Optimization of Job Shop Scheduling with Material Handling by Automated Guided Vehicle

by

Shiyang Huang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:

Guiping Hu, Major Professor

Mingyi Hong

Cameron MacKenzie

Frank Montabon

Lizhi Wang

Iowa State University

Ames, Iowa

2018

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Job Shop Scheduling with Material Handling has attracted increasing attention in both industry and academia, especially with the inception of Industry 4.0 and smart manufacturing. A smart manufacturing system calls for efficient and effective production planning. On a typical modern shop floor, jobs of various types follow certain processing routes through machines or work centers, and automated guided vehicles (AGVs) are utilized to handle the jobs. In this research, the optimization of a shop floor with AGV is carried out, and we also consider the planning scenario under variable processing time of jobs. The goal is to minimize the shop floor production makespan or other specific criteria correlated with makespan, by scheduling the operations of job processing and routing the AGVs. This dissertation includes three research studies that will constitute my doctoral work.

In the first study, we discuss a simplified case in which the scheduling problem is reformulated into a vehicle dispatching (assignment) problem. A few AGV dispatching strategies are proposed based on the deterministic optimization of network assignment problems. The AGV dispatching strategies take future transportation requests into consideration and optimally configure transportation resources such that material handling can be more efficient than those adopting classic AGV assignment rules in which only the current request is considered. The strategies are demonstrated and validated with a case study based on a shop floor in literature and compared to classic AGV assignment rules. The results show that AGV dispatching with adoption of the proposed strategy has better performance on some specific criterions like minimizing job waiting time.

In the second study, an efficient heuristic algorithm for classic Job Shop Scheduling with Material Handling is proposed. Typically, the job shop scheduling problem and material handling problem are studied separately due to the complexity of both problems. However, considering these two types of decisions in the same model offers benefits since the decisions are related to each other. In this research, we aim to study the scheduling of job operations together with the AGV routing/scheduling, and a formulation as well as solution techniques are proposed. The proposed heuristic algorithm starts from an

optimal job shop scheduling solution without limiting the size of AGV fleet, and iteratively reduces the number of available vehicles until the fleet size is equal to the original requirements. The computational experiments suggest that compared to existing solution techniques in literature, the proposed algorithm can achieve comparable solution quality on makespan with much higher computational efficiency.

In the third study, we take the variability of processing time into consideration in optimizing job shop scheduling with material handling. Variability caused by random effects and deterioration is discussed, and a series of models are developed to accommodate random and deteriorating processing time respectively. With random processing time, the model is formulated as a Stochastic Programming Job Shop Scheduling with Material Handling model, and with deteriorating processing time the model can be nonlinear under specific deteriorating functions. Based on a widely adopted dataset in existing literature, the stochastic programming model were solved with Pyomo, and models with deterioration were linearized and solved with CPLEX. By considering variable processing time, the JSSMH models can better adapt to real production scenarios.

CHAPTER 1. GENERAL INTRODUCTION

1.1 Research Background

Production scheduling is essential in achieving optimal performance on a manufacturing shop floor, and it is well known that job shop scheduling problems are computationally challenging. When material handling is not considered in the planning process, the problem is reduced to the classic Job Shop Scheduling (JSS) problem, which is difficult to solve even for small-sized problems (Pinedo, 2009). Additionally, it is important to consider transportation of materials and jobs between multiple machines or work centers. Job Shop Scheduling with Material Handling (JSSMH) problems aims to consider job shop scheduling and material handling decisions in the same framework and this brings additional modeling and computational challenges.

Using automated guided vehicles (AGVs) on shop floors has become an important trend in the manufacturing industry due to easier control as well as the elimination of human error (Carlo, Vis, & Roodbergen, 2014). AGVs are also playing significant roles in many other areas such as container terminals and warehouses, and they prove to be effective in increasing the efficiency of logistics and warehousing systems. This serves as one of the major motivations for this dissertation work. It is our intention that this dissertation would shed lights on the efficiency of adopting AGV systems, especially in scheduling of modern smart manufacturing shop floors.

On a manufacturing shop floor, each job is processed on a set of machines in certain sequence according to the job type. Nowadays job shops' control and planning are mainly done electronically, and the material handling process relies on robots or AGVs. In the body of literatures such a system is also defined as a Flexible Manufacturing System (FMS) (Browne, Dubois, Rathmill, Sethi, & Stecke, 1984; El Maraghy, 2006). The goal of planning and decision making for FMS typically focuses on minimizing the makespan (Han, Xing, Chen, Lei, & Wang, 2014; Kumar, Haleem, Garg, & Singh, 2015), and JSSMH is a representative planning scenario in the FMS. The JSSMH problem can be viewed as a combination of

JSS and a vehicle scheduling/vehicle routing (VS/VR) problem, both acknowledged as complicated optimization problems and proved to be NP-hard (Baños, Ortega, Gil, Márquez, & De Toro, 2013; Doh, Yu, & Kim, 2013). Research interests on JSSMH has been increasing and a variety of optimization methods have been proposed, since AGVs' introduction to the manufacturing shop floors in the 1990's.

Limited attention has been paid to the production scheduling problems that job processing time is variable, which has been reflected many production scenarios. When human activity is involved in job processing, the job processing time can be affected by variability of human manipulation, and jobs themselves can have inherent variability in processing time too. Variable processing time has not been considered in job shop scheduling when material handling is part of decision making. With material handling system as an integral part of production, it is essential to take this into consideration when making production decisions

The three research studies in this dissertation fit into three scenarios of JSSMH. In the first study, we focus on the AGV planning problem, in which the JSSMH problem is simplified to be a vehicle dispatching/assignment problem. The second study considers the job shop scheduling and AGV routing simultaneously, with a comprehensive JSSMH optimization model. In the third study, we consider the JSSMH under variable processing time, which brings additional difficulty to solving the scheduling problem, hence a stochastic programming model and models involving deteriorating processing time are developed based on classic JSSMH.

1.2 Introduction and Literature Review

For the AGV dispatching problem in the first study, we propose a series of AGV dispatching strategies that are based on network optimization and shorten job-waiting times. In the second study, a comprehensive JSSMH model is formulated and a heuristic algorithm is proposed to efficiently find a solution close to optimality. The model is extended to deal with variability of job processing in the third study.

The three studies are distinct according to the scenarios, but also associated with each other inherently. The literature review is also presented separately in each of the subsections.

1.2.1. AGV Dispatching

Given a predetermined job shop schedule, a set of classic AGV assignment rules were developed by Egbelu and Tanchoco (1984) that guide the response and movement of AGVs on shop floors when transportation requests arrive. Classic AGV assignment rules are executed when a vehicle becomes idle (vehicle initiated) or a job is ready to be transported (work center initiated). The AGV assignment rules decide which AGV should respond to the current transportation request when there are several idle AGVs, or which request an idle AGV should respond when there are several awaiting requests. Table 1.1 summarizes the classic AGV assignment rules. A combined strategy of RV/RW and NV/STT is most commonly adopted in practice and serves as the benchmark of comparison to the proposed strategies in our research.

Table 1.1: Classic AGV assignment rules

Work Center Initiated Assignment Rule	Vehicle Initiated Assignment Rule
Random Vehicle (RV)	Random Work Center (RW)
Nearest Vehicle (NV)	Shortest Travel Time (STT)
Farthest Vehicle (FV)	Longest Travel Time (LTT)
Longest Idle Vehicle (LIV)	Maximum Outgoing Queue Size (MOQS)
Least Utilized Vehicle (LUV)	Minimum Remaining Outgoing Queue Space (MROQS)
	First Come-First Serve (FCFS)
	Unit Load Shop Arrival Time (ULSAT)

When an AGV becomes idle or when one job is ready at the output port of a work center, decisions on AGV assignment are made based on classic rules in Table 1.1. In each assignment decision, there is a matching between one AGV and one request. In other words, the classic AGV assignment rules respond to one request at a time. Such a short decision horizon brings convenience to AGV programmers, and applying classic AGV assignment rules is effective considering the frequent and complicated material flows on the shop floor. This strategy is probably not the most efficient, however, since programmable AGV systems enable shop floor operators to accomplish material handling in a more efficient way by storing and processing more information in AGVs (Abbas, Mohamed, & Hafez, 2014).

Vehicle assignment problem has its application in more areas other than shop floors such as container terminals or smart warehouses (Confessore, Fabiano, & Liotta, 2013; J. Kim, Choe, & Ryu, 2013; L. H. Lee, Chew, Tan, & Wang, 2010; Luo & Wu, 2015; Luo, Wu, & Mendes, 2016; Vis, 2006). Furthermore, besides heuristic assignment rules, optimizations methods have also been developed to accomplish AGV movement optimization in a limited or rolling time horizon (Fauadi, Yahaya, & Murata, 2013; Fazlollahtabar, 2016). However, unlike AGV planning problems in container terminals, AGV dispatching on shop floors has a vital characteristic that makes the problem more complicated. In container terminals, containers are transported by AGVs only once, from one storage area (can be a ship) to another. For shop floors on the other hand, jobs are loaded and unloaded, usually by different vehicles, between different work centers multiple times due to sequential processing characteristic. Consequently, there are more decision variables in AGV dispatching problems on shop floors than in container terminals. Moreover, the decision variables and decision making conditions are correlated, i.e., for the same current request, different AGV dispatching decisions might lead to a different timing and sequence of future requests, which makes the problem even more complicated.

Besides the traditional heuristic-based approach, Mathematical programming-based approaches have been proposed. Multi-objective optimization was adopted by many researchers to meet multiple criteria on shop floor and container terminals (J. Kim et al., 2013; U A Umar, Ariffin, Ismail, & Tang, 2013). AGV optimization models usually include integer variables; hence, the problem could usually be described with integer programming models such as set partitioning (K. S. Kim, Chung, & Jae, 2003) and minimum cost flow networks (Confessore et al., 2013; Joe, Gan, & Lewis, 2014). Different models have resulted in different solution techniques, including arithmetic calculation (Egbelu, 1987), simulation (Wang, Guan, Shao, & Ullah, 2014), exact solution algorithms (Tanaka, Nishi, & Inuiguchi, 2010), and heuristic algorithms (Nageswararao, Rao, & Rangajanardhana, 2012). Almost all dispatching models minimize makespan or waiting time (Confessore et al., 2013; Joe et al., 2014; J. Kim et al., 2013; Pisuchpen, 2012).

In this study, we developed two AGV dispatching strategies based on assignment problems in network optimization for a shop floor where the status of vehicles as well as jobs (products) in work centers are predictable. Firstly, we consider two requests in a row when the first one has been realized and second one is predicted, hence it is expected to be more efficient than only considering current request. Secondly, we observe the status of products at all work centers, and optimize the comprehensive AGV assignment.

The case study is based on Egbelu (1987). The product batches are large enough to observe the validity of proposed AGV assignment rules, and it is appropriate to implement on simulation platforms. Results in Egbelu (1987) also acts as a reference to validate the simulation model developed in this study. The package CPLEX is utilized in JAVA-based simulation platform AnyLogic when solving the optimization models in proposed AGV dispatching strategies. In the dynamic production process, corresponding parameters keep updating, and are passed to models to be solved repeatedly. The performance of our AGV dispatching strategies are compared with classic rules in scenarios with different AGV fleet sizes, and it proved that our optimization is valid, resulting in shorter material (product) waiting time.

1.2.2. Deterministic Job Shop Scheduling with Material Handling

The JSSMH problem can be viewed as a combination of a job shop scheduling (JSS) and a vehicle scheduling (VS) or vehicle routing (VR) problems, which have both been recognized as complicated decision making problems (Baños et al., 2013; Doh et al., 2013). These two problems have been extensively studied in the existing body of literature. For JSS problems, a variety of techniques, ranging from exact methods to hybrid techniques, have been proposed since 1950's, and summarized by Albert Jones and C.Rabelo (1999) by the end of last century, and Chaudhry and Khan (2016) more recently. Typical solution techniques of JSS include classic exact algorithms like branch-and-bound (Ashour & Hiremath, 1973) and genetic algorithms (Pezzella, Morganti, and Ciaschetti 2008). VS/VR problems is also known to be NP-hard (Lenstra & Kan, 1981), and recent solution technique studies for VS/VR focus on efficient heuristics such as evolutionary algorithm (Chiang & Lin, 2013) and simulation-based approach (Villarreal, Garza-Reyes, & Kumar, 2016).

Optimization of JSSMH has mainly been studied for small size manufacturing shop floors, while recent advancement of computational resources has reinvigorated the research in the JSSMH problem. Bilge and Ulusoy (1995) formulated a nonlinear programming optimization model and proposed a heuristic time window-based algorithm to solve the problem, and following this work, various models have been proposed (Xie & Allen, 2015). Typically, JSSMH models aim to minimize production makespan, either as a sole objective function or as a vital optimization criterion in the multi-objective settings. The essence of JSSMH models consists of a set of job scheduling constraints that determines operations sequences on machines, and a set of constraints that determines the routing of AGVs. Additional constraints may be adopted considering shop floor conditions such as path constraints (Bürgy & Gröflin, 2016; Wang et al., 2014) and task preemption (Dang & Nguyen, 2017; Izabela Nielsen, Dang, Nielsen, & Pawlewski, 2014). Variations of JSSMH models include different presentation of vehicle movement (Ahmadi-Javid & Hooshangi-Tabrizi, 2017), or adoption of different modeling methodologies such as constraint programming (Novas & Henning, 2014) and Petri nets (Baruwa & Piera, 2016). The classic JSSMH problem has been proved to be NP-hard (Na, Woo, & Lee, 2016).

The solution techniques to JSSMH in the body of literature are mainly heuristic based and specifically genetic algorithms. When the JSSMH problem was firstly formulated, Bilge and Ulusoy (1995) derived a time window of job pick-up at machines, which was used to regulate the movement of vehicles. Deroussi, Gourgand, and Tchernev (2008) implemented three different metaheuristics algorithm including iterated local search, simulated annealing, and a hybrid of these two to the JSSMH problem. Reddy and Rao (2006) formulated the problem into a multi-objective model for scheduling both the vehicles and machines, and the problem was solved with evolutionary algorithms. Abdelmaguid et al. (2004) proposed a hybrid approach of heuristic and genetic algorithms that greedily search the vehicle starting operation to solve the simultaneous vehicle and machine scheduling modules. Ahmadi-Javid and Hooshangi-Tabrizi (2015) developed an algorithm with analogy to anarchic society, and the authors applied this algorithm to JSSMH considering employee timetabling in a follow up study (Ahmadi-Javid & Hooshangi-Tabrizi, 2017). Zheng, Xiao, and Seo (2016) applied Tabu Search to the JSSMH problem. Baruwa and Piera (2016) proposed a Petri-nets based model formulation for JSSMH and reported good performance. They also reported detailed CPU time of the solution, which was lacking in the body of literature.

In this study, the model formulation for JSSMH problem is based on the model proposed by Bilge and Ulusoy (1995). We applied a linearization to the formulation with conditional constraints to replace the original nonlinear constraints so that the model can be solved with commercial solvers such as CPLEX, and we added a constraint to start timing as soon as the first job is taken out of the Loading/Unloading station (L/U). The results were used as a case study validation and for comparison. Optimization results based on the proposed algorithm is compared to existing solution techniques in literature, and the performance of the proposed model is justified by its high efficiency and good solution accuracy.

Besides, to explain the mechanism of the proposed algorithm, a new visualization method is adopted based on traditional Gantt charts to present the job schedule and AGV movement simultaneously, and we use it to explain how the proposed algorithm works with examples. The new visualization contains all the information in traditional vehicle-implemented Gantt charts in which vehicles are treated as additional machines; however, the routes and schedules of AGV fleet on the shop floor are explicitly presented. Optimization results based on the proposed algorithm is compared to existing solution techniques in literature, and the performance of the proposed model is justified by its high efficiency and good solution accuracy.

1.2.3. JSSMH with Variable Processing Time

Limited attention has been paid to the production scheduling problems that job processing time is variable, which has been reflected many production scenarios. As mentioned in some previous studies in

JSSMH, when human activity is involved in job processing, the job processing time can be affected by variability of human manipulation, such as random redundant motion or slowing down due to tiredness (Fink et al., 2014; Liu, Fan, Zhao, & Wang, 2017). Jobs themselves can have inherent variability in processing time too. For example, metal products' operation time can be influenced by a series of factors (Yang, Chen, Wei, & Chen, 2018), as well as industrial chemical processes (Bonfill, Espuna, & Puigjaner, 2005). There are two common types of variation reported in the body of literature, processing time in random distribution and deteriorating processing time. However, variable processing time has not been considered in job shop scheduling when material handling is part of decision making. With material handling system as an integral part of production, it is essential to take this into consideration when making production decisions.

