RECOVERY-BASED RESCHEDULING AND
OPTIMISATION OF BATCH PRODUCTION
PROCESSES

Yaqing Tan
MSc by research

2012
UNIVERSITY OF BEDFORDSHIRE
RECOVERY-BASED RESCHEDULING AND
OPTIMISATION OF BATCH PRODUCTION PROCESSES

by

Yaqing Tan

A thesis submitted to the University of Bedfordshire in partial fulfilment
of the requirements for the degree of Master of Science by research

September 2012
ABSTRACT

Batch production processes are widely used in the process industries, applied to produce high-value added products with great varieties but in small volumes. The dynamic features of batch production processes contribute to the flexibility of the processes, but also pose big challenges to process scheduling problems. Moreover, disturbances in such a dynamic environment intensify its complexity.

In this work, scheduling and rescheduling models on batch production processes are proposed, considering parallel machines allocation, storage capacity and waiting. The rescheduling model addresses process disturbances, such as machine breakdown and rush orders, in a recovery-based approach, which uses the original schedules as a guide to diminish the deviations between new and original schedules. Genetic Algorithms (GA) and Constraint Programming (CP) are applied to solve the models, but the rescheduling model built by CP can be applied to original schedules created by any techniques. According to case studies and experiments on the proposed scheduling and rescheduling approaches, it is found that CP has a better performance for scheduling and rescheduling problems with complex constraints although it cost longer time than GA. It is also found that rush orders exerted bigger influences on the batch production process than machine breakdowns, especially when the breakdowns do not happen on the ‘bottleneck’ machines.
# TABLE OF CONTENT

ABSTRACT ....................................................................................................................................... iii  
TABLE OF CONTENT ................................................................................................................. iv 
LIST OF FIGURES ........................................................................................................................ vii  
LIST OF PUBLICATIONS ......................................................................................................... viii  
ACKNOWLEDGEMENTS .......................................................................................................... ix  

Chapter 1 Introduction ............................................................................................................. 1  
1.1 General Background ......................................................................................................... 1  
1.2 Problem Statement ......................................................................................................... 3  
1.3 Aims and Objectives ....................................................................................................... 4  
  
1.3.1 Aim ........................................................................................................................... 4  
  1.3.2 Objectives ............................................................................................................... 4  
1.4 Novel Contribution ...................................................................................................... 5  
1.5 The Structure of Thesis ............................................................................................... 5  

Chapter 2 Literature Review ................................................................................................... 7  
2.1 Introduction ...................................................................................................................... 7  
2.2 Batch Production Processes and Their Features ......................................................... 7  
2.3 Optimization Techniques for Scheduling and Rescheduling ...................................... 9  
  
2.3.1 Mixed Integer Linear Programming (MILP) ........................................................... 9  
2.3.2 Meta-heuristics ....................................................................................................... 10  
2.3.3 Constraint Programming (CP) .............................................................................. 11  
2.4 Batch Production Processes Rescheduling ................................................................ 13  
  
2.4.1 Preventive scheduling ............................................................................................. 14  
2.4.2 Reactive scheduling ............................................................................................... 15  
2.5 Limitation of Current Research ................................................................................... 17  
2.6 Conclusion ...................................................................................................................... 19  

Chapter 3 Scheduling Optimisation and Analysis of Batch Production Processes .......... 20  
3.1 Introduction ...................................................................................................................... 20  

}
3.2 Batch Production Processes Scheduling Optimisation Model ......................... 20
  3.2.1 Problem Description.................................................................................. 20
  3.2.2 Batch Processes Scheduling Model.......................................................... 21
3.3 Two Approaches for Batch Processes Scheduling Problems ..................... 23
  3.3.1 Genetic Algorithms Approach ................................................................. 23
  3.3.2 Constraint Programming Approach......................................................... 26
3.4 Case Study and Scalability Tests................................................................ 28
  3.4.1 Case study ................................................................................................ 28
  3.4.2 Optimisation Results................................................................................ 29
3.5 Analysis and Findings .................................................................................. 31
  3.5.1 Analysis on Different Approaches of GA and CP................................. 31
  3.5.2 Analysis on Influencing factors of GA Approach.................................... 33
  3.5.3 Analysis on Influencing Factors of CP Approach.................................... 34
3.6 Conclusion.................................................................................................... 35

Chapter 4 Recovery-based Rescheduling Optimisation of Batch Production
Processes.......................................................................................................... 37
4.1 Introduction .................................................................................................. 37
4.2 Recovery-based Batch Production Process Rescheduling.......................... 37
  4.2.1 Recovery-based Rescheduling Framework.............................................. 37
  4.2.2 Two Types of Disturbances ................................................................. 38
  4.2.3 Rescheduling Model............................................................................... 38
4.3 GA Approach to Solving Batch Process Rescheduling Problems .......... 40
  4.3.1 GA Rescheduling Framework................................................................. 40
  4.3.2 For Original Schedules from GA Approaches....................................... 40
  4.3.3 For Original Schedules from Non-GA Approaches............................... 42
4.4 CP Approach to Solving Batch Process Rescheduling Problems .......... 43
  4.4.1 Overview of CP Rescheduling............................................................... 43
  4.4.2 Disturbance model.................................................................................. 43
  4.4.3 Rescheduling Models............................................................................ 44
4.5 Discussion and Comparison ........................................................................ 45
4.5.1 Discussion on GA Rescheduling ................................................................. 45
4.5.2 Comparison on GA and CP Rescheduling Approaches ......................... 45
4.6 Conclusion ..................................................................................................... 46

Chapter 5 Case Study and Scalability Test for Batch Process Rescheduling

5.1 Introduction .................................................................................................... 47
5.2 Case Study of Rescheduling ........................................................................ 47
  5.2.1 Machine Breakdown Disturbances ......................................................... 47
  5.2.2 Rush Orders Disturbances ..................................................................... 53
  5.2.3 Discussion on Case Study ....................................................................... 55
5.3 Scalability test .............................................................................................. 55
  5.3.1 Machine breakdowns ............................................................................. 55
  5.3.2 Rush orders ............................................................................................ 56
5.4 Analysis .......................................................................................................... 58
5.5 Conclusion ..................................................................................................... 59

Chapter 6 Conclusion and Future Work .............................................................. 60
6.1 Conclusion .................................................................................................... 60
6.2 Future Work ................................................................................................. 62

Reference ............................................................................................................. 63
LIST OF FIGURES

Fig. 3.1 Batch production processes ................................................................. 20
Fig. 3.2 Overview of GA approach scheduling.................................................. 24
Fig. 3.3 Chromosomes crossover ................................................................... 25
Fig. 3.4 Production processes of the case study .............................................. 28
Fig. 3.5 The optimal solution from GA ............................................................... 29
Fig. 3.6 The optimal solution from CP ............................................................... 29
Fig. 3.7 Time for different number of batches ................................................. 30
Fig. 3.8 Time for different number of machines and batches ......................... 31
Fig. 3.9 Time of GA for constraints on capacity and waiting time ................... 33
Fig. 3.10 Branch numbers and constraints numbers for 9 batches on different number of machines ................................................................. 34
Fig. 3.11 Comparison on time and branches among three search strategies ....... 35
Fig. 3.12 Comparison on time and branches between search types of restart and multi point ................................................................. 35
Fig. 4.1 Recovery-based rescheduling framework.......................................... 37
Fig. 5.1 9-hour breakdown on Blender 2 from 4 to 13 .................................... 48
Fig. 5.2 8-hour breakdown of a blender on CP and GA original schedules ....... 49
Fig. 5.3 Comparison of different breakdowns on CP original schedules .......... 50
Fig. 5.4 1-hour breakdowns of storage machine on CP and GA original schedules 50
Fig. 5.5 1-hour breakdowns of packing machine on CP and GA original schedules 51
Fig. 5.6 A rush order on CP and GA original schedules .................................. 54
Fig. 5.7 Deviations and branches of different batches of machine breakdowns rescheduling ................................................................. 56
Fig. 5.8 Deviation comparison of two rush orders on different number of batches .. 57
LIST OF PUBLICATIONS


ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Dr. Wei Huang for providing me the invaluable opportunity to study MSc in the University of Bedfordshire. During my study, Dr. Huang offered careful guidance, insightful advice and continued patience, supporting me towards the completion of the work. I was impressed by the rigorous academic attitude of Dr. Huang, and I benefited a lot from his critical thinking.

I would like to thank the staffs of University of Bedfordshire who were always there to help me in my academic day-to-day activities. I would also like to thank my friends in Institution of Research in Applicable Computing (IRAC) and in Treehouse, at the University of Bedfordshire.

Special thanks to my parents offering financial and emotional support. Without them, I could not imagine myself in the position. I would also like to thank my best friend Miss. Yandong Xia for her emotional support for my time in UK.
Chapter 1  INTRODUCTION

1.1 General Background

Batch production processes are widely used in the process industries and occupy a noteworthy place in the modern industry. They are neither continuous processes (a steady inflow of raw materials results in a steady outflow of products) nor discrete processes (work on individual items) (Huang and Chen, 2007), mainly used for high-value added products manufactured in small volumes but with great varieties. Batch production processes contribute to the process industries significantly. Early in 2000, the chemical processing industry, a small branch of process industries, already accounted for 11% of overall EU manufacturing turnover (Engell et al., 2000).

Batch production processes deploy parallel machines and storage facilities to obtain magnificent flexibility and excellent adaptability to accommodate the unpredictable market changes and customer requirements, but the application of parallel machines and storage facilities also poses big challenges on the batch production process scheduling and rescheduling problems. The application of parallel machines shortens the completion time, but requires delicate planning to make full use of them; storage facilities hold the intermediate materials when the machines for next operation are all in use to alleviate the intensity of the production processes, but their capacities and predetermined maximum waiting time, which cannot be violated for the sake of safety and stable performance, demand careful and correct planning and scheduling to take full advantage of them, contributing to the efficiency of batch production processes. Parallel machines allocation, storage capacity and waiting time are three basic complex issues in the real-world batch production processes, and they distinguish the batch process problems from most studied job-shop scheduling problems (Huang and Chen, 2007, Goldman and Boddy, 1997).
The intrinsic complexity of batch production processes draws attention from various optimisation techniques to seek their contribution on the problems. Those optimisation techniques spend a great deal of effort on studying and developing themselves to fit into the problems. Mixed Integer Linear Programming (MILP), one of the classic mathematical programming, maintains its reputation in handling batch process problems by its rigorousness and simplicity. Meta-heuristics algorithms, including Genetic Algorithm (GA), Tabu Search (TS) and Simulated Annealing (SA), show their high time-efficiency performance in solving batch process problems. Constraint Programming (CP), a relatively new approach, won its place in batch process problems by its excellent capability to address complex constraints. Apart from these three techniques, some other techniques, e.g. expert system, ant colony optimisation and immune system, are also exploring their places in batch process problems and their applications are in development.

Scheduling problems usually aims at optimally allocating limited resources to handle tasks over time, including optimal sequence of task being processed on each machine, the optimal amount of materials being processed on each machine, and etc (Li and Ierapetritou, 2008). The complex and dynamic environment of batch production processes asks for higher performance from optimisation techniques in solving batch production process scheduling problems, which involves many complex constraints and diverse elements: various resources (e.g. parallel machines, raw materials and personnel), external market factors (including products demands, prices and order changes), operational level issues (machine setup times, priorities of orders and machines with deterministic processing time or with un-deterministic processing time) (Duenas and Petrovic, 2008). In the fast-changing environment, established schedules often face with unexpected turbulences and may become invalid.

Many rescheduling approaches are developed to rescue batch production processes from various chaotic situations when disturbances happen. Some generate totally new schedules to replace the original ones, while some adjust the original schedules to fit into current situations (Novas and Henning, 2010, Vieira et al., 2003). The former
approaches neglected the influences of original schedules on rescheduling problems that different original schedules response for disturbances and affect the new schedules differently. A good knowledge of the impacts of original schedules and disturbance in rescheduling problems is beneficial to the robustness of batch production processes and helps disturbed processes recover from chaos easily.

