APPLICATION OF MIXED INTEGER LINEAR PROGRAMMING IN OPTIMIZATION OF SEQUENCE-DEPENDENT FLUID PACKAGING PRODUCTION SCHEDULING

BY

NAPHAT ORMSAPSIN

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING (LOGISTICS AND SUPPLY CHAIN SYSTEMS ENGINEERING)

SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2022

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

INDEPENDENT STUDY

BY

NAPHAT ORMSAPSIN

ENTITLED

APPLICATION OF MIXED INTEGER LINEAR PROGRAMMING IN OPTIMIZATION OF SEQUENCE-DEPENDENT FLUID PACKAGING PRODUCTION SCHEDULING

was approved as partial fulfillment of the requirements for
the degree of Master of Engineering (Logistics and Supply Chain Systems Engineering)

on June 19, 2023

Member and Advisor ______________________
(Associate Professor Jirachai Buddhakulsomsiri, Ph.D.)

Member ______________________
(Pham Duc Tai, Ph.D.)

Director ______________________
(Professor Pruettha Nanakorn, D.Eng.)

Independent Study Title	APPLICATION OF MIXED INTEGER LINEAR PROGRAMMING IN OPTIMIZATION OF SEQUENCE-DEPENDENT FLUID PACKAGING PRODUCTION SCHEDULING
Author	Naphat Ormsapsin
Degree	Master of Engineering (Logistics and Supply Chain Systems Engineering)
Faculty/University	Sirindhorn International Institute of Technology/ Thammasat University
Advisor	Associate Professor Jirachai Buddhakulsomsiri, Ph.D.
Academic Years	2022

ABSTRACT

In the manufacturing process of packaging fluid, there are processes in between the assigned producing jobs called 'changeover' in which the machine or the production line must have an adjustment to prepare for the upcoming job. Therefore, scheduling requires optimization to minimize the total production time from the changeover of the sequence scheduled. The process is normally executed by a planner with effort spent to find the optimal solution with the lowest makespan as fast as possible. This research considers the job scheduling problem with sequence-dependent changeover times on a single machine to minimize the makespan of the jobs assigned. Different changeover types with varied required times will be assumed. To provide an effort improvement tool, a mixed integer linear programming (MILP) is developed to compute the optimal solution and compared with the human-executed heuristic proposed to find the solution with a better computational time in exchange for a non-optimized solution. Up to 71 jobs, the developed MILP with a tolerance 0.5% takes approximately 60 minutes to find the solution. In comparison, the heuristic cannot reduce the computational time until scheduling 71 jobs and can reduce the computational time by 87.5%. However, compared to the MILP with a tolerance of 5%, it can reduce the computational time and

the MILP with a tolerance of 0.5% and can improve the results compared to the heuristic at any number of jobs.

Keywords: Machine scheduling problem, Sequence-dependent setup time, Makespan, Heuristic, Mixed integer linear programming

ACKNOWLEDGEMENTS

My horizon in optimization problems has been widening since I was given knowledge by my professor, Associate Professor Dr. Jirachai Buddhakulsomsiri. The benefit and application of this topic were not clear from my perspective until I work on this research. Also, I would like to express my sincere gratitude to my professor for his immeasurable patience and feedback on this work that has been pushing me to continue my journey on this research. His knowledge and expertise have been essential to deliver and improve my work.

I would also like to express my appreciation to the faculty members at SIIT who have supported me since the day I joined this program. Without their help, my study life could not be this convenient and constructive in learning.

Lastly, I am also grateful to my peers in the same batch who consistently supported me academically and mentally. This has been my significant impact as energy and inspiration for learning.

Naphat Ormsapsin

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
MILP	Mixed Integer Linear Programming
SKU	Stock Keeping Unit
WO	Washout
SHM	Shampoo
CON	Conditioner
LC	Line Clearance
MILP5	MILP With 5% Solution Tolerance
MILP0.5	MILP With 0.5% Solution Tolerance

CHAPTER 1
INTRODUCTION

In manufacturing and service industries, competitive sequencing and scheduling are necessary for marketplace survival as a form of decision-making to eliminate loss internally as a producer with efficient production and externally to meet customer satisfaction (Pinedo, 2008). Optimizing production schedules can eliminate costs and elevate productivity for manufacturers. For instance, an optimal production schedule can lead to the shortest makespan (time required to produce all jobs assigned). As a result, manufacturers will pay the least fixed cost compared to the number of jobs assigned, and they will be able to produce more jobs with the same available time. Another example is that an optimal production schedule can provide the lowest cost of production. Assuming a fluid filling process where some bulk products can be filled consecutively without washing required and some that cannot, the optimal sequence can be the one with the lowest amount of washing process to minimize cost.

1.1 Problem statement

In a system with one fluid filling machine in a package with different sizes, which requires changeover when size is changed between production batches, an optimal solution of a sequence must provide the least makespan. Changeover time for each changeover is presumably given.

1.2 Objectives of the study

1. To create mixed integer linear programming to solve for the optimized solution with minimized makespan
2. To generate the sequencing solution with the least makespan
3. To develop the heuristic to solve the problem
4. To compare the computational time and the solution results between the MILP and the heuristic

1.3 Overview of Research

Overall, this research attempts to solve the production scheduling problem of a single machine system with varied changeover time to minimize the total makespan of the sequenced solution. The tool utilized will be the application of MILP and the heuristic to find the optimal solution from the given set of jobs. Therefore, the main delivery is the development of the MILP model and heuristic based on the literature review and the application of these methods in this designed system. In addition, from the solution delivery, the performance of the tools used will be analyzed to evaluate the number of jobs that the model can feasibly deliver the results.

CHAPTER 2
REVIEW OF LITERATURE

The review of literature in this study emphasizes the previous work related to the use of MILP in scheduling problems. Each literature will have a different MILP model. Furthermore, heuristics were proposed for the improvement as there were large job amount constraints that the MILP model cannot handle. The objective can vary among different research similarly to the amount and system of the machine in the system. Parameters such as due date and release date can be added into consideration.

2.1 Sequence-dependent job scheduling problem

Sequence-dependent job scheduling was first published by Gilmore and Gomory (Gilmore & Gomory, 1964) who utilized the traveling salesman problem concept to solve the problem with total setup cost as an objective. To optimize the solution to the same problem, Presby and Wolfson (Presby & Wolfson, 1967) used MILP under the constraint of a small problem size. There is a review of Allahverdi et al. (Allahverdi et al., 1999) which classified multiple literature based on the sequence dependency and batch system and provides the objective measure of each literature for solving scheduling problems. In this case, the sequence dependency is how each job relates to each other when they are placed adjacent. The changeover or setup time or cost will vary if the sequence is dependent and vice versa for a sequence-independent system as shown in Figure 1. Moreover, jobs can be classified further if they can be grouped without changeover required or not. If there is no changeover required, the system can be called a batch system, while the system will be called non-batch in case a changeover is needed for each job. This literature review guides through the mentioned configuration in different manufacturing systems including single machines, parallel machines, flow shops, and job shops. This literature was updated in 2006 and 2015 with consideration made on setup times and costs by Allahverdi et al. (Allahverdi et al., 2008) and Allahverdi (Allahverdi, 2015). The updated review further classified the setup into a normal setup between each job and family setup where jobs can be grouped into the same family with minimal to no setup required. The results of different approaches were compared by performance in each environment. Meanwhile, Yang

(Yang, 1999) also published literature with the intent to compare scheduling involving setup times in different dimensions including (i) job vs. class (family) setup (ii) sequence-dependent vs. sequence-independent (iii) separable vs. inseparable setups where separable means the setup of the next job can be performed during the time when the machine is idle.