Random processing time in production scheduling problems usually results from inaccurate data collection or uncontrollable operations. Sakawa and Kubota (2000) applied genetic algorithms to fuzzy programming for multi-objective job shop scheduling problems in which uncertain processing time and due date were introduced, and in the case study each operation had three possible realized processing times in triangular distribution. Bonfill, Espuna, & Puigjaner (2005) formulated a two-stage stochastic programming model based on job shop scheduling for chemical processes where reaction time is uniformed distributed. Such models were also described as Stochastic Job Shop Scheduling (SJSS) problems, while the material handling was not included and it was often assumed that operations could start immediately after completion of the previous operation. In reality, introducing material handling to the optimized solution of SJSS will make the problem more realistic; however, also much more complicated. Hence simulation has been commonly utilized when randomness exists in JSSMH (Xie & Allen, 2015). With a large number of experiments, simulation could help in developing heuristic shop floor management strategy (Wang et al., 2014). The strategy can also be flexible to implement operation mechanisms, such as behavior rules (Ng, Eheart, Cai, & Braden, 2011; Y. Zhang, Huang, Sun, & Yang, 2014) and optimization-based decision making (Almeder, Preusser, & Hartl, 2009; Sacone & Siri, 2009).

Deterioration reflects the phenomenon that job processing becomes longer as the production process goes. Deterioration was studied first by Gupta and Gupta (1988) in steel rolling mills. Following that, a variety of researchers studied deterioration in job shop scheduling problems in various production scenarios, such as single machine (Gawiejnowicz, Lee, Lin, & Wu, 2011), two-machine (W. C. Lee, Shiau, Chen, & Wu, 2010) and parallel machine based job shop scheduling (X. Huang, Wang, & Ji, 2014). Deterioration brought additional difficulty to optimally scheduling the jobs hence some heuristic solution techniques were also proposed (Kuo, Hsu, & Yang, 2012; Rustogi & Strusevich, 2012). In deteriorating job processing scenario, the processing time is, to a large degree, dependent on starting time of the operation, and researchers have reported multiple dependency relationships. The simplest case is that the processing time is linear to the operation start time (W. C. Lee et al., 2010), but it also common that processing time can be exponential to the processing sequence of jobs (X. Zhang, Wu, Lin, & Wu, 2018). In this study, both dependency relationships are discussed with corresponding model formulation of JSSMH.

The major contribution of this research can be summarized as follows. Firstly, we introduce variable processing time to the formulation of JSSMH. The model formulation has been derived to reflect real production practice, including the production scenario with random and deteriorating processing time. Secondly, we proposed the Stochastic Programming based JSSMH (SP-JSSMH) solution techniques to find the expected shortest makespan when job processing times are random, and solved the SP-JSSMH models with Pyomo. Thirdly, we propose a series of models for different dependency functions when deterioration exists, and the models are solvable with CPLEX including the formulation with linear dependency function and that with exponential dependency function but can be linearized by reformulating the model.

1.3 Dissertation Structure

The remainder of the dissertation is organized as follows.

Chapter 2 presents a few proposed AGV dispatching strategies in which the shop floor can be planned with a large number of jobs and potential uncertainties. The strategies are based on deterministic optimization of assignment problems in network optimization, and with these strategies, AGVs are assigned to work centers based on mathematical programming models minimizing the total waiting time of jobs in a decision horizon in which the status of vehicles as well as jobs in work centers can be predicted. The strategies are demonstrated in a case study based on a shop floor in literature and are compared with classic AGV assignment rules including random assignment and nearest vehicle/shortest travel time rule. The results show that hybrid strategies based on the proposed dispatching strategies and classic assignment rules outperform pure classic strategy in minimizing jobs' waiting time on the shop floor.

Chapter 3 presents an efficient algorithm to solve deterministic JSSMH. The proposed algorithm starts from an optimal solution under a large vehicle fleet, and iteratively reduces the number of available vehicles until the fleet size is equal to the original requirements. In each iteration, one vehicle is removed from the incumbent schedule, and remaining vehicles are reassigned to the transportation of operations according to a set of specially designed heuristic rules, all while the schedule is simultaneously adjusted due to vehicle reassignment. The algorithm stops when all operations are served and the AGV fleet size meets the job shop requirements. A quadratic optimization model is formulated to initialize the vehicle assignment. The algorithm is compared to existing solving methods in literature on optimized production makespan and solution efficiency based on the same data sets, and the results suggest that the proposed algorithm can achieve comparable solution quality on makespan with much higher efficiency.

Chapter 4 demonstrates the validity of considering variable processing time in optimization of JSSMH. A two-stage stochastic programming model is formulated to account for randomly distributed processing time, and two additional models are formulated for different deterioration scenarios. The models are validated with small job set examples, and the optimized shop floor makespans with solutions of proposed models are compared to the makespans with solutions of classic JSSMH excluding randomness or deterioration of processing time in modeling. The proposed models prove to be superior in performance with the realization of variable processing time.

The structure and relationship between the studies in this dissertation can be represented with Figure 1.1.

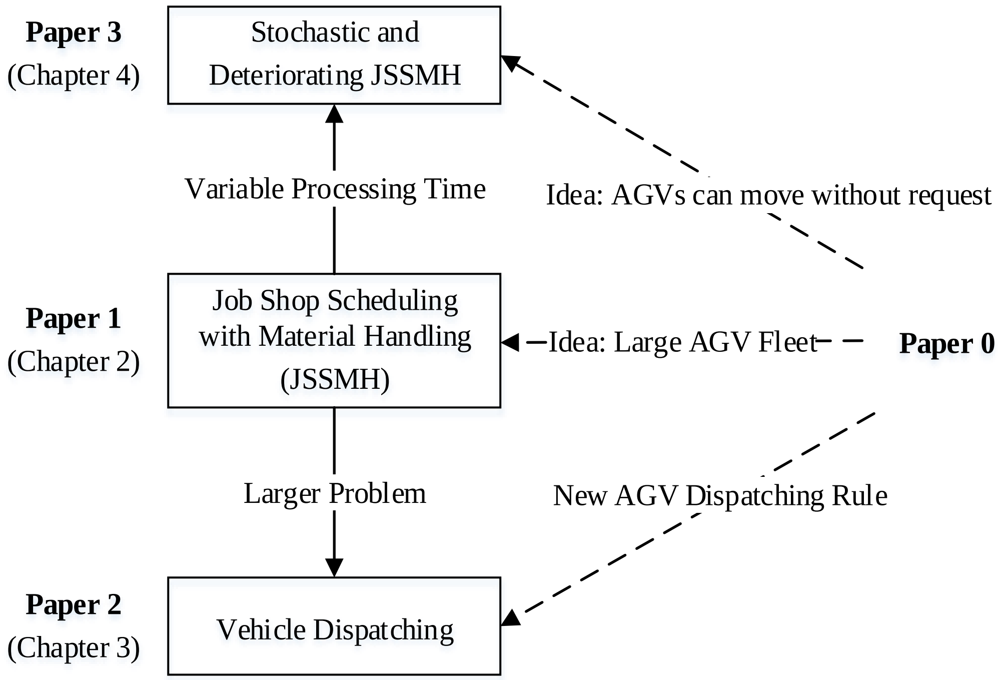


Figure 1.1: Research Structure

(S. Huang, Brown, & Hu, 2017; S. Huang & Hu, 2017a, 2017b, 2018)

Paper 0:

Huang, S., Brown, C., & Hu, G. (2017). Shop Floor AGV Assignment Optimization with Uncertain Request Arrival. In K. Coperich, E. Cudney, & H. Nembhard (Eds.), *Proceedings of the 2017 IIE Annual Conference*. Pittsburgh.

Paper 1:

Huang, S., & Hu, G. (2017b). Automated Guided Vehicle Dispatching Based on Network Optimization in Shop Floors. *International Journal of Planning and Scheduling, Under Review*.

Paper 2:

Huang, S., & Hu, G. (2017a). A Degressive Vehicle Fleet based Heuristic Algorithm for Job Shop Scheduling with Material Handling. *International Journal of Production Research, 2nd Round Review*.

Paper 3:

Huang, S., & Hu, G. (2018). Job Shop Scheduling with AGVs under Variable Processing Time, *In Progress*.

CHAPTER 2. AUTOMATED GUIDED VEHICLE DISPATCHING BASED ON NETWORK OPTIMIZATION ON SHOP FLOORS

The contents in this chapter is organized as follows: two optimization-based strategies are formulated and their application scenarios are discussed in Section 2.1 with two subsections separately. In Section 2.2, all AGV dispatching strategies are implemented in the simulation platform, and compared to each other in a case study summarized in Section 2.3. This chapter concludes with a summary of research findings and future works.

2.1 AGV Dispatching Based on Network Optimization

The complexity of AGV dispatching problems is mainly because of sequential decision making and the dependence of future decision making conditions and current decisions. The complexity increases when more shop floor components (work centers, vehicles, products etc.) are included, and classic request-by-request assignment rules are highly likely to be biased from global optimality. Assuming the i^{th} request R_i is described by $R_i=(w, p)$, meaning product p has finished processing in Work Center w, and travel time of AGV j for transporting request R_i is T_{ij}, the tree in Figure 2.1 of two sequential requests demonstrates the non-optimality of classic rule NV/STT, in which Solid arrows are real AGV assignment under NV/STT, while dash arrows are alternative assignments.

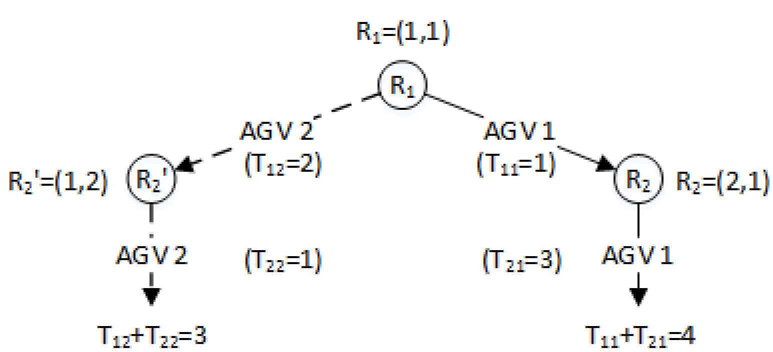


Figure 2.1 Two sequential requests and AGV assignment.

When request R_1 is generated, there are two AGVs that can be assigned, and different dispatching can lead to different time and place of the next request R_2 because of different product processing time, shop floor layout, AGV speed, etc. Assume that at the beginning, both AGVs 1 and 2 are in the same depot, and AGV 1 is known to be quicker than AGV 2 for the transportation of R_1 ($T_{11} < T_{12}$). Then AGV 1 is assigned under NV/STT and results in a new request R_2. AGV 1 also takes R_2 since it is the nearest vehicle and the associated travel time is T_{21}. Consequently, the total travel time of vehicles is (T_{11}+T_{21}). However, there is another combination of sequential AGV assignments which is marked with dash lines in Figure 1, and it leads to shorter total vehicle travel time, but such a strategy is not adopted by NV/STT since AGV 2 takes a longer time to transport R_1 than AGV 1. Classical AGV assignment rules excluding random assignment, like NV/STT adopted in this example, are not optimal because they take only one step searching the decision tree like Figure 2.1.

Thus, to search a dispatching solution consisting of sequential AGV assignments that is closer to global optimality, we should look further beyond a single current request, such that the problem can be formulated into mathematical programming models. However, the correlation between decision variables (dispatched AGVs) and parameters (dispatching decision making conditions) means that the model is highly likely to be nonlinear and difficult to solve. This is probably the reason why the adoption of classic AGV heuristic assignment rules have been the focus of shop floor AGV dispatching.

Although we cannot take too many future requests into consideration, considering more than one is still applicable because in automated shop floors, future statuses of work centers, products, and vehicles are predictable based on current status and operating parameters (Pinedo, 2009). Two strategies are proposed, and both of them consider more than one future request to shorten the material or product waiting time for transportation. The difference between decision horizons makes two formulations distinct; thus, solution techniques are different. The objective of both models is to minimize total waiting time for being loaded by a vehicle of all products. All notations for model formulation are included in Table 2.1.

Table 2.1 Notations of AGV dispatching models.

Sets	
N	Set of AGVs.
M	Set of requests in the AGV dispatching decision horizon.
W	Set of work centers.
Indices	
n	Index of an AGV, $n \in \{1,2,\dots,\lvert N\rvert\}$.
w	Index of a work center, $w \in \{1,2,\dots,\lvert W\rvert\}$.
i	i^{th} request in the optimized time horizon.
(n,i)	An assignment of AGV *n* to request *i*.
j	Index of arc assignment (n,i).
Parameters	
d_{nw}	Travel distance of AGV *n* to work center *w*.
$D_{w'w}$	Fixed distance between work center *w'* and *w*.
c_{ni}	Travel time of AGV *n* for request *i*.
e_{nw}	Waiting time of product at Work Center *w* for AGV *n*.
t_r	The r^{th} time point that AGVs' status is checked in the optimized time horizon.
v	AGV speed.
Decision variables	
x_{ni}	Binary variable. If the assignment "AGV *n* is assigned to request *i*" is adopted, $x_{ni} = 1$, otherwise $x_{ni} = 0$.

The whole production period can be divided into two periods with the time point that all products enter the shop floor and start waiting for the processing procedure. At the beginning of production period, initial products arrive on the shop floor randomly and stay in the initialization zone with unlimited capacity, hence requests for AGVs are uncertain before the arrivals finish. When all products enter the system, the randomness is eliminated, such that the succeeding transportation requests are predictable. In the first period, randomness is considered and requests are responded with classic AGV assignment rules.

In the second period, product status is predictable since processing time, vehicle speed, and vehicle routes are assumed to be fixed, therefore AGVs can be dispatched according to corresponding prediction.

2.1.1 2-request Optimization Assignment Strategy (OA2)

First, we consider one step further, i.e., we optimally dispatch AGVs for the current transportation request as well as the following request that is predictable. In the example of Figure 2.1, when request R_1 is generated, we can predict where and when the next request R_2 will be, by enumeration of AGV assignments to R_1. After that we can evaluate the outcome of assigning each AGV to corresponding request R_2 based on the assignment of AGV to R_1, and make the decision that is optimal to these two sequential requests. In real operations, such a process repeats every time a new request is generated.

We focus on two requests in a row rather than considering more sequential requests because of the complexity of enumeration brought by correlation between variables and parameters. The dependency can be demonstrated by a simple example in Figure 2.2.

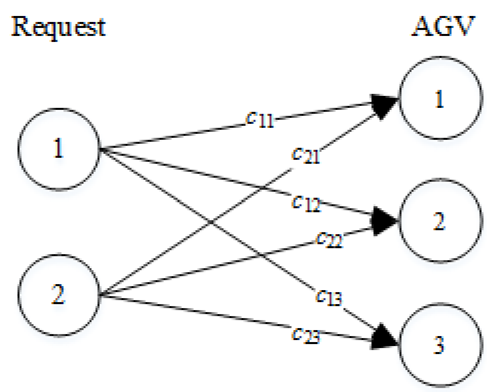


Figure 2.2 Assignment network of two sequential requests with three AGVs

There are three AGVs for two requests from two work centers. Arcs connecting requests and AGVs represent assignment of AGV. The arc weights c_{in} is the travel distance of AGV n for loading request i. The assignment is expected to minimize the total travel distance, and hence, the corresponding products' total waiting time is also minimized.

The difference between such an assignment problem and a common assignment problem is the unfixed arc weight c, and this difference is a reflection of correlation between decision variables and decision making conditions (parameters) in AGV dispatching problems. For instance, let binary x denote the assignments; then in assignment x_{11}=1 and x_{21}=1, AGV 1 is assigned first to request 1 then to request 2; and in assignment x_{12}=1 and x_{21}=1, AGV 2 is assigned to request 1 and AGV 1 is assigned to request 2 simultaneously. In these two assignments, AGV 1 travels different distances to request 2, which means c_{21} has two different values.

Such a dependency of parameters on variables for the AGV assignment optimization is quite difficult to describe by an explicit function due to nonlinear shop floor layout and timing. At any moment, we can capture the statuses (positions) of AGVs, but their distances to all other places at a certain time point after an assignment can only be described by an If-Then correspondence. For example, at time point t_1, AGV 1 is somewhere between Work Center 1 (WC1) and Work Center 2 (WC2), and its distance to WC1 is d_{11}. If AGV 1 is assigned to a work center at t_1, at time point t_2($t_2 > t_1$), AGV 1's distance to WC1 is formulated as Equation (2.1), which is correlated with the assignment at t_1.

$$c_{11} = \begin{cases} d_{11} - (t_2 - t_1)v & if\ AGV1\ is\ assigned\ to\ WC1 \\ d_{11} + (t_2 - t_1)v & if\ AGV1\ is\ assigned\ to\ WC2 \end{cases} \tag{2.1}$$

As a result, the model formulation would become very complicated if we model the problem into a pure linear programming model, in which extensive linearization is necessary for the conditional distance between AGVs and work centers. Enumeration should be the most efficient solution method if we only consider two requests in a row; however, if we consider more sequential requests, enumeration would take more time to reach the optimal solution. Consequently, we only consider two requests in a row in our optimization practice in this paper. For any AGV fleet size, we can model the situation into an assignment problem in network optimization (Bertsekas, 1998), like the generalized network in Figure 2.3.