1.2 Problem Statement

Scheduling and rescheduling play significant roles in batch production process problems, but complex real-world issues in batch production processes, such as the application of parallel machines and storage facilities, were hardly consider together. Most work took either the parallel machines or the storage facilities into consideration (Neumann et al., 2005, Blömer and Günther, 1998, Shujun, 2011, Chung et al., 2010, Wang and Cheng, 2000). Some works considered but simplified them so much that some issues did not exert their influences in the processes. The problems without complex constraints drag on the attempt to find capable optimisation techniques for the batch production process problems, since all optimisation techniques were easy to find their own niches. Moreover, many researchers stopped their research at the doorway from scheduling problems to rescheduling problems in batch production processes.

When disturbances happen, generating brand new schedules to replace the original ones is a straightforward approach, but it leaves some tasks in their half ways and causes unnecessary waste (Raheja and Subramaniam, 2002). Therefore, the current state of the processes and original schedules can be used as a guide to create new schedules. However, most rescheduling approaches considering original schedules require original schedules generated by the same optimisation techniques, since the scheduling and rescheduling models are tightly coupled and rescheduling models cannot work separately from scheduling models (Huang and Chung, 2003, Sawik, 2007, Zakaria and Petrovic, 2012). Moreover, most rescheduling work concentrated on the approaches to resolve the problems without deep think and comparison of the
impacts from different original schedules and disturbances (Huang and Chung, 2003, Novas and Henning, 2010). The optimisation techniques used for batch production process problems also concentrated much on developing their performance without comprehensive comparison among themselves.

1.3 Aims and Objectives

1.3.1 Aim
This project aims at developing a recovery-based rescheduling optimisation model of batch production processes. The model will address complex real-world issues: parallel machines allocation, storage capacity and waiting time. It will use original schedules as a guide for rescheduling and it can be applied to original schedules generated from all kinds of optimisation techniques. The research will investigate the impacts of different disturbances, as well as original schedules, on the stability and robustness of batch production processes.

1.3.2 Objectives
The proposed project has the following objectives:

I. To review state-of-the-art literatures on scheduling and rescheduling of batch production processes.

II. To develop a batch production scheduling model considering the complex real-world constraints and using different optimisation techniques, such as CP and GA, to solve it and evaluate their performance.

III. To develop a recovery-based rescheduling optimisation model for batch production processes.

IV. To investigate the impacts of different disturbances and original schedules on the stability and robustness of batch production processes.
1.4 Novel Contribution

This research establishes general models for batch process scheduling and rescheduling problems, considering three complex real-world issues—parallel machines allocation, storage capacity and waiting time. The rescheduling model is loosely coupled with the scheduling model and can use original schedules generated by different optimisation techniques as guides to create new schedules. As a recovery-bases rescheduling approach, the rescheduling model keeps the undisturbed part appear in the new schedules as same as in the original schedules, and uses the total deviations of start times and end times of operations between original schedules and new schedules as the objective.

In this research, two kinds of disturbances (machine breakdowns and rush orders) are used for rescheduling problems and their different impacts on the stability and robustness of batch production processes are investigated. Moreover, the performances of Genetic Algorithms (GA), as the representative of Meta-heuristics, and Constraint Programming (CP) are investigated to discover their advantages and disadvantages in solving batch production processes problems. The impacts of original schedules are also investigated and compared. The findings of this research provide a guide to build stable and robust original schedules and production environment that are able to recover quickly from disturbances without causing many changes.

1.5 The Structure of Thesis

This thesis has six chapters. Chapter 2 reviews related literatures in the batch production processes scheduling and rescheduling problems. Several better-known optimisation techniques and different rescheduling approaches are also discussed in the chapter. Chapter 3 develops a batch production processes scheduling model, which addresses three complex real-world constraints in batch production processes. GA and CP are applied to the model and their performances in batch production process problems are compared in the chapter. Chapter 4 presents a recovery-based
rescheduling model and two rescheduling approaches from GA and CP. The rescheduling model used original schedules as a guide to ensure minimum changes introduced to the original schedules to obtain new schedules. Case studies and scalability tests of the rescheduling approaches are carried out in the Chapter 5, in which machine breakdown disturbances and rush order disturbances are investigated. Chapter 6 concludes the whole thesis and suggests the future work.
Chapter 2  LITERATURE REVIEW

2.1 Introduction
This literature review examined the main concerns in batch production processes and batch production process problems, including their features, optimisation techniques and rescheduling approaches. The study of literature concentrated on the state-of-the-art optimisation techniques for batch production process problems and different approaches to handle unexpected disturbances. At the end of the section, a critical analysis of the literature demonstrating the gaps in batch production process problems, together with a clear emphasis of this work was provided.

2.2 Batch Production Processes and Their Features
Batch production processes are widely used in the process industries, such as chemical industry, food industry and polymer industry. They improve the abilities of the industries to accommodate diverse production requirements and promote the adaptability of the industries to sudden changes in fast-changing global economic condition. Unlike continuous or discrete production, batch production engages with a number of batches and a batch is the smallest quantity to be manufactured (Kallrath, 2002). Raw materials are portioned into batches to produce diverse products with small quantities. This production attains a remarkable flexibility with respect to the diversities and volumes of products, broadly used for low volume, high value-added products.

Some distinctive features of the batch production processes contribute to their flexibility and distinguish batch process problems from the other scheduling problems, such as the job-shop scheduling problems (Huang and Chen, 2007, Goldman and Boddy, 1997):

I. Parallel machines are prepared for some operations and free to be chosen.
II. Some machines can be shared by different batches, such as the storage facilities.

III. Real world issues are complex, such as storage capacity and waiting time.

The storage facilities are regarded as buffers between operations, sometimes, to store intermediate materials that cannot be passed to next operation, since the machines for the next operation are all in use when the current operation on the batch finishes. As characteristic features of batch production processes, storage displays the multi-stage nature of batch production processes and their intrinsic versatility to adapt to the unpredictable fluctuation of market demands. Storage is an indispensable element in the batch production processes, employing diverse storage and waiting strategies (Datta et al., 2001, Kim et al., 1996, Majozi, 2010, Neumann et al., 2005):

I. Unlimited intermediate storage (UIS)
II. Finite intermediate storage (FIS)
III. No intermediate storage (NIS)
IV. Unlimited wait (UW)
V. Finite wait (FW)
VI. Zero wait (ZW)

Usually, FIS policy is applied to solve the batch process scheduling problems with limited storage capacity. The ZW or FW mode is used for where unstable intermediate materials must be processed immediately or within a short time after the previous operation has been completed. Compared with the UIS, NIS, ZW or MIS policies, FIS and FW are more difficult and less to be considered in batch production processes problems.

FIS, FW and parallel machines allocation are three complex real-world constraints in the batch production processes problems. However, most research studied on them respectively or simplified some of them to reduce the complexity of the problems (Neumann et al., 2005, Blömer and Günther, 1998, Shujun, 2011, Chung et al., 2010, Wang and Cheng, 2000). For example, Neumann et al. (2005) mentioned FW in their
paper, FW acted as a variable related with product quantities but not as an independent constraint, and the research from Blömer and Günther (1998) was one of the few that considered FIS and parallel machines allocation together, but it did not mention FW at all.

2.3 Optimisation Techniques for Scheduling and Rescheduling

Efficient and effective schedules are crucial for batch production processes. They improve production performance by reducing waste time and allocating tasks reasonably. The quality of schedules depends on the performance of optimisation techniques. Because of the significance of batch production processes in the process industries, a number of optimisation techniques have been studied to achieve their best performances in the batch production process problems. The section discussed on Mixed Integer Linear Programming (MILP), Meta-heuristics and Constraint Programming (CP), since they are most commonly used optimisation techniques and all have their own niches in the batch production processes problems.

2.3.1 Mixed Integer Linear Programming (MILP)

MILP obtains the priority to solve batch production process problems because of its rigorousness and simplicity (Floudas and Lin, 2005). From 1995 to 1998, Pinto and Grossmann had studied continuous-time MILP models for batch production process problems (Pinto and Grossmann, 1995, Pinto and Grossmann, 1996, Pinto et al., 1998). They stated that applying heuristic rules could reduce the computational time significantly, particularly for small-size problems. However, the heuristic-based approaches performed insufficiently to achieve optimal results for large-size problems (He and Hui, 2008, He and Hui, 2007). Some other heuristic rules were used in an MILP model, which was developed for complex industrial scheduling and rescheduling problems (Roslöf et al., 2001). Those heuristic rules performed a reordering procedure, adjusting the released jobs and inserting them back into schedules. Recently, many hybrid methodologies involving MILP and CP were proposed (Maravelias and Grossmann, 2004b, Maravelias and Grossmann, 2004a,
Roe et al., 2005). These hybrid methods use MILP to decompose original problems and use CP to address sequencing problems. All of them achieved significant save for computational time. However, MILP relies too much on those complements, e.g. heuristic rules and other optimisation techniques, to obtain solutions in a reasonable time, especially when dealing with large-size or complex problems. Apart from those complements, equations and inequalities are other characters of MILP approaches, but they are not intuitive to implement (Wang et al., 2000).

2.3.2 Meta-heuristics

Meta-heuristics are regarded as suitable approaches for large-size problems considering their excellent performances in obtaining near optimal solutions within reasonable computational time (He and Hui, 2008). Genetic Algorithms (GA), Tabu Search (TS) and Simulated Annealing (SA) are three methods in Meta-heuristics with different features, but all of them shared the same disadvantage of Meta-heuristics, that is, they cannot assure the convergence and feasible solutions.

TS and SA are mainly used in the flow-shop or job-shop scheduling problems without stepping into the world of batch production processes problems, (Li and Tang, 2005, Hooda and Dhingra, 2011, Li and McMahon, 2007). He and Hui (2007) presented a GA-based heuristic approach to solve large-size scheduling problems in batch production processes. In this research, GA used less time and displayed its stability to solve problems with different objectives when compared with MILP. A GA approach for match-up rescheduling was proposed to address the new order arrival (Zakaria and Petrovic, 2012). However, the research did not consider any of the three complex real-world constraints (FIS, FW and parallel machines allocation, which were discussed in the previous section).

Constraint handling and objective function optimisation are main challenges for meta-heuristics to tackle with simultaneously (Venkatraman and Yen, 2005). Besides, Meta-heuristics seldom deals with complex constraints in batch production process problems, e.g. FIS and FW (He and Hui, 2008, Shaw et al., 2000, Shaw et al., 1999,
He and Hui, 2007). Three categories of constraints handling methods were proposed for GA, but all of them were based on rank-based chromosomes selection schemes that picked up the top ones to produce next generations, which could not eliminate infeasible solutions thoroughly to guarantee the optimal solutions and handle constraints effectively (Al Jadaan et al., 2009, Venkatraman and Yen, 2005). Even though GA is inherently deficient in constraint handling, some researchers still concentrated on develop effective methods for GA to address complex constraints. A GA approach with a directed search component and special elements was introduced to solve scheduling problems in batch production processes, but the tailored approach only address FW constraints and could not expand its application to dealing with other constraints, such as FIS and parallel machines allocation (Wang et al., 2000). Penalty functions, one of the generic methods for constraint handling, were employed to tackle with storage level in batch processes and achieved acceptable results (Shaw et al., 2000, Al Jadaan et al., 2009).

With the wide application of meta-heuristics, a GA solver and a TS solver were implemented in the Global Optimization Toolbox of MATLAB (The MathWorks, 2010a). The GA solver supports custom GA variant by modifying or defining options and functions. Besides, it also supports linear, nonlinear, mixed integer and bound constraints, but all those constraints have to be written in a certain format. The SA solver implemented a random search method. The annealing process, temperature schedule and acceptance criteria can be defined through provided options. Moreover, both solvers are available if custom data types are used.

2.3.3 Constraint Programming (CP)

Constraint Programming (CP) stands out because of its excellent capability to deal with all kinds of constraints (Mendez et al., 2006, B.P.Das et al., 1999). In CP, the relationships among variables are stated in the form of constraints, which demonstrates its intrinsic adaptability to solve constrained problems (Banaszak et al., 2009). The CP approach consisted of two parts: models and search strategies. During
the search procedures, a constructive search and a constraint propagation are used (IBM, 2012, Zeballos et al., 2011). In the first phrase, the constraint propagation removes unfitted values from domains that contain all possible values for the results to reduce the search space; then, the constructive search experiments on the remaining values in the domains to find feasible solutions or prove that no feasible solutions are for the problems.

