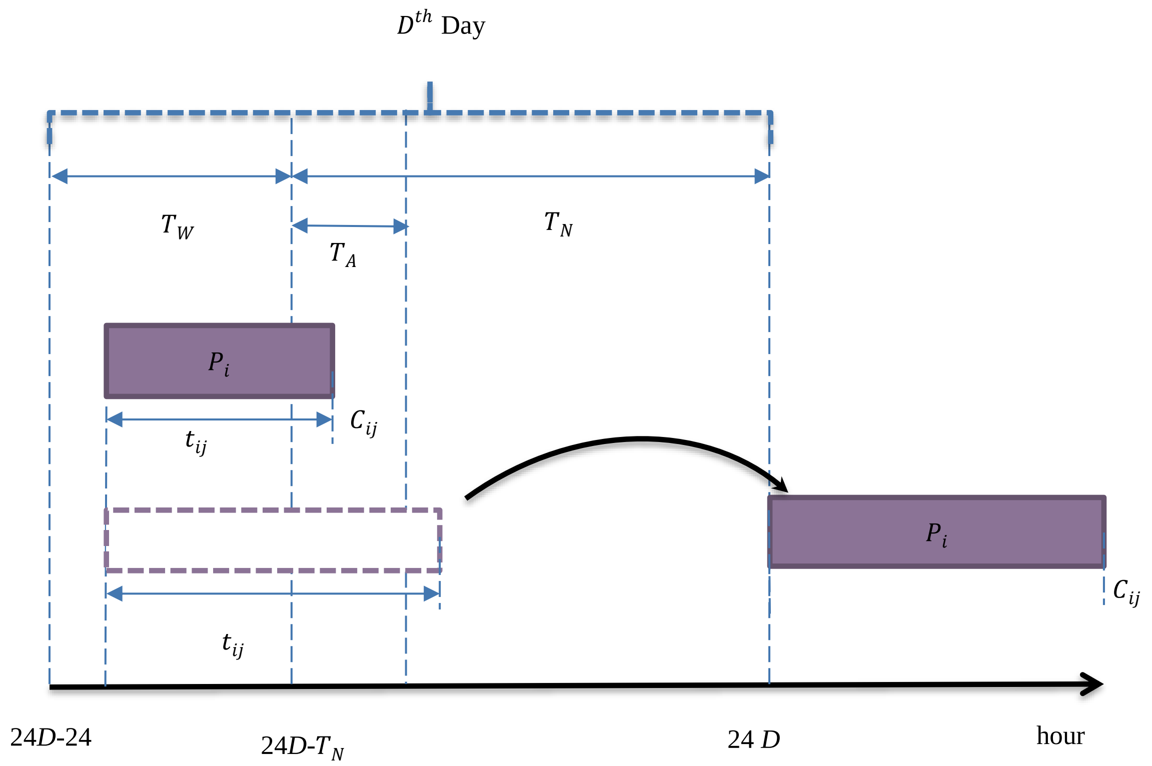

Figure 3.4 Two situations of uninterruptible operations ($j = 3$).

Concrete curing, as one of two uninterruptible operations, is a very special operation that can be executed automatically once the concrete is poured and no labours are needed. After casting, a fast cure could be finished within a few hours, while steam curing generally needs 12-16 hours. The completion time of curing (see Figure 3.5) could be computed as:

$$C_{ij} = \begin{cases} T^* & \text{if } T^* > 24(D+1) \text{ or } T^* < 24D + T_W \\ 24(D+1) & \text{if } 24D + T_W \le T^* \le 24(D+1) \end{cases}, \quad j = 4 \tag{3.10}$$

$$T^* = C_{i(j-1)} + t_{ij} \tag{3.11}$$

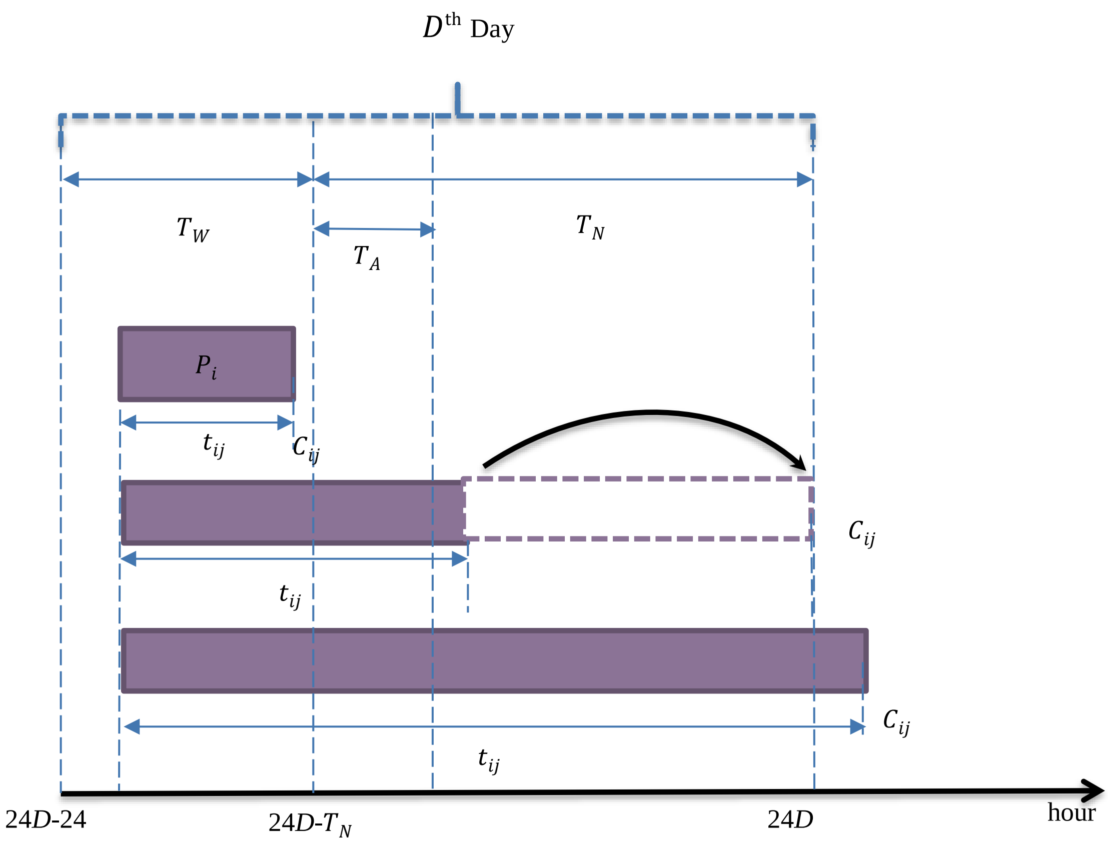

Figure 3.5 Three situations of uninterruptible operations (j=4).

After formulating the completion time of operations, the multi-objective of minimum makespan (i.e. maximum completion time) and total penalty costs of E/T can be simultaneously developed. The maximum completion time can be computed as:

$$f_1(x) = C_{nm} \tag{3.12}$$

The total penalty costs of E/T can be formulated as:

$$f_2(x) = \sum_{i=1}^{n} \varepsilon_i \times \max\left(0, d_i - c_i\right) + \sum_{i=1}^{n} \tau_i \times \max\left(0, c_i - d_i\right) \tag{3.13}$$

Where d_i denotes the due date of PC i which also means desired completion time for PC i. ε_j and τ_i respectively denote the unit cost of inventory and unit late delivery cost for PC i.

The mathematical model used to minimise a multi-objective is shown in Equation (3.14):

$$Minimise\ \ Z = \left(f_1(x), f_2(x) \right) \tag{3.14}$$

Subject to $x \in X$

Where Z denotes the objective vector; x represents the decision vector; and X represents the feasible area. The goal of optimisation in this study is to search for alternative production schedules, as opposed to trade-off surface. More details of the precast production scheduling models are available in Chan and Hu (2001, 2002a), Benjaoran et al. (2005), Ko and Wang (2011), and Wang and Hu (2017).

3.4 Multi-objective precast production scheduling model with lot delivery (MOPPSM-LD)

Apart from MOPPSM proposed by Ko and Wang (2011), MOPPSM-LD was also used to test the performances of proposed metaheuristics in our study. Table 3.1 presents the comparison between and MOPPSM and MOPPSM-LD. Based on MOPPSM, two improvements were introduced in MOPPSM-LD. Firstly, we improved MOPPSM by considering lot delivery of PCs in MOPPSM-LD. As mentioned in Chapter 1, lot delivery is widespread in industries because of the realistic requirement to the customer order and cost saving for transportation. If waiting time of lot delivery is neglected, the calculation results of completion time would be less than in the real case, and the early and late delivery of PCs will occur. Secondly, Wang and Hu (2017) improved MOPPSM by extending the traditional 6 operations of make-to-order precast fabrication to 8 operations: (1) mould assembly, (2) reinforcement setting, (3) concrete pouring, (4) concrete curing, (5) demoulding, (6) PC finishing, (7) PC storing, and (8) PC delivery, as depicted in Figure 3.6. In the traditional six-operation MOPPSM model, scheduling considered just those processes directly related to production, PC storing and transportation processes were not included even though it is one of main bottlenecks limiting the productivity of an entire construction project (Liu, Zhang, and Li, 2014). Therefore, Wang and Hu (2017) modified the scheduling model from the perspective of the whole PCs supply chain and integrated these processes into the calculation. Accordingly, we also considered 8 operations in the proposed MOPPSM-LD model according to the improvement suggestion of Wang and Hu (2017).

Table 3.1 Comparison between MOPPSM and MOPPSM-LD.

Model	Operation number	Lot delivery	Objectives
MOPPSM	6	ignored	Makespan, penalty
MOPPSM-LD	8	considered	Makespan, penalty

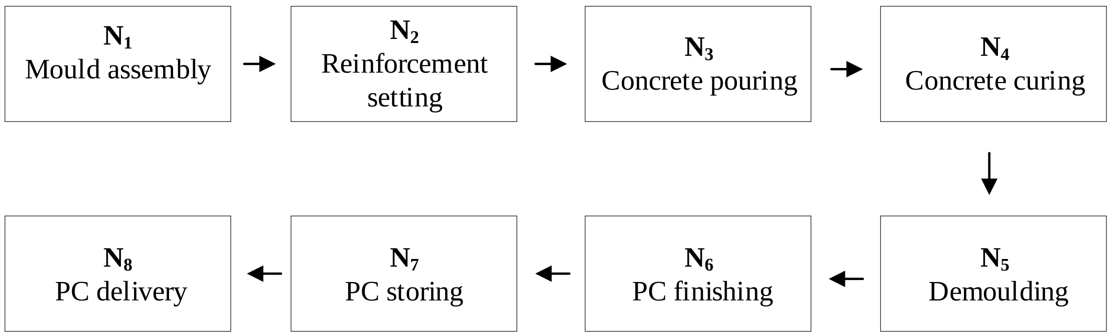

Figure 3.6 Operation processes of precast concrete production and delivery.