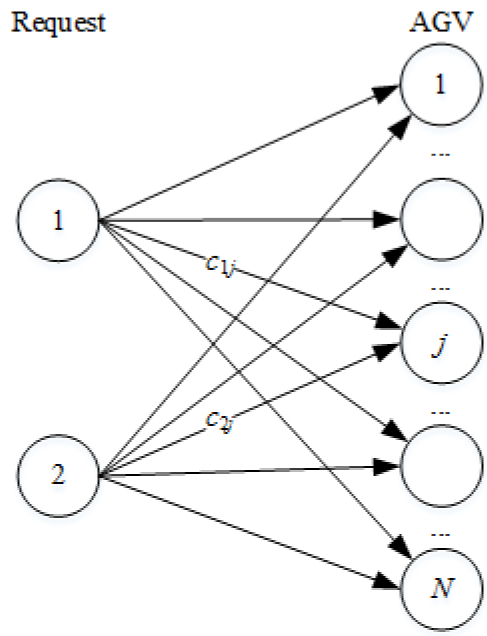


Figure 2.3 Assignment network 2 sequential requests and *N* AGVs

Equations (2.2) to (2.5) consist of a standard formulation of the assignment problem in Figure 2.3. We consider two requests in a row; therefore *i* equals to 1 or 2 in our case.

$$\min \sum_{(i,j)} c_{ij} x_{ij} \tag{2.2}$$

s.t.

$$\sum_{i} x_{ij} \geq 1 \quad \forall j \tag{2.3}$$

$$\sum_{j} x_{ij} = 1 \quad \forall i \tag{2.4}$$

$$c_{ij} = f\left(x_{ij}\right) \tag{2.5}$$

Equation (2.2) is the objective function minimizing the total waiting time of the two products. Constraint (2.3) and (2.4) ensure that at the decision making time point each AGV can be assigned to multiple work centers but each work center can only take one AGV. Equation (2.5) means the arc weights are dependent on decision variables with an implicit relationship.

Model represented by Equation (2.2) to (2.5) can be easily solved on simulation platforms by enumeration due to a limited number of variables and simple model formulation, and can be programmed in centralized AGV controlling systems or in each AGV by simple searching loops. Unlike classic AGV assignment rules, OA2 also enables assigning requests to AGVs that have not arrived at any work center, and new tasks are saved in an AGV's memory, such that once an AGV completes its current job, it can immediately start the next trip. Culler and Long (2016) developed similar systems with customized AGV. The operation mechanism of the AGV system proposed in this paper is further introduced in Section 2.4.

2.1.2 All-work-center Optimization Assignment Strategy (OAW)

Besides assigning AGVs for current requests generated by products ready for transportation, for products in processing, AGVs can be assigned for future requests. If AGVs can be assigned without requests generated by ready products, some ready products might fail to request an AGV with immediate response since all AGVs are on the way to other work centers. We still define request $R_i=(w, p)$ which is from Work Center *w* by Product *p*, and example in Table 2.2 explains how such an "ignorance" happens. There are three AGVs and three work centers on the shop floor. At time $t_1=0$, request (1,1) is observed, while Product 2 is in Work Center 2, and Product 3 is in Work Center 3. Since processing times are fixed, it can be predicted that request (2,2) will be ready at time $t_3=2$, and request (3,3) will be ready at time $t_4=3$. AGVs are assigned for all these three requests, with different travel times according to each product's processing route. It can be observed in Table 3 that before any of the AGVs arrive at their next destination, a new request (1,4) is generated at $t_2=1.5$; however, since all AGVs are busy, this is "ignored" until an AGV becomes idle.

Table 2.2 Considered certain requests with unconsidered requests in between

Time	Request	Planned AGV assignment	AGV travel time to next destination
0	(1,1)	AGV 1	2
1.5	(1,4)	No AGV is assigned	-
2	(2,2)	AGV 2	2.5
3	(3,3)	AGV 3	2.5

As a result, when we observe that all work centers are busy, we can optimize the AGV assignment for these certain requests, and temporarily "ignore" the requests generated by products entering work centers after current assignments. The ignored requests will be responded to after optimized transportations are completed. Compared to responding with assignment of one AGV until single requests are generated, assigning AGVs to a group of potential requests is expected to reduce the total waiting time of most products, although some products might experience longer waiting time. Different processing and transportation time on the shop floor lead to different consequence of adopting such an AGV assignment strategy. Intuitively, quicker transportation and slower processing can take more advantage of this strategy, while slower transportation and quicker processing would lead to more "ignorance" and finally enlarge the total product waiting time.

In this strategy, the dispatching is determined by an optimization model, and the optimization-based assignment initiates when all work centers are detected to be busy for the first time. If work centers can process multiple products simultaneously, the optimization is for products that are getting ready as the earliest at each work center. The dispatching and transportation order is executed strictly according to the optimization result until the last optimized transportation starts. Before that, if a new transportation request is generated, AGVs are assigned according to classic assignment rules when the vehicles become idle. When all optimized transportation is completed, the optimization process repeats.

The optimization model in the OAW strategy actually solves the assignment problem in Figure 2.4, in which arc weight e_{nw} equals the waiting time of the product at Work Center w if the corresponding vehicle n is assigned to it. It should be noted that since in AGVs are can be assigned without existing requests, the nodes no longer represent requests and AGVs like in Figure 2.3 but AGVs and Work Centers.

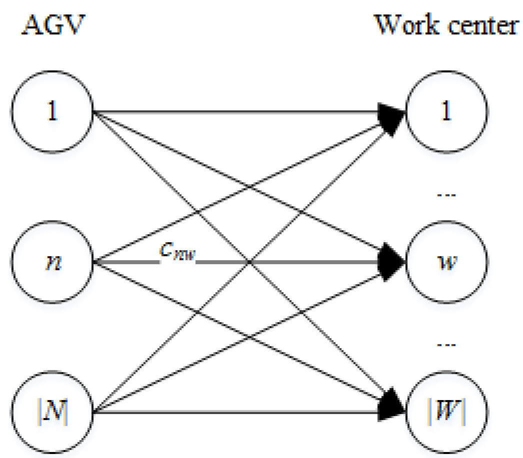


Figure 2.4 Network of assignment of OAW

Definition of link weights in Figure 2.4 relies on accurate record of agents' real-time status. Therefore, to implement this strategy in an AGV system, the remaining time of a work center *w* having one job ready for pickup t_w^r, and remaining time of AGV *n* becoming idle t_i^R should be monitored and recorded. In modern shop floors, this information can be easily collected, hence link weights c_{nw} in Figure 2.4 can be calculated by Equation (2.6).

$$c_{nw} = \begin{cases} (t_n^r - t_w^R) + \dfrac{d_{nw}}{v} & if\ t_w^R \le t_n^r \\ max\left\{0, \dfrac{d_{nw}}{v} - (t_w^R - t_n^r)\right\} & if\ t_w^R > t_n^r \end{cases} \tag{2.6}$$

For any possible assignment of AGV *n* to Work Center *w*, a vehicle and a job always become ready earlier than another; hence, Equation (2.6) differentiates the two cases. If the processing of job in Work Center *w* finishes after AGV *n* becoming idle ($t_w^R \ge t_n^r$), the waiting time of this job is the summation of the time difference and AGV's travel time. If AGV *n* becoming idle happens earlier ($t_w^R < t_n^r$), the waiting time is the travel time of AGV's remaining trip to the work center, or 0 if the AGV has arrived and waited at the work center.

With link weights calculated with Equation (2.6), Equations (2.7) to (2.9) can be formulated as a typical linear integer programing model of assignment problem in network optimization.

$$\min \sum_{(n,i)} c_{nw} x_{nw} \tag{2.7}$$

s.t.

$$\sum_{n} x_{nw} = 1 \quad \forall w \tag{2.8}$$

$$\sum_{w} x_{nw} = 1 \quad \forall n \tag{2.9}$$

Equation (2.7) is the objective function minimizing the total waiting time of jobs in the decision horizon. Equation (2.8) and (2.9) are the constraints that ensure in one optimization only one AGV can be assigned to each work center and each AGV can only have one destination.

Models (2.7) to (2.9) on shop floor scale can be quickly solved by commercial solvers like CPLEX. The operation mechanism of the AGV system in practice and simulation is further introduced in Section 2.4.

2.2 Architecture of Shop Floor Simulation for AGV Dispatching

A simulation model for a shop floor is constructed based on data from Egbelu (1987) in AnyLogic, shown in Figure 2.5. The shop floor operates one 8-hour shift per day with eight work centers on the shop floor, and five types of jobs are produced. Each type of job has unique processing routes and processing times at each work center. Table 6 includes the job types and processing routes.

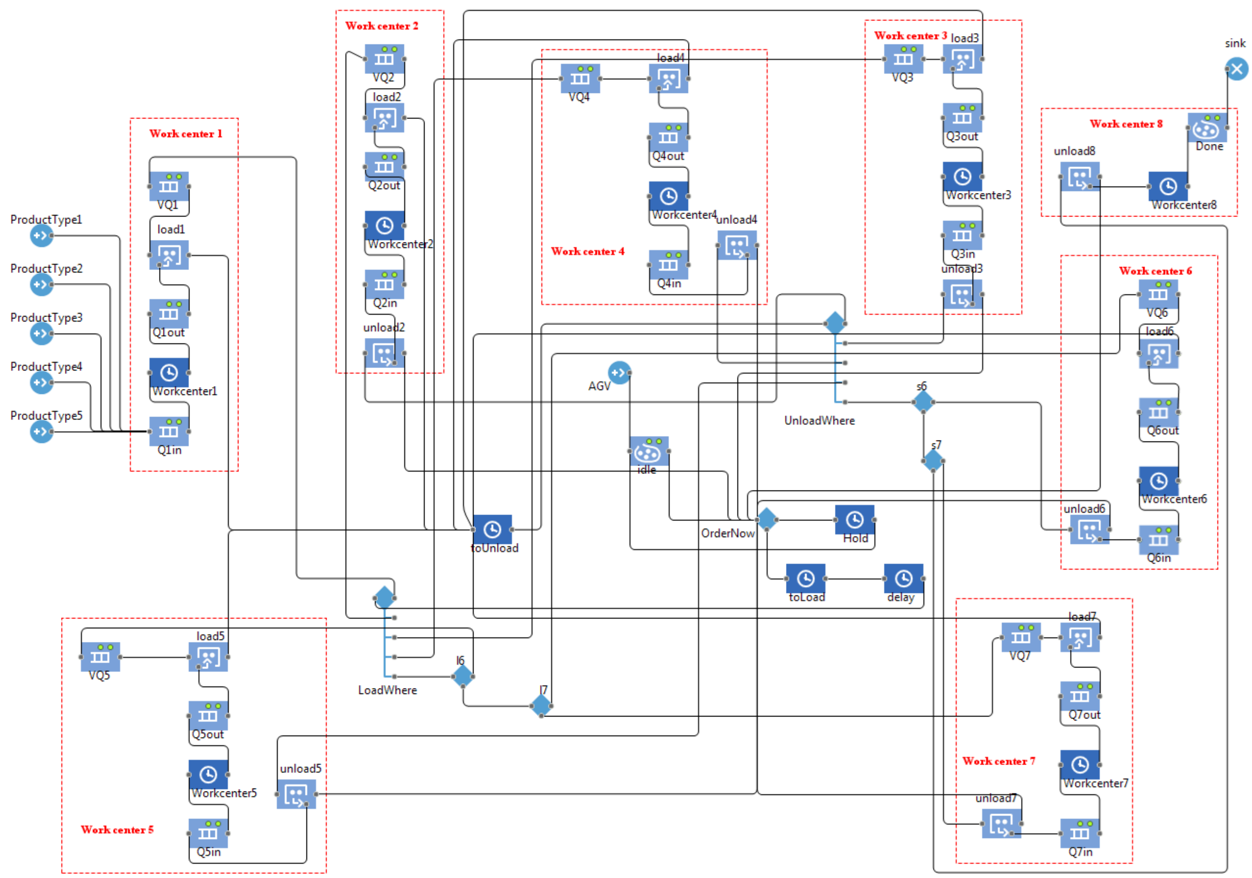


Figure 2.5 Simulation model of shop floor in AnyLogic

All products must go through Work Center 1 at the beginning and never come back, and this means unloading does not happen at this work center. Moreover, products finish all processing at Work Center 8, but the processing time at this work center is always 0. Besides the core processing machine, Work Centers 2 to 7 consist of AGV loading and unloading ports with corresponding queues, and a queue for AGVs that arrive earlier than product ready for transportation. There is no product transported by AGVs out of Work Center 8; therefore, there is no AGV queuing area at Work Center 8, either.

At the beginning, all AGVs are kept at Work Center 1, which serves as the depot of vehicles. When products are ready at the loading port of work centers, transportation requests are generated. Destination of an AGV with loaded product is determined by the product type, and once the product is unloaded, the AGV decide whether to stop and stay idle at the current work center, or go to another work center to load additional products. If there is a transportation task assigned to it by optimization during its last trip and saved in its memory, it will go to the corresponding work center for product loading. If multiple tasks are saved in the memory, the AGV will follow a first-come-first-serve rule to decide the next destination.

Table 2.3 Attributes of jobs on shop floor

Job type	Processing route	Processing time per unit load (T/minutes)
1	1,3,2,5,8	1.0, 5.0, 10.0, 7.0, 0.0
2	1,6,5,4,7,8	1.0, 8.0, 5.0, 10.0, 7.0, 0.0
3	1,4,6,8	1.0, 9.0, 9.0, 0.0
4	1,7,2,3,8	1.0, 10.0, 5.0, 10.0, 7.0, 0.0
5	1,2,6,3,5,7,4,8	1.0, 8.0, 7.0, 9.0, 10.0, 8.0, 5.0, 0.0

The processing time for all products at each work center are assumed to be fixed values, and we make this the basis of our AGV dispatching optimization, since only with fixed processing time, the statuses of products and vehicles are predictable.

In reality, the processing time is not always a fixed value, but it is quite likely to be a random distribution. We take the fixed processing time as an assumption to formulate the models; however, in the case study we relaxed this assumption by replacing the fixed processing time T in Table 2.3 with a uniform distribution $U[T-1,T+1]$ to make the scenario closer to reality. Good performance of the proposed models on uncertain processing time is a proof of robustness to production uncertainty.

Figure 2.6 and Figure 2.7 demonstrate how OA2 and OAW strategies work on the shop floor. At the beginning, AGVs are dispatched by RV/STT, and the optimization based dispatching strategies are not activated until all jobs enter the shop floor and randomness from job arrivals are eliminated. When OA2 and OAW are activated, models are called repeatedly and solved with solution enumeration or commercial solvers, and solutions are transformed into transportation tasks distributed to corresponding AGVs.