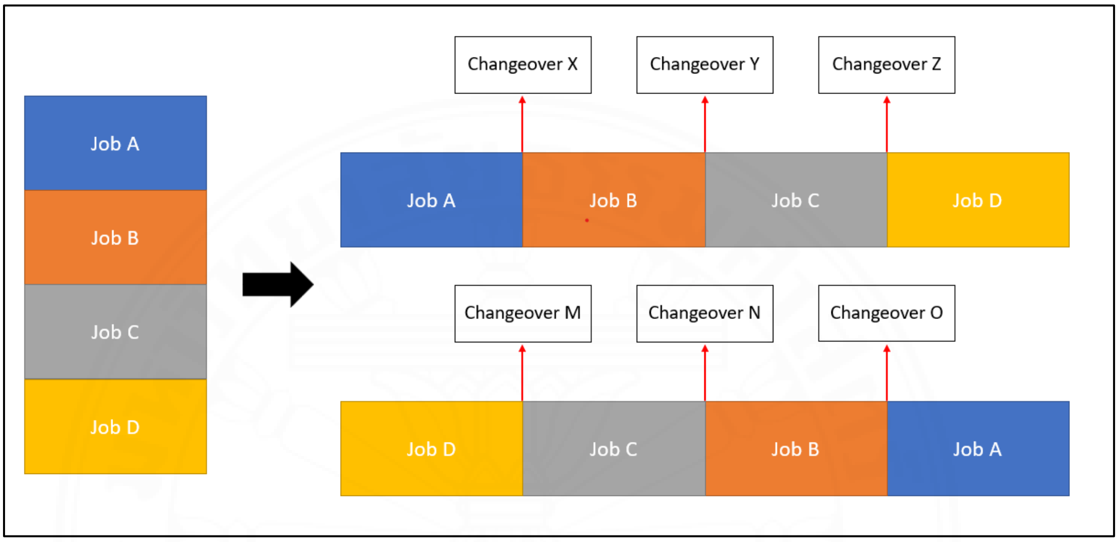

Figure 2.1 Sequence-dependent changeover

2.2 Literature with application of MILP on job scheduling

The first work found was from Vélez-Gallego et al. (Vélez-Gallego et al., 2016) with MILP developed to solve job scheduling for a single machine with arbitrary release dates and sequence-dependent setup times to minimize the makespan. Since the problem is nondeterministic polynomial hard, a beam search technique is proposed for searching high-performance solutions with fast computation. Bianco (Bianco et al., 1988) also developed a MILP model to minimize the makespan of a single machine system with sequence dependence and anticipatory setup times. A heuristic algorithm of the upper bound, lower bound, and dominance criteria was proposed. Kelly and Zyngier (Kelly & Zyngier, 2007) also presented a MILP model to minimize the cost of the sequence-dependent changeover for uniform discrete-time scheduling problems using memory operation logic variables. This can be applied to both batch and non-batch setups. Yalaoui and Nguyen (Yalaoui & Quy Nguyen, 2021) created a graph-based and sequence-based MILP model with a parallel machine with sequence-dependent setup times and release dates to minimize the makespan. Choobineh et al.

(Choobineh et al., 2006) also formulated the MILP model on sequence-dependent setup times on a single machine as well as developing an m-objective tabu search algorithm for sequencing n jobs. Antonioli et al. (Antonioli et al., 2022) created the MILP model with sequence-dependent setup times to minimize the total tardiness of the scheduling. With hybrid metaheuristics, the performance measures were evaluated and compared among proposed methods using the relative deviation index and success rate. Kucukkoc (Kucukkoc, 2019) used MILP in additive manufacturing machine scheduling problems to minimize makespan with various machine configurations. The calculation was performed in the CPLEX solver. For the flow shop system, Meng et al. (Meng et al., 2019) utilized the MILP with parallel configuration to minimize the makespan. 8 MILP models were formulated and tested to solve under sequence-dependent setup times, no-wait, and with blocking. Al-harkan and Qamhan (Al-harkan & Qamhan, 2019) created the MILP model under parallel machines with non-zero arbitrary release dates and non-anticipatory sequence-dependent setup times to minimize the makespan. A hybrid metaheuristic based on variable neighborhood search hybrid and simulated annealing was used for better computational time. Mousakhani (Mousakhani, 2013) performed MILP under a job shop system with sequence-dependent setup times to minimize tardiness. A metaheuristic under iterated local search was newly proposed and then perform comparative analyses with tabu search and variable neighborhood search algorithm previously mentioned. Naderi and Salmasi (Naderi & Salmasi, 2012) emphasized the application of MILP in sequence-dependent group scheduling in a flow shop system to minimize the makespan. In addition, a metaheuristic hybridizing genetic and simulated annealing algorithm was developed to compare the performance with the MILP solution. Xiao and Zheng (Xiao & Zheng, 2010) used MILP in the manufacturing and assembling processes of semiconductors, where the system is a two-stage hybrid flow shop with a sequence-dependent setup to minimize makespan. A heuristic was created with the rule designed by the authors. Lastly, Kongsri and Buddhakulsomsiri (Kongsri & Buddhakulsomsiri, 2020) developed a MILP on parallel machine scheduling with sequence dependence to minimize makespan and tardiness.

Table 2.1 Literature review summary

		Objective				System				Parameter		Setup type	
Author	**Heuristic**	**Make span**	**Tardines s**	**Cost**	**Profit**	**Single**	**Parallel**	**Serial**	**job shop**	**Due date**	**Release date**	**Anticip atory**	**Non-anticipa tory**
(Vélez-Gallego et al., 2016)	Beam search	x				x					x		x
(Kelly & Zyngier, 2007)				x		x	x	x				x	
(Yalaoui & Quy Nguyen, 2021)	Branch-and-bound algorithm	x					x				x		x
(Choobin eh et al., 2006)	Tabu search	x	x			x				x			x
(Antoniol i et al., 2022)	Order-Scheduling Modified Due-Date heuristic	x	x				x			x			x
(Kucukko c, 2019)		x				x	x	x		x		x	
(Meng et al., 2019)		x					x						x
(Al-harkan & Qamhan, 2019)	Two-stage hybrid variable neighborho od search hybrid and simulated annealing	x					x				x		x
(Mousakh ani, 2013)	Tabu search	x	x						x	x			x
(Naderi & Salmasi, 2012)	Genetic and simulated annealing algorithm	x						x					x
(Xiao & Zheng, 2010)	Bottleneck station	x						x			x		x
(Kongsri & Buddhaku lsomsiri, 2020)	Predetermin ed batch size based on business and assign high volume product to least flexible machine	x	x				x			x			x
(Bianco et al., 1988)	Upper, lower bound and dominance criteria	x				x					x	x	x

CHAPTER 3
METHODOLOGY

3.1 Data preparation

3.1.1 Original production plan

Initially, the original production plan is obtained from the actual production plan. The production plan contains multiple components as follows and related to shampoo/conditioner bottle filling industry.