The assumptions for MOPPSM can be totally applied in MOPPSM-LD except slight modifications, as presented as follows.

(1) Each PC has to be operated on all workstations in the same sequence $1,2\ldots m$;

(2) Each workstation operates only one PC for each time;

(3) Each PC is operated on only one workstation for each time;

(4) Normal working time is 8 hours per day and off-working time is 16 hours per day. The allowable labour overtime is limited within 4 hours per day;

(5) Interruptible operations include mould assembly, reinforcement setting, demoulding and PC finishing, while uninterruptible operations include concrete pouring, concrete curing, PC storing and PC delivery;

(6) More than two PCs can be curing simultaneously since concrete curing is a parallel operation which requires almost no labours and tools;

(7) The set-up time of all operations are sequence-independent and are considered in the operation times;

(8) The operation orders of all the PCs are the same on each workstation, and this is common determined sequence for all PCs.

For the interruptible operations (*j*=1, 2, 5, 6), the calculation of completion time was similar to the traditional method, as showed in Equation (3.6). The completion time of the casting (*j*=3) can be deduced from Equation (3.9). Curing and storing are another two special uninterruptible operations that doesn't need workers. Similar to PC curing which can be executed automatically once the concrete is poured, PC storing can also be executed when the PC fabrication is completed. Therefore, Equation (3.10) is slightly modified to compute the completion time of both curing and storing as following:

$$C_{ij} = \begin{cases} T^* & \text{if } T^* > 24(D+1) \text{ or } T^* < 24D + T_W \\ 24(D+1) & \text{if } 24D + T_W \le T^* \le 24(D+1) \end{cases}, \quad j = 4, 7 \tag{3.15}$$

PC delivery is the last process which could begin just after the ending of PC storing. However, if the PC could not be delivered to the construction site within the allowable overtime, it would be postponed to the next working day to cut down overtime costs. Assuming there are *f* different customer orders, each order is denoted by R_k $(k = 1, 2, ..., f)$, and each order R_k includes n_k amount of PCs. Assuming the n_k PCs in an order R_k are expected to be delivered together to its customer by a specific due date, the completed PCs of an order have to wait for delivery until all the PCs of R_k are completed. The completion time of delivery could be described by Equation (3.16):

$$C_{ij} = \begin{cases} \max C_7(R_k) + t_{ij} & \text{if } \max C_7(R_k) + t_{ij} \le 24D + T_w + T_A \\ 24(D+1) + t_{ij} & \text{if } \max C_7(R_k) + t_{ij} > 24D + T_w + T_A \end{cases}, \quad j = 8 \tag{3.16}$$

Where $\max C_7(R_k)$ is the maximum completion time of PC storing (j=7) for all the PCs in k^{th} customer order R_k ($k = 1, 2, ..., f$). And $\max C_7(R_k)$ can be calculated by Equation (3.17):

$$\max C_7(R_k) = \max\{C_{(n_1+n_2+\cdots+n_{k-1}+1)7}, C_{(n_1+n_2+\cdots+n_{k-1}+2)7}, ..., C_{(n_1+n_2+\cdots+n_{k-1}+n_k)7}\} \tag{3.17}$$

The makespan and penalty costs of E/T, as depicted in as Equations (3.18) and (3.19) respectively, were chosen as two conflicting objectives for the proposed MOPPSM-LD.

$$f_1(x) = C_{nm} \tag{3.18}$$

$$f_2(x) = \sum_{k=1}^{f} \left\{ \max\left(0, d_k - C_{R_k}\right) \times \sum_{i=1}^{n_k} \varepsilon_i \right\} + \sum_{k=1}^{f} \left\{ \max\left(0, C_{R_k} - d_k\right) \times \sum_{i=1}^{n_k} \tau_i \right\} \tag{3.19}$$

Where d_k denotes the due date of PC k which also means desired completion time for PC k. ε_j and τ_i respectively denote the unit cost of inventory and unit late delivery cost for PC i.

The mathematical model used to minimise a multi-objective is shown in Equation (3.20):

$$\mathit{Minimise}\ \ Z = \left(f_1(x), f_2(x) \right) \tag{3.20}$$

Subject to $x \in X$

Where Z denotes the objective vector; x represents the decision vector; and X represents the feasible area. The optimisation goal in this study is to search for alternative production schedules, as opposed to trade-off surface.

CHAPTER 4

PROPOSED META-HEURISTIC ALGORITHMS FOR PRECAST PRODUCTION SCHEDULING PROBLEM

The MOPPSM and MOPPSM-LD considered in our study are NP-hard problem that beyond the optimising capability of exact methods, such as Lagrangian relaxation and branch and bound. Thus, in this research work, we employed two metaheuristic algorithms, i.e. multi-objective variable neighbourhood search (MOVNS) and a non-dominated sorting genetic algorithm (NSGA-II), to find Pareto optimal solutions for the MOPPSM and MOPPSM-LD. Moreover, multi-objective genetic local search (MOGLS) algorithm was successfully employed to solve MOPPSM (Ko and Wang 2011). Therefore, we considered MOGLS as the benchmark algorithm for the performance comparison of MOVNS and NSGA-II in optimising MOPPSM and MOPPSM-LD. Figure 4.1 illustrates the flowchart of the three algorithms and the detail of them has described in the following subsections.

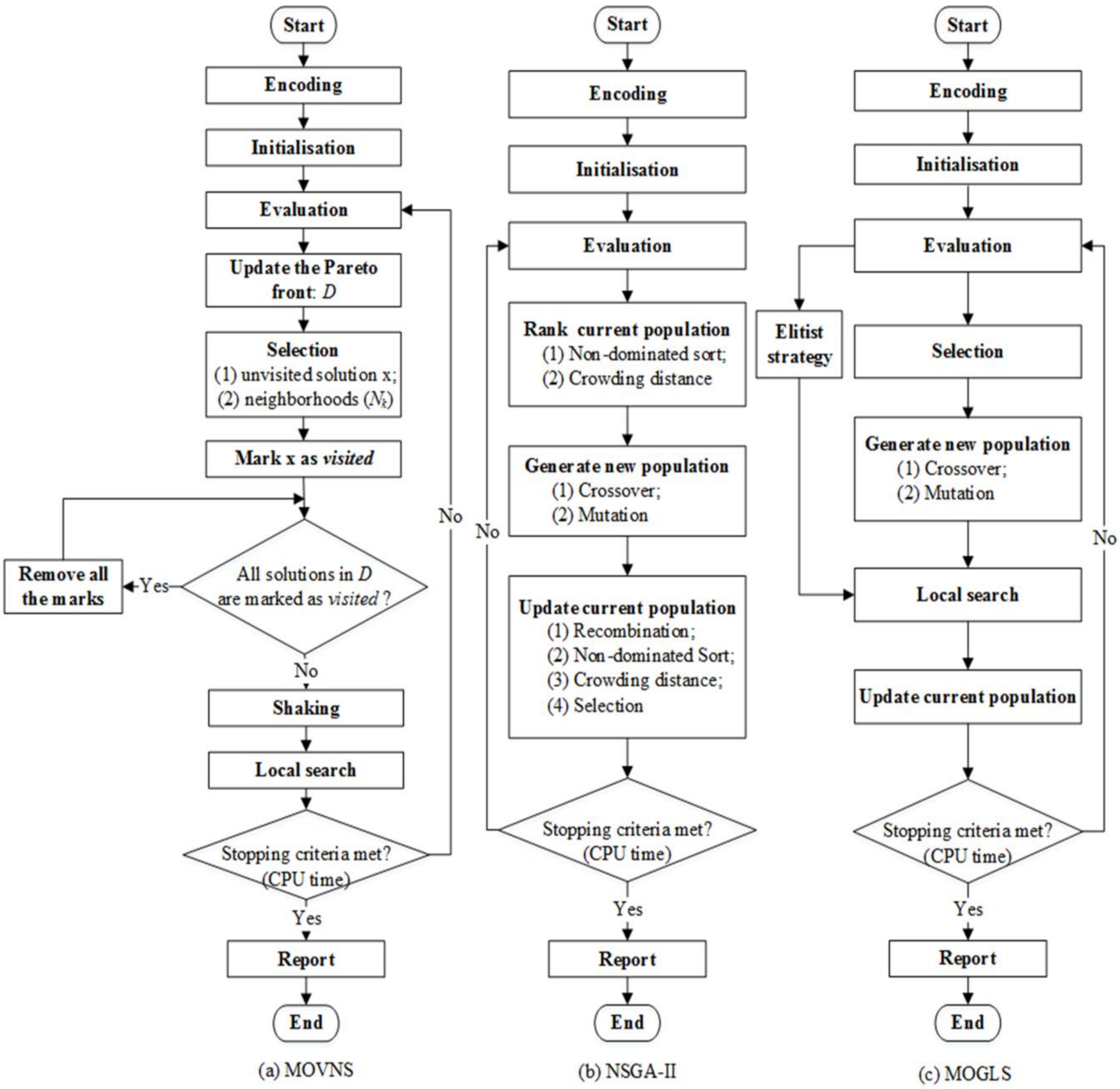

Figure 4.1 Flow chart of the three metaheuristics.

4.1 The proposed MOVNS for the MOPPSM and MOPPSM-LD

Variable neighbourhood search (VNS) is the local search based metaheuristic that was firstly proposed by Mladenoviã and Hansen (1997). It is based on the principle of systematic changes of neighbourhood in both, the descent phase down to find a local optimum, and the perturbation phase to escape from the corresponding local minimum valley. In this way, VNS search for increasingly distant neighbourhoods of present incumbent solutions to obtain the promising neighbouring solutions. The VNS algorithm has been used successfully to solve many production scheduling problems since 2001 (Adibi, Zandieh, and Amiri 2010; Hansen and Mladenovíc 2001; Lei and Guo 2011, 2014; Lei 2015, 2017). To the best of our knowledge, however, VNS has not been applied to precast production scheduling problems. Accordingly, the objective of this study was to extend the application of VNS in solving precast production scheduling problems. Furthermore, two conflicting objectives are considered in our model to remedy the defect that the application of VNS in multi-objective optimisation problems is scanty (Arroyo, Ottoni, and Oliveira 2011). The multi-objective VNS (MOVNS) algorithm in this study was developed based on the algorithm of Geiger (2004) and the flowchart of MOVNS is depicted in Figure 4.1 (a). The detailed steps of the developed MOVNS algorithm outline as follows.