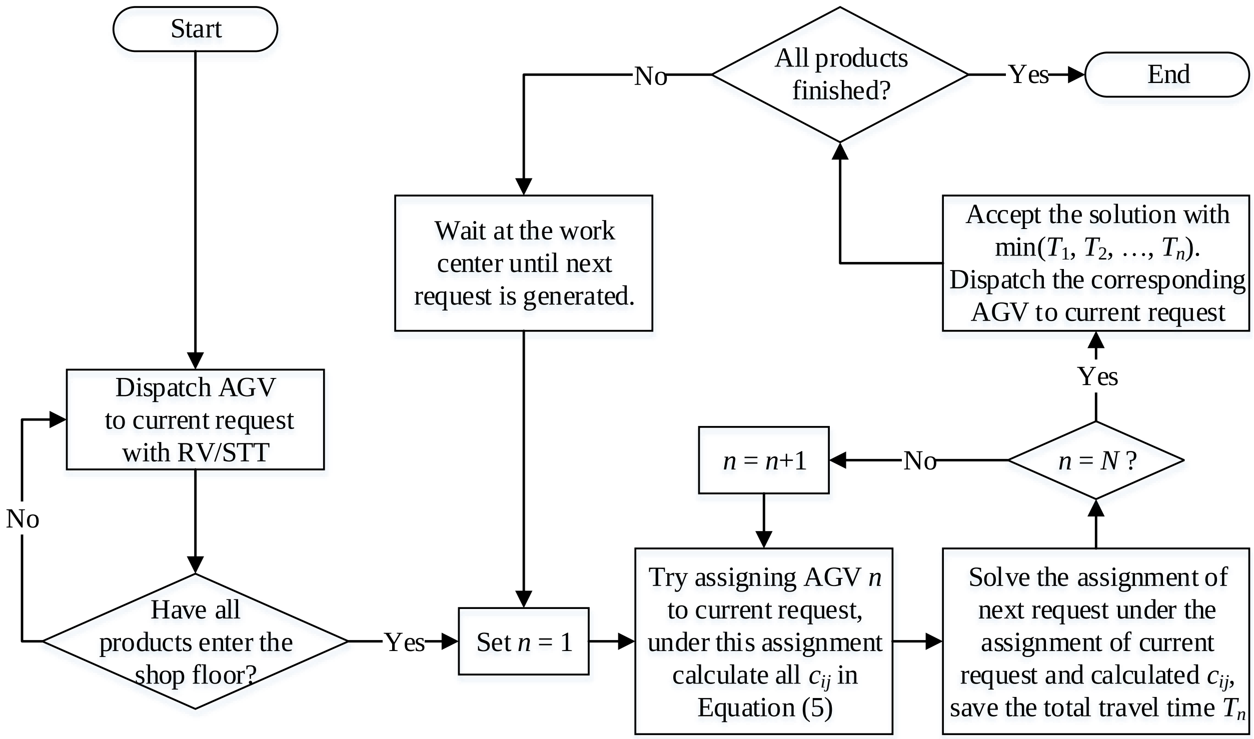


Figure 2.6 OA2 mechanism on shop floor

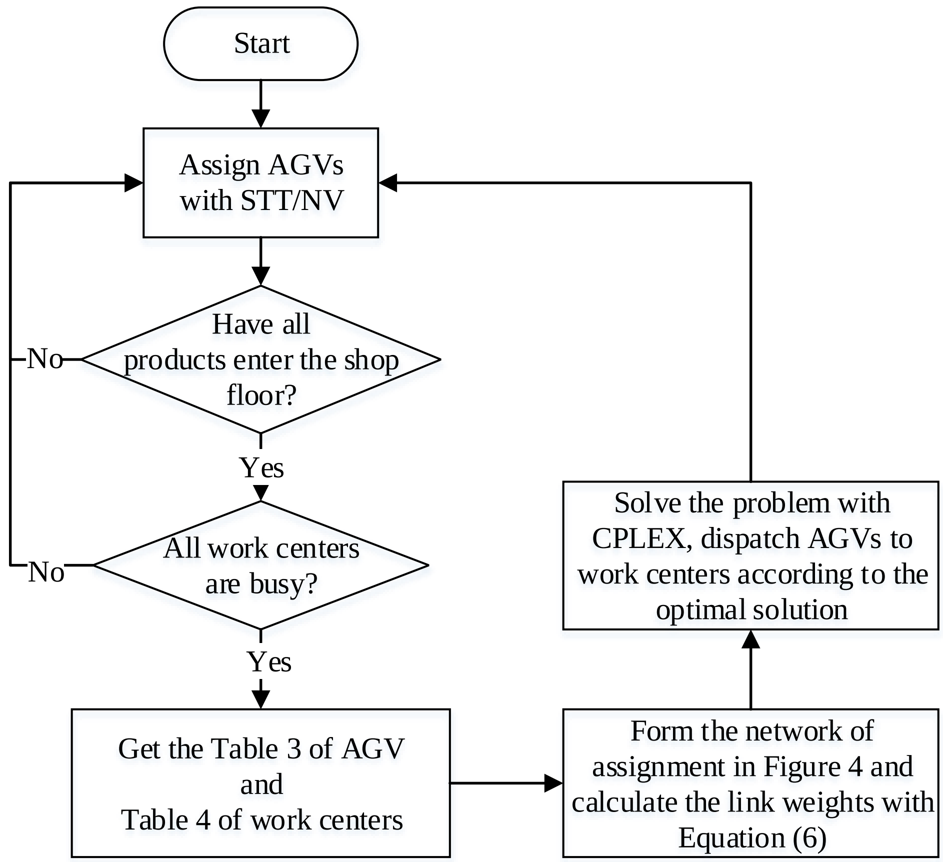


Figure 2.7 OAW mechanism on shop floor

In Egbelu (1987), the optimal AGV fleet sizes are calculated with different AGV assignment rules, and all of the combinations of fleet size and assignment rules should complete all jobs in 8 hours. Thirteen AGVs can complete all jobs on time with the RV/RW rule and nine AGVs complete all jobs on time with NV/STT. Simulation experiments are carried out in our model, and resulting makespans show that with thirteen AGVs and the RV/RW strategy adopted, all jobs are completed in approximately 8 hours, as well as with nine AGVs and the NV/STT strategy. There is only limited data for validation, but the consistency of makespans proves that the simulation model of the shop floor is a good replication of the reality, and with this model, AGV strategies can be compared in the case study.

2.3 Case Study Result

A case study is carried out for the simulation model described in Section 2.4 to evaluate the optimization models described in Section 2.3. All AGV dispatching strategies, including OA2, OAW, and classic AGV assignment rules RV/RW and NV/STT, are implemented and compared. For each given AGV fleet size, all strategies are tested with 20 replication simulation experiments, and the makespan in each experiment and waiting time of each job are recorded. Figure 2.8 and Figure 2.9 show how average makespans and jobs' waiting times fluctuate with AGV fleet size changing, and the fluctuations reflect characteristics of different AGV dispatching strategies, which can be used to evaluate their performances on the shop floor.

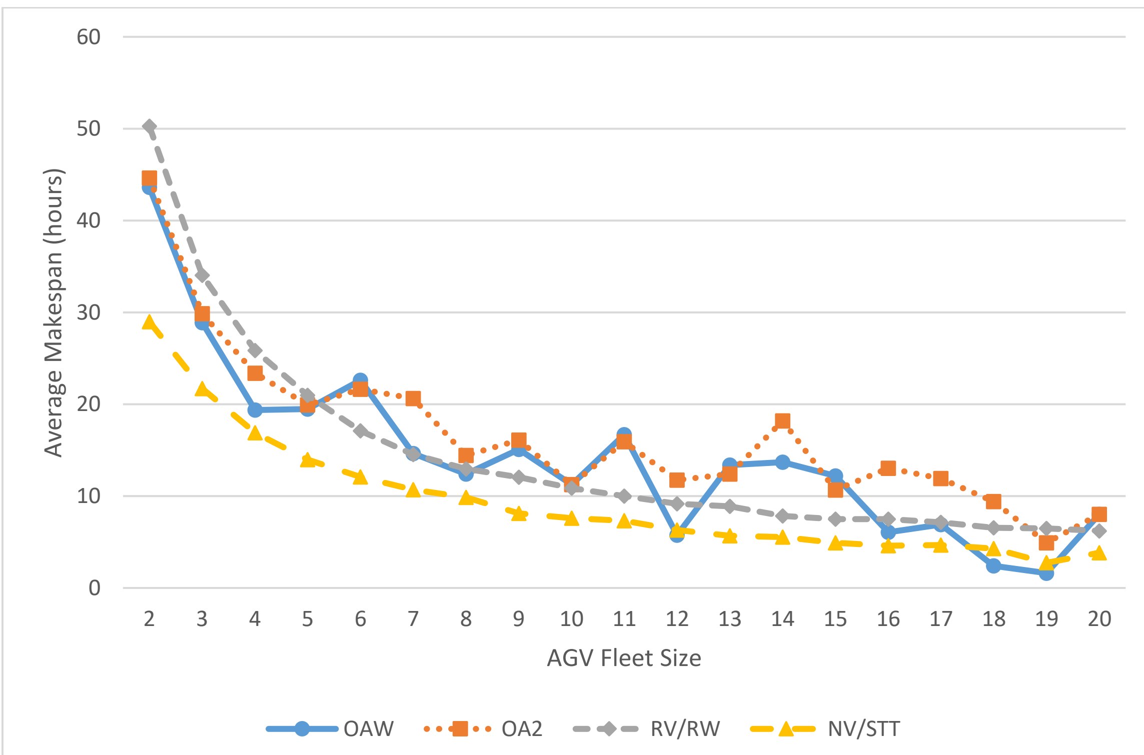


Figure 2.8 Shop floor makespan of all AGV dispatching strategies

Except for rare cases, the NV/STT strategy always leads to shortest makespan, but when the AGV fleet size grows, the makespan under other AGV assignment strategies get close to makespan under NV/STT. This can be partly explained by the definition of makespan, which is finish time of the last product. When there are only limited number of products on the shop floor, more AGVs are likely to be idle compared to busy production period, hence NV/STT rule can maximally reduce the waiting time of these products since there are more choices. On the other hand, in the entire production horizon, impact of long waiting time of products in busy production period is not reflected in the makespan because long waiting time can be made up by following transportation.

For most realistic shop floors, where minimizing makespan is usually the management objective, other AGV dispatching strategies may not be attractive; however, if some other criteria are valued on shop floors, the situation becomes different.

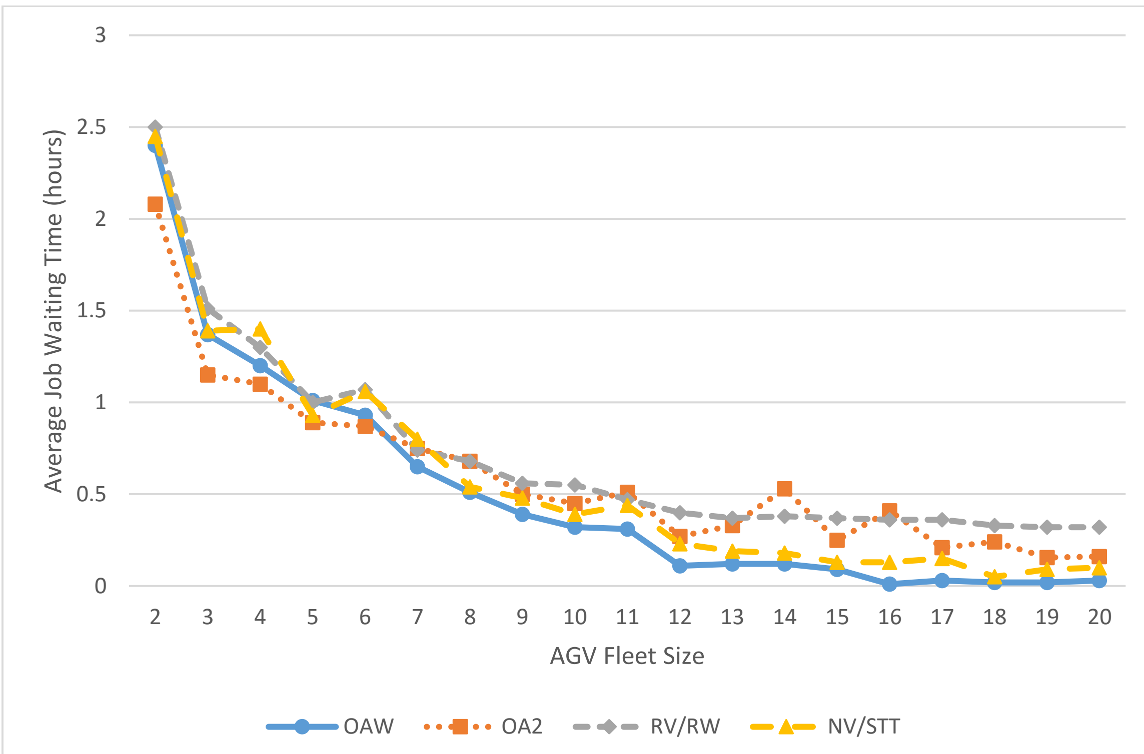


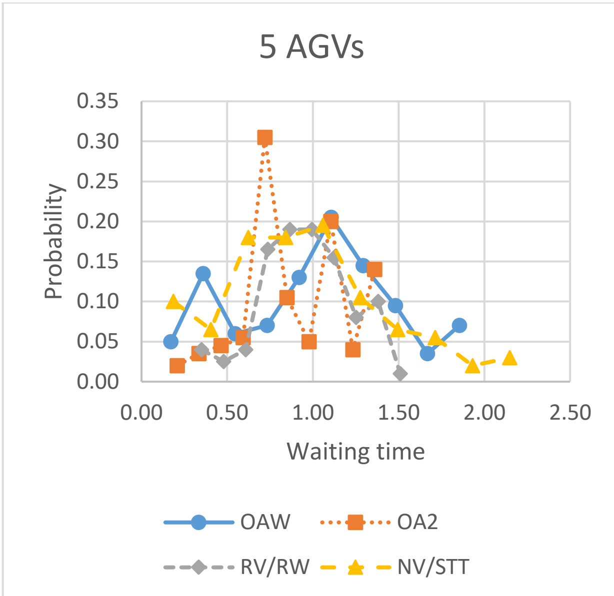
Figure 2.9 Jobs' average waiting time of all AGV dispatching strategies

From Figure 2.9, it can be observed that AGV dispatching strategies OA2 and OAW based on network optimization shorten the products' waiting time in different scenarios, respectively. Relatively speaking, with a large number of transportation requests on the shop floor, the waiting times that proposed strategies can save is quite significant. Figure 2.9 leads to an empirical conclusion that the threshold of an AGV fleet size differentiating the validity of OA2 and OAW lies approximately at the number of work centers with both loading and unloading port.

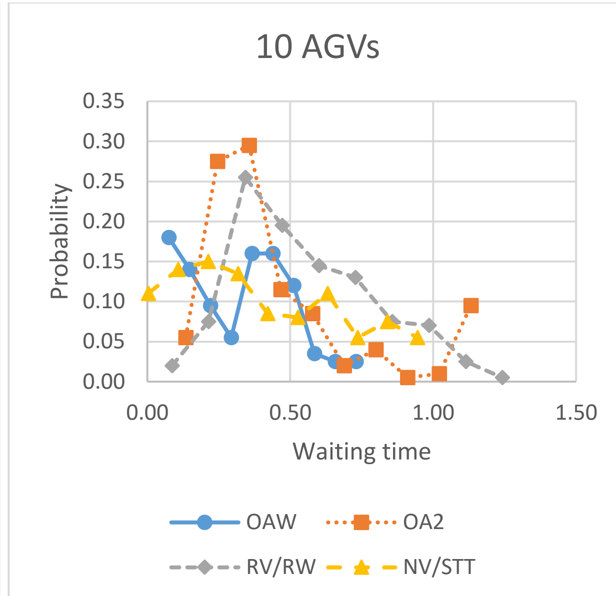
When an AGV fleet is small, OA2 leads to shortest average waiting time of products, but its performance becomes worse when the AGV fleet size grows. This is foreseeable since OA2 only focus on two transportation requests that are the closest to the current time point of decision making, and all possible dispatching are enumerated. The growing fleet size means more complicated future scenario and larger bias from global optimality by OA2.

For large AGV fleet sizes, OAW is the best among all strategies on controlling product waiting time and the trend is quite stable. The theoretic evidence is that although the optimization in OAW still cannot guarantee global optimality, it reaches the local optimality in a moderate-length period. It better utilizes the growing feasible solution set when AGV fleet size increase compared to other AGV dispatching strategies. We can also observe that OAW is never the worst among all strategies under all AGV fleet sizes.

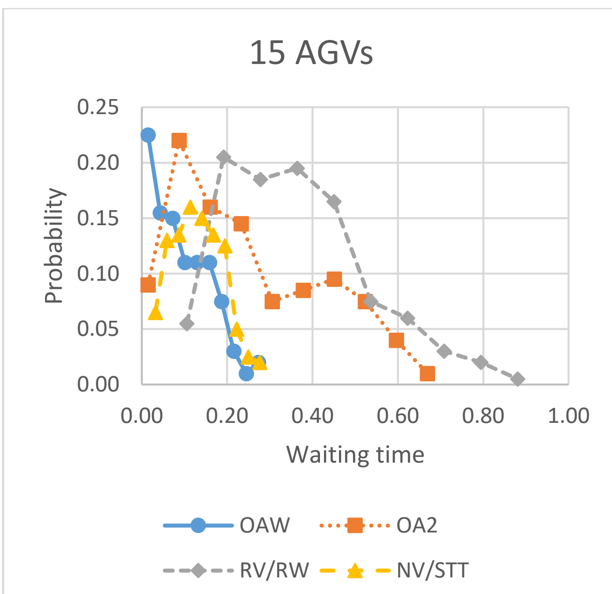
By observing the products' waiting time distribution under different AGV dispatching strategies in Figure 2.10, we can summarize more positive characteristics of the proposed strategies, and they are extremely important when some special management objectives are pursued on the shop floor, such as keeping all products' waiting times under a tolerable threshold, etc.



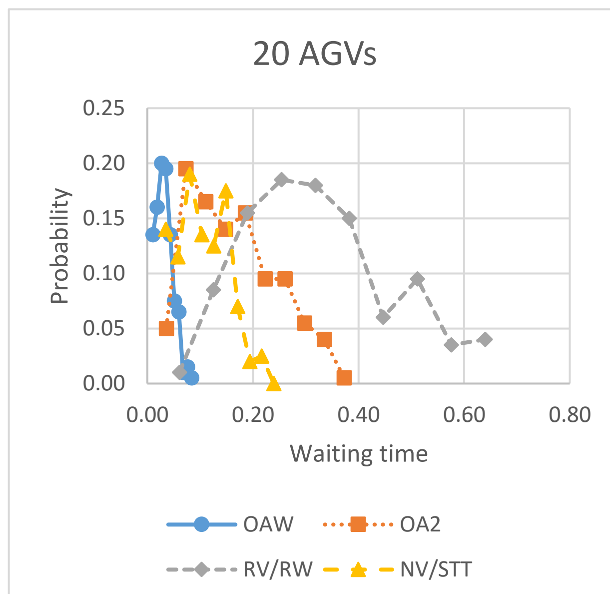
(a)



(b)



(c)



(d)

Figure 2.10 Waiting time distribution under all AGV dispatching strategies

In Figure 2.10 (a), OA2 under small AGV fleet size is superior to other strategies according to its shortest longest waiting time of products and high probability of short waiting time. Such a superiority of OA2 is less significant when AGV fleet size increases but OAW shows its advantage. In Figure 2.10 (b), (c), and (d), OAW has the shortest longest waiting time and aggregating short waiting time in all AGV fleet size scenarios, and the more AGVs there are, the more superior OAW is for the given shop floor. Theoretically speaking, classic AGV assignment rules including RV/RW and NV/STT can never eliminate the possibility that certain products beyond their one-step decision making horizon wait extremely long, especially for shop floors with large number of products and work centers; however, the proposed strategies avoid this scenario to a large degree.