- Process order number
- Start date and time
- End date and time
- SKU
- Planned production amount
- Bulk material (filling fluid)
- Product group
- Package size
- Processing time
- ID (1 to n number of jobs)

PO #	Start Date	S.Time	End Date	E.Time	FG Material	Planned	Bulk Mat	Product.Grp	Size	ID
908750900	01/10/2023	16:29	01/10/2023	19:49	80689004	2518	90149954	PT-SH-CAP-400	400	1
908750901	01/10/2023	20:22	01/10/2023	22:01	82305594	1250	90516379	PT-SH-CAP-400	400	2
908750902	01/10/2023	22:36	01/10/2023	23:48	80715171	1380	91351428	PT-SH-CAP-400-X6	400	3
908751880	01/11/2023	0:21	01/11/2023	2:56	80728171	2964	90294992	PT-SH-CAP-400-X6	400	4
908751878	01/11/2023	3:29	01/11/2023	4:47	80728181	1484	90077039	PT-SH-CAP-400-X6	400	5
908751869	01/11/2023	5:20	01/11/2023	6:36	80728178	1460	90294989	PT-SH-CAP-400-X6	400	6
908751876	01/11/2023	7:17	01/11/2023	7:57	80707365	786	91343534	PT-SH-CAP-300-X6	300	7
908751883	01/11/2023	8:30	01/11/2023	9:12	80722898	798	90562095	PT-CN-CAP-300-X3	300	8
908751871	01/11/2023	11:37	01/11/2023	14:58	80680957	1992	91342404	PT-CN-CAP-150-X3	150	9
908751873	01/11/2023	18:41	01/11/2023	20:21	80732998	996	91173033	RJ-CN-CAP-150-X3	150	10
908751891	01/11/2023	21:02	01/12/2023	3:33	80689182	3932	90094162	PT-SH-CAP-150-X3	150	11
6404647629	01/12/2023	4:06	01/12/2023	9:44	80697837	3928	90294992	PT-SH-CAP-150-X3	150	12
6426903481	01/12/2023	10:25	01/12/2023	13:36	82318078	786	90294992	PT-SH-CAP-300-X6	300	13
6411125021	01/12/2023	13:54	01/12/2023	14:35	82317307	786	90294992	PT-SH-CAP-300-X3	300	14
6411125022	01/12/2023	15:54	01/12/2023	16:35	82317807	798	91342404	PT-CN-CAP-300-X3	300	15
6404633075	01/12/2023	17:16	01/12/2023	19:37	80690092	838	90294992	PT-SH-CAP-90-X6	90	16
6382618595	01/12/2023	20:10	01/12/2023	22:31	80689324	838	90294989	PT-SH-CAP-90-X6	90	17
6388900574	01/12/2023	23:04	01/13/2023	8:35	80690527	3385	90094319	PT-SH-CAP-90-X6	90	18
6404647648	01/13/2023	9:16	01/13/2023	10:24	82318020	694	90256697	PT-SH-CAP-200-X6	200	19
6411119076	01/13/2023	11:27	01/13/2023	12:36	82318021	694	90077039	PT-SH-CAP-200-X6	200	20
6385782560	01/13/2023	13:09	01/13/2023	13:39	82251624	295	96236588	PT-SH-CAP-200-X6	200	21
6385782587	01/13/2023	14:20	01/13/2023	14:36	82251616	296	96236588	PT-SH-CAP-400-X6	400	22
6436358030	01/13/2023	15:09	01/13/2023	15:47	80728172	738	90345619	PT-SH-CAP-400-X6	400	23
6436358365	01/13/2023	16:20	01/13/2023	18:55	80728171	2964	90294992	PT-SH-CAP-400-X6	400	24
6347004210	01/13/2023	19:30	01/13/2023	21:09	82305595	1250	90505833	PT-SH-CAP-400	400	25
6379324936	01/13/2023	21:27	01/14/2023	0:46	82305597	2500	90505833	PT-SH-CAP-400	400	26
6340243180	01/14/2023	1:04	01/14/2023	4:25	82305225	2520	90505833	PT-SH-CAP-400	400	27
6401378563	01/14/2023	4:58	01/14/2023	6:37	82305596	1247	90835421	PT-SH-CAP-400	400	28
6411131992	01/16/2023	21:38	01/16/2023	0:00	80690092	838	90294992	PT-SH-CAP-90-X6	90	29
6452653464	01/16/2023	22:41	01/16/2023	0:00	82318085	786	90294992	PT-SH-CAP-150-X6	150	30
6436253117	01/16/2023	23:18	01/16/2023	0:00	80700591	786	91351428	PT-SH-CAP-300-X3	300	31
6459108940	01/16/2023	23:21	01/16/2023	0:00	80728172	738	90345619	PT-SH-CAP-400-X6	400	32
6336788929	01/16/2023	23:23	01/16/2023	0:00	82318014	694	90077039	PT-SH-CAP-400-X6	400	33
6388900584	01/16/2023	23:39	01/16/2023	0:00	82317299	399	91342404	PT-CN-CAP-300-X6	300	34
6449316391	01/17/2023	20:39	01/17/2023	0:00	80689003	2518	90116777	PT-SH-CAP-400	400	35
6426889376	01/17/2023	20:41	01/17/2023	0:00	80680823	1964	90294989	PT-SH-CAP-150-X3	150	36
6414311678	01/17/2023	21:38	01/17/2023	0:00	80693911	841	90345619	PT-SH-CAP-90-X6	90	37
6323947172	01/17/2023	22:19	01/17/2023	0:00	80689004	1259	90149954	PT-SH-CAP-400	400	38
6455874547	01/17/2023	22:49	01/17/2023	0:00	80690963	694	90145455	PT-SH-CAP-210-X6	210	39
6410888890	01/17/2023	23:19	01/17/2023	0:00	80690160	786	90145455	PT-SH-CAP-300-X6	300	40
6468628839	01/18/2023	20:39	01/18/2023	0:00	82305225	2520	90505833	PT-SH-CAP-400	400	41
6426890325	01/18/2023	21:11	01/18/2023	0:00	80697837	1964	90294992	PT-SH-CAP-150-X3	150	42
6407893051	01/18/2023	21:11	01/18/2023	0:00	80709129	1964	91882746	RJ-CN-CAP-150-X3	150	43
6388900491	01/19/2023	20:39	01/19/2023	0:00	80689004	2518	90149954	PT-SH-CAP-400	400	44
6414313386	01/19/2023	21:25	01/19/2023	0:00	80728181	2968	90077039	PT-SH-CAP-400-X6	400	45
6392000373	01/19/2023	21:38	01/19/2023	0:00	80689324	838	90294989	PT-SH-CAP-90-X6	90	46
6414305912	01/19/2023	22:19	01/19/2023	0:00	80732998	996	91173033	RJ-CN-CAP-150-X3	150	47
6404646033	01/20/2023	20:38	01/20/2023	0:00	80680957	1992	91342404	PT-CN-CAP-150-X3	150	48
6411021407	01/20/2023	20:39	01/20/2023	0:00	82305225	2520	90505833	PT-SH-CAP-400	400	49
6404647636	01/20/2023	21:25	01/20/2023	0:00	80728171	2964	90294992	PT-SH-CAP-400-X6	400	50
6426677320	01/20/2023	21:38	01/20/2023	0:00	80690092	838	90294992	PT-SH-CAP-90-X6	90	51
6414305913	01/20/2023	22:19	01/20/2023	0:00	80732998	996	91173033	RJ-CN-CAP-150-X3	150	52
6392012113	01/20/2023	22:20	01/20/2023	0:00	82305595	1250	90505833	PT-SH-CAP-400	400	53
6407892975	01/20/2023	23:18	01/20/2023	0:00	82318078	786	90294992	PT-SH-CAP-300-X6	300	54
6347004211	01/21/2023	3:41	01/21/2023	7:00	82305597	2500	90505833	PT-SH-CAP-400	400	55
6468642994	01/21/2023	5:19	01/21/2023	7:00	80689004	1259	90149954	PT-SH-CAP-400	400	56