Step 0 **Encoding**: This study encodes the solution of MOVNS by PC sequencing. Every scheduling solution was represented by a single stringlike entity namely a chromosome. Each chromosome consisted of many genes, and its value and positions denoted the PC serial number and its schedule sequence for production. The encoding schemas of MOPPSM and MOPPSM-LD are related to different constraints of customer orders. Ko and Wang (2011) developed MOPPSM based on the assumption that only one PC is ordered for each different due date, as described in Table 4.1. By given each one of PCs in Table 4.1 a serial number from array (1, 2, 3, 4, 5, 6, 7, 8), the encoding schema of the proposed MOVNS algorithm for MOPPSM can be described in Figure 4.2. That is, the solution with PC serial number (4, 5, 7, 3, 2, 6, 8, 1) represents the PC type (#4, #5, #7, #3, #2, #6, #8, #1) which can be considered as a production schedule. However, for MOPPSM-LD, lot delivery is required since a customer order as a sample described in Table 4.2 generally involves not only one PC but multiple types and multiple numbers of PCs. 5 different types of PCs, as shown in Table 4.2, are ordered by all customers with order amount 1, 2, 1, 3, and 1, respectively. The encoding schema of the proposed MOVNS algorithm for MOPPSM-LD can be described in Figure 4.3, which shows an example of one individual in an case with eight PCs, where 8 non-repetitive integer (4, 5, 7, 3, 2, 6, 8, 1) generated in [1, 8] would stand for the PCs sequence (#3, #4, #4, #2, #2, #4, #5, #1).

Table 4.1 A sample of PC order for MOPPSM.

PC type	Order amount	Due date	PC serial No.
#1	1	15	1
#2	1	21	2
#3	1	22	3
#4	1	10	4
#5	1	41	5
#6	1	14	6
#7	1	31	7
#8	1	29	8

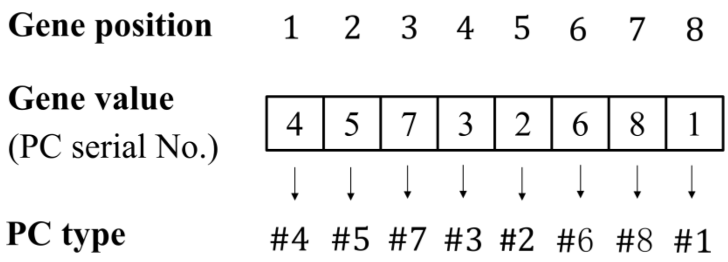

Figure 4.2 Chromosome representation of MOPPSM.

Table 4.2 A sample of PC order for MOPPSM-LD.

PC type	Order amount	Due date	PC serial No.
#1	1	15	1
#2	2	21	2
			3
#3	1	10	4
#4	3	31	5
			6
			7
#5	1	29	8

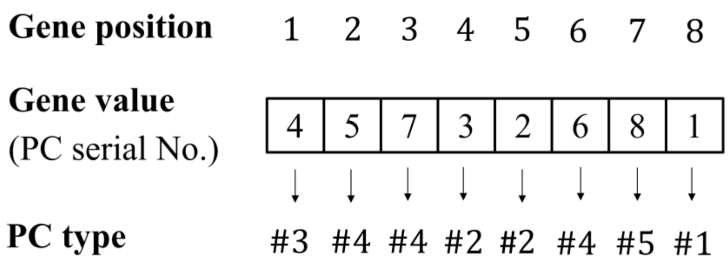

Figure 4.3 Chromosome representation of MOPPSM-LD.

Step 1 **Initialisation**: Choose a stopping criterion, define the set of neighbourhood structure (N_k , k = 1 ~ k_{max}), and randomly generate initial population of N_{pop} chromosomes to represent as precast production schedules. Each chromosome represents a solution, which is schedule of precast component sequence in MOPPSM or MOPPSM-LD.

Step 2 **Evaluation**: Evaluate objective values for MOPPSM, i.e. makespan and total penalty costs of E/T, of each chromosome in current population pop using Equations (3.12) and (3.13), respectively. For optimising MOPPSM-LD, the Equations (3.18) and (3.19) are respectively used to evaluate objectives makespan and total penalty costs of E/T.

Step 3 **Update the Pareto front:** The tentative set *D* where all non-dominated solutions are separately stored from current population is updated according to the concept of domination. A solution p is said to dominate solution q if and only if: (1) $f_i(p) \le f_i(q)$ $\forall i \in \{1,2,...,q\}$ and (2) $f_i(p) < f_i(q)$ $\exists i \in \{1,2,...,q\}$.

Step 4 **Selection**: Randomly select an unvisited base solution from *D*, and randomly select a neighbourhood structure N_k from the following two common neighbourhood structures.

(1) **Insertion neighbourhood (N_1):** Randomly select two positions r_1 and r_2 (where $r_1 < r_2$) in the solution representation and then remove the PC serial No. at position r_2 and insert it before r_1 in the scheduling string, as shown in Figure 4.4 (a).

(2) **Swap neighbourhood (N_2):** Randomly select two positions r_1 and r_2 in the solution representation and then swap the two PCs at the r_1 and r_2 in the scheduling string, as shown in Figure 4.4 (b).

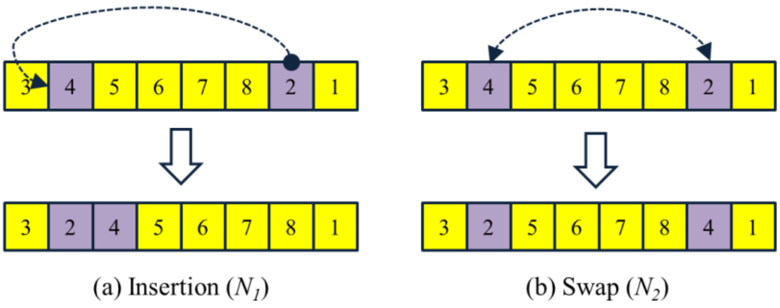

Figure 4.4 Neighbourhood structures.

Step 5 **Mark**: The selected base solution is marked as visited for avoiding to be selected in the next iterations. If all solutions in the tentative set D have been marked as visited, then all the marks will be removed.

Step 6 **Shaking:** Generate randomly a solution **x'** from the N_k neighbourhood of current solution **x**;

Step 7 **Local search:** Apply a complete local search in the N_k neighbourhood of **x'**, denote the obtained local optimum with **x″**.

Step 8 **Termination:** If the algorithm reaches maximum CPU time, end the algorithm. Otherwise, return to Step 2 (iteratively update the Pareto front using generated solution **x″**).

Step 9 **Report**: PC sequences represented by Pareto optimal solutions, i.e. solutions in the tentative set D of final iteration, are reported as optimum schedules for MOPPSM or MOPPSM-LD.

4.2 The proposed NSGA-II for the MOPPSM and MOPPSM-LD

As noted in aforementioned section, the application and popularity of NSGA-II has informed the choice of the algorithm for the present study. The NSGA-II, first introduced by Deb et al. (2002), has been demonstrated as one of the most applicable and popular evolutionary algorithms for solving multi-objective optimisation problems. In the algorithm, parent population is ranked to create Pareto fronts by using the fast non-domination sorting and crowding distance procedures. Then, the algorithm applies binary tournament selection, crossover and mutation operators to generate an offspring population for the next generation. At last, the best individuals in terms of diversity and non-dominance are saved as the near-optimal solutions. The main components of the algorithm are summarised in Figure 4.1 (b).

The main steps of the NSGA-II algorithm outline as follows.

Step 0 **Encoding**: This study encodes the solution of NSGA-II in solving MOPPSM and MOPPSM-LD by permutation representation, which can be described as Figure 4.2 and Figure 4.3. In our adopted permutation representation, a non-repetitive integer numbers (genes values) are randomly generated in each chromosome to represent directly the production sequence or schedule of PCs.

Step 1 **Initialisation:** Randomly generate initial population of N_{pop} chromosomes to represent as precast production schedules. Each chromosome represents a solution, which is schedule of precast component sequence in our precast production scheduling problem.

Step 2 **Evaluate Current Population**: Evaluate objective values for MOPPSM, i.e. makespan and total penalty costs of E/T, of each chromosome in current population pop using Equations (3.12) and (3.13) respectively. For optimising MOPPSM-LD, the Equations (3.18) and (3.19) are respectively used to evaluate objectives makespan and E/T penalty.

Step 3 **Rank Current Population:** The current generation population is ranked by following steps:

(1) **Non-dominated Sort**: Each chromosome of *pop* is assigned a rank by using the fast non-domination sorting procedure described as below.

(1.1) Initialise front counter: $r = 0$.

(1.2) Increase: $r = r + 1$.

(1.3) Find non-dominated solutions from *pop* according to the concept of domination.

(1.4) Assign rank r to these non-dominated solutions.

(1.5) Remove these non-dominated solutions from *pop*.

(1.6) Repeat (1.2) to (1.5) until *pop* is empty.

(2) **Crowding Distance**: The crowding distance value for each chromosome is calculated as below.

(1.1) Initialise distance of all z individuals to be zeros: $d_i = 0$ *for* $i = 1, 2, ..., Z$.

(1.2) For objective function f_k (f_k is makespan or penalty cost), sort the set in ascending order.

(1.3) Let d_1 and d_Z be infinite distance: $d_1 = d_Z = \infty$.

(1.4) For $j = 2,3,\ldots,Z-1$, let $d_j = d_j + \dfrac{(f_k^{(j+1)} - f_k^{(j-1)})}{f_k^{\max} - f_k^{\min}}$.

Step 4 **Generate New Population**:

(1) **Crossover**: Generate an offspring population $pop_{crossover}$ by following steps.

(1.1) **Selection**: Select a pair of parent solutions from the population using binary tournament selection operator based on non-domination rank and crowding distance.

(1.2) **Crossover**: Implement a two-point cut crossover operator, as shown in Figure 4.5 (a), to the selected pair of parents to generate two new child solutions.

(1.3) **Evaluation:** Objective values of MOPPSM problem are evaluated for the two new child solutions according to Equations (3.12) and (3.13). For optimising MOPPSM-LD, the Equations (3.18) and (3.19) are used respectively to evaluate objectives.

(1.4) **Loop**: Repeat (1.1) to (1.3) until $N_{crossover}$ child solutions are generated through crossover operator among $N_{crossover}$ selected parent solutions.

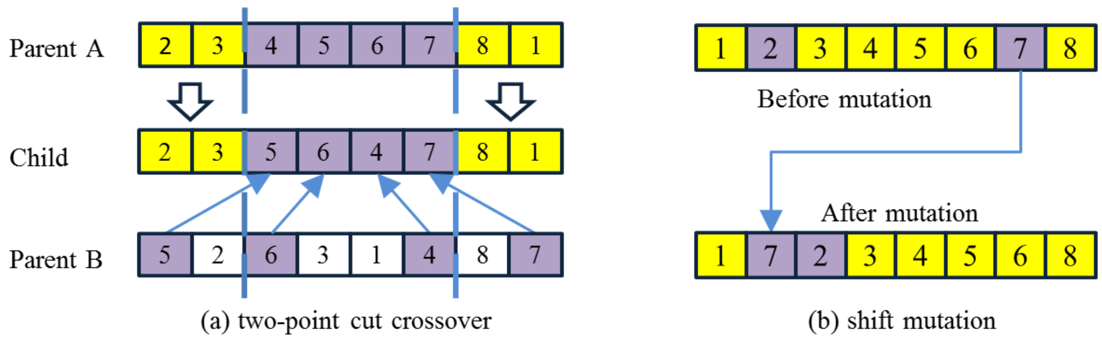

Figure 4.5 Genetic operators' schemas.