A CP approach handling various waiting and storage policies was developed to investigate different search strategies (Zeballos et al., 2011). The approach proved that different search strategies have different time efficiency and impact on the computational time remarkably. The research also compared time-efficiency between CP and MILP, and showed that the computational time used by MILP increased much faster than the CP one when the number of batches increased. Although the research claimed that several problems could not obtain optimal solutions within 900s, CP is still regarded as a suitable technique to solve large-size problems with many constraints (He and Hui, 2008, Zeballos et al., 2011). A CP approach was developed for batch process rescheduling problems on pipeless plants (Huang and Chung, 2003). It considered the FIS, FW and parallel machines allocation constraints, but it was built by adding some constraints on a CP scheduling approach and it could only be applied to original schedules that were from CP approaches. A CP approach combined with a domain representation for a support framework was introduced, aiming at repair-based reactive rescheduling problems, but the work did not consider complex real-world constraints (Novas and Henning, 2010). Nevertheless, the research on CP application in batch production process problems is still not sufficient. Most work was conducted by using MILP and meta-heuristics approaches, and CP was more likely used as a supplement to MILP (B.P.Das et al., 1999, Shaw et al., 2000, Zeballos et al., 2011, He and Hui, 2007).

A CP Optimizer is provided by IBM ILOG CPLEX Optimization Studio. It contains some basic scheduling building blocks to facilitate scheduling problems (IBM, 2012). Time intervals are used as variables to represent activities, operations and tasks to be
completed; sequences are another kind of variables that store the orders of operations on each machine. Many useful constraints and functions, e.g. precedence constraints, no overlap constraints, cumulative expression constraints, are provided to collaborate with the two variables to address various situations in batch production processes problems. Besides, the CP Optimizer offers three search strategies to use in the constraint propagation: depth first, restart and multipoint (IBM, 2012). Depth first, a kind of tree search algorithm, searches each branch thoroughly and will not move to another branches until the current one has been fully explored, which makes it less efficient since it cannot easily recover from poor branches. Restart improves depth first strategy that it restarts after a certain number of fails by using depth first search. Multipoint is different from the above two, which creates a set of solutions and combines them to produce better solutions. Although multi-point search is more diversified than depth-first search or restart search, it cannot guarantee the feasible solutions.

2.4 Batch Production Processes Rescheduling

In the highly dynamic industrial environment like batch production processes, the optimal production performance cannot be guaranteed, even with the perfect optimal schedules, since most schedules were generated on the assumption that everything related with production will remain unchanged in the whole time horizon and they did not consider unexpected disturbances, which often occur and may turn the original schedules into inefficient or even infeasible ones (Mendez et al., 2006). Disturbances generally come from external market factors and operational levels, classified into four types: model inherent disturbances representing by heat transfer coefficients, process inherent disturbances by un-deterministic processing time problems, external disturbances by changes related with orders, and discrete disturbances by machine breakdowns (Li and Ierapetritou, 2008, Mendez et al., 2006, Vieira et al., 2003). In addition, different performance measures are used for rescheduling, concerning efficiency, stability and cost of the schedules, respectively (Vieira et al., 2003). Usually, makespan and tardiness are for efficiency; time
deviations and sequence differences between new and original schedules are for stability; CPU cost, setup cost and transportation are for cost. Two types of approaches have been developed to handle unexpected events during manufacturing process: preventive scheduling and reactive scheduling (Li and Ierapetritou, 2008).

2.4.1 Preventive scheduling

Preventive scheduling usually deals with uncertainties of parameters, preparing itself for disturbances like process inherent disturbances and external disturbances. Preventive scheduling approaches are constituted by stochastic scheduling methods, robust optimisation methods and fuzzy programming methods (Floudas and Lin, 2004).

Stochastic scheduling treats disturbances as stochastic variables and transforms original deterministic scheduling model into stochastic model. Because of the character, stochastic scheduling is seldom used to solve problems involving machine breakdowns. A GA approach with a new random key representation was proposed to deal with machine breakdown rescheduling problems (Lei, 2011). Lei (2011) compared the approach with a Simulated Annealing and a particle swarm optimisation as a further support for his new method. The study enlightened the application of GA in stochastic scheduling for random machine breakdown problems. An approximation algorithm utilized a ‘black box’ distribution to handle stochastic scheduling (Shmoys and Sozio, 2007). The methods enlarged the ways of dealing with stochastic scheduling problems, but it did not consider machine breakdowns.

As for robust optimisation methods, most work relies on the scenario-based framework. A new robust optimisation methodology had been applied to MILP problems to produce robust solutions and alleviate the significant increase of size causing by the increase of scenarios (Lin et al., 2004). Moreover, the novel method improved the ability of addressing uncertain parameters noticeably. A robust counterpart optimisation method was also developed to handle uncertain parameters without causing dramatic increase of the problem size (Gharehgozli et al., 2009).
Three robust counterpart optimisation formulations were used in the work. Those papers proved that robust optimisation methods have been in a continuous development from scenario-based approaches.

Fuzzy techniques focus on searching robust fault-tolerant results, where all constraints are satisfied but leaving flexibilities for the disturbances. Fuzzy sets serve as the prominent element in fuzzy programming methods. A work proposed an overview of fuzzy sets and released a new idea on fuzzy constraint-directed approach (Dubois et al., 2003). The work tackled with flexible constraints and incomplete or imprecise information, which are two important issues for fuzzy programming. A new mixed-integer goal programming (MIGP) model was developed for a parallel-machine scheduling problem (Gharehgozli et al., 2009). The model was under the hypothesis of fuzzy processing time’s knowledge and an effective and accessible methodology was developed to solve the fuzzy model. However, fuzzy techniques rely on the historical information about disturbances.

In fact, all the three preventive scheduling approaches require historical data to guide their work and leave huge margins in the schedules to absorb the disturbances. The process inherent disturbances and external disturbances are usually mild disturbances and easy to eliminate. But the discrete disturbances like machine breakdowns are difficult to address by preventive scheduling approaches.

### 2.4.2 Reactive scheduling

Reactive scheduling aims at taking effective measures when disturbances happen, not relying on historical data as preventive scheduling does. It modifies the existing schedules and generates new schedules to cope with the changes (Floudas and Lin, 2004). Machine breakdowns and order modifications are two main types of disturbances that were addressed in reactive scheduling approaches in the literature (Li and Ierapetritou, 2008, Vin and Ierapetritou, 2000, Mendez et al., 2006, Novas and Henning, 2010). According to the different ideologies and strategies, reactive
scheduling is divided into three types: rescheduling, online scheduling and dynamic scheduling (Vieira et al., 2003).

Rescheduling is regarded as an event-driven action, triggered by disturbances. Vin and Ierapetritou (2000) proposed a two-stage rescheduling scheme in multiproduct batch plants by formulating as an MILP problem. They claimed that the event of machine breakdowns was more critical than the event of rush orders arrival. However, their claim is not always true and it depends on the setting of the problems. Later, a partial modification rescheduling in resource-constrained multistage batch plants was developed in an MILP framework (Mendez and Cerdá, 2004). Their research improved the computational performance in resource-constrained batch plants and obtained the schedule in a reasonable CPU time. Some meta-heuristic methods are introduced to resolve rescheduling problems. A hybrid meta-heuristic method combined artificial immune system algorithm and GA was developed (Abdullah et al., 2007). The artificial immune system algorithm created partial schedules to replace disturbed parts of the original schedule and GA evolved those partial schedules to gain the best fitness. However, the study was confined to job shop rescheduling without further research on batch production processes problems. A reactive rescheduling framework was created to solve repair-based rescheduling problems (Novas and Henning, 2010). The work concentrated on introducing minimum changes to the original schedule and measured total deviation of the start times between the original schedules and new schedules. The work gave another aspect in addressing reactive rescheduling problems in batch production processes, but the rescheduling model was only an extension from the scheduling model with some additional constraints. Actually, most rescheduling models are extensions from the scheduling model, and therefore they only was applied to original schedules using as same optimisation techniques as the rescheduling models used.

Online scheduling offers real-time modifications to the former schedules. A two-stage stochastic integer programming model emphasizing on real-time scheduling was developed (Sand and Engell, 2004). This contribution went against the common
paradigm that establishes off-line schedules with some margins for online adjustments. However, only four types of uncertainties were considered and all of those four uncertainties would not cause momentous impact on original schedules. Online scheduling is most commonly approach that was embedded into information system to adapt to the changing environment in the batch plants. A real-time execution architecture containing three simulation models was built (Saenz de Ugarte et al., 2009). The work demonstrated the way to integrate GA with real-time event simulation models into an information system and the complete process of how a decision system works. Online scheduling is suitable for all disturbances but it requires too much from computers and CPU.

Being different with other two scheduling approaches, dynamic scheduling usually runs without a firm schedule in advance and decisions are made locally in real-time (Ouelhadj and Petrovic, 2009). Three different policies basing on MILP mixed formulations were compared to investigate the performance of dynamic scheduling (Sawik, 2007). In consequence, the non-reschedule policy behaved the worst while it cost least time of the three. It is not surprising, since frequently-used dispatching rules in non-reschedule policy are fast and easy to implement. This research indicated a disadvantage of dynamic scheduling that its performances are remarkably influenced by the reschedule policy and if a wrong policy is chosen, the rescheduling would perform poorly.

2.5 Limitation of Current Research

According to the literature, three complex constraints- FIW, FW and parallel machines allocation-from the intrinsic features of batch production processes were seldom considered together. Though MILP, Meta-heuristics and CP displayed their applications in the batch production processes, Meta-heuristics and CP were mainly used as supplements for MILP. Moreover, most of work concentrated on improving or developing novel approaches to handle the batch production process scheduling
problems without comparisons on the advantages and disadvantages of different optimisation techniques.

In addition, research on batch production process rescheduling problems was left far behind. Most batch process rescheduling approaches created new schedules to replace the original ones when disturbances happen. Some approach used original schedules as a guide to achieve minimum changes in new schedules, but original schedules and rescheduling approaches were required to use the same techniques, since most rescheduling models were tightly coupled with scheduling models by adding some constraints on the scheduling models.

Moreover, most researchers failed to consider the influence of original schedules and disturbances on the stability and robustness of batch production processes. They did not think about the waste caused by deviations between original schedules and new schedules, and only focused on the capability to address different disturbances. Furthermore, machine breakdown disturbances were regarded as one kind of disturbances without sub-categories to different kinds and positions of machines.

In this work, Finite Intermediate Storage (FIS), Finite Waiting (FW) and parallel machines allocation were considered and addressed in both scheduling and rescheduling approaches. GA, as the representative of meta-heuristics, and CP were used to solve the scheduling and rescheduling problems of batch production processes, and their performance in solving the problems were compared. A recovery-based rescheduling model was proposed to use original schedules as a guide to generate new schedules with minimum changes to the original schedules. The rescheduling model is loosely coupled with the scheduling model and could be applied to any original schedules, regardless of the optimisation techniques they used. Machine breakdown disturbances and rush order disturbances were investigated to obtain the understanding of their impacts on the batch production processes. Moreover, the machine breakdown disturbances were studied on different types of machines.
2.6 Conclusion

The literature review introduced features of batch production processes and three complex real-world constraints of batch production problems. MILP, Meta-heuristics and CP were discussed as major optimisation techniques to solve batch production processes problems. Two types of approaches to handle disturbances in the production process were investigated. The research on literature revealed that few works addressed the three complex constraints together and most work on rescheduling did not consider the impacts of original schedules and disturbances.
Chapter 3  SCHEDULING OPTIMISATION AND ANALYSIS OF BATCH PRODUCTION PROCESSES

3.1 Introduction

This chapter presented a batch production process scheduling model with three complex constraints, Finite Intermediate Storage (FIS), Finite Waiting (FW) and parallel machines allocation. Genetic Algorithms (GA) and Constraint Programming (CP) were applied to solve the model. Their performances were investigated to find out their capabilities in dealing with batch production process scheduling problems. The investigation went deep to gain the knowledge of the techniques for their better use in the field.