6407892973	01/21/2023	5:22	01/21/2023	7:00	82317370	982	90256697	PT-SH-CAP-150-X3	150	57
6414307809	01/23/2023	20:39	01/23/2023	0:00	80689003	2518	90116777	PT-SH-CAP-400	400	58
6414302996	01/23/2023	21:38	01/23/2023	0:00	80693910	841	90169470	PT-SH-CAP-90-X6	90	59
6411118298	01/23/2023	22:35	01/23/2023	0:00	80693909	986	90169470	PT-SH-CAP-150-X3	150	60
6449309706	01/23/2023	22:51	01/23/2023	0:00	82318103	694	90294992	PT-SH-CAP-200-X6	200	61
6404647624	01/23/2023	23:19	01/23/2023	0:00	80690161	786	90145455	PT-SH-CAP-300-X6	300	62
6411119283	01/23/2023	23:39	01/23/2023	0:00	82317299	399	91342404	PT-CN-CAP-300-X6	300	63
6468642995	01/24/2023	17:19	01/24/2023	0:00	82305225	5040	90505833	PT-SH-CAP-400	400	64
6436358568	01/24/2023	20:41	01/24/2023	0:00	82305597	2500	90505833	PT-SH-CAP-400	400	65
6468625879	01/24/2023	20:41	01/24/2023	0:00	82305595	2500	90505833	PT-SH-CAP-400	400	66
6436358029	01/24/2023	21:38	01/24/2023	0:00	80690092	838	90294992	PT-SH-CAP-90-X6	90	67
6414305892	01/24/2023	21:38	01/24/2023	0:00	80689324	838	90294989	PT-SH-CAP-90-X6	90	68
6420401783	01/24/2023	22:19	01/24/2023	0:00	80732998	996	91173033	RJ-CN-CAP-150-X3	150	69
6404661043	01/24/2023	22:35	01/24/2023	0:00	80693913	986	90345619	PT-SH-CAP-150-X3	150	70
6455874012	01/24/2023	22:43	01/24/2023	0:00	80728178	1460	90294989	PT-SH-CAP-400-X6	400	71

Figure 3.1 Sample production plan

3.1.2 Changeover time calculation logic

To determine the changeover time between each job connection from the original production plan, a changeover time determination logic is provided below. The flowchart indicates the condition required to determine the changeover type. SKU will be compared between the job before and the job after and the characteristics of both jobs will result in different changeover types.

1. Do they have the same product group?
   - 1.1. If yes, do they use the same bulk material?
     - 1.1.1. If yes, are they the same SKU?
       - 1.1.1.1. If yes, then we must perform “line clearance” to change the process order.
       - 1.1.1.2. If not, then we must perform “1D changeover” to change artwork of the package.
     - 1.1.2. If no and the job before is conditioner, then we need to perform **“washout conditioner”**
     - 1.1.3. If no and the job before is shampoo, then we need to perform **“washout shampoo”**
   - 1.2. If not, do they have the same package size?
     - 1.2.1. If yes and the job before is shampoo, then we need to perform **“washout shampoo”**
     - 1.2.2. If yes and the job before is conditioner, then we need to perform **“washout conditioner”**

1.2.3. If no and they have the same bulk material, then we need to perform a **"3D changeover without washout"**

1.2.4. If no and they have different bulk material where bulk before is conditioner, we must perform **"washout conditioner"**

1.2.5. If no and they have different bulk material where bulk before is shampoo, we must perform **"washout shampoo"**

Table 3.1 Changeover time for each changeover type

Changeover type	Time (min)
3D WO SHM	25
3D WO CON	30
3D no WO	19
1D	14
LC	10
WO SHM	17
WO CON	21

3.1.3 Changeover time for each SKU crossover

In Excel where the Excel Open Solver will be used, figure 3 was created to allow data connection between SKU of job before and job after for changeover time determination.

From SKU	From product group	From bulk	From size	To SKU	To product group	To bulk	To size	Concatenate	Changeover type	Changeover time
80689004	PT-SH-CAP-400	90149954	400	80689004	PT-SH-CAP-400	90149954	400	8068900480689004	LC	10
80689004	PT-SH-CAP-400	90149954	400	82305594	PT-SH-CAP-400	90516379	400	8068900482305594	WO SHM	17
80689004	PT-SH-CAP-400	90149954	400	80715171	PT-SH-CAP-400-X6	91351428	400	8068900480715171	WO SHM	17
80689004	PT-SH-CAP-400	90149954	400	80728171	PT-SH-CAP-400-X6	90294992	400	8068900480728171	WO SHM	17
80689004	PT-SH-CAP-400	90149954	400	80728181	PT-SH-CAP-400-X6	90077039	400	8068900480728181	WO SHM	17
80689004	PT-SH-CAP-400	90149954	400	80728178	PT-SH-CAP-400-X6	90294989	400	8068900480728178	WO SHM	17

Figure 3.2 Changeover time for SKU crossover in Excel

3.1.4 Setup time

Setup time will be included as one of the parameters for the MILP model constraint. Therefore, the table was created and pulled data from the original production plan and changeover (setup) time from the changeover time for SKU crossover Excel table.