(2) **Mutation**: Generate an offspring population $pop_{mutation}$ by following steps.

(2.1) **Selection**: Select a solution from the population using binary tournament selection with crowded-comparison operator. If (rank number of chromosome X is smaller than that of chromosome Y) or (rank number of chromosome X is equal to that of chromosome Y, and crowding distance of X is bigger than Y), the X will be chosen by binary tournament selection operator.

(2.2) **Mutation**: Implement mutation operator, shown in Figure 4.5 (b), to the selected solution to create a new solution.

(2.3) **Evaluation:** Evaluate objective values for the new child solution according to Equations (3.12) and (3.13), or Equations (3.18) and (3.19).

(2.4) **Loop**: Repeat the (2.1) to (2.3) until $N_{mutation}$ parent solutions are selected to mutate $N_{mutation}$ child solutions.

Step 5 **Update Current Population:** The current population *pop* is updated for a further run of the algorithm by following steps.

(1) **Recombination:** The current population *pop* is combined with its offspring populations $pop_{crossover}$ and $pop_{mutation}$: pop **=[** pop, $pop_{crossover}$, $pop_{mutation}$].

(2) **Non-dominated Sort:** Each individual in recombination population *pop* is assigned a rank based on non-domination criteria.

(3) **Crowding Distance:** Calculate the crowding distance value for each individual in recombination population *pop*.

(4) **Selection:** Once recombination population *pop* is sorted based on descending crowding distance and ascending non-domination rank. The individuals of new generation need to be selected from current population. The new population is generated by filling every front subsequently and the extra individuals are deleted. If the population exceeds N by adding all the individuals with rank r, then rank r individuals are selected according to their crowding distance in the descending order. The selection will end once the population size reach to N_{pop}: $pop = pop$ (1: N_{pop}).

Step 6 **Termination Test**: If the algorithm reaches maximum generations, terminate the algorithm and return the Pareto optimal solutions in current population. Otherwise, return to Step 3.

Step 7 **Report**: Similar to last step (as shown in *section 4.1*) of MOVNS, non-dominated solutions in *pop* of final iteration, are reported as optimum or near optimum schedules for MOPPSM or MOPPSM-LD.

4.3 The comparative algorithm MOGLS for the MOPPSM and MOPPSM-LD

Many real world problems consider multiple objectives which unable to be optimised by classical genetic algorithm (GA). Thus, Murata and Ishibuchi (1995) proposed a MOGA, by adding GA with modified selection operation and elite strategy, so that it can find non-dominated solutions for the multi-objective optimisation problems. Later, Ishibuchi and Murata (1998) generated a hybrid MOGA, namely multi-objective genetic local search (MOGLS) algorithm, by applying local search step to all new solutions generated from the crossover and mutation procedures in the MOGA. MOGLS was proved to be an efficient algorithm that can outperform two other multi-objective methods, i.e. the vector evaluated GA and the constant weight GA, in solving MOPPSM (Ko and Wang, 2011). Hence, this research applied MOGLS as a comparative algorithm to search for Parato optimal schedules for both MOPPSM and MOPPSM-LD problems. Evolutionary process of MOGLS is represented in Figure 4.1 (c).

The steps of MOGLS applied in solving MOPPSM or MOPPSM-LD problem is explained as follows.

Step 0 **Encoding:** As described in Figure 4.2 and Figure 4.3, this study encodes the precast production scheduling problem by permutation representation because of the ease with which it encodes the identity of the PCs to be scheduled in each gene. In our adopted permutation representation, a non-repetitive integer numbers (genes values) are randomly generated in each chromosome to represent directly the production sequence or schedule of PCs.

Step 1 **Population initialisation:** Based on GA theory, N_{pop} chromosomes are required to be randomly generated in the initial population to represent as potential solutions. For production scheduling problem, each chromosome stands for a possible production schedule solution of PCs at hand.

Step 2 **Evaluation:** Since the aim of MOGLS is to find non-dominated solutions rather than to determine a single final solution, the fitness values for each obtained schedule have to be evaluated by using a weighted sum of multiple objective functions: Equation (4.1).

$$f(x) = \frac{1}{\omega_1 f_1(x) + \omega_2 f_2(x)} \tag{4.1}$$

$$\omega_1 + \omega_2 = 1 \tag{4.2}$$

$$\omega_i = \frac{random_i}{random_1 + random_2} \tag{4.3}$$

Where ω_1, ω_2, determined as Equation (4.2), are the weights of the optimisation objectives of makespan and penalty costs of E/T respectively and $random_1$, $random_2$, $random_i$ are nonnegative random integers.

Step 3 **Selection:** The RouletteWheel selection is utilised to choose the fitter chromosomes for evolving better generations, where the chance of them to be chosen is proportional to the fitness value evaluated in step 2. Specifically, $(N_{pop} - N_{elite})$ pairs of parent chromosomes are selected in current population by repeating the following procedures:

(1) Randomly generate the weight values ω_1, ω_2 and calculate the fitness

by using Equations (4.2) and (4.3) for all chromosomes.

(2) Select a pair of parent chromosomes according to the selection probability calculated by using Equation (4.4).

$$P(x) = \frac{f(x) - f_{\min}(X)}{\sum_{x \in X} \{ f(x) - f_{\min}(X) \}} \tag{4.4}$$

Step 4 **Crossover and mutation:** As explained in Figure 4.4, two-point cut crossover and shift mutation genetic operators are used for all of the selected ($N_{pop} - N_{elite}$) pairs of parent chromosomes respectively to form new offspring. If no crossover or mutation was performed, offspring is the exact copy of parents.

Step 5 **Elitist strategy:** Update the tentative set of non-dominated solutions according to the fitness evaluation in step 2. The ($N_{pop} - N_{elite}$) solutions created in step 4 are added with the N_{elite} solutions which are randomly selected from tentative set of the non-dominated solutions.

Step 6 **Local search:** Mutation operator is adopted to search local area for all N_{pop} solutions of the present population as recommend by Ko and Wang (2011). For every solution, the search direction of its local search is determined by the weight values that are same as the weight values using for selecting its parent solutions.

Step 7 **Population update:** In this process, the current population is renewed by the improved N_{pop} solutions so that the next generation can continuously include new solutions for evolution.

Step 8 **Termination test:** If terminate condition is satisfied, end the evolutionary process and output the final set of non-dominated solutions, which provides precast production schedulers some options of selecting the best schedule according to their preference. Otherwise, repeat the evolutionary process from step 2 to step 7 for further evolution by using updated population above. Specifically, a pre-specified maximum number of iteration is used as terminate condition in this research.

Step 9 **Report**: PCs sequences decoded from optimum solutions, i.e. solutions of final iteration, are reported as optimum schedules for MOPPSM or MOPPSM-LD.

CHAPTER 5

EXPERIMENTS

This paper analysed the efficiency of the two proposed metaheuristics, i.e. multi-objective variable neighbourhood search (MOVNS) algorithm and non-dominated sorting genetic algorithm II (NSGA-II), and the comparative multi-objective genetic algorithm local search (MOGLS) algorithm in solving the MOPPSM and MOPPSM-LD problems. 10 problem cases of MOPPSM and 5 cases of MOPPSM –LD was generated to test the performances of these three algorithms and all the three algorithms were programmed in MATLAB software and executed on an Intel Core i5 3.3 GHz personal computer with 4 GB of memory. In order to ensure a fair comparison, all algorithms were ran in the same computer until they were terminated by the time limitation method (based on elapsed CPU times) which is widely used criterion in the performance comparison of different metaheuristics (Arroyo, Ottoni, and Oliveira 2011; Duarte et al. 2015; Selvi and Manimegalai 2015; Palubeckis 2017).

5.1 Problem cases

This section generated 10 cases for MOPPSM problem and 5 cases for MOPPSM-LD problem.

5.1.1 MOPPSM: case 1-10

This study mainly validated the performance of the MOVNS, NSGA-II and MOGLS algorithms in solving MOPPSM based on three data sets, in which the first 6 PCs types (out of 26 PCs types) were taken from Chan and Hu (2002a, b), the next 10 PCs types were taken from Benjaoran, Dawood, and Hobbs (2005) and the left 10 PCs types were taken from Wang and Hu (2017), as shown in Table 5.1. Based on this data set, Table 5.2 illustrates 10 cases randomly generated by assigning different amounts to all *n* PCs (PCs) to be produced in precast production process. For example, case 1 assign each type of PCs with specific amount [0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0], respectively. Accordingly, the size of case 1 is 10 PCs which is calculated as 10=0+0+0+0+0+0+1+1+1+1+1+1+1+1+1+1+0+0+0+0+0+0+0 +0+0+0. case 2 assign each type of PCs with specific amount [1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1], respectively. Accordingly, the size of case 2 is 20 PCs which is calculated as 20=1+1+1+0+1+0+1+0+1+0+1+1+1+1+1+1+0+1+1+1+1+1+1 +0+1+1. With the same manner, the size of case 3 to case 10 are 30 PCs, 40 PCs, 50 PCs, 60 PCs, 70 PCs, 80 PCs, 90 PCs and 100 PCs, calculated as follows:

30=0+0+0+0+0+0+4+2+2+5+3+3+2+2+5+2+0+0+0+0+0+0+0+0+0+0.

40=2+1+2+1+2+1+1+2+1+2+1+2+1+1+2+1+2+1+2+2+1+2+2+2+2+1.

50=2+2+2+2+2+2+2+2+2+2+3+2+2+2+2+2+2+1+2+2+2+2+2+1+1+2.

60=2+3+4+1+5+1+4+1+2+5+2+1+2+2+2+1+3+4+1+2+1+2+1+4+2+2.

70=1+3+5+2+3+4+1+2+3+1+2+3+1+3+6+2+5+2+3+1+4+1+4+2+5+1.

80=1+2+1+5+4+2+6+2+3+2+4+3+5+4+4+2+4+3+1+5+3+4+1+4+3+2.

90=6+3+4+5+7+5+6+4+5+10+7+5+4+5+12+2+0+0+0+0+0+0+0+0+0+0.

100=1+5+2+5+7+5+9+4+5+3+7+6+5+5+2+4+2+5+3+2+1+5+1+3+2+1.

Table 5.1 Production data of PCs for 10 cases of MOPPSM.