3.2 Batch Production Processes Scheduling Optimisation Model

3.2.1 Problem Description

In batch production processes, raw materials are processed in batches and parallel machines are used for the same operation, such as blending, mixing, packing and etc. Different final products are produced in different routines, which are made up with different operations, Fig. 3.1. Storage facilities are used specifically to hold the intermediate production when they are waiting for the next operation, but the waiting time is predetermined and limited by the capacities of storage facilities for safety concern.

![Batch production processes diagram]

Fig. 3.1 Batch production processes
3.2.2 Batch Processes Scheduling Model

a. Basic notations

I: total number of batches;
J: total number of operations;
i: one of the batches in the production;
j: one of the operations in the process;
\( j_m \): one of the parallel machines for operation j;
t: time for one moment;
\( M_j \): total number of machines for operation j;
\( T_{ij} \): processing time of operation j for batch i;
\( M_j \): total number of parallel machines for operation j;
\( ST_{ij} \): start time of operation j for batch i;
\( ET_{ij} \): end time of operation j for batch i;
\( W_{j,m} \): binary parameter for the parallel machine \( j_m \). When machine \( j_m \) is in use, \( W_{j,m} = 1 \); when machine \( m \) is not in use, \( W_{j,m} = 0 \);
\( MT \): maximum waiting time in the storage facilities;
\( InQ_{i,j,m} \): quantity of input materials of operation j for batch i on machine m;
\( OutQ_{i,j,m} \): quantity of output materials of operation j for batch i on machine m;
\( SQ_{i,j,m} \): quantity of materials at the start time of operation j for batch i on machine m.
For the first operation, \( SQ_{i,j,m} = 0 \);
\( EQ_{i,j,m} \): quantity of materials at the end time of operation j for batch i on machine m;
\( C_{j,m} \): capacity of machine m for operation j;
\( MakeSpan \): end time of the last operation.

b. Objective function and precedence constraints

Objective function

The objective of the scheduling model aims at obtaining the minimum makespan, like many other scheduling problems in batch production processes (Neumann et al., 2005, Li and Tang, 2005, Kim et al., 2000, Blömer and Günther, 1998).

\[
\text{Min } MakeSpan
\]

(3-1)

Precedence constraints

The production processes must follow the recipes that some operations should be processed before another operation.
\[ ET_{i,j} \leq ST_{i,j+n} \]  
(3-2)

For each batch, the end time of last operations is the start time of current operations, and the start time plus the processing time is the end time of current operation.

\[ ET_{i,j-1} = ST_{i,j} \]  
(3-3)

\[ ST_{i,j} + T_{i,j} = ET_{i,j} \]  
(3-4)

c. Specialised constraints

Finite Intermediate Storage (FIS), FW (Finite Wait) and parallel machines allocation are also included in the model. They are representatives of the disparities between job shop scheduling problems and batch process scheduling problems.

Machines allocation constraints

Materials are assigned to available machines for the same operation to speed up the production processes. Since one machine can only work on one batch at a time, machines allocation constraints limit that any batches require for the same machine do not overlap and the number of machines in use cannot exceed the total number of machines for the operation.

\[ ET_{i,j} \leq ST_{i+n,j} \text{ or } ET_{i-n,j} \leq ST_{i,j} \]  
(3-5)

\[ \sum_{m=1}^{M} W_{j,m} \leq M_j \]  
(3-6)

FW constraints

Waiting policies tackle with operations without fixed processing time, particularly storage. FW constraints demonstrate that the intermediate products can only stay in the machine for a finite time.

\[ ET_{i,j} - ST_{i,j} = T_{i,j} \leq MT_j \]  
(3-7)

FIS constraints
FIS constraints handle with the capacity of machines. They restrict that the amount of materials in a machine cannot exceed its capacity to ensure the steady performance. The quantities of materials on each machine are calculated at the start time and the end time of each operation. The quantity of the machine at the start time is the quantity of the last operation at the end time plus the quantity of input materials; the quantity of the machine at the end time is the quantity of the current operation at the start time minus the quantity of output materials.

\[
SQ_{i,j,m} = EQ_{i,j-1,m} + InQ_{i,j,m}
\]  \hspace{1cm} (3-8)

\[
EQ_{i,j,m} = SQ_{i,j,m} - OutQ_{i,j,m}
\]  \hspace{1cm} (3-9)

\[
SQ_{i,j,m} \leq C_{j,m}
\]  \hspace{1cm} (3-10)

\[
EQ_{i,j,m} \leq C_{j,m}
\]  \hspace{1cm} (3-11)

FW and FIS influence each other: FW restrict the amount of products that are produced simultaneously by setting finite waiting time. FW and FIS will alleviate the burdens on the other machines and provide stable material flows in the processes if they are handled properly.

### 3.3 Two Approaches for Batch Processes Scheduling Problems

#### 3.3.1 Genetic Algorithms Approach

a. Genetic Algorithms Model Overview

The Genetic Algorithms (GA) model was built on the MATLAB R2012b with some helps from Genetic Algorithm Solver in Global Optimization Toolbox (The MathWorks, 2010b). The model was constituted by four main parts: setting, objective function, chromosomes and calculation, Fig. 3.2. The setting part served as the interface of the model, preparing all data and parameters, such as processing time, number of batches, population of chromosomes and etc. It connected with the objective function part and the chromosomes creation in the chromosomes part. The objective function part received chromosomes from the chromosome part and passed
them to the calculation part; then received the results from the calculation part and passed them to the setting part to make final decisions. The chromosomes part and the calculation part were two most important components in the model, displaying the distinctions of GA in solving batch process scheduling problems.

b. Chromosomes Part

The chromosomes part engages with four activities for chromosomes: creation, selection, crossover and mutation. In this work, chromosomes are generated by the permutation encoding method, in which each integer represents a specific batch (Obitko, 1998). The creation assigns the integers randomly in chromosomes with the same presence frequency to generate the initial chromosomes. The presence of an integer in a chromosome means that an operation of the corresponding batch will be carried out. The initial chromosomes go through the cycles of selection, crossover and mutation to evolve and obtain the expecting results.

The tournament selection is applied to choose a group of chromosomes in the current generation to serve as parents to produce next generation through crossover. It selects chromosomes in Tournament size (usually 4) at random and picks up the best one of them to be a parent. Crossover works on a pair of parents together. It chooses one or more integers randomly in one parent chromosome, but at least two integers should be left to make the crossover work. The selected integers are kept in the same positions in both chromosomes and the un-chosen integers are replaced by the integers from the other chromosome following the same sequence as in the original chromosome. The
crossover retains the same frequencies of each integer in one chromosome, ensuring the validity of chromosomes. An illustration is presented below (Fig. 3.3):

\[\text{parent1}[3\ 2\ 1\ 1\ 3\ 2\ 3\ 1\ 2] \rightarrow \text{child1}[1\ 2\ 3\ 3\ 1\ 2\ 1\ 3\ 2]\]
\[\text{parent2}[2\ 1\ 3\ 3\ 1\ 2\ 1\ 2\ 3] \rightarrow \text{child2}[2\ 3\ 1\ 1\ 3\ 2\ 3\ 2\ 1]\]

Fig. 3.3 Chromosomes crossover

Mutation is a separate procedure from selection and crossover, and it is designed in a rather simple way. First, two randomly selected points divided a chosen chromosome into three parts. Then the middle part is reversed to obtain the next generation. Crossover rate and mutation rate determine the number of chromosomes involving crossover and mutation, respectively.

c. Calculation Part

The calculation part is made up with two phrases: first phrase decodes the chromosomes and transforms them to another format that contains the information of different operation for each batch; second phrase uses the information to calculate the makespans of each chromosome and deal with three complex constraints- parallel machines allocation, Finite Wait (FW) and Finite intermediate Storage (FIS).

All machines for the same operation are numbered first to prepare for parallel machines allocation constraints. Machines with a small number have the priority when serving intermediate products. For example, intermediate products are assigned to Machine 1 first, and if Machine 1 is in use, they will be re-assigned to Machine 2 instead, and so on. Penalty functions are used to implement FW and FIS constraints, since the nonlinear constraints options in MATLAB only apply to the variables that are genes in the chromosomes (The MathWorks, 2010b). Exterior penalty functions are used here, which seek for optimal solutions starting from infeasible solutions to feasible solutions (Coello Coello, 2002). Since all constraints in the problems are “death constraints” that cannot be violated at all, the penalty value is set in a greater order of magnitude. All constraints are examined when each operation ends. Once a
constraint is violated, a penalty value will be added to the object. The more constraints are violated, the larger the results will be. After iterations, the number of violations will decrease and the feasible solutions will be found. But in some extreme cases, feasible solutions could not be found. The pseudo code of how the penalty functions work is presented as follow:

\[
\begin{align*}
\text{Read the sequence of all operations.} \\
\text{For each operation} \\
\quad \text{Allocate to the available machines.} \\
\quad \text{Calculate its start time and end time.} \\
\quad \text{If end time-start time} > \text{predetermined waiting time, then} \\
\quad \quad \text{Add penalty value to the objective function.} \\
\quad \text{End if} \\
\quad \text{Calculate the quantities in the storage machine.} \\
\quad \text{If the quantities} > \text{capacity, then} \\
\quad \quad \text{Add penalty value to the objective function.} \\
\quad \text{End if} \\
\text{End loop}
\end{align*}
\]

3.3.2 Constraint Programming Approach

a. Constraint Programming Model Overview

Constraint Programming (CP) model is built on the IBM ILOG CPLEX Optimization Studio (IBM, 2012). Comparing with GA model, CP model is much simpler and it only contains a model file, a data file and a setting file. Data files store data for the model, and are able to read and write data from MS Excel spreadsheet files. The CP model is mainly written in the model files and influenced by different parameters in the setting files.

b. Model Files

Interval variables and sequence variables are two distinctive elements used in CPLEX Optimization Studio for scheduling problems (IBM, 2012). They provide remarkable helps to build CP models. Interval variables represent the time period consumed by an operation, storing the end time, start time and processing time for each operation. In this work, interval variables engaged with the implementation of FIS and FW. Sequence variables record the order of operations on each machine, and were used to
handle parallel machines allocation constraints in the work. The ‘endAtStart’ constraint provided by CPLEX was used implement precedence constraints, indicating that the next operation starts when the current operation ends.

FW constraints were implemented easily by using the minimum size and maximum size properties of interval variables. The maximum size property was set as the maximum waiting time. FIS constraints implementation required the cooperation of interval variables and the cumulative. The functions calculated the amount of materials in each machine in the whole time horizon by adding the predetermined amount of materials at the start time of the related interval variable and deducing at the end time. The ‘alwaysIn’ constraint, an embedded constraint of CPLEX, was used to restrain the amount of materials in the machines to meet the requirement of capacities.

In CPLEX, ‘tuple’, a special data structure, is provided to cluster closely related data together (IBM, 2012). The ‘tuple’ data structure was used to store parallel machines information, exhibiting available machines for each operation, and the sequence variables displayed the orders of interval variables on the different machines. According to those data, ‘noOverlap’ constraint, another embedded constraint of CPLEX, was used to ensure that one machine only worked on one batch and one batch was only processed by one machine simultaneously.

c. Setting Files
Search strategy options in the setting files impact the performance and the results of CP models. Three search strategies-depth first, restart and multipoint-are provided by CPLEX Studio (IBM, 2012). Depth first, a kind of tree search, searches each branch thoroughly; restart, an improvement from depth first strategy, restarts after a certain number of fails by using depth first; multipoint is different from the above two, which creates a set of solutions and combines them to produce better solutions.
3.4 Case Study and Scalability Tests

3.4.1 Case study

A case study from a published paper is used as an example to investigate the performance of scheduling models from GA and CP (Huang and Chen, 2007). According to Fig. 3.4, 60 tonne Raw material A and 60 tonne Raw material B are blended with equal proportion in the two identical blenders with a fixed capacity of 5 tonne each. After blending for 2 hours, all products are passed on to the storage facility with 15 tonne capacity, waiting to be packed. The intermediate products could only wait in the storage machine for 6 hours at most. Packing machines pack 2500 packs per hour and perform on different sizes of packs, but the packing size only changed on hourly basis. The case study requires final products as 20 tonne 1-kg-pack-size products, 20 tonne 2-kg-pack-size products and 20 tonne 3-kg-pack-size products, respectively.