Table 3.2 Example of changeover time for each job crossover

From\To	**Job 1**	**Job 2**	**Job 3**
Job 1	15 min	16 min	14 min
Job 2	20 min	12 min	11 min
Job 3	17 min	19 min	5 min

3.2 MILP Model

Table 3.3 MILP model notation

Index sets	
$j, k \in N = \{1, 2, \ldots, n\}$	Indices of jobs, where N denotes the set of jobs. j means a job that goes first, and k is a job that goes after job j
$N_0 = \{0, 1, 2, \ldots, n\}$	N_0 denotes the set of jobs including a dummy job
Parameters	
S_{jk}	The setup time (changeover) of change from job j to job k (In this system, the makespan is equivalent to the total setup time as there is only a single machine)
Decision variables	
X_{jk}	1 if job k is scheduled after job j, or 0 otherwise
C_j	The completion time of job j
C_{max}	Maximum completion time among job j (makespan)

Objective function:

min C_{max} (3.1)

{Minimize makespan (C_{max})}

Constraints:

$$\sum_{j\in N_0:j\neq k} X_{jk} = 1 \ \forall k \in N \quad (3.2)$$

{Each job has 1 job before}

$$\sum_{k\in N_0:j\neq k} X_{jk} = 1 \ \forall j \in N \quad (3.3)$$

{Each job has 1 job after}

$$\sum_{k\in N_0:j\neq k} X_{jk} - \sum_{h\in N_0:h\neq j} X_{hj} = \ 0 \ \forall j \in N \quad (3.4)$$

{Job flow balance: The job cannot turn back to produce the already done one}

$$\sum_{k\in N} X_{0k} = 1 \quad (3.5)$$

{Specify first job of the machine must have only one job selected}

$$C_k - C_j + V(1 - X_{jk}) \geq S_{jk} \forall j \in N_0, \forall k \in N : j \neq k \quad (3.6)$$

{Ensure completion time of job after is equal to completion time of job before + setup time of job after. V is a large constant in case the job j and k are not connected}

$$C_0 = 0 \quad (3.7)$$

{Set dummy job completion time to zero}

$$X_{jk} \in \{0,1\} \forall j \in N_0, \forall k \in N : j \neq k \quad (3.8)$$

{Define binary decision variable}

$$C_j \geq 0 \ \forall j \in N \quad (3.9)$$

{Positive completion time}

$$C_{max} = \sum_{j,k\in N_0,} (X_{jk} \times S_{jk}) \quad (3.10)$$

{Total completion time is equal to sum of product of changeover time and X_{jk}}

3.3 Excel Open Solver

Apart from the data prepared in Excel, here is the MILP model preparation in Excel for Excel Open Solver. Firstly, a decision variable X_{jk} table was created as shown in table 4 below. The row and column header are the same list of jobs from the original production plan. The values for the decision variable are then used for other related constraints.

Table 3.4 Example of decision variable X_{jk} in Excel (Sequence job 1>2>3)

j\k	Dummy job	Job 1	Job 2	Job 3	$\sum_{k \in N_0 : j \neq k} X_{jk}$
Dummy job	0	1	0	0	1
Job 1	0	0	1	0	1
Job 2	0	0	0	1	1
Job 3	1	0	0	0	1
$\sum_{j \in N_0 : j \neq k} X_{jk}$	1	1	1	1	

Second, completion time is also considered the decision variable in the model. Table 5 shown will be designated for the location of decision variables C, C_0, C_{max} .

Table 3.5 Example of decision variable C, C_0, C_{max} table in Excel

Job	Completion time (min)
Dummy job (j or k = 0)	0
Job 1	20
Job 2	10
Job 3	30
Makespan (C_{max})	30

Third, a matrix for constraint equation 6 was created to compare the summation of changeover time in the prepared data section.

Table 3.6 Left-hand-side of equation (3.6) table in Excel

j\k	Dummy job	Job 1	Job 2	Job 3
Dummy job	$V + C_0 - C_0$	$V + C_1 - C_0$	$V + C_2 - C_0$	$V + C_3 - C_0$
Job 1	$V + C_0 - C_1$	$V + C_1 - C_1$	$V + C_2 - C_1$	$V + C_3 - C_1$
Job 2	$V + C_0 - C_2$	$V + C_1 - C_2$	$V + C_2 - C_2$	$V + C_3 - C_2$
Job 3	$V + C_0 - C_3$	$V + C_1 - C_3$	$V + C_2 - C_3$	$V + C_3 - C_3$

Table 3.7 Right-hand-side of equation 6 table in Excel

j\k	Dummy job	Job 1	Job 2	Job 3
Dummy job	0 min	0 min	0 min	0 min
Job 1	0 min	15 min	16 min	14 min
Job 2	0 min	20 min	12 min	11 min
Job 3	0 min	17 min	19 min	5 min

Finally, the decision variables, objective function, and constraints were inputted in the Open Solver and the optimization results can be obtained. The MILP will calculate the completion time for each job and reconcile it into constraint 6 to find the optimal solution. The sequence results will be validated by manual calculation to confirm that the output measure is correlated with the sequenced plan proposed.

3.4 Python CPLEX Solver

To improve the computational time of the MILP model even further, a CPLEX solver using Python was done. Firstly, it is required to import the Pandas library to import necessary data from Excel and Docplex to utilize CPLEX solver for MILP optimization. There are global variables including n (number of jobs + 1), N (index set of jobs), N_0 (index set of jobs including dummy job), V (big M for constraint equation 6), and A (matrix of X_{jk}). The decision variables were set to include X_{jk} (binary), C (stored in N_0 with positive value), and C_{max} (denoted as z). The objective function is set to minimize z. And like Excel solver, the constraints were added to the model. Moreover, the optimal solution tolerance is varied between 5% and 0.5% for computational time comparison.

3.5 Heuristic

Using the Excel file, the heuristic is developed based on the logic of trying to group the same package size together (as changing size has the highest changeover time) and then trying to connect the same bulk material across different sizes as much as possible. The logic will try to find the "cores" or the structure of the sequencing by knowing which size should be produced first or later from connecting the same bulk together (same bulk means there is no washing process, which will save changeover time). Then fill the one that is not the sequence core with the "fillers".

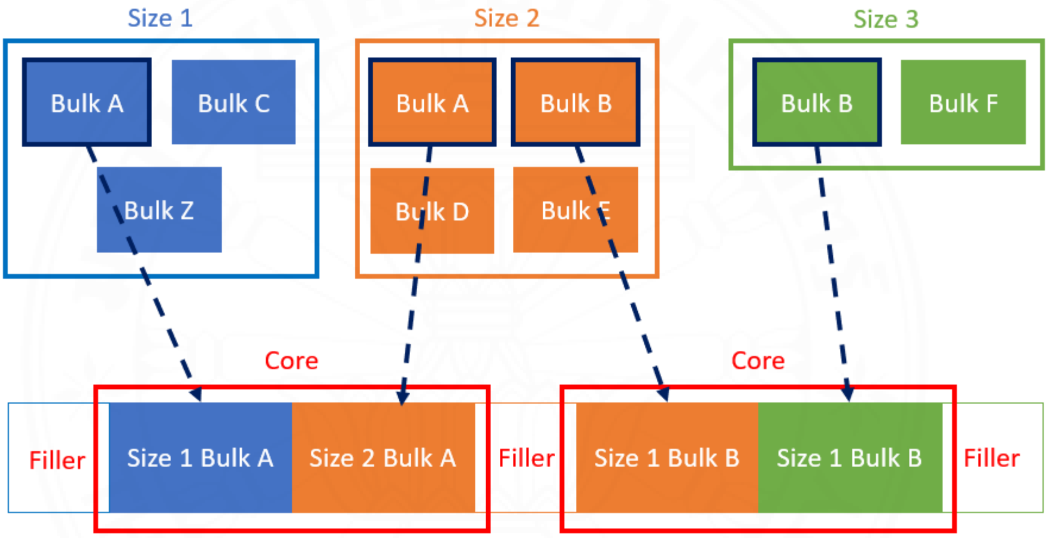

Figure 3.3 Sequence core and filler heuristic

1. Preparation phase (Prepare data for the heuristic)
   a. Concatenate the bulk material code and the size to create a unique combination of bulk and size.
   b. Highlight duplicated bulk material using conditional formatting function: This is for Excel to filter the duplication by color.
2. Filtering phase (Filter the filler out for later scheduling)
   a. Filter to have only the duplicated bulk material. The unduplicated bulk material will be stored as pool 1 for later scheduling.
   b. Filter the duplicated concatenation of bulk material and size. Then sort ascending.