Consequently, we can conclude that if the primary objective of the shop floor in this case study is controlling the products' waiting time, OAW and OA2 strategies can be considered instead of the commonly adopted RV/RW and NV/STT strategies. This is especially true for shop floors like what is in this case study, where processing times in work centers are fixed or quite stable, and minimizing products' waiting time for transportation also means minimizing products' total time spent in the production system.

2.4 Conclusion

In this paper, two AGV dispatching strategies based on network optimization of assignment problems are developed for shop floors. Classic AGV assignment rules make decisions for each single request, while the basic idea of our optimization based AGV dispatching strategies considers one more step further than classic intuitive AGV assignment rules, such that the system can be more efficient. The two strategies have different dispatching decision horizons, and the case study results show that the two strategies also have different performance in minimizing a product's waiting time for transportation with various AGV fleet sizes. In practice, if a shop floor has a small sized AGV fleet (empirically this means the number of AGVs is fewer than number of work centers), adopting an OA2 strategy will shorten the products' waiting time, while for shop floors with a large AGV fleet (empirically this means number of AGVs is larger than the number of work centers), OAW can save more waiting time of products. Minimizing waiting time of products for transportation is significant for products such as heated steel and frozen food that cannot be exposed to room temperature or natural environments for too long.

If OA2 and OAW are implemented on shop floors, one technique characteristic must be paid enough attention for useful application. There cannot be too many sources of randomness in the system, especially in vehicle traveling, product processing, and job arrivals. If vehicle traveling time or product processing time are not fixed values, they should be limited in a narrow interval. This is one of the major assumptions of this paper, and without this, the optimization models can lead to significant bias on dispatching solution efficiency, which might be even worse than random assignments. For job arrivals, there are two conditions that must be met to successfully implement OA2 and OAW strategies. First, all jobs enter the system and get started shortly after production begins. If the first condition is not met, there must be a long delay between pairs of entering jobs such that in this time interval, statuses of agents in the shop floor are predictable. With these two conditions, the AGV dispatching strategies based on deterministic optimization in this paper are valid, therefore they can be regarded as the limitation of the work so far, but still adoptable in applications if the conditions are met and the production scenario asks for short job waiting time.

CHAPTER 3. A VEHICLE REDUCING ALGORITHM FOR JOB SHOP SCHEDULING WITH MATERIAL HANDLING

This chapter is organized as follows: the mathematical formulation for the JSSMH problem and an example of visualization of simultaneous job and vehicle schedule are described in Section 3.1. In Section 3.2, the proposed algorithm of this study is introduced and presented with an example. In Section 3.3, computational experiments are carried out to validate the proposed algorithm, and the optimization results are compared to existing solution techniques in the body of literature. The chapter concludes with a summary of research findings.

3.1 Model Formulation for Job Shop Scheduling with Material Handling

The JSSMH problem addressed in this study can be described as following: on a shop floor, a set of jobs J is processed on a set of machines, and each machine can only process one job at a time. Each job j has a unique processing route consisting of a set of operations I_j to complete the manufacturing process, and for each operation i, a fixed time p_i is required. A fleet of AGVs is available on the shop floor to handle jobs at the L/U or after the completion of each operation at the machine. A fixed loaded travel time t_i is incurred for each job before the start of next operation i. If one AGV takes operation h and i successively, the deadheading trip takes another fixed period τ_{hi}. The objective is to achieve the shortest makespan which is defined by completion time of the last operation on the shop floor.

The JSSMH problem can be formulated as a linear programming model based on Bilge and Ulusoy (1995). In the formulation there is not any specific subscript representing jobs for variables and parameters because all operations are sequentially indexed. There are no subscripts representing AGVs either because the routes of AGVs are represented by distinct visiting sequences.

Table 3.1 and 3.2 include all necessary notations in modeling of JSSMH, and a linearized model of JSSMH is formulated with Equation (3.1) to (3.16).

Table 3.1: Notations of sets and parameters

J	Set of jobs.
n_j	Number of operations of job *j*.
N_j	Total number of operations of the jobs indexed before *j*.
n	Total number of operations of all jobs. $n = \sum_{j \in J} n_j$.
I	Index set of all operations. $I = \{1,2, \dots, n\}$.
I_j	Set of operations associated with job *j*.
$\bar{I}_i$	Index set of operations excluding operation *i* and succeeding operations of the same job.
$\underline{I}_h$	Index set of operations excluding operation *h* and preceding operations of the same job.
K	AGV fleet size.
p_i	Processing time of operation *i*.
t_i	Travel time to loaded trip heading for operation *i*.
τ_{hi}	Travel time of deadheading trip from machine of operation *h* to machine of operation *i*.

Table 3.2: Notations of variables

Z	Job shop makespan.
c_i	Completion time of operation *i*.
T_i	Completion time of loaded trip for operation *i*.
q_{rs}	Binary variable. $q_{rs} = 1$, if $c_r < c_s, r \neq s$
x_{hi}	Binary variable. $x_{hi} = 1$, if a vehicle is assigned for deadheading trip from operation h to i.
x_{oi}	Binary variable. $x_{oi} = 1$, if a vehicle starts from L/U to operation *i* as its first trip.
x_{ho}	Binary variable. $x_{ho} = 1$, if a vehicle returns to L/U from operation h as its last trip.
D_{jih}	Auxiliary variable for time between AGV handling of operation *i* and *h* that both belong to job *j*.
S_{jh}	Auxiliary variable for time between AGV handling of operation *h* and the first operation of job *j*.
st_i	Auxiliary variable for start time of operation *i*.

A mixed integer programming (MILP) model is formulated for the JSSMH with Equations (3.1) to (3.16) as the following. The optimal solution will include the routes of AGVs, the job processing sequences, and operations completion time.

$$\min Z \tag{3.1}$$

subject to:

$$Z \geq c_{N_j+n_j} \qquad \forall\, j \in J \tag{3.2}$$

$$c_i - c_{i-1} \geq p_i + t_i \qquad \forall\, i, i-1 \in I_j, j \in J \tag{3.3}$$

$$c_{N_j+1} \geq p_{N_j+1} + t_{N_j+1} \qquad \forall\, j \in J \tag{3.4}$$

$$\begin{cases} (1 + H\tau_{rs})c_r \geq c_s + p_r - Hq_{rs} \\ (1 + H\tau_{rs})c_s \geq c_r + p_s - H(1 - q_{rs}) \end{cases} \qquad \forall\, r \in I_j, s \in I_k, j, k \in J, j \neq k \tag{3.5}$$

$$x_{oi} + \sum_{h \in \bar{I}_i} x_{hi} = 1 \qquad \forall\, i \in I \tag{3.6}$$

$$x_{ho} + \sum_{i \in \underline{I}_h} x_{hi} = 1 \qquad \forall\, h \in I \tag{3.7}$$

$$\sum_{i \in I} x_{oi} \leq K \tag{3.8}$$

$$\sum_{i \in I} x_{oi} - \sum_{h \in I} x_{ho} = 0 \tag{3.9}$$

$$T_i \leq c_i - p_i \qquad \forall\, i \in I \tag{3.10}$$

$$T_i - t_i \geq c_{i-1} \qquad \forall\, i, i-1 \in I_j, j \in J \tag{3.11}$$

$$D_{jih} = T_h + \tau_{h,i-1} \; if \; x_{h,i} = 1 \qquad \forall\, i, i-1 \in I_j, h \in \bar{I}_i \; j \in J \tag{3.12a}$$

$$D_{jih} = 0 \; if \; x_{h,i} = 0 \qquad \forall\, i, i-1 \in I_j, h \in \bar{I}_i \; j \in J \tag{3.12b}$$

$$T_i - t_i \geq x_{oi}\tau_{o,i-1} + \sum_{h \in \bar{I}_i} D_{jih} \qquad \forall\, i, i-1 \in I_j, j \in J \tag{3.12c}$$

$$S_{jh} = T_h + \tau_{ho} \; if \; x_{h,N_j+1} = 1 \qquad \forall\, h \in \bar{I}_{N_j+1}, j \in J \tag{3.13a}$$

$$S_{jh} = 0 \; if \; x_{h,N_j+1} = 0 \qquad \forall\, h \in \bar{I}_{N_j+1}, j \in J \tag{3.13b}$$

$$T_{N_j+1} - t_{N_j+1} \geq \sum_{h \in \bar{I}_{N_j+1}} S_{jh} \qquad \forall\, j \in J \tag{3.13c}$$

$$x_{oi}T_{\mathrm{i}} = x_{oi}\tau_{oi} \qquad \forall\, i \in I \tag{3.14}$$

$$x, q \in \{0,1\} \tag{3.15}$$

$$T, c, Z > 0 \tag{3.16}$$

Equations (3.1) to (3.5) are based on a typical Job Shop Scheduling (JSP) model (Pinedo, 2009), while an additional parameter t_i is used to consider necessary transportation time of a job from one machine to another for a pair of consecutive operations. When jobs finish their last operation, they are immediately removed from the machine, and AGVs do not handle them back to L/U, hence the makespan is defined as the finish time of the last operation on the shop floor in Equation (3.2). Binary variable x represents the routes of AGVs, which indicates the sequential relationship of each operation. Equations (3.6) and (3.7) regulate the strict one-by-one following relationship between each pair of operations. Equation (3.8) defines that the number of AGV routes is limited by AGV fleet size. Equation (3.9) ensures that for each AGV, there must be a starting trip as well as an ending trip. Equation (3.10) means the operation must begin after the job arrival to the machine. Note that Equation (3.10) is not an equation because it is possible that in an optimal schedule, an early-arriving job waits at the machine until another job whose operation arrives later to start first. The operation sequence of one job is ensured in Equation (3.11). Equations (3.12) and (3.13) are linearized conditional constraints to replace the nonlinear constraints by Bilge and Ulusoy (1995), which indicate the impact of previous trips on the next trip of each AGV. Equation (3.14) is used to start timing when a vehicle leaves the L/U with the first job it conveys, and such a constraint means a default initial condition that AGVs are at the L/U until they leave for the first job handling task. Sometimes the trip of vehicles between L/U and machines is not considered (Khayat, Langevin, & Riopel, 2006); however, we decide to include these trips in the optimization thus reflecting the production reality (Y. J. Xiao, Zheng, & Jia, 2014).

The scheduling model defined in Equations (3.1) to (3.16) can be solved by commercial solvers to get the optimal schedule for small sized problem. However, it either takes a long time or becomes computationally intractable when the problem size increases, which is why an efficient solution technique is necessary.

3.2 A Heuristic Algorithm Based on Degressive Vehicle Fleet for JSSMH

The job shop planning configuration is based on the case study in Bilge and Ulusoy (1995), which were also used by Abdelmaguid et al. (2004) Khayat, Langevin, and Riopel (2006), Umar et al. (2015), Zheng, Xiao, and Seo (2016) and Ahmadi-Javid and Hooshangi-Tabrizi (2017) for model formulation and algorithm validation. Table 3.3 and 3.4 include Layout 1 and Job Set 1 as an example.

Table 3.3 Layout 1

	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

Table 3.4 Job Set 1

	1	2	3
Job 1 (J1):	1.M1(8)	2.M2(16)	3.M4(12)
Job 2 (J2):	4.M1(20)	5.M3(10)	6.M2(18)
Job 3 (J3):	7.M3(12)	8.M4(8)	9.M1(15)
Job 4 (J4):	10.M4(14)	11.M2(18)	-
Job 5 (J5):	12.M3(10)	13.M1(15)	-

In Layout 1, there are 4 machines and 1 Loading/Unloading station on the shop floor. Each job is initially at L/U, and each job must follow the production sequence defined in Table 3.4 with corresponding processing times in the parenthesis. For example, M1(8) means the job is processed by M1, and the processing time is 8 minutes including loading, processing and unloading. The items in Table 3.4 are indexed to keep consistent with modeling notation in the formulation defined in Section 3.1.

3.2.1 Proposed Visualization of Job and Vehicle Scheduling

In this section, we begin with a new visualization method for job and vehicle scheduling to explain the mechanism of the proposed algorithm. In the existing body of literature, the activity of vehicles for material handling on shop floors is presented by treating them as machines. Additional timelines are added for vehicles and time blocks are marked with job names and travel types (Abdelmaguid et al., 2004; Baruwa & Piera, 2016), which is good to present the vehicle schedules but the presentation of vehicle routes relies on text markers. The impact of vehicle movement on the job scheduling cannot be easily read from the schedule, hence modifying the vehicle routing and observing the outcome is inconvenient. The proposed method improves the visualization of vehicle scheduling and routing, with Gantt chart implemented with arrows representing vehicle routes.

The scheduling of the job set in Table 3.4 on shop floor represented by Table 3.3 with 2 AGVs is solved on NEOS server by CPLEX, and we present the result in Figure 3.1.

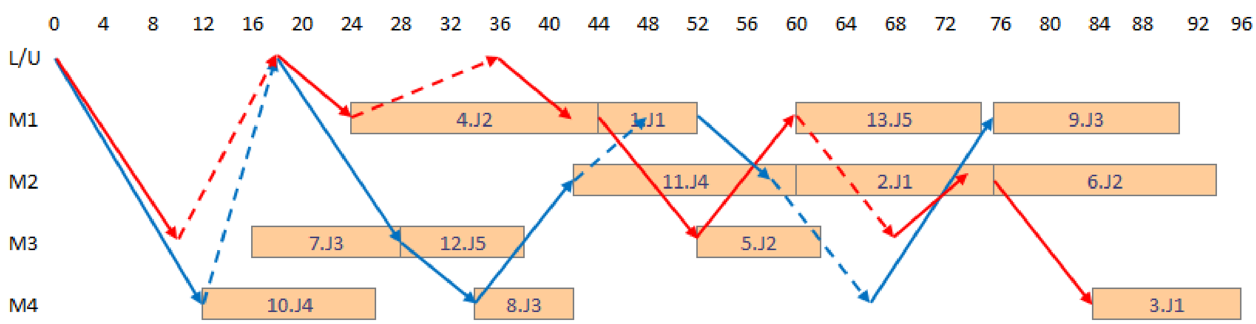


Figure 3.1: An example of schedule of Job Set 1 and AGV route in Layout 1 (2 AGVs)

In Figure 3.1, each operation is marked with its index and job name, and unlike existing literatures, we add arrows on the Gantt chart of jobs to represent the movement of vehicles so that the interaction of jobs, machines, and vehicles can be observed simultaneously. In this example, arrows in different colors represent different AGVs. Solid arrows are for loaded trips and dashed arrows are for deadheading trips. AGVs do not stop in the middle of a path, hence all arrows in Figure1 start and end at machines. Note that initially AGVs are all standby at the L/U.

The length of an arrow does not reflect the travel time of AGVs, but its projection on the time axis does. If arrows for a single AGV are always connected, it means the next trip starts immediately when last one finishes. If interruptions happen between arrows, the vehicle waits at the current machine until the next trip starts. Less and shorter interruptions in the schedule usually indicate a higher vehicle utilization. Vehicle utilization can be measured by many criteria (Beamon, 1998), and in this research, the utilization U_a of a single AGV a is evaluated by Equation (3.17) with makespan Z and traveling time TT.

$$U_a = \frac{TT_a^L + TT_a^D}{Z} \qquad \forall\, a \in A \tag{3.17}$$

In Equation (3.17), A is the set of available AGVs. TT_a^L and TT_a^D stand for the traveling time of a loaded and a deadheading trip respectively for AGV a. With a given makespan Z, the relative utilization of AGVs can be directly compared with total traveling time.

In Figure 3.1, heads and tails of solid arrows are always connected with the starting and finishing point of an operation of the same job, because a loaded AGV cannot change the transported job in the middle of its trip. If an arrow adheres with operations, it means the corresponding AGV does not wait for either loading or unloading at a machine. For example, for the Blue AGV handling Operation 7 and 8 for Job J3 between M3 and M4, it picks up or drops off the job as soon as it arrives M3 and M4, respectively. For an example of AGV or job waiting, the solid arrow of the second to the last trip of Red AGV is for handling of Job J2, while it does not adhere to either Operation 5 or 6. This means when Operation 5 finishes, the assigned Red AGV has not arrived. When the corresponding Job J2 is conveyed to M2 from M3, the machine is occupied by Job J1, and does not finish until 2 minutes after J2's arrival, hence Operation 6 does not start until Operation 2 finishes.

With the AGV route embedded job schedule visualization, we can discuss the proposed algorithm to solve JSSMH.

3.2.2 Degressive Vehicle Fleet Algorithm (DVFA)

For two AGV fleets in similar fleet sizes, with the scheduling of one AGV fleet, the scheduling of the other can be found quickly by adjusting the assignments of operations to AGVs. This heuristic is adopted in the proposed DVFA.