The case study was built on an hourly basis, aiming at the minimum of makespan, which is the total time for completing all the operations. Raw materials were divided into 12 batches and operated in a cyclical mode to achieve the product demand. The models were performed on a PC with AMD Athlon (tm) 2.91 GHz Processor and 1.75 GB of RAM. A group of parameters was set for GA to obtain the best results after several experiments: crossover rate was set as 0.2; mutation rate as 0.01; population size as 300; termination condition was that the algorithm stops if the average change in the objective function value over 500 generations is less than 1e-6. For CP, the auto search type that automatically selects the most suitable search type from the three search strategies was used. The restart fail limit was 100; restart
growth factor was 1.15; number of search points in multipoint search was 30; time limit, fail limit and solution limit were all 2147483647. Two criteria were put forwards to evaluate the approaches to compare the performance of GA approach and CP approach (He and Hui, 2007):

I. The optimal solutions from each algorithm.
II. The computational time for the optimal solutions.

3.4.2 Optimisation Results

a. Optimal solutions

Fig. 3.5 The optimal solution from GA

Fig. 3.6 The optimal solution from CP
Both approaches obtained 19 hours as the smallest makespan, which was as the same as the result from Huang and Chen (2007) by using a different approach. But the schedules attained by GA approach and CP approach were different. According to Fig. 3.5 and Fig. 3.6, GA approach assigned blending tasks to the blenders evenly, but CP approach tended to use Blender 1 so much that Blender 2 was idle for some time. Apart from different tasks assignment strategy, the schedules obtained by GA approach varied from different runs while those from CP approach remained the same. Moreover, CP approach spent triple time than GA approach. The small case above only provided a superficial look at differences between two models and a deep discussion was made in the Section 3.5.

b. Scalability Tests

![Fig. 3.7 Time for different number of batches](image)

Scalability tests were carried out to explore the performance of two models dealing with different number of batches and machines. Fig. 3.7 displayed total running time consumed by GA and CP when they handled the case study but with different number of batches. Since the time for the same problem varied on GA approach, the averages of ten figures for the same problem was used as running time for GA approach. Running time trend line of GA approach was obviously made up with two phrases: for smaller number of batches (3 batches, 6 batches and 9 batches), the time stayed the same and less time was used comparing to CP approach; the time jumped rapidly from 12 batches to 15 batches. When it came to 18 batches, GA approach could not find solutions as the same as CP approach did within 500 generations. Unlike GA approach, the time used by CP approach did not change much.
Fig. 3.8 presents the running time of GA and CP approaches with different batches on the different numbers of machines, and it revealed the advantages and disadvantages of GA and CP dealing with those problems. In most circumstances, GA approach cost less time than CP approach. The consuming time had few changes when GA model worked on 6 batches, 9 batches and 12 batches. In the experiment with 9 batches to 18 batches, running time of GA approach decreased when the number of machines increased and then stayed fluctuated. Although GA beat CP in time consumption, GA approach could not attain the feasible solution for the problem that 4 machines worked on 18 batches. As for CP, running time roared when machine numbers increased from 4 batches to 8 batches and stayed stable after that, except for the production of 9 batches that displayed a sudden drop when the machine number was 16.

3.5 Analysis and Findings

3.5.1 Analysis on Different Approaches of GA and CP

The Matlab platform provides GA Solver in Global Optimization Toolbox to facilitate the application of GA. However, the GA Solver could not be applied to this research directly. The default data types for GA Solver are double and binary string,
but cell arrays were used in this research (The MathWorks). Therefore, a creation function, a crossover function and a mutation function were created to work on the custom data type. Besides, some build-in functions that produce next generation were also changed to accommodate the custom functions. For example, the default crossover function combined pairs of parents to one child (The MathWorks, 2010b), but the custom crossover function mixed the parents to generate two children (Fig. 3.3).

GA approach relies on crossover and mutation to evolve chromosomes from generation to generation to access optimal results. However, the unpredictable changes of chromosomes did not always produce better offspring for the next generation. Therefore, GA could not obtain the same optimal solutions as CP did for some problems. On the other hand, constraints information, such as parallel machine allocation constraints, FW constraints and FIS constraints, is difficult to embed into chromosomes except for using tailored chromosomes. Penalty functions are often used as a generic method to solve constrained problems. The sensitivity of penalty value influenced the performance of GA model to find the optimal solutions (Coello Coello, 2002). Moreover, penalty functions could not implement constraints to the problems in nature, since they obtained the optimal solutions by reducing the times of constraints violations but not forbidding the occurrence of the violations.

CP Optimizer in CPLEX Studio provides some basic building blocks to deal with scheduling problems, including interval variables, cumulative functions, ‘endAtStart’ constraint and etc. Those blocks facilitate the creation of CP scheduling models. For example, interval variables are designed to be optional because some operations may not be processed. They consider all possibilities in the scheduling problems without impacting the final decisions. With the help of those blocks, scheduling problems were realised much more easily in the CP approach, comparing with GA approach.

CP approach is a tree search algorithm, searching on the tree that contains all the solutions for the problems until the optimal ones are found or proved that no feasible
solutions exist. But this holistic search owns a disadvantage that it takes long time to confirm the results. Its fixed search strategy accounted for the fact that the solutions for the same problem were the same.

3.5.2 Analysis on Influencing factors of GA Approach

According to Fig. 3.7 and Fig. 3.8, the batch numbers exerted greater impact on GA approach than machine numbers, and the huge contrasts on performances of GA model handling smaller and larger number of batches revealed that it was the elasticity of constraints rather than batches or machines that influenced the performance of GA approach. When the constraints are loose, the elasticity is high and GA approach performs well; when the constraints are tight, the elasticity is low and GA approach performs poorly. To verify the relationship between running time and constraint elasticity, FW constraints and FIS constraints were investigated, Fig. 3.9. Running time decreased when the constraints turned to be loose, for example, when the storage capacity becomes larger and the waiting time becomes longer. When the constraints are so loose that constraints had no impact on the problems, the running time fluctuated. Fig. 3.9 confirmed that GA approach was affected by the elasticity of constraints in the problem. In other words, the tighter the constraints were, the more difficult for GA to work out the optimal solutions. Therefore, GA approach could not perform well in problems with complex, tight constraints, such as batch process scheduling problems in the work.

![Fig. 3.9 Time of GA for constraints on capacity and waiting time](image)

33
3.5.3 Analysis on Influencing Factors of CP Approach

a. Investigation on Number of Branches

The sudden drop of running time when 9 batches were processed on 16 machines in Fig. 3.8 was explained by the tree search nature of CP approach, since the problem owned fewer branches. Although the problem of 9 batches on 16 machines contained more variables and constraints than the problems on 8 machines and 12 machines, it only contained 3197 branches while the problem on 8 machines had 441,576 branches and the problem on 12 machines owned 427,284 branches. The left picture of Fig. 3.10 displayed the number of branches for 9 batches on the different machine numbers, which looked the same with the picture of 9 batches in Fig. 3.8. The right picture of Fig. 3.10 showed that the constraint numbers increased linearly with machine numbers, which proved that branches, rather than constraints, determined the running time of CP model.

![Graph showing branches and constraints for 9 batches on different number of machines](image)

b. Investigation on Search Strategies

Since the branches are related to the tree search nature of CP approach and the tree search procedures are determined by the search strategies, three search strategies—depth first, restart and multipoint—were investigated to find out their relationship with branch numbers and their performance in solving batch production processes problems. They were tested on the problems that use 4 machines to produce 12 batches, 15 batches, 18 batches, 21 batches and 24 batches. Fig. 3.11 displayed the computational time of the three search strategies on different batches.
According to Fig. 3.11, depth first cost most time and the time increased exponentially when the number of batch increased. Comparing with depth first, restart and multi point used such little time that could not see their trends on the same diagram with depth first search. The computational time of restart and multi point search were displayed in Fig. 3.12, without the computational time of depth first search.

Fig. 3.11 Comparison on time and branches among three search strategies

Fig. 3.12 Comparison on time and branches between search types of restart and multi point

Fig. 3.12 showed a sudden jump in multi point search from 18 batches to 21 batches. In constrast, the branch number of restart search increased linearly and cost the least time among the three, which indicated that restart search strategy has excellent robustness with different elasticities of constraints and could be the proper search strategy in the batch process scheduling problems.

3.6 Conclusion

Batch production process scheduling problems have gained more attention from industry and academia. A batch production processes model was proposed dealing
with complex real-world constraints, such as waiting time, storage capacity and parallel machines allocation. GA and CP approaches were applied to solve the batch processes model and investigated. Discussions and experiments were carried out to analyse their disparities. It was found that CP has a better performance for batch production process scheduling problems with complex real-world constraints although it costs longer time than GA and the restart search type in CP approach performed better than another two search strategies.
Chapter 4 RECOVERY-BASED RESCHEDULING
OPTIMISATION OF BATCH PRODUCTION
PROCESSES

4.1 Introduction
Disturbances are inevitable in the production processes and they will invalidate the original schedules sometimes. A recovery-based rescheduling model of batch production processes was built in the chapter. The rescheduling model was built on GA and CP approaches, respectively, and dealt with two types of disturbances. In this approach, the undisturbed part of original schedules was kept in the new schedules and the total deviation on the start time and end time of all operations replaced the makespan to serve as the objective.

4.2 Recovery-based Batch Production Process Rescheduling

4.2.1 Recovery-based Rescheduling Framework

![Diagram of Recovery-based rescheduling framework]

Fig. 4.1 Recovery-based rescheduling framework

The recovery-based rescheduling framework is made up with three elements: original schedules from scheduling models, disturbances and rescheduling models. A machine breakdown rescheduling model and a rush order rescheduling model are built to deal with machine breakdown disturbances and rush order disturbances, respectively. When the original schedules are disturbed by the disturbances and become invalid, the original schedules and the disturbances information are delivered to the...
rescheduling models to generate new schedules. The rescheduling model analyses the disturbances information and uses a specific rescheduling model to solve the problems accordingly. As a recovery-based rescheduling approach, original schedules are used as a guide to create schedules and the un-disturbed part of original schedules remains unchanged in the new schedules. Disturbances only happen once and two types of disturbances are not considered to happen simultaneously.

4.2.2 Two Types of Disturbances

Machine breakdowns and rush orders are two types of disturbances considered in this research. They only happen once in the production process and do not happen simultaneously. Machine breakdown disturbances are the representation of discrete disturbances and they happen at any time on any machines; rush order disturbances represent external disturbances, and their arrival time and due time are considered. The repeat disturbances were not the focus of this research.

4.2.3 Rescheduling Model

a. Additional Notations

$r$: one of the rush orders;
$MaxT$: time horizon of schedule;
$NewST_{i,j}$: new start time of operation $j$ for batch $i$;
$NewET_{i,j}$: new start time of operation $j$ for batch $i$;
$BkST_{j,m}$: start time of breakdown on machine $m$;
$BkET_{j,m}$: end time of breakdown on machine $m$;
$ArrT_r$: arrival time of a rush order;
$DueT_r$: due time of a rush order;
$DisValue_{i,j}$: deviation on the start time and end time of operation $j$ for batch $i$.

b. Objective Functions and Time-Bound Constraints

Objective Functions

Deviations on the start time and end time of operations between original schedules and new schedules are the main concern for the recovery-based rescheduling. As a recovery-based approach, the rescheduling model aimed at minimising the total
deviation and used the total deviations between original schedules and new schedules as the objective to measure the changes in the new schedules.

\[
DisValue(O_{i,j}) = \left[ \left| NewST(O_{i,j}) - ST(O_{i,j}) \right| + \left| NewET(O_{i,j}) - ET(O_{i,j}) \right| \right] 
\]  
(4-1)

\[
\text{Min } \sum_{i=1}^{I} \sum_{j=1}^{J} DisValue(O_{i,j}) 
\]  
(4-2)

Time-Bound Constraints

Although the rescheduling model aimed at minimum total deviation, the makespan is still an important element in the scheduling and rescheduling problems that the end time of any operation cannot exceed the time horizon.

\[
NewET_{i,j} \leq MaxT 
\]  
(4-3)

c. Disturbances

In the rescheduling model, the work done before the disturbances happen is kept unchanged in the new schedules.