c. Create column “unique?” to check if the previous concatenation is similar to the current row or not. Starting from 0, if the previous concatenation is similar to the current row, add 1 from it. This point will not be correct if the concatenation column was not sorted before.

d. Filter only the “unique?” column = 0 to obtain a set of only unique concatenations of bulk material and size. The others are stored as pool 2 for later scheduling.

e. From number 2d, filter the duplicated bulk material and sort ascending by size and then by bulk material respectively. For the job with unique bulk material, stored as pool 3 for later scheduling.

3. Sequencing phase (sequence the core after filtering fillers out)

    a. Starting from the smallest size for the smallest bulk number, is there any similar bulk in the next size?

        i. If yes, then connect the similar bulk together

        ii. If no, then go to number 3b

    b. Consider the next bulk for the size. Is there any bulk that can connect to the next size?

        i. If yes, then connect the similar bulk together

        ii. If no, then go to number 3c

    c. Consider the next size. Is there any bulk that can connect to the next size?

        i. If yes, then connect the similar bulk together and move to the next size

        ii. If no, then go to number 3d

    d. Is all the job connected?

        i. If yes, then go to number 3e

        ii. If no, then repeat the process starting from number 3a

    e. Is there > 1 connection group?

        i. If yes, starting from the smallest size of any end, search for the unconnected job for the connection with the end of another group starting from the smallest size first

        ii. If no, then go to number 4

4. Filling phase (fill the fillers after finishing the sequence core)

a. From the sequence core, combine unconnected jobs with jobs from pools 1, 2, and 3 together. Then sort ascending by size then bulk.
b. From the sequence core, the connected job will indicate “the end gap between package size. Fill the jobs from number 4a into the middle gap of each size by size
c. If there is a job with a size not connected, fill in the end

CHAPTER 4
RESULTS

4.1 Excel Open Solver

Initially, the production scheduling problem using the MILP model was executed on Excel Open Solver. The computational time is 38696 seconds (10.8 hrs) and the makespan result is 1039.45 minutes as the optimized solution for 10 jobs. The computational time and sequenced plan are shown in Figures 5 and 6 respectively.

Figure 4.1 Excel Open Solver computational results for 10-job problem

PO #	Start Date	S.Time	End Date	E.Time	FG Material	Plann	Line #	Bulk Ma	FIL-RK	Product.Grp	Size	Changeover t	Changeover ti	Processing time (Pik	ID	Seque
908751880	01/11/2023	0:21	01/11/2023	2:56	80728171	2964	B9	90294992	#N/A	PT-SH-CAP-400-X6	400			154.65	4	1
908750902	01/10/2023	22:36	01/10/2023	23:48	80715171	1380	B9	91351428	1F1R	PT-SH-CAP-400-X6	400	WO SHM	17	72.00	3	2
908751869	01/11/2023	5:20	01/11/2023	6:36	80728178	1460	B9	90294989	#N/A	PT-SH-CAP-400-X6	400	WO SHM	17	76.17	6	3
908750900	01/10/2023	16:29	01/10/2023	19:49	80689004	2518	B9	90149954	#N/A	PT-SH-CAP-400	400	WO SHM	17	200.33	1	4
908750901	01/10/2023	20:22	01/10/2023	22:01	82305594	1250	B9	90516379	1F1H	PT-SH-CAP-400	400	WO SHM	17	99.45	2	5
908751878	01/11/2023	3:29	01/11/2023	4:47	80728181	1484	B9	90077039	1H1F	PT-SH-CAP-400-X6	400	WO SHM	17	77.43	5	6
908751883	01/11/2023	8:30	01/11/2023	9:12	80722898	798	B9	90562095	451V	PT-CN-CAP-300-X3	300	3D WO SHM	25	41.63	8	7
908751876	01/11/2023	7:17	01/11/2023	7:57	80707365	786	B9	91343534	1F1Q	PT-SH-CAP-300-X6	300	WO CON	21	40.13	7	8
908751873	01/11/2023	18:41	01/11/2023	20:21	80732998	996	B9	91173033	412Y	RJ-CN-CAP-150-X3	150	3D WO SHM	25	100.65	10	9
908751871	01/11/2023	11:37	01/11/2023	14:58	80680957	1992	B9	91342404	4712	PT-CN-CAP-150-X3	150	WO CON	21	201.30	9	10

Figure 4.2 Optimized sequenced plan of 10-job problem by Excel Open Solver

4.2 Python CPLEX Solver

As it is noticeable that Excel Open Solver requires an excessive amount of time to perform optimization on the 10-job problem. Thus, the CPLEX solver is used to compare results with the Excel Open Solver. The results can be seen in Table 8. It can be observed that the CPLEX solver significantly improves the calculation time from the Excel Open Solver, with the 10-job problem using only 1 second while Excel Open Solver uses 10.8 hours.

Table 4.1 Computation time and total setup time results between MILP5 and MILP0.5

#job	Computational time (s)		Total setup time (min)	
	MILP5	MILP0.5	MILP5	MILP0.5
10	1	1	177	177
20	1	1	343	339
30	1	2	489	489
40	1	6	663	650
50	5	75	782	761
60	5	224	905	880
71	18	3492	1033	1008

Comparing MILP with solution tolerance of 5% and 0.5%, as expected, the computational time for MILP5 is better than MILP0.5, but the solution has a higher total setup time.

With this result, the results from the CPLEX solver will be further compared with the developing heuristics to improve the computation further while compensating for the degraded output. Meanwhile, it is also worth considering a method to improve the CPLEX solver.

4.3 Heuristic

From Table 9, the results showed increasing computational time as the number of jobs increases. For the total setup time, it will be compared with other methods in the later section.

Table 4.2 Computation time and total setup time results of heuristic

#job	Computational time (s)	Total setup time (min)
10	78	182
20	230	353
30	320	503
40	393	670
50	385	791
60	410	914
71	438	1053

CHAPTER 5
DISCUSSION

In terms of computation time between the heuristic, MILP5, and MILP0.5, the MILP5 reflects the fastest computation. Within the frame of job numbers of 10 to 71, the computation time for MILP0.5 increases exponentially as almost an hour is needed for optimizing 71 jobs. This is because of the narrower tolerance window of MILP0.5 allowing it to accept the solution quicker. For the heuristic, even though it has the slowest computation compared to those from MILP. The characteristic of computation time inclination is linear as there may certain points where it can compute quicker than MILP5.