PCs type	Operation time of each process (hour)						Due dates (h)	E&T penalties	
	N_1	N_2	N_3	N_4	N_5	N_6		Earliness	Tardiness
1	1	0.8	1.2	12	1.5	0.5	28	2	10
2	1.7	2	2	12	1.5	2.5	28	2	10
3	0.4	0.5	0.6	12	0.5	0	28	1	10
4	0.3	0.4	0.5	12	0.4	1	28	1	10
5	1.5	1.8	1.2	12	1.5	1.5	52	2	10
6	1.5	1.6	1.5	12	1.8	0.8	32	2	10
7	2	1.6	2.4	12	2.5	1	112	2	10
8	3.4	4	4	12	2.4	5	112	2	10
9	0.8	1	1.2	12	0.8	0	112	1	10
10	0.6	0.8	1	12	0.6	2	112	1	10
11	3	3.6	2.4	12	2.4	3	208	2	10
12	3	3.2	3	12	3	1.6	128	2	10
13	1.3	0.9	2.4	12	1.9	1.8	144	2	10
14	1.7	1.4	1.1	12	0.9	0.7	144	2	20
15	2.2	1.8	1.2	12	2.3	0.7	144	1	20
16	1.6	3.2	2.3	12	2.1	2.7	240	1	20
17	1.5	2	0.5	8	1	0.5	164	2	10
18	1	2	0.4	8	1	0.5	140	2	10
19	1	1.5	0.5	8	0.5	0.5	164	2	10
20	0.5	1	0.3	8	0.3	0.5	160	2	10
21	1	0.8	1	8	1.5	0.5	160	2	10
22	0.5	2	0.4	8	0.5	0.5	164	2	10
23	1.5	2	0.5	8	1	0.4	140	2	10
24	0.5	2	0.3	8	0.6	0.3	164	2	10
25	1.5	1.8	1.2	8	1.5	1.5	140	2	10
26	0.4	0.5	0.6	8	0.5	0.5	164	2	10

Table 5.2 Problem cases with different size.

PCs type	case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8	case 9	case 10
1	0	1	0	2	2	2	1	1	6	1
2	0	1	0	1	2	3	3	2	3	5
3	0	1	0	2	2	4	5	1	4	2
4	0	0	0	1	2	1	2	5	5	5
5	0	1	0	2	2	5	3	4	7	7
6	0	0	0	1	2	1	4	2	5	5
7	1	1	4	1	2	4	1	6	6	9
8	1	0	2	2	2	1	2	2	4	4
9	1	1	2	1	2	2	3	3	5	5
10	1	0	5	2	2	5	1	2	10	3
11	1	1	3	1	3	2	2	4	7	7
12	1	1	3	2	2	1	3	3	5	6
13	1	1	2	1	2	2	1	5	4	5
14	1	1	2	1	2	2	3	4	5	5
15	1	1	5	2	2	2	6	4	12	2
16	1	1	2	1	2	1	2	2	2	4
17	0	0	0	2	2	3	5	4	0	2
18	0	1	0	1	1	4	2	3	0	5
19	0	1	0	2	2	1	3	1	0	3
20	0	1	0	2	2	2	1	5	0	2
21	0	1	0	1	2	1	4	3	0	1
22	0	1	0	2	2	2	1	4	0	5
23	0	1	0	2	2	1	4	1	0	1
24	0	0	0	2	1	4	2	4	0	3
25	0	1	0	2	1	2	5	3	0	2
26	0	1	0	1	2	2	1	2	0	1
Problem size	10 PCs	20 PCs	30 PCs	40 PCs	50 PCs	60 PCs	70 PCs	80 PCs	90 PCs	100 PCs

5.1.2 MOPPSM-LD: case 11-15

Table 5.3 lists the operation time and E/T penalties for case 11 to case 15 of MOPPSM-LD. We can see that operation time of the six fabrication processes (i.e. N_1-N_6) and E/T penalties are same with those presented in Table 5.1. Then, all the PCs were assumed to have the same 10 hours storing (N_7) according to Wang and Hu (2017). Normally, the time spent on delivery (N_8), i.e. last operation of MOPPSM-LD, greatly depends on how far the customer's destination. Hence, the delivery time could be listed in Tables 5.4-5.8 in which customer orders are described.

Tables 5.4-5.8 list the information of customer orders for cases 11-15 of MOPPSM-LD problem. The information includes the number of customer orders, PCs amount, delivery times and due dates. Specifically, Table 5.4 and 5.5respectively list the 10 different customer orders with PCs number 20 and 40, while Tables 5.6-5.8 list the 26 different customer orders with PCs number 60, 80 and 100 respectively. Furthermore, if we take Table 5.4 as an example, we can see that first customer's (c1) order includes 1 PC amount of both PC type 1 and PC type 20. Moreover, in Tables 5.4-5.8, different delivery times are given based on the assumption that all the customer orders are from different areas and they expect to receive the PCs by different due dates. The smallest delivery time and due date are 0.5 hours and 50 hours respectively, while the biggest delivery time and due date are 13 hours and 250 hours respectively.

Table 5.3 Operation time and E/T penalties for MOPPSM-LD.

PCs type	Operation time of each process (hour)							E/T penalties	
	N_1	N_2	N_3	N_4	N_5	N_6	N_7	Earliness	Tardiness
1	1	0.8	1.2	12	1.5	0.5	10	2	10
2	1.7	2	2	12	1.5	2.5	10	2	10
3	0.4	0.5	0.6	12	0.5	0	10	1	10
4	0.3	0.4	0.5	12	0.4	1	10	1	10
5	1.5	1.8		12	1.5	1.5	10	2	10
6	1.5	1.6	1.5	12	1.8	0.8	10	2	10
7	2	1.6	2.4	12	2.5	1	10	2	10
8	3.4	4	4	12	2.4	5	10	2	10
9	0.8	1	1.2	12	0.8	0	10	1	10
10	0.6	0.8	1	12	0.6	2	10	1	10
11	3	3.6	2.4	12	2.4	3	10	2	10
12	3	3.2	3	12	3	1.6	10	2	10
13	1.3	0.9	2.4	12	1.9	1.8	10	2	10
14	1.7	1.4	1.1	12	0.9	0.7	10	2	20
15	2.2	1.8	1.2	12	2.3	0.7	10	1	20
16	1.6	3.2	2.3	12	2.1	2.7	10	1	20
17	1.5	2	0.5	8	1	0.5	10	2	10
18	1	2	0.4	8	1	0.5	10	2	10
19	1	1.5	0.5	8	0.5	0.5	10	2	10
20	0.5	1	0.3	8	0.3	0.5	10	2	10
21	1	0.8	1	8	1.5	0.5	10	2	10
22	0.5	2	0.4	8	0.5	0.5	10	2	10
23	1.5	2	0.5	8	1	0.4	10	2	10
24	0.5	2	0.3	8	0.6	0.3	10	2	10
25	1.5	1.8	1.2	8	1.5	1.5	10	2	10
26	0.4	0.5	0.6	8	0.5	0.5	10	2	10

Table 5.4 Customer orders in case 11: 10 orders and 20 PCs.

PCs type	Customer orders									
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
1	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	1	0	0	0
15	0	0	0	0	0	1	0	0	0	0
16	0	0	0	0	1	0	0	0	0	0
17	0	0	0	1	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0
19	0	1	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0
N_8	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
due date	50	100	150	200	250	50	100	150	200	250

Table 5.5 Customer orders in case 12: 10 orders and 40 PCs.

PCs type	Customer orders									
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
1	1	0	0	0	0	0	1	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0
7	0	0	0	1	0	0	1	0	0	0
8	1	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	2	0	0	1	0
10	0	0	0	1	0	0	0	0	0	1
11	0	1	0	0	2	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	1	0	0	0
14	0	1	0	1	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	1	0	1
17	0	0	0	1	0	0	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0
19	1	0	1	0	0	0	0	0	2	0
20	0	0	0	0	0	0	0	0	0	0
21	0	2	0	0	1	0	0	1	0	0
22	0	0	0	0	1	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	1	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
N_8	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
due date	50	100	150	200	250	50	100	150	200	250

Table 5.6 Customer orders in case 13: 26 orders and 60 PCs.

Products type	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26
	Customer orders																									
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	1	0	0	0	1	0	0	0	0
12	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
15	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
17	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
21	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
24	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
25	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
N_8	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11	11.5	12	12.5	13
due date	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	150

Table 5.7 Customer orders in case 14: 26 orders and 80 PCs.

Products	Customer orders																									
type	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
5	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
14	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0
15	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
16	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
18	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
19	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
20	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
21	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	0
24	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0
25	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0
26	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
N_8	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	10.5	11	11.5	12	12.5	13
due date	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	150

Table 5.8 Customer orders in case 15: 26 orders and 100 PCs.

Products	Customer orders																									
type	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	c26
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
7	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
9	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	1	0	1	0	0	2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
11	0	1	0	2	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0
12	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	2	1	0	0
13	0	0	0	1	0	2	0	0	1	0	1	1	1	0	0	0	0	0	0	0	2	0	0	0	1	0
14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0
15	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
16	0	0	2	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	2	1	0	0
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0
20	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
21	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
22	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
26	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N_8	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10	11	11	11.5	12	12.5	13
due date	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	50	100	150	200	250	150

5.2 Performance Measures

For each case of MOPPSM or MOPPSM-LD, this research compared the non-dominated solutions (approximated Pareto fronts) obtained by the three tested algorithms, i.e. MOVNS, NSGA-II and MOGLS, in terms of solution quality. Let S_1, S_2 and S_3 be the sets of non-dominated solutions obtained by the tested algorithms MOVNS, NSGA-II and MOGLS, respectively. Since the optimal Pareto front for each MOPPSM or MOPPSM-LD case is unknown, a reference set (the best known Pareto front) is used (Ishibuchi, Yoshida, and Murata 2003). The reference set, denoted by S^*, is generated by gathering all non-dominated solutions of MOVNS, NSGA-II and MOGLS. Then, the performance of an algorithm was measured in terms of the quality of the solutions in non-dominated solution set S_k (k = 1, 2, 3) relative to the solutions in reference set S^*. To compare the performance of the algorithms, the spread metric and distance metrics have been employed for evaluating the quality of non-dominated solution set S_k (k = 1, 2, 3). These two types of performance metrics are briefly explained as follows:

- *Spread metric.* Spread (SP) metric was employed in many studies (Zitzler 1999, Kaige, Murata, and Ishibuchi 2003, Deb 2010, Ko and Wang 2011) to estimate the extent or diversity of the final non-dominated solution set S_k (k = 1, 2, 3) obtained by each algorithm. Metric SP represents the diagonal of the minimum N-dimension hyper rectangle in the solution space (Ko and Wang 2011). The bigger the value of $SP(S_k)$, the better the solutions of S_k. Metric SP can be calculated by Equation (5.1).

$$SP = \sqrt{\sum_{i=1}^{N} (\max_{j=1}^{|S_k|} f_i^*(x_j) - \min_{j=1}^{|S_k|} f_i^*(x_j))^2} \tag{5.1}$$

Where $x_j \in S_k$, for j=1,2,…, $|S_k|$, S_k denotes the set of non-dominated solutions obtained by algorithm k (k = 1, 2, 3). $f_i^*(\cdot)$ is the ith objective normalised using the reference set S^* in the N-dimensional objective space. The normalisation was accomplished based on the following equation.

$$f_i^*(x) = 100 \times \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}} \tag{5.2}$$

where f_i^{max} and f_i^{min} are the maximum and minimum values of the ith objective in the reference set S^* respectively.