If \( ET_{i,j} \leq BkST_{j,m} \) or \( ET_{i,j} \leq ArrT \),

\[
NewET_{i,j} = ET_{i,j} \text{ and } NewST_{i,j} = ST_{i,j} 
\]  
(4-4)

Machine breakdown disturbances

A machine cannot serve any tasks when a breakdown happens, and any operations requiring the machine should finish before the start time of the breakdown or begin after the end time of the breakdown.

\[
NewET_{i,j} \leq BkST_{j,m} \text{ or } NewST_{i,j} \geq BkET_{j,m} 
\]  
(4-5)

Rush order disturbances

The arrival time and the due time limit the options for rush orders. They start after receiving the orders and must be delivered before the due time.
\[ ST_{r,j} \geq ArrT_r \]  \hspace{1cm} (4-6) \\
\[ ET_{r,j} \leq DueT_r \]  \hspace{1cm} (4-7)

4.3 **GA Approach to Solving Batch Process Rescheduling Problems**

4.3.1 **GA Rescheduling Framework**

GA rescheduling framework was made up with a scheduling model, a disturbances part and a rescheduling model. Disturbances information was stored in the disturbances part, including their types, their emerging time, etc. Those information and original schedules were passed on to the rescheduling model to create new schedules, when the disturbances happen. The rescheduling model came from the scheduling model, with some modifications. They had the same elements and all the elements were linked in the same ways. Since GA fundamentally works on the chromosomes, the rescheduling model had different strategies to tackle with original schedules from GA and other optimisation techniques: for those from GA approaches, chromosomes are the focus; for those from other approaches, penalty functions are the solution to the problems.

4.3.2 **For Original Schedules from GA Approaches**

a. **Disturbances Part**

The disturbances part was constituted by a machine breakdowns part and a rush orders part, serving as a disturbance input to the rescheduling model. The start time of the breakdowns and the arrival time of rush orders were analysed to find out the first disturbed operation in the chromosome of the original schedules, but analysis of machine breakdowns just considered the operations on the broken machine not like the analysis of rush orders that considered operations on all the machines.

b. **Rescheduling Model**
The rescheduling model had four parts, similar with the scheduling model: rescheduling setting part, rescheduling objective function part, rescheduling calculation part and rescheduling chromosome part. Rescheduling setting part and rescheduling objective function part were almost as same as the setting part and objective function part in the scheduling model, except for some minor modifications. Rescheduling setting part built a bridge between disturbances part and rescheduling objective function part to obtain more data required for rescheduling, such as disturbances information and the chromosome of the original schedules; rescheduling objective function part linked to the machine breakdowns rescheduling calculation part or rush orders rescheduling calculation part, respectively, basing on different information from the disturbances part.

Special chromosome creation and mutation approaches were built in the rescheduling chromosome part to achieve the recovery-based rescheduling approach, since the chromosome for the original schedule were divided into an un-disturbed part with a disturbed part and the two parts could not be mixed together in the creation and mutation procedures. The undisturbed part stayed the same but the disturbed part engaged with the continuous selection, crossover and mutation to produce next generation and obtain the final results. Rescheduling creation part accounted the frequencies of each integer in the un-disturbed part to acquire the information for rest operations and transferred the information to the language of chromosomes to fill in the rest part of chromosome to make them complete. Machine breakdown disturbances used the same integers as before, but rush order disturbances needed new integers in the chromosomes to represent new batches of rush orders. Crossover part and selection part for scheduling model were used here without any changes, since they would not mess up integers in the two parts. Rescheduling mutation part left the undisturbed part of chromosomes aside and only worked on the rest part. It reversed the middle part of the disturbed part to obtain mutated chromosomes. Apart from chromosome creation, machine breakdown disturbances and rush order disturbances were handled in the same ways in the rescheduling chromosome part.
The rescheduling calculation part modified the second phrase of scheduling calculation part without any changes in the first phrase. In the rescheduling part, the approaches for two disturbances were different. Additional penalty functions were applied to tackle with due dates of the rush orders in the rescheduling part. Penalty value would be added to the objective function, if the end time of the last operations of any batches of rush orders exceeded the due dates, working in the same way as the penalty functions for FIS and FW in scheduling model. Machine breakdown disturbances removed all the operations in the period of machine breakdowns to leave a completed idle time for broken machines. The pseudo code for the procedure was presented as follow:

Decode the chromosome to obtain the time schedule.
For each operation
    If the operation was assigned to the broken machine
        If start time of broken machine < end time of the operation < end time of broken machine, then
            Start time of the operation = end time of the broken machine
        End if
        If start time of broken machine < start time of the operation < end time of broken machine, then
            Start time of the operation = end time of the broken machine
        End if
    End if
End loop

4.3.3 For Original Schedules from Non-GA Approaches
Since schedules for non-GA approaches did not own the chromosomes, another approach was developed to handle the problems. In this approach, disturbances part did not need to analyse disturbances information like for GA approach original schedules. It passed all information directly to the rescheduling model. Rescheduling setting part sent disturbances information with the original schedules to the rescheduling calculation part through rescheduling objective function part. Rescheduling objective function part remained the same with the one for GA approach original schedules. The rescheduling chromosome part remained as same as in scheduling model when dealing with machine breakdown disturbances but new
integers were added into chromosomes for rush orders disturbances. The applications of penalty functions was the main changes in rescheduling model for non-GA approach original schedules, which was centralized in the rescheduling calculation part. Penalty functions were applied to implement the constraints that the undisturbed part of the original schedules was kept the same in the new schedules. The deviations from the rest part were added up as the objective value.

4.4 CP Approach to Solving Batch Process Rescheduling Problems

4.4.1 Overview of CP Rescheduling

The CP approach was constituted by four CP models: scheduling model, disturbance model, machine breakdown rescheduling model and rush order rescheduling model. The disturbance model and the two rescheduling models were used in the scheduling model as its submodels. The terms of master model and submodel are used in CPLEX Studio to describe the components of decomposed models that are big, long and complex (IBM, 2012). Master model is the basis of the whole model and usually contains flow control script in C++ language to connect with its submodels.

C++ language is used in CPLEX Studio for flow control and introduces a scripting block in the master model to link its submodels. A set of C++ classes was applied to connect the scheduling model with the other CP models by defining submodels, passing data, etc. The example of setting up an existing disturbance model as a submodel of the scheduling model was displayed below:

```c++
var mainSource = new IloOplModelSource("disturbance.mod");
var mainDef = new IloOplModelDefinition(mainSource);
var mainCp = new IloCP();
var mainData = new IloOplDataSource("disturbance.dat");
var mainOpl = new IloOplModel(mainDef,mainCp);
mainOpl.addDataSource(mainData);
mainOpl.generate();
...
mainOpl.end().
```
Class *IloOpModelSource()* extracted all input source from the disturbance model for the submodel, and class *IloOpModelDefinition()* received the source to establish a model definition that prepared information to instantiate the disturbance model as a submodel of the scheduling model. *IloCP()* was used to indicated that CP Optimizer was used as the search engine for the disturbances model. *IloOpDataSource()* read data from data files of the disturbance model and added the data to the disturbances submodel by using the method *addDataSource()*.*IloOpModel()* combined information of model definitions and search engines to establish the disturbances submodel in the scheduling model. Method *generate()* opened the submodel and *end()* closed it after use. Machine breakdowns rescheduling submodel and rush orders rescheduling submodel were all established in the similar ways.

### 4.4.2 Disturbance model

The disturbance model received and stored disturbance information. ‘Machine’ and ‘order’ were used as keywords to establish the specific rescheduling submodel, e.g. when the disturbance information contained the keyword ‘machine’, the machine breakdown rescheduling model would be used to solve the problems. The start time and end time for broken machines were store in a ‘tuple’ data; the arrival time and due time of rush orders were store by using interval variables properties. But rush order disturbances required more information, such as processing time for each operation, quantities of each batch and etc.

### 4.4.3 Rescheduling Models

The rescheduling models were like scheduling model but with some modifications. Comparing with machine breakdown rescheduling model, the rescheduling model for rush orders was simple. The scheduling model and the disturbance model passed all information to rescheduling model, and a CP specialised constraint ‘imply’ is provided by CPLEX to implement the constraint that the undisturbed part was as same as the original schedules.
Machine breakdown rescheduling model was complex. The utility of the machine at the start time of breakdowns were set as 100, which stated the 100% utility of the machine, and the end time of breakdowns was set as 0 for the machine was not in use, and the the durations of the same utility on one machine were calculated to obtain the efficiency of the machine over time. The different utilities and their durations worked together to realise machine breakdowns. The ‘imply’ constraint was also used here to keep what had been done unchanged, and the interval variables were forbidden to start, end or overlap the time intervals with the utility of 0.

4.5 **Discussion and Comparison**

4.5.1 Discussion on GA Rescheduling

Different GA rescheduling approaches were developed to cope with original schedules obtained from different techniques. For those from GA approach, chromosomes were the focus of the rescheduling and specialised functions for creation and mutation were developed to handle the problems. The disturbances part had to analyse the disturbances information to transform it into the language of chromosomes before sending it to the rescheduling part.

The rescheduling approach for the original schedules from non-GA approach was simpler, because no changes in the chromosomes part were needed. A number of penalty functions were used to restrict the undisturbed part of original schedules appeared in the new schedules, but the effect of penalty functions and the feasible solutions of the problems could not be guaranteed. Worse yet, the application of penalty functions exerted more constraints on GA, which increased the elasticity of constraints in the problem and made it more difficult to attain the optimal solutions.

4.5.2 Comparison on GA and CP Approach Rescheduling

GA rescheduling applied different strategies to deal with original schedules generated by different techniques, but CP rescheduling approach tackled with all original schedules in the same way, no matter which technique they were generated. Both of
GA and CP approaches added new elements to work with the scheduling part. But CP received remarkable supports from CPLEX Studio. Featured data types, functions and constraints were provided and reduced the work on developing CP rescheduling models. But MATLAB did not offer any specific supports for the problems and the penalty functions were still used as the methods to implement new constraints in the rescheduling problems. Furthermore, CP approach integrated four separate models as a complete rescheduling model with one access in the scheduling model by using the C++ classes provided by CPLEX Studio. The approach is simple and compact, but GA rescheduling approach involved with several different parts and their connections were complex.

4.6 Conclusion

A batch production process recovery-based rescheduling model was built in this chapter to deal with disturbances of machine breakdown and rush order. GA and CP approaches were used to solve the model and they addressed the problems in different ways. Different strategies were applied in GA approach to handle original schedules from GA approach or non-GA approach. But CP approach treated all original schedules in the same way. Furthermore, GA approach could not handle the new constraints for rescheduling well because of the drawbacks of penalty functions. Therefore, CP approach was found to be the appropriate technique for rescheduling problems of batch production processes.
Chapter 5  CASE STUDY AND SCALABILITY TEST
FOR BATCH PROCESSES RESCHEDULING
PROBLEMS

5.1 Introduction

This chapter explored the impacts of different disturbances and original schedules on
the batch production process rescheduling problems. Machine breakdown
disturbances were divided as three types of disturbances and each one focuses on a
particular kind of machines. Rush order disturbances took the due dates into
consideration. The experiments in the case study investigated the impacts of
disturbances with different emerging time and the experiments in the scalability tests
aimed at the impacts of disturbances with the increase of batches.

5.2 Case Study of Rescheduling

The same example used for scheduling model was used here as a case study to
explore the impacts of different disturbances and original schedules on the batch
production process rescheduling problems. The case study investigated different
types of machine breakdown disturbances and rush order disturbances with due dates.
A GA schedule (Fig. 3.5) and a CP schedule (Fig. 3.6) were used as original
schedules in this study to find out their attributes of robustness and their interaction
with different kinds of disturbances, since tasks assignment strategies, especially on
parallel machines, were different in the GA approach and the CP approach.