Table 5.1 Computation time for heuristic, MILP5, and MILP0.5

	Computation time (seconds)					
#jobs	**Heuristic**	**Heuristic %diff**	**MILP5**	**MILP5 %diff**	**MILP0.5**	**V**
10	78	7700.0%	1	0.0%	1	506
20	230	22900.0%	1	0.0%	1	966
30	320	15900.0%	1	-50.0%	2	1412
40	393	6450.0%	1	-83.3%	6	1923
50	385	413.3%	5	-93.3%	75	2276
60	410	83.0%	5	-97.8%	224	2903
71	438	-87.5%	18	-99.5%	3492	3302

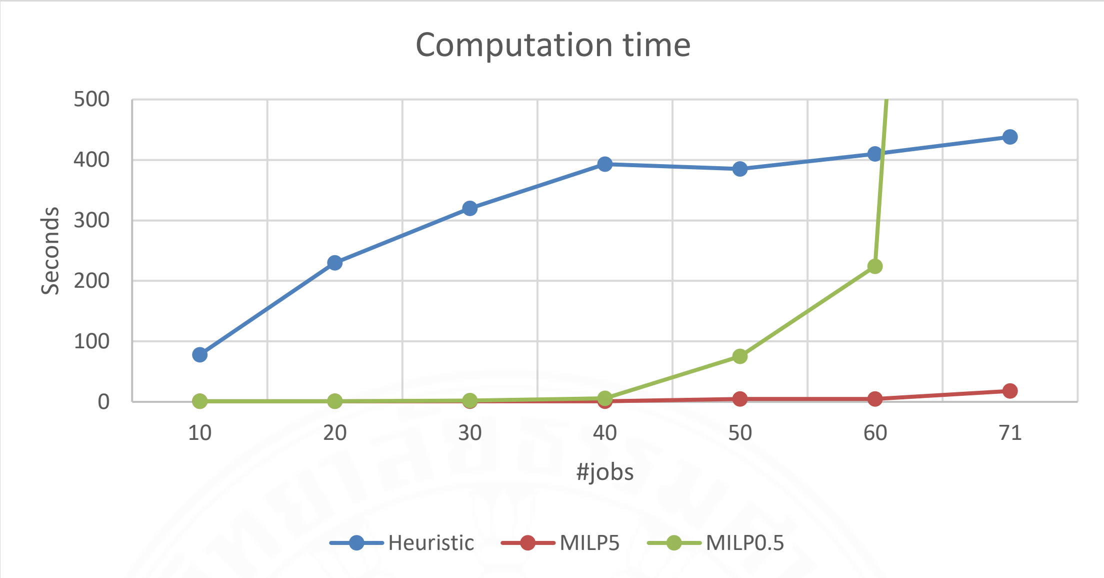

Figure 5.1 Computation time for heuristic, MILP5, and MILP0.5

In terms of the solution results, a clear trend can be seen as MILP0.5 can give the best solution followed by MILP5 and the heuristic developed at any number of jobs. In addition, there is no clear trend of the %difference compared to optimized answers from MILP0.5. From this, it depends on the business to choose the suitable method for job scheduling, whether it wants solution optimality, cost, or human effort. Currently, with this problem setup, the MILP has the best proposal as it provides the optimal solution with adjustable tolerance based on preference. However, if the CPLEX cost is the concern or more jobs need to be sequenced, then a study must be conducted further to determine the performance of the proposed heuristic at a larger job set.

Table 5.2 Total setup time results for heuristic, MILP5, and MILP0.5

	Total setup time (minute)					
#jobs	**Heuristic**	**Heuristic %diff**	**MILP5**	**MILP5 %diff**	**MILP0.5**	**V**
10	182	2.8%	177	0.0%	177	506
20	353	4.1%	343	1.2%	339	966
30	503	2.9%	489	0.0%	489	1412
40	670	3.1%	663	2.0%	650	1923
50	791	3.9%	782	2.8%	761	2276
60	914	3.9%	905	2.8%	880	2903
71	1053	4.5%	1033	2.5%	1008	3302

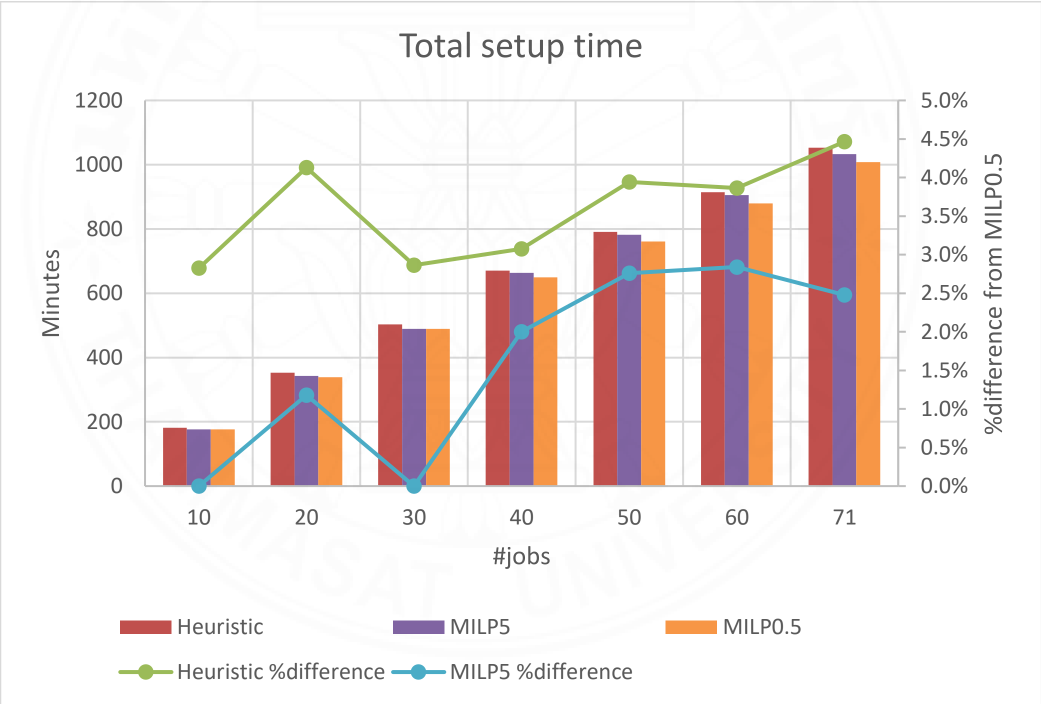

Figure 5.2 Total setup time result and %difference from MILP0.5

CHAPTER 6
CONCLUSION

In this study, the production scheduling of a job for a single shampoo/ conditioner filling machine is optimized using various proposed tools including the Excel MILP solver, the CPLEX MILP python solver, and the heuristic developed to minimize the makespan or the total production time for the manufacturing process. Firstly, the MILP model was developed and executed using an Excel solver, and the computation time is infeasible. Therefore, the CPLEX solver is used instead. Then the heuristic is developed to compare with the MILP model to provide an alternative method to reduce computation time and software cost. As a result, the MILP with a solution tolerance of 0.5% provides the most optimized solution. However, the MILP with a solution tolerance of 5% provides the fastest computation. For the heuristic, even though it has the worst results in terms of the solution optimality and the computation time, the computation time though has a linear trend. This means that, at a larger problem size, the heuristic may be computed faster than MILP.