- *Distance metrics.* The distance metric was utilised for evaluating the performance of solution set S_k . More specifically, the minimum distance (D_{min}), average distance ($D1_R$) and maximum distance (D_{max}) were used to investigate the distance from each solution of reference set S^* to its nearest solution in S_k . In this manner, the convergence and diversity of a Pareto front set can be both considered. The distance metrics are widely applied for multi-objective scheduling problems (Ishibuchi, Yoshida, and Murata 2003; Armentano and Claudio 2004; Arroyo and Armentano 2004; Framinan and Leisten 2008; Arroyo, Ottoni, and Oliveira 2011; Lei 2015; Lei and Zheng 2017). Unlike, spread metric, distance metric consider not only diversity but also consider convergence of the obtained solution set. The smaller values of $D_{min}(S_k)$, $D1_R(S_k)$ and $D_{max}(S_k)$, computed by Equations (5.3)-(5.5), indicate the better solutions of S_k.

$$D_{min} = min_{y \in S*}\{min\{d_{xy}, x \in S_k\}\} \tag{5.3}$$

$$D1_R = \frac{1}{|S*|}\sum_{y \in S*} min\{d_{xy}, x \in S_k\} \tag{5.4}$$

$$D_{max} = max_{y \in S*}\{min\{d_{xy}, x \in S_k\}\} \tag{5.5}$$

where d_{xy} represents the Euclidean distance, computed by Equation (5.6), between a non-dominated solution x and a reference solution y in the N-dimensional normalised objective space,

$$d_{xy} = \sqrt{(f_1^*(y) - f_1^*(x))^2 + \cdots + (f_N^*(y) - f_N^*(x))^2} \tag{5.6}$$

5.3 Parameters settings

With respect to parameters setting, Table 5.9 lists the main parameters involved in each algorithm. As can be seen in Table 5.9, firstly, the population size (N_{pop}) needs to be set for all algorithms. Secondly, both NSGA-II and MOGLS involves crossover rate (p_c) and mutation rate (p_m). Lastly, MOGLS also involves elite number (N_{elite}) and local search terminated value (k_t). Since different parameter values may be appropriate for each of the three algorithms (i.e. MOVNS, NSGA-II and MOGLS), the different combinations of parameter values of N_{pop} (50, 100, 150), p_c (0.5, 0.7, 0.9) and p_m (01, 0.3, 0.5) were examined by the preliminary experiments for all related algorithms in all cases of MOPPSM problem, similar to the parameters analyses in the work of Ishibuchi, Yoshida, and Murata (2003). In other words, MOVNS was tested in three different N_{pop} while NSGA-II and MOGLS were tested with 27 combinations of the above three parameters in all the addressed cases. For the MOGLS, the values of N_{elite} and k_t were respectively set to 4 and 2 for all cases of MOPPSM and MOPPSM-LD problems according to Ko and Wang (2011). Moreover, maximum CPU times in seconds (*s*) were set as the termination criterion as shown in Table 5.10 for all tested algorithms in each case. The maximum CPU time for small-size cases less than 50 PCs were set based on the average convergence time of all algorithms while the maximum CPU time for the mid- and large-size ones up to 100 PCs were limited to 600 s.

Based on the termination criterions shown in Table 5.10, each algorithm was applied to each addressed case 10 times for each combination of the parameter values. The indicators $D1_R$ and D_{max} were implemented in those experiments to evaluate the algorithm performance with different combination parameter values. Enormous experiments indicated that the combinations of parameter values: N_{pop} =150 for the MOVNS, (N_{pop}, p_c, p_m) = (150, 0.9, 0.3) for the NSGA-II and (N_{pop}, p_c, p_m) = (100, 0.7, 0.5) for the MOGLS, enable related algorithms to obtain optimal or near-optimal solutions in most cases, in terms of $D1_R$ and D_{max} values. Therefore, all the algorithms were controlled by these parameter values in order to carry out the experiments.

Table 5.9 Parameter settings of all algorithms.

Parameters	**MOVNS**	**NSGA-II**	**MOGLS**
N_{pop}	(50, 100, 150)	(50, 100, 150)	(50, 100, 150)
pc	-	(05, 0.7, 0.9)	(05, 0.7, 0.9)
pm	-	(0.1, 0.3, 0.5)	(0.1, 0.3, 0.5)
N_{elite}	-	-	4
k_t	-	-	2

Table 5.10 The maximum CPU time for terminating the tested algorithms in each case.

Cases	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6-10	Case 11	Case 12	Case 13-15
CPU time (s)	15	70	120	300	600	600	70	300	600

5.4 Computational Results and Comparison

This section showed the performances of different algorithms in solving MOPPSM and MOPPSM-LD. Using the parameters settings and maximum CPU times determined in *section 4.3***,** the three algorithms were run 20 independent times (replications) for all the 10 cases of MOPPSM and 5 cases of MOPPSM-LD. The sets S_1 , S_2 and S_3 contain the non-dominated solutions found among all the runs. The spread metric SP and distance measures D_{min} , $D1_R$ and D_{max} were calculated for these non-dominated solution sets.

5.4.1 Results and Comparison for solving MOPPSM

The experimental results of solving MOPPSM with the three algorithms are displayed in Tables 5.11 and 5.12, which present the average values of SP , D_{min} , $D1_R$ and D_{max} for each group of cases and for each algorithm, and the bold font indicates the best result among all algorithms. As can be seen in Tables 5.11 and 5.12, the average values of all indicators in all cases showed that MOVNS was superior to NSGA-II and MOGLS. More specifically, regarding the SP indicator in Table 5.11, MOVNS algorithm outperformed NSGA-II and MOGLS in all addressed cases. With respect to $D1_R$ indicator, MOVNS outperformed NSGA-II in all case sizes while it can win the MOGLS algorithm in almost all cases except for the 100 PCs case. The MOGLS can achieve the best value of $D1_R$ measure ($D1_R$ = 7.23) while the MOVNS was at $D1_R$ = 7.36. In addition, for the values of D_{min} shown in Table 5.12, MOVNS achieved better than the other two algorithms except for the 10 PCs case. However, for the 10 PCs case, all the three algorithms can be accomplished an equivalent value at D_{min} = 0. For the values of D_{max} , MOVNS also outperformed NSGA-II and MOGLS except for the 40 PCs case. In the case of 40 PCs, NSGA-II and MOGLS acquired the same value of D_{min} = 11.73.

Table 5.11 Comparison of the three algorithms in solving MOPPSM using SP and $D1_R$ indicators.

Cases	SP			$D1_R$		
	MOVNS	NSGA-II	MOGLS	MOVNS	NSGA-II	MOGLS
Case 1	**145.38**	139.34	143.63	**0.36**	2.86	1.09
Case 2	**156.45**	75.77	64.79	**2.33**	3.25	3.71
Case 3	**144.82**	113.07	110.23	**3.58**	11.56	12.33
Case 4	**167.38**	153.75	147.52	**2.64**	4.28	3.58
Case 5	**143.07**	74.35	84.49	**5.36**	30.69	25.51
Case 6	**151.06**	114.03	121.14	**16.58**	35.46	20.06
Case 7	**148.95**	74.04	103.21	**6.93**	23.74	8.02
Case 8	**314.88**	47.80	79.47	**6.70**	20.34	11.54
Case 9	**178.94**	82.69	61.83	**11.67**	28.77	31.24
Case 10	**293.94**	155.44	96.76	7.36	12.79	**7.23**
average	**184.49**	103.03	101.31	**6.35**	17.37	12.43

Table 5.12 Comparison of the three algorithms in solving MOPPSM using D_{min} and D_{max} indicators.

Cases	D_{min}			D_{max}		
	MOVNS	NSGA-II	MOGLS	MOVNS	NSGA-II	MOGLS
Case 1	0.00	0.00	0.00	**4.33**	14.34	16.64
Case 2	**0.12**	0.45	0.91	**47.53**	61.32	67.37
Case 3	**0.48**	1.14	2.57	**12.63**	44.02	49.54
Case 4	**0.00**	0.18	0.05	11.90	**11.73**	**11.73**
Case 5	**1.00**	4.02	4.00	**24.96**	67.71	72.18
Case 6	**2.83**	14.50	7.73	**59.35**	81.46	65.41
Case 7	**0.52**	6.51	1.32	**44.71**	91.32	70.96
Case 8	**1.45**	7.81	3.44	**31.72**	84.24	77.56
Case 9	**2.52**	3.87	5.72	**28.51**	71.38	79.25
Case 10	**2.56**	4.52	3.29	**34.19**	70.77	67.95
average	**1.15**	4.30	2.90	**29.98**	59.83	57.86

Figure 5.1 illustrates the graphical results of all indicators. Overall, it shows that the MOVNS obviously outperformed NSGA-II and MOGLS while the NSGA-II seemed to have similar performance to MOGLS for solving the MOPPSM. Moreover, focusing on the SP and D_{max} indicators, the MOVNS was more powerful than the other two algorithms when the size of the MOPPSM problem case becomes larger.