5.2.1 Machine Breakdown Disturbances

This research aimed at investigating the impacts from different types of machine
breakdowns in the batch production processes and offered the knowledge of the
contributions from different types of machines on the reliability and robustness of the
batch production processes. The research explored the biggest tolerances for each type of disturbances in the batch production processes and experimented on different start times of the disturbances. According to the different types of machines, three kinds of breakdowns were investigated:

I. Breakdowns on the one of the parallel machines.
II. Breakdowns on the machines without deterministic processing time.
III. Breakdowns on the machines with deterministic processing time

a. Breakdowns on Parallel Machines

Fig. 5.1 9-hour breakdown on Blender 2 from 4 to 13

The breakdowns on the two blenders represented the breakdowns on the parallel machines. The idle time of Blender 2 in the Fig. 3.6 revealed that the original schedule would not change much, if the breakdown intervals of Blender 1 did not exceed the total idle time of Blender 2. According to the Fig. 3.6, 8 hours and 9 hours were tested as breakdown intervals on one blender on the CP original schedule to find out the maximum length of breakdowns that did not lengthen the makespan. The 9-hour breakdown was found to be the maximum but only happened once when the breakdown was from 4 to 13 (Fig. 5.1) and 10-hour breakdown was too long to attain the makespan of 19 hours.
The 8-hour breakdown was implemented on CP and GA original schedules to investigate its influences. According to the chart in Fig. 5.2, experiments on both original schedules obtained the makespan of 19 hours in some cases but from different routines. The rescheduling based on CP original schedule (ReCP) attained 19 hours at the start of the study while the rescheduling based on GA original schedule (ReGA) achieved that at the end of the study. In fact, ReGA did not attain the makespan of 19 hours with a breakdown of 8 hours on one blender, since the two blenders finished their work at 12 in the original schedule and breakdown interval from 8 to 16 only impacted the two blender for 4 hours. Some experiments were carried out to search the maximum breakdown time that the GA approach original schedule could bear and proved to be 4 hours. Comparing with the CP approach original schedule, the GA original schedule displayed less tolerance for the breakdowns. 22 hours were the longest makespan in the research, obtained on the GA original schedule. The trends of makespans from both approaches were different: the makespans of ReCP underwent a climb from 19 to 21 and followed a fall back to 19, but those of ReGA dropped from 22 to 19 progressively.

The deviations of each experiment were showed in the chart in Fig. 5.2 and they were in the similar trend like the makespans. The deviations at the start of the ReGA were much higher than those of ReCP, but the deviations of ReGA went beneath the trend line of ReCP after the Experiment No. 9 with the breakdown interval from 8 to 16, indicating that the robustness of GA original schedule was worse than CP original.
schedule at the early stage. The breakdowns after 11 had no impacts on the makespan of ReGA, since one blender was sufficient to handle the rest tasks after 11 to ensure the makespan of 19 hours, but this for ReCP had waited to the 12 hour.

This experiment split the 8-hour breakdown to two 4-hour breakdowns with the same start on the two blenders to explore their impacts on the batch production processes and compared with the 8-hour breakdown. The table in Fig. 5.3 showed that the makespan of 19 hours was only achieved once when the breakdowns were 4 hours on two blenders. The chart in Fig. 5.3 demonstrated that the deviations of 4-hour breakdowns on two blenders were higher than 8-hour breakdown on one blender, and the fluctuations were huge for breakdown of 4 hours, though, with the same trend.

b. Breakdowns on Machines with Un-deterministic Processing Time

![Fig. 5.3 Comparison of different breakdowns on CP original schedules](image)

![Fig. 5.4 1-hour breakdowns of storage machine on CP and GA original schedules](image)
The storage machine represented the machines with un-deterministic processing time. 1-hour breakdown was tested first because no idle time was found in the storage machine in both original schedules. Based on the table of Fig. 5.4, the 1-hour breakdown was already too harsh for both original schedules to handle - they only attained the makespan of 19 hours rarely and the experiments with 2-hour breakdown intervals proved it. The deviations of ReCP fluctuated remarkably with an increasing trend at the first several experiments and reached the climax in the Experiment No. 9 with the breakdown interval of 10-11, while those of ReGA had a bigger fluctuation from Experiment No.1 to Experiment No. 2 and arrived at their peak of a deviation of 102 at Experiment No. 3. Both deviation trend lines dropped after their climax, but the deviations of ReCP dropped linearly while two more fluctuations after Experiment No.11 appeared in the trend line of ReGA. Even with the fluctuations, their deviations were still less than those of ReCP.

c. Breakdowns on Machines with Deterministic Processing Time

<table>
<thead>
<tr>
<th>No.</th>
<th>Interval</th>
<th>CP Makespan</th>
<th>CP Disparity</th>
<th>GA Makespan</th>
<th>GA Disparity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3-4</td>
<td>20</td>
<td>48</td>
<td>20</td>
<td>48</td>
</tr>
<tr>
<td>2</td>
<td>4-5</td>
<td>21</td>
<td>64</td>
<td>21</td>
<td>78</td>
</tr>
<tr>
<td>3</td>
<td>5-6</td>
<td>20</td>
<td>40</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>6-7</td>
<td>21</td>
<td>78</td>
<td>20</td>
<td>42</td>
</tr>
<tr>
<td>5</td>
<td>7-8</td>
<td>20</td>
<td>42</td>
<td>20</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>8-9</td>
<td>21</td>
<td>69</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>9-10</td>
<td>20</td>
<td>39</td>
<td>20</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>10-11</td>
<td>21</td>
<td>63</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>9</td>
<td>11-12</td>
<td>20</td>
<td>36</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>12-13</td>
<td>20</td>
<td>30</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>13-14</td>
<td>20</td>
<td>24</td>
<td>21</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>14-15</td>
<td>20</td>
<td>18</td>
<td>20</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>15-16</td>
<td>20</td>
<td>12</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>16-17</td>
<td>20</td>
<td>9</td>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>17-18</td>
<td>20</td>
<td>6</td>
<td>20</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>18-19</td>
<td>20</td>
<td>3</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 5.5 1-hour breakdowns of packing machine on CP and GA original schedules

The packing machines were the delegates of the machines with deterministic processing time. According to the table in the Fig. 5.5, rescheduling for packing machines breakdowns could not attain the makespan of 19 hours like other two types of machines albeit the breakdowns were only 1 hour. The deviations trend lines of packing machine breakdown looked similar with the deviations of storage machine, except that they were in a decreasing trend from the beginning. The deviations of
ReCP had fluctuations at the first couples of experiments and the fluctuations were shrinking; the deviations of ReGA had one similar fluctuation with ReCP at the start of the experiments and they went beneath the trend line of ReGA for the most of time.

d. Discussion on Machine Breakdowns on Different Types of Machines
Based on the study of breakdowns on different types of machines, breakdowns on parallel machines were found to be the least detrimental disturbances and parallel machines contributed most to the robustness of batch production processes. In the study, the two blenders together could resist a maximum breakdown of 9 hours and maximum breakdowns of 4 hours on each simultaneously, without lengthening the makespan of original schedules. But the deviations and fluctuations of 4-hour breakdowns on two blenders were higher and bigger than 8-hour breakdown on one blender, because the parallel machines lost their advantages that supported each other as a buffer. Moreover, the two 4-hour breakdowns happening at the same time could be regarded as the breakdowns on the machines with deterministic processing time and the machines with deterministic processing time was the worst one to handle disturbances in this research that they could bear any breakdowns. The machines with un-deterministic processing time were in the second place of robustness contributors with the breakdowns of 1 hour in the study.

Although parallel machines contributed most to the robustness of batch production processes, the consequences after they broken down were difficult to predict. In this study, the trends of deviations of storage machine breakdowns and packing machines breakdowns were nearly a line at the late phrase in the experiments, particularly those from ReCP. Predictable consequences ease the intensity to response the disturbances quickly and leave more time for mangers seek out solutions for the current situations. The deviations of ReCP on packing machines were typical examples of predictable consequences. They decreased in decreasing sequences, which provided enough information to estimate the consequences of different disturbances. The concise patterns of the disturbances on the storage machine and packing machines was mainly attributed to that they did not have alternatives.
In general, early breakdowns have greater impacts than late breakdowns, regardless of the types of breakdowns. The deviations for the early experiments were relatively large, especially those from ReGA. GA original schedules assigned the tasks evenly on the parallel machines without leaving any idle time (Fig. 3.5), which inevitably provided fewer available choices for rescheduling and obtained the huge deviations. The even task assignment enabled the parallel machines to complete their tasks earlier so that late breakdowns would not impact them at all. However, some later breakdowns in ReCP obtained fairly big deviations. Few choices of task assignment to keep the same with original schedules were blamed for the exceptions on blenders; the strategy of packing machines to address the 2-hour tasks first were responsible for the exceptions on the storage machines, since the strategy slowed material flows from storage machines to packing machines, leading to full used storage machines and stuck intermediate materials in the storage. Once a disturbance happened in this period, the state of full use was seriously broken and difficult to recover, causing the huge deviations. Comparing with ReGA, 1-hour tasks were operated in the period, which accelerated the flows from storage machine and mitigated the burden of storage machines. In addition, the deviations of ReGA displayed few ascending trends and they always decreased much more smoothly than those of ReCP.

The fluctuation that was made up with one up and one down appearing in all trend lines had little relationship with the disturbances and they were from the case study intrinsically, since the processing time on the blenders was 2 hours. If the breakdown happened at an odd number time points, one task in process was be disturbed undoubtedly, which explained the bigger deviations; if the breakdown happened at an even number time points, tasks were not in process, which obtained the smaller deviations.

5.2.2 Rush Orders Disturbances

This study investigated the impacts of rush orders with due dates on the batch production processes. The length between receiving time and due time was regarded
as the processing time for the rush orders, which, together with due time, determined the start time to work on the rush orders to handle out the rush order before the due time. In this study, an order of 10 tonne 1-kg-pack-size products was received as the rush order.

The processes only accepted rush orders with a reasonable processing time that was sufficient to complete the task. The processing time was analysed before confirming the acceptance of the rush orders to evaluate the impacts of the rush order. The shorter the processing time is, the more difficult to finish the tasks in time and more effort would be paid in the rescheduling. 5 hours were the shortest processing time for one batch and obtained by adding the shortest processing time for each operation together, but the feasible solutions for rescheduling could not be attained with a processing time of 5 hours. 7 hours were proved to be the least processing time for the rush orders after several experiments and rush orders were processed right after receiving with the processing time of 7 hours.