Usually the target fleet size is much less than the number of jobs. In DVFA, we start from a feasible solution with an AGV fleet in a size same as the number of jobs, in which the feasible solution can be derived by assigning one AGV to the operations of one job. AGV fleet size is iteratively reduced until the targeted size is reached. In each iteration, the operations need efficient reassignment to vehicles, and the makespan increasing due to degressive AGV fleet and consequent operation reassignment should be controlled. In initialization, the AGV fleet size is equal to job set size, such that a “theoretic optimal schedule” (TOS) can be acquired by letting each AGV uniquely follows a job in its entire production horizon on the shop floor. In TOS, makespan is equal to the solution from just solving Equations (3.1) to (3.5) as a job shop scheduling problem with the additional parameter of considering necessary transportation time. Figure 3.2 shows such a TOS solution of Job Set 1 on Shop Floor Layout 1, in which Red AGV follows J1, Blue AGV follows J2, Green AGV follows J3, Purple AGV follows J4, and Golden AGV follows J5.

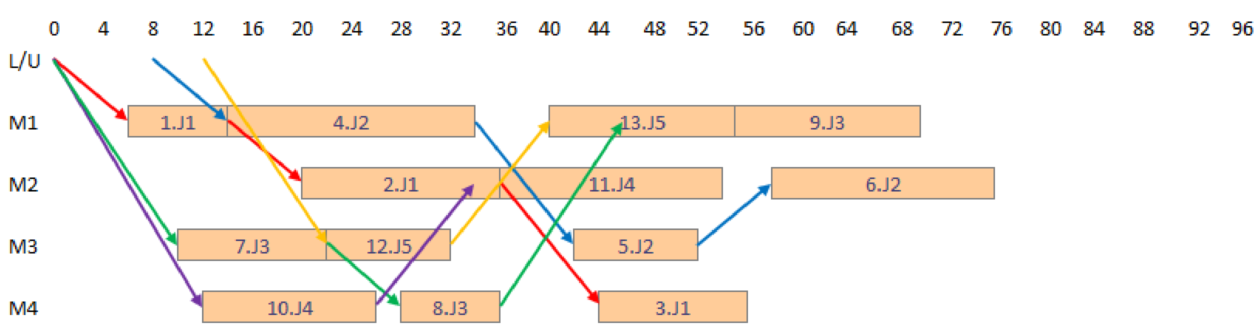


Figure 3.2 Example of schedule of Job Set 1 and AGV route in Layout 1 (5 AGVs)

Beginning with TOS, we can reduce the AGV number and reassign operations. Figure 3.3 introduces a general framework of the heuristic algorithm designed for JSSMH based on reducing the AGV number iteratively.

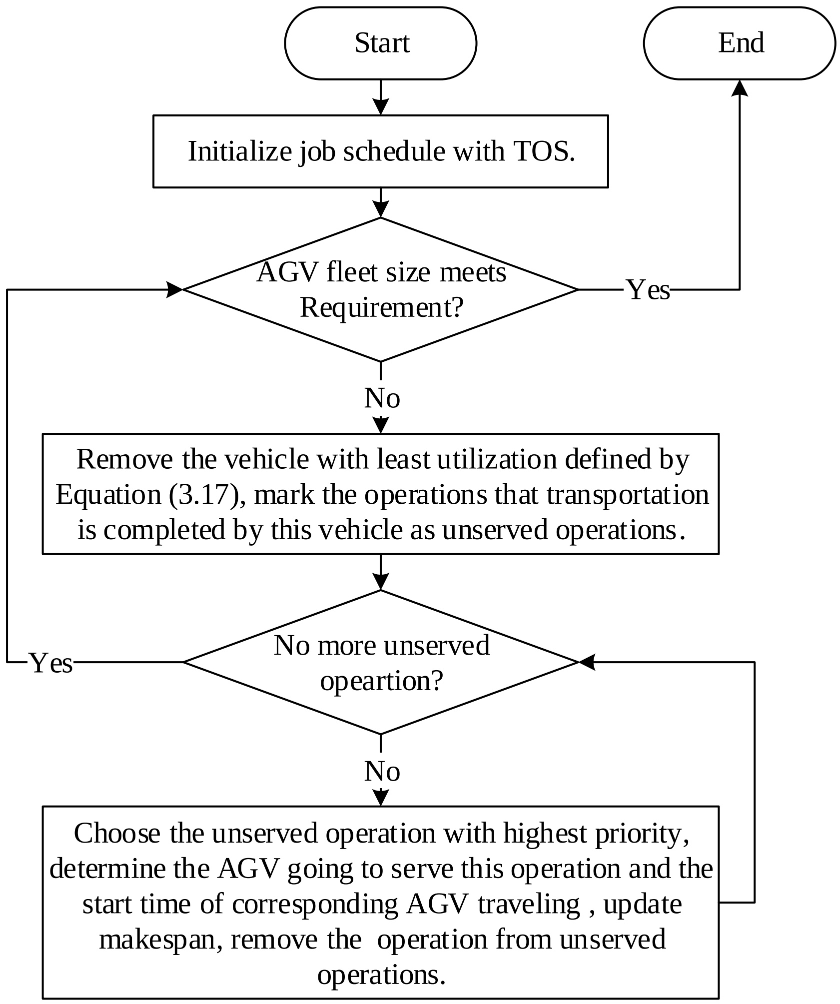


Figure 3.3 General framework of Degressive Vehicle Fleet Algorithm

Generally speaking, the proposed DVFA tries to adjust the schedule for unserved operations, while keeping served operations on time. In other words, the algorithm approaches to an optimal scheduling solution, and ensure the feasibility of incumbent scheduling solutions.

The detailed steps of the proposed DVFA are presented below:

Step 0: Initialize the scheduling priority of each operation i as $Prio_i$., $Prio_i = +\infty, \forall i \in I$. Get targeted AGV fleet size as A_0, solve TOS and get the minimum AGV fleet size $|A|$ that satisfies TOS. A is the set of available AGV. Go to Step 1.

(Solve the pure job scheduling problem in Equation (3.1) to (3.5) as a relaxation of JSSMH. This is the optimal solution with $K = |J|$, that the AGV fleet size equals to job set size.)

Step 1: For current AGV fleet, calculate vehicle utilization U_a with Equation (3.17). Define set *Unserv* with operations taken by AGV a_r, where $a_r = \underset{a}{\operatorname{argmin}}\, U_a$. Set $Prio_i = 0, \forall i \in Unserv$. Get current shop floor makespan Z, remove a_r from A. Go to Step 2.

(Reset parameter K and make constraints in Equation (3.6) unsatisfied. For operation i in *Unserv*, $x_{oi} + \sum_{h \in \bar{I}_i} x_{hi} = 0$)

Step 2: For each operation $i \in Unserv$, if $i \in I_j$, $i + 1 \in I_j$, and $i + 1 \notin Unserv$, set $Prio_i = 1$. Sort operations in *Unserv* according to operation start time. Get rank of sorted operation i as $Rank_i$, set $Prio_i = Prio_i + Rank_i$. Go to Step 3.

(Operation i in *Unserv* are assigned priorities to satisfy the constraint in Equation (3.6). Due to the constraint in Equation (3.11), to minimize the impact of completion time of previous unserved operation i on following served operation (i+1) of the same job, operation i has higher priority. Unserved operations with earlier start time also deserve higher priority to minimize the impact on following operations.)

Step 3: Find the Operation i_0 that $i_0 = \underset{i \in Unserv}{\operatorname{argmin}}\, Prio_i$. Get the operation start time st_{i_0}. For each vehicle $a \in A$, get the completion time T_{i_a} of a travel for operation i taken by vehicle a that is closest to st_{i_0} as well as the time $T_{i_0}^a$ that AGV a completes transporting i_0 if starting from T_{i_a} (i.e. $x_{i_a i_0} = 1$ in the optimization model), and thus $T_{i_0}^a = T_{i_a} + \tau_{i_a, i_0} + t_{i_0}$. Notate completion time of the travel for previous scheduled operation i_a' that is right after operation i_a as T_{i_0}'. If it was now after transportation of i_0 by vehicle a, the expected transportation completion time would be $T_{i_a'}^a = T_{i_0}^a + \tau_{i_0, i_a'} + t_{i_a'}$. Go to Step 4.

(Following constraints in Equation (3.12) and (3.13), calculate the arrival time of operation *i* with highest priority if AGV a is assigned.)

Step 4 (operation delaying): Notate the operation right after i_0 on the same machine as i_0^M

- If $\exists a \in A$ that $T_{i'_a}^a \leq T'_{i_0}$ and $T_{i'_a}^a + p_{i_0} \leq st_{i_0^M}$, assign a to i_0 at T_{i_a}, followed by original schedule, remove i_0 from $Unserv$ and go to Step 6.

- If $\exists a \in A$ that $T_{i'_a}^a \leq T'_{i_0}$ and $st_{i_0^M} \leq T_{i'_a}^a + p_{i_0} \leq st_{i_0^M} + \frac{1}{2} p_{i_0^M}$, assign a to i_0 at T_{i_a}, update start time of operation i_0^M: $st_{i_0^M} \leftarrow T_{i'_a}^a + p_{i_0}$; remove i_0 from $Unserv$ and go to Step 6.

- If $\exists a \in A$ that $T_{i'_a}^a \leq T'_{i_0}$ but $T_{i'_a}^a + p_{i_0} \geq st_{i_0^M} + \frac{1}{2} p_{i_0^M}$, or $\nexists a \in A$ that $T_{i'_a}^a \leq T'_{i_0}$, assign a to i_0 at T_{i_a} and go to Step 5.

- If $\nexists a \in A$ that $T_{i'_a}^a \leq T'_{i_0}$, assign a to i_0 at T_{i_a}, go to Step 5.

(An AGV is assigned to the operation *i* with highest priority to satisfy the constraint in Equation (3.6), which has the least impact of following operations. The delaying of starting operations caused by this AGV reassignment is calculated based on constraints in Equation (3.10) and (3.11).)

Step 5 (operation swapping): Notate the operation right after i_0 on the same machine as i_0^M

- $T_{i_0^M} \leftarrow st_{i_0} - p_{i_0^M}$, and $st_{i_0^M} \leftarrow T_{i_0^M}$; add i_0^M into $Unserv$ and set $Prio_{i_0^M} = 0$. Let the vehicle serving i_0^M skip this mission including the loaded and deadheading trip. Go to Step 6.

(An AGV is assigned to the operation *i* with highest priority to satisfy the constraint in Equation (3.6), which has the least impact of following operations by swapping operations on the same machine, keeping the constraint in Equation (3.5) satisfied. The delaying of starting operations is calculated based on constraints in Equation (3.10) and (3.11).)

Step 6: If $Unserv$ is empty, return current job schedule and vehicle assignments and go to Step 7; otherwise go to Step 2.

(One of the unsatisfied constraints for operations in Equation (3.6) is now satisfied, with corresponding constraints for AGVs in Equation (3.12) and (3.13) satisfied.)

Step 7: If $|A| > A_0$, go to Step 1; otherwise stop.

(Check if parameter K is reset to meet the requirement of AGV fleet size.)

A few principles should be emphasized to ensure the algorithm validity and efficiency. First, each operation can be marked as unserved only once at the most. This allows the algorithm to speed up and prevents it from entering an endless loop. Second, if multiple operations have the same priority in Step 3, or if multiple vehicles meet the condition, the break-even rules are adopted with the rank in Table 3.5 and 3.6. If one rule cannot break even, then move down to next rule.

Table 3.5 Break-even rules for selecting operations with same priority

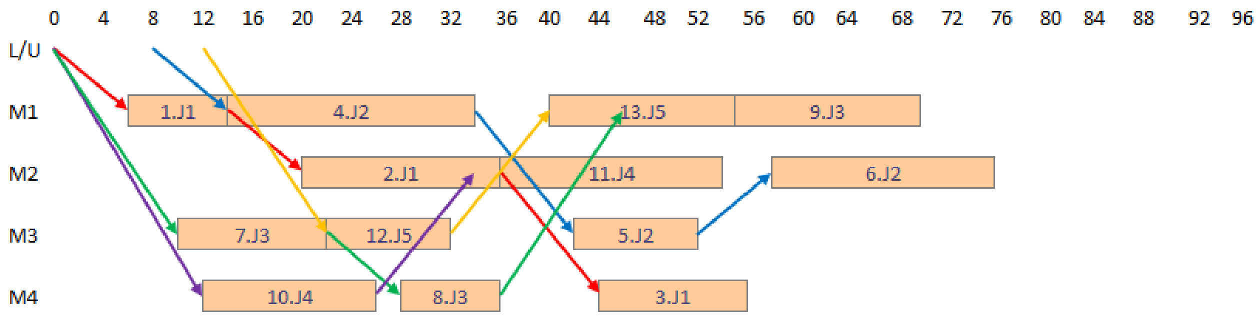
Rank	Rule
1	Select the operation closer to first operation of a job.
2	Select the operation of a job with less operations.
3	Select the operation with smaller index.

Table 3.6 Break-even rules for selecting vehicles meeting same condition

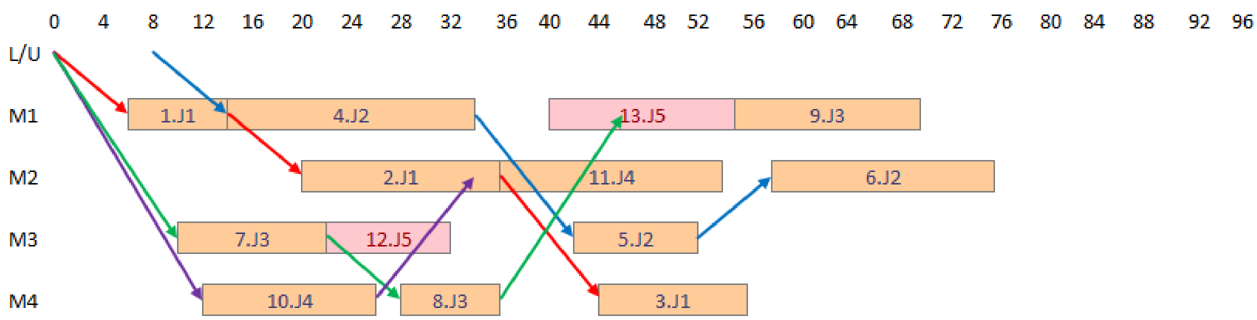
Rank	Rule
1	Select the vehicle that can arrive early for the newly assigned operation.
2	Select the vehicle with less utilization.
3	Select the vehicle serving less operations
4	Select the vehicle with smaller index.

3.2.3 Example of Solving JSSMH with DVFA

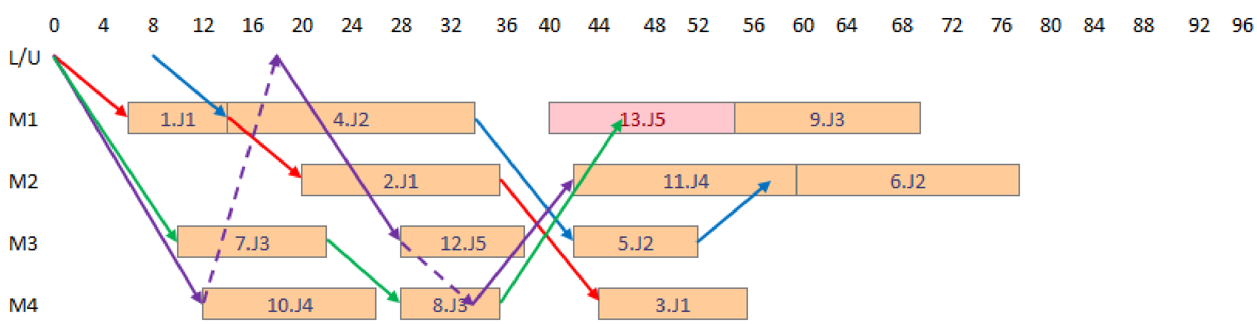
In this section we present an example of applying DVFA to the JSSMH problem with a schedule in Figure 3.1, in which the targeted AGV fleet size is 2. Figure 3.4 (a) to (e) has shown how the AGV fleet size is reduced to 4 from 5 step by step with DVFA, and Figure 3.5 (a) and (b) include the optimized job schedule with 3 and 2 AGVs. The algorithm is expected to result in a schedule similar to Figure 3.1 in Figure 3.5 (b) in terms of minimized makespan



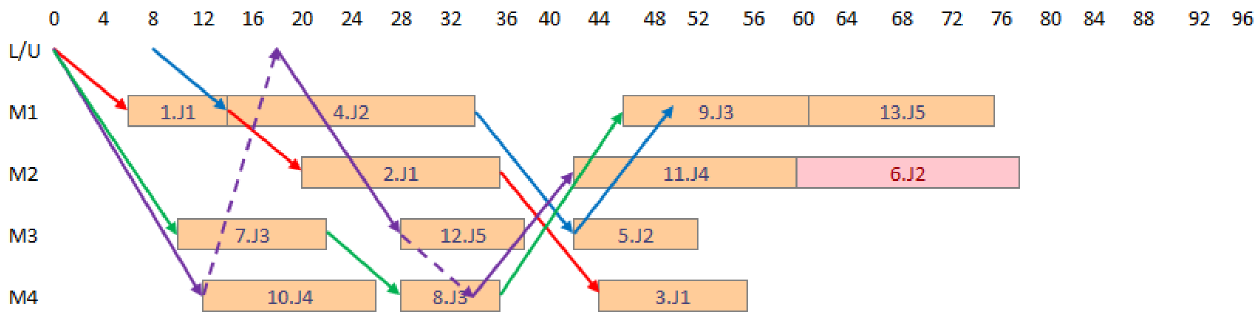
(a) TOS with 5 AGVs following jobs.