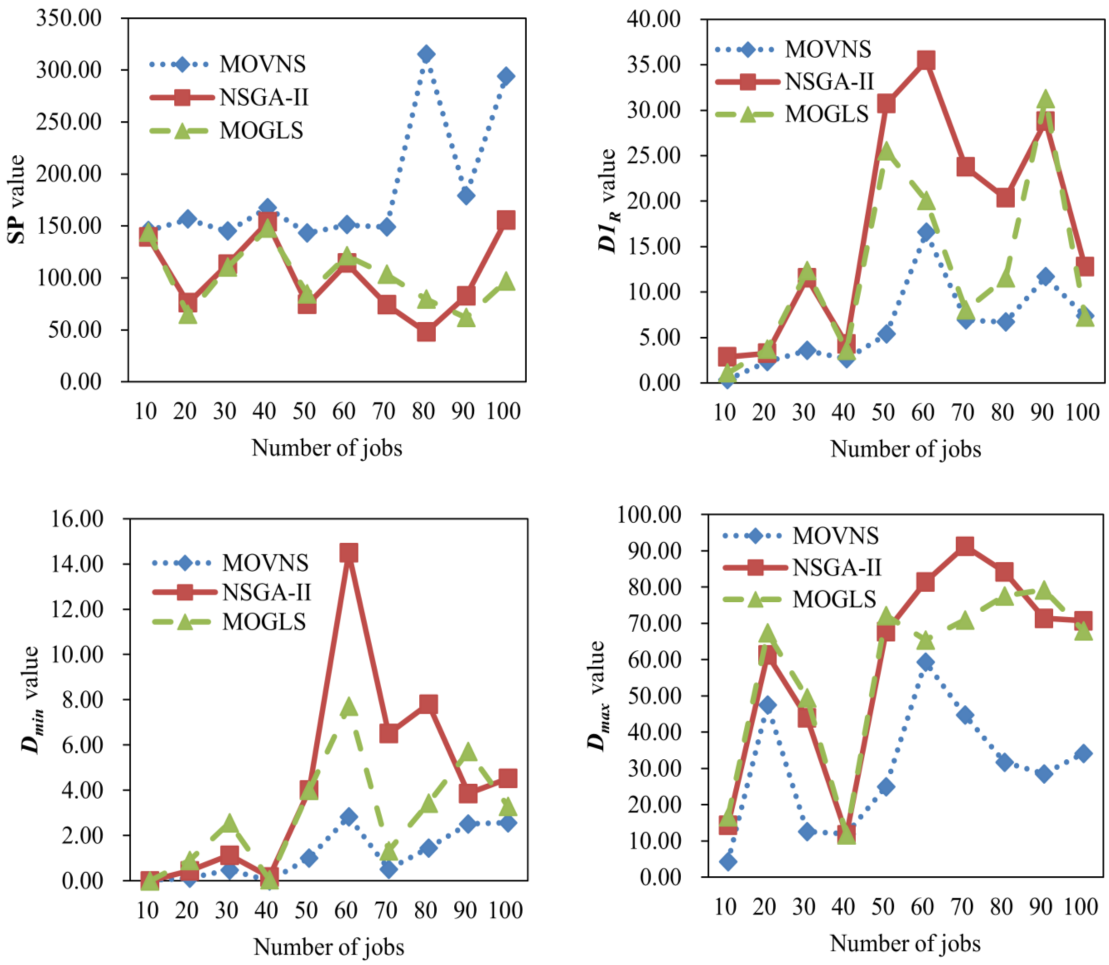

Figure 5.1 Performance comparison of the three algorithms in solving MOPPSM (case 1-10) using the SP, D_{min}, $D1_R$ and D_{max} indicator.

5.4.2 Results and Comparison for solving MOPPSM-LD

Tables 5.13 and 5.14 present the average values of SP, D_{min}, $D1_R$ and D_{max} of 20 times optimising each of MOPPSM-LD cases by MOVNS, NSGA-II and MOGLS algorithms. The bold font numbers in the Tables indicate the best results among all algorithms. With respect to all four SP, D_{min}, $D1_R$ and D_{max} indicators, MOVNS can achieve the best value in all cases except for the case 11, in which NSGA-II achieved best $D1_R$ value (i.e. 11.31) and best D_{max} value (i.e. 23.87). Furthermore, the average values of SP, D_{min}, $D1_R$ and D_{max} of all cases show that the performance of NSGA-II was as good as MOGLS, and MOVNS was superior to both NSGA-II and MOGLS. For average SP value, MOVNS (SP=115.20) was 35.25% overcome MOGLS (SP=85.18). In a similar way, MOVNS was superior to MOGLS 32.78% in average $D1_R$ value, 57.76% in average D_{min} value, and 14.75% in average D_{max} value.

Table 5.13 Comparison of the three algorithms in solving MOPPSM-LD using the SP and $D1_R$ indicators.

Cases	SP			$D1_R$		
	MOVNS	NSGA-II	MOGLS	MOVNS	NSGA-II	MOGLS
Case 11	**146.41**	130.64	119.19	20.85	**11.31**	13.53
Case 12	**131.74**	124.89	116.41	**6.13**	14.70	7.99
Case 13	**63.81**	46.77	37.14	**20.97**	26.43	36.85
Case 14	**90.44**	60.45	68.49	**19.76**	35.80	37.02
Case 15	**143.59**	110.23	84.65	**11.00**	27.21	21.70
average	**115.20**	94.59	85.18	**15.74**	23.09	23.42

Table 5.14 Comparison of the three algorithms in solving MOPPSM-LD using the D_{min} and D_{max} indicators.

Cases	D_{min}			D_{max}		
	MOVNS	NSGA-II	MOGLS	MOVNS	NSGA-II	MOGLS
Case 11	**5.82**	7.46	10.44	37.73	**23.87**	24.17
Case 12	**1.25**	7.60	3.89	**39.09**	56.46	40.84
Case 13	**8.83**	13.26	18.10	**66.37**	85.66	81.28
Case 14	**2.88**	8.00	7.35	**56.98**	75.45	64.39
Case 15	**6.38**	15.59	19.82	**42.17**	60.90	73.61
average	**5.03**	10.38	11.92	**48.47**	60.47	56.86

Figure 5.2 illustrates the graphical results of all indicators. Overall, it shows that the MOVNS obviously outperformed NSGA-II and MOGLS while the NSGA-II seemed to have similar performance to MOGLS for solving the

MOPPSM-LD.

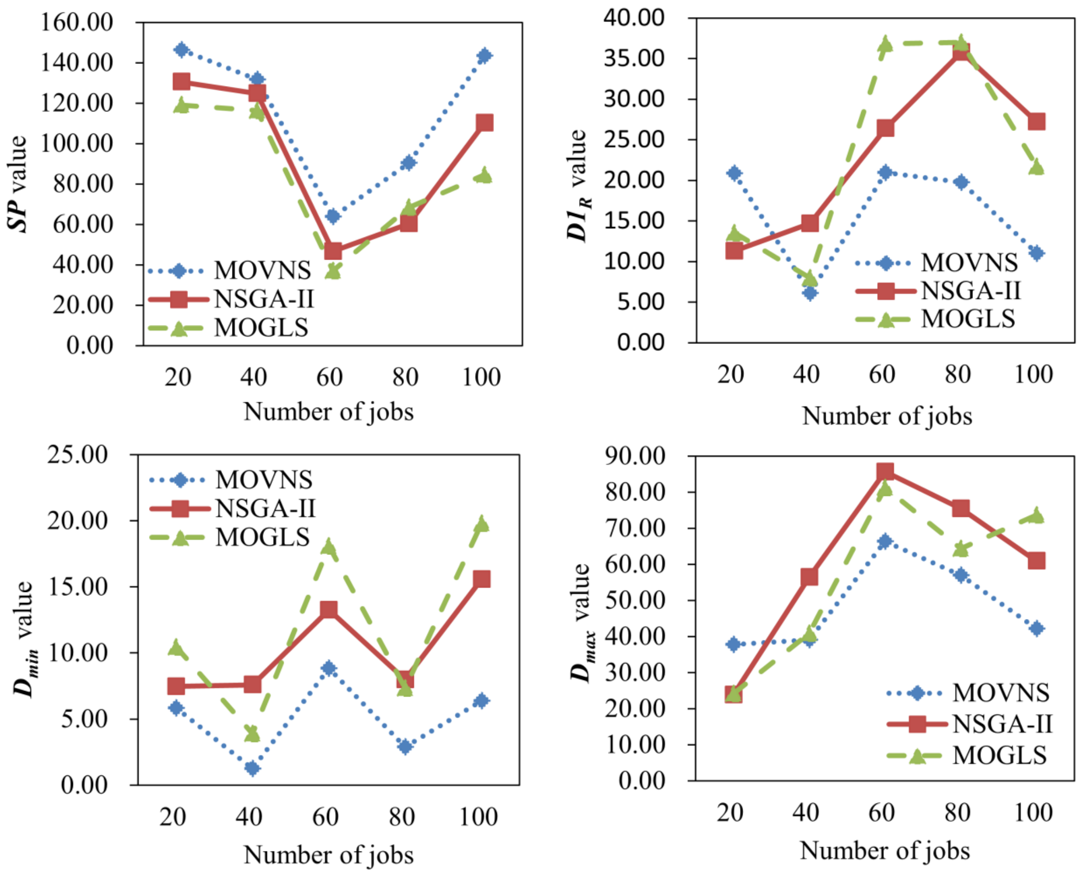

Figure 5.2 Performance comparison of the three algorithms in solving MOPPSM-LD (case 11-15) using the SP, D_{min}, $D1_R$ and D_{max} indicator.

CHAPTER 6

CONCLUSION AND RECOMMENDATION

6.1 Conclusions

Precast production scheduling problem is one of several important aspects for decision making in real-world construction industries. This work considered the MOPPSM based on six precast production processes with two conflicting objectives: the makespan and the total penalty costs of E/T. This research also extended MOPPSM to MOPPSM-LD by considering lot delivery with the complex customer orders. The multi-objective variable neighbourhood search (MOVNS) and the non-dominated sorting genetic algorithm II (NSGA-II) were developed to optimise the MOPPSM and MOPPSM-LD. In addition, this study also contributed the extension on the size of the MOPPSM problem case and the application of new performance indicators, called distance metrics. The performances of the two proposed metaheuristic algorithms were validated by 15 problem cases and compared with the multi-objective genetic local search (MOGLS) algorithm. The experiment results showed that MOVNS outperformed NSGA-II and MOGLS while the NSGA-II can be considered as an equally comparative algorithm to MOGLS.

6.2 Suggestions and future work

For the future study, it involves that the MOVNS algorithm would be extended to more complicated MOPPSM by considering more objectives and more complex manufacturing disturbances situations, such as rush order arrival, due date change, uncertain operation time, as well as buffer size.

REFERENCES

Adibi, M. A., M. Zandieh, and M. Amiri. 2010. “Multi-Objective Scheduling of Dynamic Job Shop using Variable Neighbourhood Search.” *Expert Systems with Applications* 37 (9): 282–287. doi:10.1016/j.eswa.2009.05.001.

Al-Bazi, A., and N. Dawood, 2008. “Enterprise Simulation Of The Precast Concrete Manufacturing Industry.” International Conference on Information Technology in Construction. Santiago, Chile, July 15-17.

Amiri, M., M. Zandieh, M. Yazdani, and A. Bagheri. 2010. “A Variable Neighbourhood Search Algorithm for the Flexible Job-Shop Scheduling Problem.” International Journal of Production Research 48 (19): 5671–5689. doi:10.1080/00207540903055743.

Armentano, V. E., and J. E. C. Arroyo. 2004. “An Application of a Multi-Objective Tabu Search Algorithm to a Bicriteria Flowshop Problem.” Journal of Heuristics 10 (5): 463–481. doi:10.1023/B:HEUR.0000045320.79875.e3.

Arroyo, J. E. C., and V. E. Armentano. 2004. “A Partial Enumeration Heuristic for Multi-Objective Flow Shop Scheduling Problems.” Journal of the Operational Research Society 55 (9): 1000–1007. doi:10.1057/palgrave.jors.2601746.

Arroyo, J. E. C., R. Ottoni, and A. P. Oliveira. 2011. “Multi-Objective Variable Neighbourhood Search Algorithms for a Single Machine Scheduling Problem with Distinct Due Windows.” Electronic Notes in T heoretical Computer Science 281: 5–19. doi:10.1016/j.entcs.2011.11.022.

Benjaoran, V., N. N. Dawood, and B. Hobbs. 2005. “Flow Shop Scheduling Model for Bespoke Precast Concrete Production Planning.” Construction Management and Economics 23 (1): 93–105. doi:10.1080/0144619042000287732.