REFERENCES

Al-harkan, I. M., & Qamhan, A. A. (2019). Optimize unrelated parallel machines scheduling problems with multiple limited additional resources, sequence-dependent setup times and release date constraints. IEEE Access, 7, 171533–171547. https://doi.org/10.1109/access.2019.2955975

Allahverdi, A. (2015). The third comprehensive survey on scheduling problems with setup times/costs. European Journal of Operational Research, 246(2), 345–378. https://doi.org/10.1016/j.ejor.2015.04.004

Allahverdi, A., Gupta, J. N. D., & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. Omega, 27(2), 219–239. https://doi.org/10.1016/s0305-0483(98)00042-5

Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187(3), 985–1032. https://doi.org/10.1016/j.ejor.2006.06.060

Antonioli, M. P., Rodrigues, C. D., & Prata, B. de. (2022). Minimizing total tardiness for the order scheduling problem with sequence-dependent setup times using hybrid matheuristics. International Journal of Industrial Engineering Computations, 13(2), 223–236. https://doi.org/10.5267/j.ijiec.2021.11.002

Bianco, L., Ricciardelli, S., Rinaldi, G., & Sassano, A. (1988). Scheduling tasks with sequence-dependent processing times. Naval Research Logistics, 35(2), 177–184. https://doi.org/10.1002/1520-6750(198804)35:2<177::aid-nav3220350203>3.0.co;2-v

Choobineh, F. F., Mohebbi, E., & Khoo, H. (2006). A multi-objective Tabu Search for a single-machine scheduling problem with sequence-dependent setup times. European Journal of Operational Research, 175(1), 318–337. https://doi.org/10.1016/j.ejor.2005.04.038

Gilmore, P. C., & Gomory, R. E. (1964). Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. Operations

Research, 12(5), 655–679. https://doi.org/10.1287/opre.12.5.655

Kelly, J. D., & Zyngier, D. (2007). An improved MILP modeling of sequence-dependent switchovers for discrete-time scheduling problems. Industrial & Engineering Chemistry Research, 46(14), 4964–4973. https://doi.org/10.1021/ie061572g

Kongsri, P., & Buddhakulsomsiri, J. (2020). A mixed integer programming model for unrelated parallel machine scheduling problem with sequence dependent setup time to minimize makespan and total tardiness. 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA). https://doi.org/10.1109/iciea49774.2020.9102086

Kucukkoc, I. (2019). MILP models to minimise makespan in additive manufacturing machine scheduling problems. Computers & Operations Research, 105, 58–67. https://doi.org/10.1016/j.cor.2019.01.006

Meng, L., Zhang, C., Shao, X., Zhang, B., Ren, Y., & Lin, W. (2019). More MILP models for hybrid flow shop scheduling problem and its extended problems. International Journal of Production Research, 58(13), 3905–3930. https://doi.org/10.1080/00207543.2019.1636324

Mousakhani, M. (2013). Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness. International Journal of Production Research, 51(12), 3476–3487. https://doi.org/10.1080/00207543.2012.746480

Naderi, B., & Salmasi, N. (2012). Permutation flowshops in group scheduling with sequence-dependent setup times. European J. of Industrial Engineering, 6(2), 177. https://doi.org/10.1504/ejie.2012.045604

Pinedo, M. (2008). Scheduling: Theory, algorithms, and systems. Prentice Hall.

Presby, J. T., & Wolfson, M. L. (1967). An algorithm for solving job sequencing problems. Management Science, 13(8). https://doi.org/10.1287/mnsc.13.8.b454

Vélez-Gallego, M. C., Maya, J., & Montoya-Torres, J. R. (2016). A beam search heuristic for scheduling a single machine with release dates and sequence dependent setup times to minimize the makespan. Computers & Operations Research, 73, 132–140. https://doi.org/10.1016/j.cor.2016.04.009

Xiao, J., & Zheng, L. (2010). A MILP-based batch scheduling for two-stage hybrid flowshop with sequence-dependent setups in semiconductor assembly and test manufacturing. 2010 IEEE International Conference on Automation Science and Engineering. https://doi.org/10.1109/coase.2010.5584514

Yalaoui, F., & Quy Nguyen, N. (2021). Identical machine scheduling problem with sequence-dependent setup times: MILP formulations computational study. American Journal of Operations Research, 11(01), 15–34. https://doi.org/10.4236/ajor.2021.111002

Yang, W.-H. (1999). Survey of scheduling research involving Setup Times. International Journal of Systems Science, 30(2), 143–155. https://doi.org/10.1080/002077299292498

APPENDIX

APPENDIX A

PYTHON CPLEX SOLVER CODE

```
import pandas as pd
from docplex.mp.model import Model
#variable setting
n=29
N=[i for i in range(1,n)]
N0=[0]+N
V=1412
A=[(j,k)for j in N0 for k in N0 if j!=k]
#print(' N=',N)#for debugging purpose
#print(' N0=',N0)#for debugging purpose
#print(' A=',A)#for debugging purpose
#import data from excel
SP = pd.read_excel (io='./Project final.xlsx',sheet_name='pythonsetup',index_col=0)
print(SP) #for debugging purpose
#print(' SP[1,1]=',SP.iloc[1,1])#for debugging purpose

#CPLEX model
md1=Model('MILP')

#step3 add decision variables
x=md1.binary_var_dict(A,name='x')
C=md1.continuous_var_list(N0,name='C',lb=0)
z=md1.continuous_var(name='z',lb=0)

#add MILP solution tolerance
md1.parameters.mip.tolerances.mipgap.set(float(0.005))

#objective function
md1.minimize(z)

#constraint 1:
for k in N:
	md1.add_constraint(md1.sum(x[j,k] for j in N0 if j!=k)==1)
#constraint 2:
for j in N:
	md1.add_constraint(md1.sum(x[j,k] for k in N0 if j!=k)==1)
#constraint 3:
for j in N:
	md1.add_constraint((md1.sum(x[j,k] for k in N0 if j!=k)-md1.sum(x[h,j] for h in
N0 if h!=j)==0))
#constraint 4:
md1.add_constraint(md1.sum(x[0,k] for k in N)==1)
#constraint 5:
```

```
for j in N0:
    for k in N:
        if j!=k:
            md1.add_constraint(C[k]-C[j]+V*(1-x[j,k])>=SP.iloc[j,k])
#constraint 6:
md1.add_constraint(C[0]==0)

#constraint 7:
for j in N:
    md1.add_constraint(C[j]>=0)

#constrain 8:
md1.add_constraint(z == md1.sum(SP.iloc[j,k]*x[j,k] for j,k in A))

print(md1.export_to_string())
#step6 solve the instance
solution=md1.solve(log_output=True)
print(solution)
```

BIOGRAPHY

Name	Naphat Ormsapsin
Education	2018: Bachelor of Engineering (Chemical Engineering) Sirindhorn International Institute of Technology Thammasat University