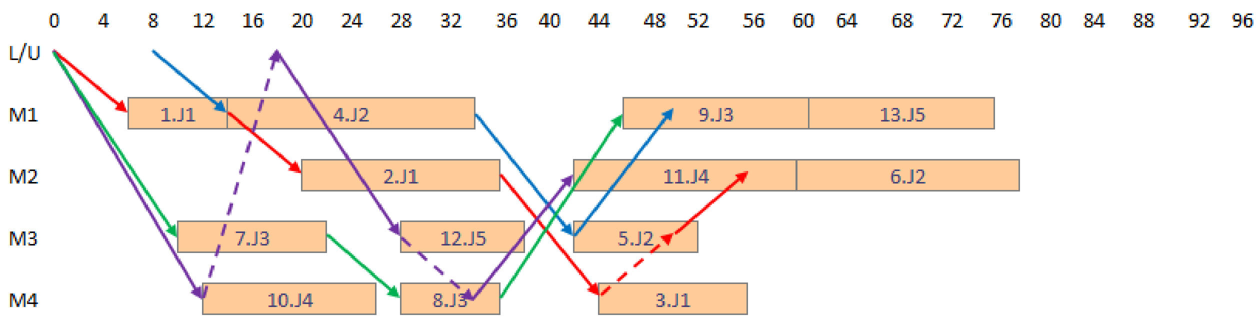
(b) Golden AGV removed.



(c) Purple AGV serves Operation 12 instead of Golden AGV.



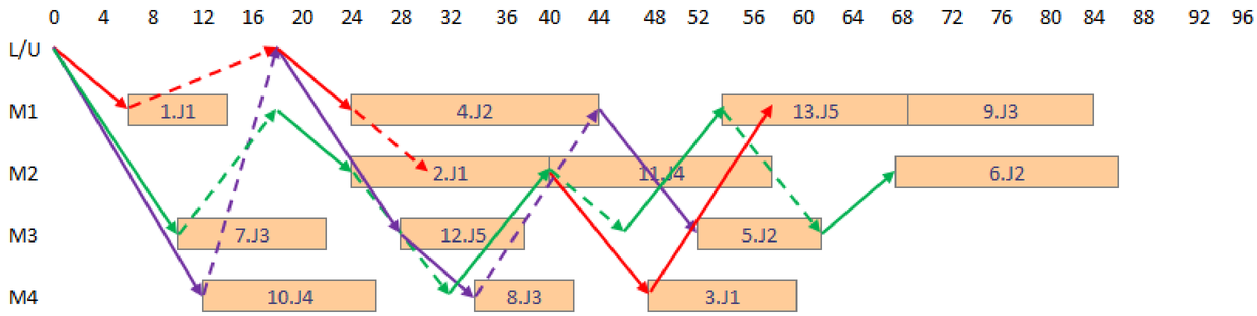
(d) Operation 9 and 13 are swapped, Blue AGV takes Operation 13, Operation 6 becomes unserved.



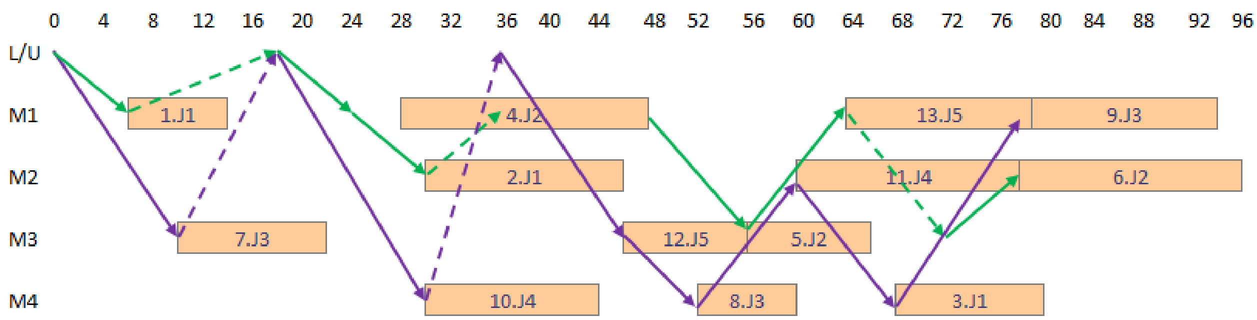
(e) Red AGV serves Operation 6 and complete scheduling.

Figure 3.4 DVFA illustration for Job Set 1 and Shop Floor Layout 1
(reducing AGV fleet size from 5 to 4).

In Figure 3.4 (a), 5 AGVs are assigned to operations and each AGV follows a job. The TOS is achieved and the makespan is equal to 76. With Equation (3.17) the utilization of AGVs can be calculated, and Golden AGV has the lowest utilization, hence it is removed from the schedule, and Operation 12 and 13 are marked as unserved in Figure 3.4 (b). Then Purple AGV is assigned to start Operation 12 as shown in Figure 3.4 (c) after it finishes handling Job 4 to start Operation 10. After that Purple AGV continuing handling Job 3 to start Operation 11 is found to be the most efficient, although is delayed due to previous reassignment. Notice that Operation 6 is also delayed, hence the makespan is increased to 78. In Figure 3.4 (d), the Blue AGV is assigned to Operation 13. Since keeping current sequence of Operation 9 and 13 would cause a long delaying, Operation 9 and 13 are swapped. Now unlike previous reassignment of Purple AGV, Blue AGV going back to continue handling Job 2 is not efficient, hence Operation 6 is marked as unserved. Red AGV is found to be the best to take Operation 6 and keep current makespan, as shown in Figure 3.4 (e).



(a) Job schedule with 3 AGVs based on DVFA.



(b) Job schedule with 2 AGVs (target) based on DVFA.

Figure 3.5 DVFA result for Job Set 1 in Layout 1 (3 and 2 AGVs).

3.2.4 Optimization-based Algorithm Initialization

Currently based on TOS, the initial vehicle assignment scheme is that each vehicle uniquely follows a job; however, we expect that when a vehicle is removed, the reassignments affect the vehicle's assigned transportation tasks to the least degree. Therefore, with the TOS, we formulate an optimization model to maximize the total idle time of vehicles. During idle time, if additional transportation mission presents, it is more likely that a vehicle is able to take over the transportation without affecting its original succeeding missions. Such an idle time maximization model is formulated in Equations (3.18) to (3.21). In the formulation, additional binary variable y_{ku} is used to indicate whether vehicle k is assigned to operation u in the TOS or not.

$$\max_{y} \sum_{k\in\{1,2,\dots,K\}} \sum_{u\in I} \sum_{\substack{v\in I \\ st_v<st_u}} (T_u - t_u - T_v - \tau_{vu}) y_{ku} y_{kv} \tag{3.18}$$

Subject to

$$\sum_{k\in\{1,2,\dots,K\}} y_{ku} = 1 \qquad \forall\, u \in I \tag{3.19}$$

$$T_u \geq T_v + (t_u + \tau_{vu}) y_{kv} \; if \; y_{ku} = 1 \qquad \forall\, k \in \{1,2,\dots,K\}, u \in I, v \in I, st_v < st_u \tag{3.20}$$

$$y \in \{0,1\} \tag{3.21}$$

Equation (3.18) is the objective function maximizing the total idle time of vehicle assignment, and the idle time is calculated by the difference on start time of two operations assigned to the same vehicle. Equation (3.19) regulates that each operation can only be assigned to one vehicle. Equation (3.20) ensures the feasibility of vehicle assignment. If operation u and v are both assigned to vehicle k, i.e. $y_{ku} = y_{kv} = 1$, the time between arrival of the two operations must be long enough for vehicle k to travel.

It should be noted that Model (3.18) to (3.21) must be feasible, since one intuitive feasible solution is the schedule that each vehicle follows a job along all its operations, like the case in Figure 3.4(a). With this initialization boosting method implemented, the DVFA takes longer computation time to solve the quadratic model (3.18) to (3.21), but the performance on minimizing makespan is expected to be improved.

3.3 Computational Experiments and Analysis

Computational experiments and comparisons have been conducted on makespan under 2 AGVs with other algorithms, based on shop floor layouts and job sets data of Bilge and Ulusoy (1995). There are 4 shop floor layouts and 10 job sets, and their combinations result in 40 experimental data sets.

The proposed DVFA is implemented in 3 steps. **Firstly**, the pure job scheduling problem with minimum necessary transportation time (Equation1 (3.1) to (3.5)) is solved with CPLEX on NEOS, as the basis of initialization. **Secondly**, model (3.18) to (3.21) is solved with CPLEX as well to initialize the AGV assignment to operations. **Finally,** the vehicle-reducing iterations are executed in R version 3.1.3 (R Core

Team, 2015) on a personal computer with Intel Xeon 2.40 GHz CPU and 16 GB RAM. Therefore, the computation time of DVFA is the summation of time used by the 3 steps.

The solution methods referred to from the literature did not report the corresponding computation time except for Baruwa and Piera (2016), hence the comparison on algorithm efficiency only takes place between their work, CPLEX and proposed DVFA. The integrated comparison is presented in Table 3.7, and the cases are named with EX*mn* representing the shop floor case of Job Set *m* and Lay out *n*. As references, B is the makespan of the same job set by Bilge and Ulusoy (1995), and similarly U is for Ulusoy, Sivrikaya-Şerifoğlu, and Bilge (1997), A is for Abdelmaguid et al. (2004), R is for Reddy and Rao (2006), D is for Deroussi, Gourgand, and Tchernev (2008), Z is for Zheng, Xiao, and Seo (2016), and Ba is for Baruwa and Piera (2016). The integrated JSSMH problems are solved as a whole with CPLEX on NEOS (Czyzyk, Mesnier, & Moré, 1998; Dolan, 2001; Gropp & Moré, 1997), notated with CPLEX in Table 3.7.

Table 3.7 Comparison results for the 40 test shop floor cases.

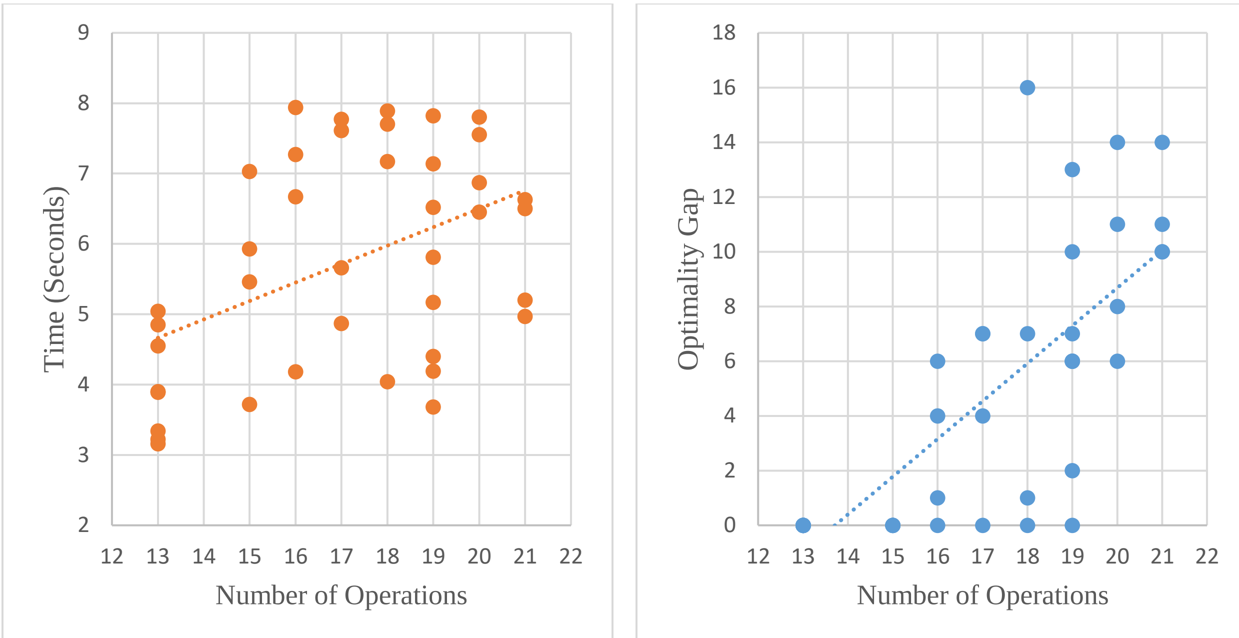
	Makespan									Computation Time (s)		
Case	B	U	A	R	D	Z	Ba	CPLEX	DVFA	Ba	CPLEX	DVFA
EX11	96	96	96	96	96	96	96	96	96	138.5	30.58	4.85
EX21	105	104	102	100	102	100	100	100	100	282.4	730.77	3.72
EX31	105	105	99	99	99	99	99	99	100	27.7	176.83	7.94
EX41	118	116	112	112	112	112	112	112	118	255.4	50803.3	4.4
EX51	89	87	87	87	87	87	87	87	87	18.4	136.43	3.34
EX61	120	121	118	118	118	118	118	118	134	74.7	7927.26	4.04
EX71	119	118	115	111	111	111	111	Fail	117	549.3	-	5.81
EX81	161	152	161	161	161	161	161	161	161	1300	27.79	7.55
EX91	120	117	118	116	116	116	116	116	123	57	22.09	7.61
EX101	153	150	147	147	147	146	146	146	157	115.5	7138.1	4.97
EX12	82	82	82	82	82	82	82	82	82	39.2	4.34	3.22
EX22	80	76	76	76	76	76	76	76	76	100.5	5.44	5.46
EX32	88	85	85	85	85	85	85	85	91	44.9	8.3	4.18
EX42	93	88	88	87	87	87	87	87	89	268.7	3118.96	4.19

EX52	69	69	69	69	69	69	69	69	69	98.7	17.82	4.55
EX62	100	98	98	98	98	98	98	98	98	66.6	10.18	7.89
EX72	90	85	79	79	79	79	79	79	85	2303	11915	5.17
EX82	151	142	151	151	151	151	151	151	151	2.7	14.77	6.87
EX92	104	102	104	102	102	102	102	102	109	284	9.69	7.77
EX102	139	137	136	135	135	135	135	135	145	3252	161.63	6.63
EX13	84	84	84	84	84	84	84	84	84	145.1	8.14	5.04
EX23	86	86	86	86	86	86	86	86	86	96.6	95.98	7.03
EX33	86	86	86	86	86	86	86	86	86	617.3	6.68	7.27
EX43	95	91	89	89	89	89	89	89	99	216.5	3997.25	6.52
EX53	76	75	74	74	74	74	74	74	74	139.4	83.23	3.9
EX63	104	104	104	103	103	103	103	103	104	902.6	23.33	7.17
EX73	91	88	86	83	83	83	83	83	90	2403	33725.1	7.14
EX83	153	143	153	153	153	153	153	153	153	9.3	14.45	7.8
EX93	110	105	106	105	105	105	105	105	109	54.1	10.17	4.87
EX103	143	143	141	139	138	137	139	137	147	66.6	290.78	6.5
EX14	108	103	103	103	103	103	103	103	103	510.2	27.67	3.89
EX24	116	113	108	108	108	108	108	108	108	475.9	3698.61	5.93
EX34	116	113	111	111	111	111	111	111	115	414.9	832.66	6.67
EX44	126	126	126	126	121	121	121	121	121	452	22554.1	3.68
EX54	99	97	96	96	96	96	96	96	96	223.2	176.06	3.16
EX64	120	123	120	120	120	120	120	120	127	370.2	1760.19	7.7
EX74	136	128	127	126	126	126	126	Fail	139	3598	-	7.82
EX84	163	163	163	163	163	163	163	163	163	295.8	4681.18	6.45
EX94	125	123	122	122	120	120	120	120	120	1266	61.69	5.66
EX104	171	164	159	158	159	157	157	157	171	822.2	79885	5.2

The makespan of B, U, A, R, D, and Z are based on Zheng, Xiao, and Seo (2016). Note that CPLEX failed on EX71 and EX74, which means CPLEX cannot find the optimal solution.

Summarized from Table 3.7, the performance of DVFA on solving JSSMH is comparable to other techniques in terms of solution accuracy and efficiency. Generally speaking, DVFA is capable of achieving optimal and near-optimal solutions with an overall optimality gap of 3.9%. DVFA solves 16 of the 40 cases to global optimality, and for those cases that DVFA cannot solve to global optimality, the average optimality gap is 6.5%. The worst case is 13.6%, when solving EX61. However, DVFA is able to result in a solution in significantly shorter time compared to CPLEX and Ba, and this could serve a reason to adopt DVFA in practice. By any solution methods, a job set has a makespan of at most 177 minutes, and the average makespan of all job sets solved with all methods is about 109 minutes; however, the solving time of Ba and CPLEX can reach as long as 60 and 1331 minutes, respectively. Such a high ratio of solving time and makespan is not reasonable in practice, unless job shops know the information of job sets in advance and there is enough time for them to complete scheduling before they start working. Considering the fact that shop floors are likely to be responsible for multiple job sets in a given production horizon, the long solving time of JSSMH might furthermore limit the application of Ba and CPLEX. The DVFA proposed in this paper has advantage in this sense, that a schedule close to optimality can be acquired in short time, hence the production can be executed quickly even if there is not much allowed time for scheduling, such as online and real time scheduling scenarios.