Behnamian, J., and Fatemi Ghomi S.M.T. 2014. Multi-objective fuzzy multiprocessor flowshop scheduling. Applied Soft Computing, 21, 139–148. Doi: 10.1016/j.asoc.2014.03.031.

Behnamian, J. 2016 . "Survey on fuzzy shop scheduling" Fuzzy Optimization and Decision Making, September, Volume 15, Issue 3, pp 331–366. Doi: 10.1007/s10700-015-9225-5.

Chan, W. T., and H. Hu. 2001. "An Application of Genetic Algorithms to Precast Production Scheduling." Computers and Structures 79 (17): 1605–1616. doi:10.1016/S0045-7949(01)00036-0.

Chan, W.T., and H. Hu. 2002a. "Production Scheduling for Precast Plants using a Flow Shop Sequencing Model." Journal of Computing in Civil Engineering 16 (3): 165–174. doi:10.1061/(ASCE)0887-3801(2002)16:3(165).

Chan, W.T., and H. Hu. 2002b. "Constraint Programming Approach to Precast Production Scheduling." Journal of Construction Engineering and Management 128 (6): 513–521. doi:10.1061/(ASCE)0733-9364(2002)128:6(513).

Campbell, H. G., R. A. Dudek, and M. L. Smith. 1970. "A Heuristic Algorithm for the n Job, m Machine Sequencing Problem." Management science 16 (10): B630-B637. doi:10.1287/mnsc.16.10.B630.

Dannenbring, D. G. 1977. "An Evaluation of Flow Shop Sequencing Heuristics." Management science 23 (11): 1174–1182. doi:10.1287/mnsc.23.11.1174.

Dawood N.N. (1993). Knowledge elicitation and dynamic scheduling using a simulation model: An application to the precast manufacturing process. In Proceedings of the Civil-Comp93, Part 4: Knowledge based systems for civil and structural engineering (p. 73).

Dawood N.N. & Neale R.H. (1993). A capacity planning model for precast concrete building, Building and Environment, 28(l), pp. 8 1-95.

Dawood N.N. (1995). Scheduling in the precast concrete industry using the simulation modeling approach. Building and Environment, Vol. 30, No. 2, pp. 197-207.

Dawood N.N. (1996). A simulation model for eliciting scheduling knowledge: an application to the precast manufacturing process. Advances in Engineering Software, 25, 215-223.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA II." IEEE Transactions on Evolutionary Computation 6 (2): 182–197. doi:10.1109/4235.996017.

Deb, K. 2010. Multi-Objective Optimization using Evolutionary Algorithms. New York: Wiley.

Duarte, A., J. J. Pantrigo, E. G. Pardo, and N. Mladenovic. 2015. "Multi-Objective Variable Neighbourhood Search: An Application to Combinatorial Optimization Problems." Journal of Global Optimization 63 (3): 515–536. doi:10.1007/s10898-014-0213-z.

Eskandari, H., and A. Hosseinzadeh. 2014. "A Variable Neighbourhood Search for Hybrid Flow-Shop Scheduling Problem with Rework and Set-up Times." Journal of the Operational Research Society 65 (8): 221–1231. doi:10.1057/jors.2013.70.

Framinan, J. M., and R. Leisten. 2008. "A Multi-Objective Iterated Greedy Search for Flow Shop Scheduling with Makespan and Flowtime Criteria." OR Spectrum 30 (4): 787–804. doi:10.1007/s00291-007-0098-z.

Geiger, M. J. 2004. “Randomized Variable Neighbourhood Search for Multi Objective Optimization.” Paper Presented at the 4th EU/ME Workshop: Design and Evaluation of Advanced Hybrid Meta-Heuristics, Nottingham, United Kingdom, 4–5th November 2004.

Gupta, J. N. 1971. “A Functional Heuristic Algorithm for the Flow Shop Scheduling Problem.” Journal of the Operational Research Society 22 (1): 39–47. doi:10.2307/3008015.

Hansen, P., and N. Mladenović. 2001. “Variable Neighbourhood Search: Principles and Applications.” European Journal of Operational Research 130 (3): 449–467. doi:10.1016/S0377-2217(00)00100-4.

Ishibuchi, H., and T. Murata. 1998. “Multi-Objective Genetic Local Search Algorithm and Its applications to Flow Shop Scheduling.” IEEE Transactions on Systems, Man and Cybernetics 28 (3): 392–403. doi:10.1109/5326.704576.

Ishibuchi, H., T. Yoshida, and T. Murata. 2003. “Balance between Genetic Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling.” IEEE Transactions on Evolutionary Computation 7 (2): 204–223. doi:10.1109/TEVC.2003.810752.

Johnson S.M. (1954). Optimal two-and three-stage production schedules with set up times included. Naval Research Logistics Quarterly, 1, 61–8.

Kaige, S., T. Murata, and H. Ishibuchi. 2003. “Performance Evaluation of Memetic EMO Algorithms using Dominance Relation-Based Replacement Rules on MOO Test Problem.” Paper Presented at the IEEE International Conference on Systems, Man and Cybernetics, Washington, USA, 5–8th October 2003.

Ko, C. H., and S. Wang. 2010. “GA-based decision support systems for precast production planning.” Automation in Construction 19: 907–916. doi:10.1016/j.autcon.2010.06.004.

Ko, C. H., and S. Wang. 2011. “Precast Production Scheduling using Multi-Objective Genetic Algorithms.” Expert Systems with Applications 38 (7): 8293–8302. doi:10.1016/j.eswa.2011.01.013.

Ko, C. H. 2016. “Impact of the Buffer Size on Precast Fabrication.” Paper Presented at the 24th Annual Conference of the International Group for Lean Construction, Boston, MA, USA.

Konczak, A., and J. Paslawski. 2015. “Decision Support in Production Planning of Precast Concrete Slabs Based on Simulation and Learning from Examples.” Procedia Engineering 122: 81–87. doi:10.1016/j.proeng.2015.10.010.

Lei, D., and X. P. Guo. 2011. “Variable Neighbourhood Search for Minimizing Total Tardiness on Flow Shop with Batch Processing Machines.” International Journal of Production Research 49 (2): 519–529. doi:10.1080/00207540903536130.

Lei, D., and X. P. Guo. 2014. “Variable Neighbourhood Search for Dual-Resource Constrained Flexible Job Shop Scheduling.” International Journal of Production Research 52 (9): 2519–2529. doi:10.1080/00207543.2013.849822.

Lei, D. 2015. “Variable Neighbourhood Search for Two-Agent Flow Shop Scheduling. Computers & Industrial Engineering 80: 125–131. doi:10.1016/j.cie.2014.11.024.

Lei, D., and Y. Zheng. 2017. “Hybrid Flow Shop Scheduling with Assembly Operations and Key Objectives: A Novel Neighbourhood Search.” Applied Soft Computing 61: 122–128. doi:10.1016/j.asoc.2017.07.058.

Leu, S. S., and S. T. Hwang. 2002. “GA-based Resource-Constrained Flow-Shop Scheduling Model for Mixed Precast Production.” Automation in Construction 11 (4): 439–452. doi:10.1016/S0926-5805(01)00083-8.

Liu, Z., Zhang, Y., and Li, M. 2014. “Integrated scheduling of readymixed concrete production and delivery.” Automat. Constr., 48, 31–43. doi.org/10.1016/j.autcon.2014.08.004.

Mladenović, N., and P. Hansen. 1997. “Variable Neighborhood Search”. Computers and Operations Research 24 (11): 1097–1100. doi:10.1016/S0305-0548(97)00031-2.

Othman, M. K. F., W. M. N. W. Muhammad, N. A. Hadi, and M. A. Azman. 2017. “The Significance of Coordination for Industrialised Building System (IBS) Precast Concrete in Construction Industry.” Paper Presented at the International Symposium on Civil and Environmental Engineering, Wuhan, China, 20–21st December 2016.

Palubeckis, G. 2017. “A Variable Neighbourhood Search and Simulated Annealing Hybrid for the Profile Minimization Problem.” Computers and Operations Research 87: 83–97. doi:10.1016/j.cor.2017.06.002.

Palmer, D. S. 1965. “Sequencing Jobs through a Multi-Stage Process in the Minimum Total Time: A Quick Method of Obtaining a Near Optimum.” Journal of the Operational Research Society 16 (1): 101–107. doi:10.1057/jors.1965.8.

Prata, B. D. A., A. R. Pitombeira-Neto, and C. J. D. M. Sales. 2015. “An Integer Linear Programming Model for the Multi-Period Production Planning of Precast Concrete Beams.” Journal of Construction Engineering and Management 141(10): 1–4. doi:10.1061/(ASCE)CO.1943-7862.0000991.

Selvi, S., and D. Manimegalai. 2015. “Multi-Objective Variable Neighbourhood Search Algorithm for Scheduling Independent Jobs on Computational Grid.” Egyptian Informatics Journal 16 (2): 199–212. doi:10.1016/j.eij.2015.06.001.

Tharmmaphornphilas, W., and N. Sareinpithak. 2013. “Formula Selection and Scheduling for Precast Concrete Production.” International Journal of Production Research 51 (17): 5195–5209. doi:10.1080/00207543.2013.795250.

Wang, Z., and H. Hu. 2017. “Improved Precast Production–Scheduling Model Considering the Whole Supply Chain.” Journal of Computing in Civil Engineering 31 (4): 1–12. doi:10.1061/(ASCE)CP.1943-5487.0000667.

Wang, Z., H. Hu, and J. Gong. 2018. “Framework for Modelling Operational Uncertainty to Optimize Offsite Production Scheduling of Precast Components.” Automation in Construction 86: 69–80. doi:10.1016/j.autcon.2017.10.026.

Warszawski, A., and E. Ishai. 1982. “Long Range Planning of Prefabrication Industry in a National Economy (Summary).” Building and Environment 17 (1): 47–54. doi:10.1016/0360-1323(82)90008-7.

Yang, Z., Z. Ma, and S. Wu. 2016. “Optimized Flow Shop Scheduling of Multiple Production Lines for Precast Production.” Automation in Construction, 72 (3), 321–329. doi:10.1016/j.autcon.2016.08.021.

Vern K. & Gunal A. (1998). Use of simulation for construction elements manufacturing. In Proceedings of the winter simulation conference, Vol. 2, pp.1273–1277.

Zitzler, E. 1999. “Evolutionary Algorithm for Multi-Objective Optimization: Methods and Applications.” PhD, Swiss Federal Institute of Technology (ETH), Switzerland.

Zandieh, M., and M. A. Adibi. 2010. "Dynamic Job Shop Scheduling using Variable Neighbourhood Search." International Journal of Production Research 48 (8): 2449–2458. doi:10.1080/00207540802662896.

VITAE

Name Mr. Lehuang Zong

Student ID 5910120115

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering (Industrial Engineering)	Jiangxi University of Science and Technology, Ganzhou, China	2014

Scholarship Awards during Enrolment

Thailand's Education Hub for ASEAN Countries (TEH-AC)