The processing time of 7 hours with different receiving time were investigated. From the table of Fig. 5.6, both approaches achieved the deviations of 0 at the end of the study, but ReCP attained the makespan of 23 hours, one hour less than the shortest makespan of ReGA. However, neither of them achieved the makespan of 19 hours like in the original schedules. The changes in the makespans of the two approaches were also different. ReCP fluctuated at the start and stayed at 23 hours till the last, but

<table>
<thead>
<tr>
<th>No.</th>
<th>Interval</th>
<th>CP Mktspan</th>
<th>Disparity</th>
<th>GA Mktspan</th>
<th>Disparity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-7</td>
<td>23</td>
<td>174</td>
<td>23</td>
<td>106</td>
</tr>
<tr>
<td>2</td>
<td>1-8</td>
<td>24</td>
<td>231</td>
<td>23</td>
<td>198</td>
</tr>
<tr>
<td>3</td>
<td>2-9</td>
<td>23</td>
<td>162</td>
<td>23</td>
<td>228</td>
</tr>
<tr>
<td>4</td>
<td>3-10</td>
<td>23</td>
<td>207</td>
<td>23</td>
<td>264</td>
</tr>
<tr>
<td>5</td>
<td>4-11</td>
<td>23</td>
<td>144</td>
<td>23</td>
<td>163</td>
</tr>
<tr>
<td>6</td>
<td>5-12</td>
<td>24</td>
<td>210</td>
<td>23</td>
<td>159</td>
</tr>
<tr>
<td>7</td>
<td>6-13</td>
<td>23</td>
<td>162</td>
<td>23</td>
<td>144</td>
</tr>
<tr>
<td>8</td>
<td>7-14</td>
<td>23</td>
<td>150</td>
<td>23</td>
<td>135</td>
</tr>
<tr>
<td>9</td>
<td>8-15</td>
<td>23</td>
<td>147</td>
<td>23</td>
<td>158</td>
</tr>
<tr>
<td>10</td>
<td>9-16</td>
<td>23</td>
<td>150</td>
<td>23</td>
<td>196</td>
</tr>
<tr>
<td>11</td>
<td>10-17</td>
<td>23</td>
<td>123</td>
<td>23</td>
<td>81</td>
</tr>
<tr>
<td>12</td>
<td>11-18</td>
<td>23</td>
<td>99</td>
<td>23</td>
<td>60</td>
</tr>
<tr>
<td>13</td>
<td>12-19</td>
<td>23</td>
<td>75</td>
<td>23</td>
<td>35</td>
</tr>
<tr>
<td>14</td>
<td>13-20</td>
<td>23</td>
<td>53</td>
<td>24</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>14-21</td>
<td>23</td>
<td>39</td>
<td>24</td>
<td>21</td>
</tr>
<tr>
<td>16</td>
<td>15-22</td>
<td>23</td>
<td>12</td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td>17</td>
<td>16-23</td>
<td>23</td>
<td>0</td>
<td>24</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 5.6 A rush order on CP and GA original schedules
ReGA attained the makespan of 23 hours at the early experiments and gained to 24 hours at the late experiments. The trend lines of deviations from both approaches looked similar with those in the machine breakdowns on the storage machines, but the gaps between the deviations from ReGA and ReCP were not as big as in the previous experiments on machine breakdowns.

5.2.3 Discussion on Case Study

The experiments in the case study displayed the similarities and dissimilarities of the machine breakdown disturbances and rush orders disturbances. The deviations of rush order disturbances looked similar with the combination of machine breakdown disturbances on storage machines and packing machines, but the deviation gaps between ReGA and ReCP were not as big as in the machine breakdown disturbances, indicating that the differences in original schedules had less influence on the rush orders rescheduling problems than the machine breakdowns rescheduling problems, because the rush order disturbances exerted holistic impact on the processes. In addition, the rescheduling problems dealing with rush orders could not obtain the original makespan of 19 hours and had bigger deviations, which demonstrated that the rush order disturbances exerted greater impacts on the processes.

The original schedules also exerted different impacts on the rescheduling problems. Usually, the deviations of ReGA were higher than those of ReCP at the start of each study and went down all the way through and they had some fluctuations at the end of the study but most of them did not exceed those of ReCP. Based on those facts, balanced original schedules, such as GA original schedules, had less tolerance for the early disturbances, but the tolerances increased with the time went by.

5.3 Scalability test

5.3.1 Machine breakdowns

The rescheduling scalability tests on machine breakdown disturbances investigated their impacts on the production processes of different batches. Those experiments
would not add machines, since all the machines turned into parallel machines in that
cased. All of the experiments were carried out on the CP original schedule, and the
breakdown intervals with the least deviations in the case study were selected to use
here to obtain obvious changes for different batch numbers: 4-13 for the blender, 5-6
for the storage machine and 9-10 for the packing machines.

According to Fig. 5.7, the makespans of all types of machine breakdowns increased
with batch numbers, but the makespans of packing machine were always longer than
those of the storage machine and the blenders, which corresponded with the
discussion for machine breakdowns that the packing machines were lack of choices to
handle the disturbances and the original schedules simultaneously. The makespans
and deviations of the storage machine and the blenders were almost in the same trend,
particularly the trend line of deviations that climbed at 9 batches and dropped back to
the figure at 6 batches.

5.3.2 Rush orders

The rescheduling scalability tests on rush order disturbances aimed at the impacts of
rush orders on the different number of batches. A rush order of 10 tonne 1-kg-pack-
size products with receiving time of 4 and due time of 11 (R4-11) was used here. The
6+2 in the batch row stated that there were 6 batches originally, with 2 batches of rush orders.

<table>
<thead>
<tr>
<th>No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batches</td>
<td>6+2</td>
<td>9+2</td>
<td>12+2</td>
<td>15+2</td>
<td>18+2</td>
<td>21+2</td>
<td>24+2</td>
</tr>
<tr>
<td>Disparity</td>
<td>72</td>
<td>120</td>
<td>144</td>
<td>219</td>
<td>288</td>
<td>285</td>
<td>394</td>
</tr>
<tr>
<td>Makespan</td>
<td>15</td>
<td>19</td>
<td>23</td>
<td>27</td>
<td>31</td>
<td>35</td>
<td>39</td>
</tr>
</tbody>
</table>

Fig. 5.8 Deviation comparison of two rush orders on different number of batches

The makespans of rush orders rescheduling increased in line, like those of machine breakdowns rescheduling, but the deviations were totally different from machine breakdowns rescheduling. The deviations increased with the batches, except for the terrace from No.5 to No.6. A rush order of 10 tonne 1-kg-pack-size products with receiving time of 8 and due time of 15 (R8-15) was used to investigate the causes for the terrace. The order had a similar deviation with the R4-11 in the research to diminish the differences of the two rush orders.

<table>
<thead>
<tr>
<th>No.</th>
<th>Batches</th>
<th>4-11</th>
<th>8-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6+2</td>
<td>72</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>9+2</td>
<td>120</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>12+2</td>
<td>144</td>
<td>147</td>
</tr>
<tr>
<td>4</td>
<td>15+2</td>
<td>219</td>
<td>198</td>
</tr>
<tr>
<td>5</td>
<td>18+2</td>
<td>288</td>
<td>246</td>
</tr>
<tr>
<td>6</td>
<td>21+2</td>
<td>285</td>
<td>291</td>
</tr>
<tr>
<td>7</td>
<td>24+2</td>
<td>394</td>
<td>355</td>
</tr>
</tbody>
</table>

Fig. 5.8 Deviation comparison of two rush orders on different number of batches
The R8-15 obtained the same makespans with R4-11 in each experiment, but the deviations were different. According to the Fig. 5.8, the R8-15 had smaller deviations than R4-11, which proved that the later the rush order came, the less the disturbances influenced. Moreover, the terrace did not appear in the trend line of the R8-15, which explained that the terrace was only a coincident without nothing inherent causes.

5.4 Analysis

The machine breakdowns and rush orders influenced the batch production system in different ways. The disturbances of machine breakdowns was influenced by the types of the machines and the disturbances of rush orders was determined by receiving time and due time. Rush orders exerted huge impacts on the makespan and longer makespan would be used for rush orders if the original schedules were sensible ones without much idle time. On the contrary, machine breakdowns had limited impacts on the makespans and sometime they were able to keep the same makespan with the original schedules.

Even though the deviations of rush orders were larger than machine breakdowns, the differences of rush order deviations between ReGA and ReCP were less, which indicated that the original schedules did not influence the performance of rush orders rescheduling as much as for machine breakdowns rescheduling. Comparing with Fig. 5.6 and Fig. 5.7, the rush orders could be regarded as the fatal disturbances to the batch production processes, and their impacts increased with the increase of batch numbers.

The impacts of different types of machines on the process were different. The packing machine influenced the processes in the similar way with rush orders that its deviations increased with the increase of batches. The storage machine and the blenders did not have lasting influence on the processes. When the number of batches arrived at a certain level, the breakdowns from the storage machine and blenders had no impacts on the processes. The different trends of deviation of different types of machines mainly relate to their different tolerances to the disturbances. Since the
packing machines were always in full use and could not stand any disturbances, they were regarded as the ‘bottle neck’ of the processes. The breakdowns on the ‘bottle neck’ were reflected on the results directly and exerted lethal impacts on the processes. When there are huge amount of batches, the makespan is determined by the ‘bottle neck’ rather than any other machines, which explained that the impact of breakdowns in the storage machine and blenders did not increase with the increase of the batches but the impacts of breakdowns in packing machines did.

5.5 Conclusion

The rescheduling model was investigated in the chapter. A case study and scalability tests were carried out to discover the impacts of different disturbances and original schedules. In the case study, machine breakdown disturbances and rush order disturbances were investigated, respectively. Scalability tests studied the impacts of disturbances in the production of different number of batches. According to the results, rush orders were the most influential impacts of the processes and machine breakdowns could be neglected when many batches are to be processes, expect for the ones on the ‘bottle neck’ machines. In addition, the balanced original schedules were found to be good at handling the late disturbances.
Chapter 6  CONCLUSION AND FUTURE WORK

6.1 Conclusion

Batch production processes have won a place in the process industries because of their inherent flexibility and adaption to the rapidly changing consumers’ needs and market. They are mainly used for high-value added products with great diversities but in a small volume. Parallel machines, storage capacity and waiting time are most significant features of batch production processes, and all of them are needed to be considered in the batch production processes problems.

Although many optimisation techniques were applied to solve batch production process problems, the researchers seldom considered the three features of batch production processes together and few comparative studies were conducted on the performances of different techniques. Furthermore, most of them only studied on batch production process scheduling problems but the rescheduling problems were left without many attentions. Most rescheduling model were built on the scheduling model with some additional constraints and they could not extent themselves to work on original schedules that came from other optimisation techniques. The influences of disturbances and original schedules on batch production processes were neglected.

In this work, batch production processes scheduling and rescheduling models were generated. Both of models considered three complex constraints-parallel machines allocation, storage capacity and waiting time policies. Genetic Algorithms (GA) and Constraint Programming (CP) optimisation techniques were applied to solve the models and their performances in batch process scheduling problems were investigated. The investigation looked into the nature of GA and CP, and discovered the causes of the differences. The rescheduling model deals with two kinds of disturbances and could be applied to all original schedules from various optimisation techniques. As a recovery-based rescheduling approach, the rescheduling model uses
the original schedules as a guide to create new schedules, and the total deviations of start times and end times of operations between original schedules and new schedules were used as the objective to ensure that the minimum changes were introduced to the original schedules. A case study and scalability tests were conducted to investigate the impacts of different disturbances and original schedules on the batch production processes. Those experiments compared the impacts of the two disturbances: machine breakdown disturbances and rush order disturbances. Machine breakdown disturbances were analysed on three different types of machines.

The work on the scheduling model displays that CP had a better performance for batch production scheduling problems with complex real-world constraints although it cost longer time. Additionally, CP approach performed remarkably in the work on the rescheduling model. Its notable compatibility to work on original schedules that came from all kinds of optimisation techniques, demonstrating its significances in the batch production processes problems.

According to the case studies and scalability tests on the rescheduling problems, rush orders exerted bigger influences on the batch production processes than machine breakdowns, especially when the machine breakdowns did not happen on the ‘bottleneck’ machines. Therefore, receiving rush orders requires comprehensive considerations of the current situation and asks for higher prices; mass production is needed to alleviate the disturbances when the machines break down. In addition, the ‘bottle neck’ machines deserve special attentions, since their breakdowns will cause huge scale of disturbances in the processes. Parallel machines contribute the stability and robustness of batch production remarkably and are encouraged to use in the processes. Apart from the disturbances, the original schedules also affect the rescheduling problems. The balanced original schedules, e.g. GA original schedules, have less tolerance of the early disturbances but they handled the late disturbances well. The balanced original schedules were still preferred, since they could finish the tasks earlier and leave less space for the disturbances.
6.2 Future Work

The case study used in the research only involved four machines in three categories. Although three complex constraints were considered in the study, it was still too simple from the real-world circumstance for batch production processes because of the lack of machines. More machines will be added to the next work and longer processes will be investigated.

This work dealt with machine breakdown disturbances and rush order disturbances separately. The work on two kinds of disturbances happening simultaneously will be carried out in the future, which is a more challenging task and requires high efficiency from the optimisation techniques. Moreover, more types of disturbances will be studied, such as the process inherent disturbances. The efficiency was the main concern in this research but the profit aspect was not considered. Therefore, some elements like products price and fine for the delay of products will be considered in the next work to make it more realistic.
Reference