The main reason of DVFA efficiency should be attribute to the logic defined in the algorithm, and in the program, it is likely to be a set of simple conditional judgement statements. Like all other solution techniques, the performance of DVFA is influenced by problem size, and for JSSMH, the number of operations can be used as an indicator of problem size. Figure 3.6 (a) and (b) record the solving time and optimality gap of DVFA on different numbers of operations of job sets. Essentially, both the solving time and the optimality gap increase when there are more operations in the optimized job set (as shown by the trend lines). Specifically, for small job sets with less operations, DVFA is able to reach the global optimality in short time.



(a) (b)

Figure 3.6 DVFA performance against operation number

3.4 Conclusion

In this study, the JSSMH problem that jobs and vehicles are scheduled and routed simultaneously is studied, and a heuristic algorithm is proposed to solve the problem instead of exact commercial solvers to achieve a good quality solution in short time.

The algorithm starts with the scenario that the vehicle fleet size is large enough that the job schedule solved by the pure JSP model can be achieved. A quadratic optimization model is formulated to initialize the job and vehicle schedule, then vehicle fleet size is iteratively reduced. In each iteration, whenever one vehicle is removed from the system, the operations served by the removed vehicle are reassigned to remaining vehicles according a series of heuristic rules. The algorithm ends when all of the operations are served by vehicles and the number of remaining vehicles is equal to the original requirement.

The major contribution of this research can be summarized as follows. Firstly, we linearized the JSSMH model of Bilge and Ulusoy (1995) with conditional constraints to replace the original nonlinear constraints, and added on a constraint to start timing as soon as the first job is taken out of the

Loading/Unloading station (L/U). We can solve the reasonable sized problem to optimality with CPLEX, which is used as a reference in a case study. Secondly, a new visualization method is proposed based on traditional Gantt charts to present the job schedule and AGV movement simultaneously, with which we explain how the proposed algorithm works. Different with treating vehicles as additional machines in Gantt charts in the existing body of literature, the proposed method explicitly presents the interaction between vehicles and jobs. Thirdly, a heuristic algorithm is proposed to solve JSSMH more efficiently. The algorithm includes an initialization with a vehicle fleet size same as the number of jobs. During each iteration, one vehicle is removed from the system, and a set of heuristic rules guide the operation reassignment to vehicles (or vehicle reassignment to operations). Finally, we designed an algorithm initialization boosting mechanism with an optimization model that can significantly improve the solution quality. The initialization counter-intuitively maximizes the idle time of vehicles, such that it is more likely to accommodate additional operations during the vehicle reduction step without affecting original transportation schedule.

CHAPTER 4. AGV-BASED JOB SHOP SCHEDULING WITH MATERIAL HANDLING UNDER VARIABLE PROCESSING TIME

This chapter is organized as follows: The SP-JSSMH model considering random processing time is introduced in Section 4.1. In Section 4.2, the JSSMH model is modified to incorporate deteriorating processing time. All proposed models are validated with small datasets in Section 4.3 and a systematic case study based on data in the body of literature is included in Section 4.4.

4.1 A Two-Stage Stochastic Programming for JSSMH with Random Processing Time

The JSSMH problem we focus on in this chapter can be stated as follows: on a shop floor, a job set J is processed on a set of machines, where each machine can only process one job at a time. Each job j has a unique processing route consisting of a set I_j of operations to complete its manufacturing procedure, and for each operation i, a random processing time $p(\xi_i)$ is required where $p(\xi_i)$ follows a specific distribution. A fleet of AGVs is configured on the shop floor to handle each job after completion of an operation. A fixed loaded travel time t_i is incurred for each job before the start of the next operation i, and deadheading trips of vehicles take another fixed period τ_{hi} depending on the vehicles' previous trip to operation h. The scheduling objective is to achieve the minimum expectation of makespan that is defined as the completion time of the last operation on the shop floor.

A two-stage stochastic programming model is formulated to minimize the expected makespan over a number of scenarios, and we notate it as SP-JSSMH. The job sequences on each machine and AGV routes are defined as first-stage variables, and both of them do not change under uncertainty. The job arrival time at machines, processing completion time, and makespan of each scenario are regarded as second-stage variables that are dependent on scenario realization. Notice that the processing start time is a hidden second-stage variable that is executable. Correspondingly, compared to the model in Chapter 3, both notation and model formulation are modified in this chapter to form into a SP model. The notations are included in Tables 4.1 and 4.2. Compared to Tables 3.1 and 3.2, additional and modified notations are bolded. The formulation of SP-JSSMH is presented with Equation (4.1) to (4.16).

Table 4.1 Notations of Sets and Parameters for SP-JSSMH

J	Set of jobs.
$\boldsymbol{S}$	Set of scenarios
n_j	Number of operations of job *j*.
n	Number of operations, $n = \sum_{j\in J} n_j$.
J	Index set of operations. $I = \{1,2, \dots, n\}$
I_j	Set of indices associated with job *j*.
N_j	Total number of operations of the jobs indexed before *j*. $N_1 = 0$
$\bar{I}_i$	Index set of operations excluding operation *i* and succeeding operations of the same job.
$\underline{I}_h$	Index set of operations excluding operation *h* and preceding operations of the same job.
K	Number of vehicles.
t_i	Travel time to loaded trip heading for operation *i*.
τ_{hi}	Travel time of deadheading trip from machine of operation *h* to machine of operation *i*.
$\boldsymbol{\omega_s}$	Probability of scenarios *s*
$\boldsymbol{p_i^s}$	Processing time of operation *i* in scenarios *s*.
H	A large number

Table 4.2 Notations of Variables for SP-JSSMH

$\boldsymbol{Z^s}$	The makespan of scenarios *s*.
$\boldsymbol{c_i^s}$	Completion time of operation *i* in scenarios *s*.
$\boldsymbol{T_i^s}$	Completion time of loaded trip for operation *i* in scenarios *s*.
q_{rs}	Binary variable. $q_{rs} = 1$ if operation *r* and *s* belong to different jobs and are on the same machine and *r* is processed earlier than *s*.
x_{hi}	Binary variable. $x_{hi} = 1$ if a vehicle is assigned for deadheading trip from operation *h* to *i*.
x_{oi}	Binary variable. $x_{oi} = 1$ if a vehicle starts from L/U to operation *i* as its first trip.
x_{ho}	Binary variable. $x_{ho} = 1$ if a vehicle returns to L/U from operation *h* as its last trip.
$\boldsymbol{D_{jih}^s}$	Auxiliary variable for time between AGV handling of operation *i* and *h* that both belong to job *j* in scenarios *s*.
$\boldsymbol{C_{jh}^s}$	Auxiliary variable for time between AGV handling of operation *h* and the first operation of job *j* in scenarios *s*.

$$\min \sum_{s} \omega_s Z^s \tag{4.1}$$

subject to:

$$Z^s \geq c^s_{N_j+n_j} \qquad \forall s, \quad j \in J \tag{4.2}$$

$$c_i^s - c_{i-1}^s \geq p_i^s + t_i \qquad \forall s, \quad i, i-1 \in I_j, j \in J \tag{4.3}$$

$$c^s_{N_j+1} \geq p^s_{N_j+1} + t_{N_j+1} \qquad \forall s, \quad j \in J \tag{4.4}$$

$$\begin{cases} (1 + H\tau_{uv})c_u^s \geq c_v^s + p_u^s - Hq_{uv} \\ (1 + H\tau_{uv})c_v^s \geq c_u^s + p_v^s - H(1 - q_{uv}) \end{cases} \qquad \forall s, \quad u \in I_j, v \in I_k, j, k \in J, j \neq k \tag{4.5}$$

$$x_{oi} + \sum_{h \in \bar{I}_i} x_{hi} = 1 \qquad \forall\, i \in I \tag{4.6}$$

$$x_{ho} + \sum_{i \in \underline{I}_h} x_{hi} = 1 \qquad \forall\, h \in I \tag{4.7}$$

$$\sum_{i \in I} x_{oi} \leq K \tag{4.8}$$

$$\sum_{i \in I} x_{oi} - \sum_{h \in I} x_{ho} = 0 \tag{4.9}$$

$$T_i^s \leq c_i^s - p_i^s \qquad \forall s, \quad i \in I \tag{4.10}$$

$$T_i^s - t_i \geq c_{i-1}^s \qquad \forall s, \quad i, i-1 \in I_j, j \in J \tag{4.11}$$

$$\begin{cases} D_{jih}^s - T_h^s - \tau_{h,i-1}x_{h,i} \leq H(1 - x_{h,i}) \\ D_{jih}^s - T_h^s - \tau_{h,i-1}x_{h,i} \geq -H(1 - x_{h,i}) \\ D_{jih}^s - \tau_{h,i-1}x_{h,i} \leq Hx_{h,i} \\ D_{jih}^s - \tau_{h,i-1}x_{h,i} \geq -Hx_{h,i} \end{cases} \qquad \forall s, \quad i, i-1 \in I_j, h \in \bar{I}_i \; j \in J \tag{4.12}$$

$$T_i^s - t_i \geq x_{oi}\tau_{o,i-1} + \sum_{h \in \bar{I}_i} D_{jih}^s \qquad \forall s, \quad i, i-1 \in I_j, j \in J \tag{4.13}$$

$$\begin{cases} C_{jh}^s - T_h^s - \tau_{ho}x_{h,N_j+1} \leq H\left(1 - x_{h,N_j+1}\right) \\ C_{jh}^s - T_h^s - \tau_{ho}x_{h,N_j+1} \geq -H\left(1 - x_{h,N_j+1}\right) \\ C_{jh}^s - \tau_{ho}x_{h,N_j+1} \leq Hx_{h,N_j+1} \\ C_{jh}^s - \tau_{ho}x_{h,N_j+1} \geq -Hx_{h,N_j+1} \end{cases} \qquad \forall s, \quad h \in \bar{I}_{N_j+1}, j \in J \tag{4.14}$$

$$T_{N_j+1}^s - t_{N_j+1} \geq \sum_{h \in \bar{I}_{N_j+1}} C_{jh}^s \qquad \forall s, \quad j \in J \tag{4.15}$$

$$x_{oi}T_i^s = x_{oi}\tau_{oi} \qquad \forall s, \quad i \in I \tag{4.16}$$

$$x, q \in \{0,1\} \tag{4.17}$$

$$T, c, Z > 0 \tag{4.18}$$

Similar to the model in Chapter 3, the structure of SP-JSSMH is not significantly changed. Equations (4.1) to (4.5) represent a typical Job Shop Scheduling (JSP) model (Pinedo, 2009), but the variables and parameters are specified for different processing time scenarios. The additional parameter t_i is to consider necessary transportation time of a job from one machine to another for a pair of consecutive operations. Unlike variable job processing time, such a travel time accomplished by AGVs are relatively constant and usually not influenced significantly by environmental factors.

When jobs finish their last operation, they are immediately removed from the machine. AGVs do not handle the completed jobs back to L/U, hence the makespan is defined as the finish time of the last operation on the shop floor in all scenarios. Binary variable *x* represents the routes of AGVs, which indicates the sequential relationship of each operation. Equations (4.6) and (4.7) regulate that each operation can only follow one another operation. Equation (4.8) limits the number of AGV routes by

AGV fleet size. Equation (4.9) ensures that for each AGV, there must be a starting trip as well as an ending trip.

Equation (4.10) means an operation can begin only after the job arrival to the machine. The operation sequence of one job is ensured in Equation (4.11). Equations (4.12) to (4.15) are linearized constraints to replace the nonlinear constraints by Bilge and Ulusoy (1995) containing variable product, which indicate the impact of previous trips on the next trip of each AGV.

Equation (4.16) is used to start timing when a vehicle leaves the L/U with the first job it conveys. Such a constraint means a default initial condition that AGVs are at the L/U until they leave for the first job handling task.

4.2 Job Shop Scheduling with Material Handling with Deterioration

Deterioration is the effect that processing becoming difficult with the production proceeding, usually reflected by elongating processing time. When deterioration exists, the optimization of JSSMH could become more complicated with processing time dependency function implemented. In this section with discuss two types of dependency separately and propose different formulations for corresponding Deteriorating Job Shop Scheduling with Material Handling (D-JSSMH).

4.2.1 Linear Deterioration of Processing Time

Lee et al. (2010) described a deteriorating job processing time that was linearly dependent on the operation start time. Based on the notations in Table 4.1 and 4.2, remove the scenario subscripts and let variable s_i denote the start time of operation i, p_i^0 and p_i denote the basic and realized processing time of operation i, and λ denote the deterioration rate, the linear deteriorating processing time is described in Equation (4.19).

$$p_i = p_i^0 + \lambda s_i \tag{4.19}$$

Correspondingly, the completion time of an operation determined by start time and realized processing time is calculated in Equation (4.20).

$$c_i = s_i + p_i = p_i^0 + (1+\lambda)s_i \tag{4.20}$$

Therefore, the comprehensive model of D-JSSMH with linear deteriorating processing time can be formulated in Equation (4.21) to (4.38) as the following.

$$\min Z \tag{4.21}$$

subject to:

$$Z \geq (1+\lambda)s_{N_j+n_j} + p^0_{N_j+n_j} \qquad \forall\, j \in J \tag{4.22}$$

$$s_i \geq (1+\lambda)s_{i-1} + p^0_{i-1} + t_i \qquad \forall\, i, i-1 \in I_j, j \in J \tag{4.23}$$

$$s_{N_j+1} \geq t_{N_j+1} \qquad \forall\, j \in J \tag{4.24}$$

$$\begin{cases}(1+H\tau_{uv})(s_u + p_u^0 + \lambda s_u) \geq (1+\lambda)s_v + p_v^0 + p_u^0 + \lambda s_u - Hq_{uv} \\ (1+H\tau_{uv})(s_v + p_v^0 + \lambda s_v) \geq (1+\lambda)s_u + p_u^0 + p_v^0 + \lambda s_v - H(1-q_{uv})\end{cases} \qquad \forall\, u \in I_j,\ v \in I_k,\ j,k \in J, j \neq k \tag{4.25}$$

$$x_{oi} + \sum_{h \in \bar{I}_i} x_{hi} = 1 \qquad \forall\, i \in I \tag{4.26}$$

$$x_{ho} + \sum_{i \in \underline{I}_h} x_{hi} = 1 \qquad \forall\, h \in I \tag{4.27}$$

$$\sum_{i \in I} x_{oi} \leq K \tag{4.28}$$

$$\sum_{i \in I} x_{oi} - \sum_{h \in I} x_{ho} = 0 \tag{4.29}$$

$$T_i \leq s_i \qquad \forall\, i \in I \tag{4.30}$$

$$T_i - t_i \geq s_{i-1} + p^0_{i-1} + \lambda s_{i-1} \qquad \forall\, i, i-1 \in I_j, j \in J \tag{4.31}$$

$$D_{jih} = T_h + \tau_{h,i-1}\ if\ x_{h,i} = 1 \qquad \forall\, i, i-1 \in I_j, h \in \bar{I}_i\ j \in J \tag{4.32a}$$

$$D_{jih} = 0\ if\ x_{h,i} = 0 \qquad \forall\, i, i-1 \in I_j, h \in \bar{I}_i\ j \in J \tag{4.32b}$$

$$T_i - t_i \geq x_{oi}\tau_{o,i-1} + \sum_{h \in \bar{I}_i} D_{jih} \qquad \forall\, i, i-1 \in I_j, j \in J \tag{4.33}$$

$$S_{jh} = T_h + \tau_{ho}\ if\ x_{h,N_j+1} = 1 \qquad \forall\, h \in \bar{I}_{N_j+1}, j \in J \tag{4.34a}$$

$$S_{jh} = 0\ if\ x_{h,N_j+1} = 0 \qquad \forall\, h \in \bar{I}_{N_j+1}, j \in J \tag{4.34b}$$

$$T_{N_j+1} - t_{N_j+1} \geq \sum_{h \in \bar{I}_{N_j+1}} S_{jh} \qquad \forall\, j \in J \tag{4.35}$$

$x_{oi}T_{\mathrm{i}} = x_{oi}\tau_{oi}$ $\quad \forall\, i \in I \quad$ (4.36)

$x, q \in \{0,1\}$ (4.37)

$T, c, Z > 0$ (4.38)

Compared to the model formulation of SP-JSSMH, except for the scenario-based variables and constraints, the model of D-JSSMH in Equation (4.21) to (4.38) has two major difference. First the variable of operations completion time is replaced with start and realized processing time; second the equation groups for linearly calculating the time between consecutive AGV trips are simplified with conditional constraints.

4.2.2 Exponential Deterioration of Processing Time

In the study of X. Zhang et al. (2018), the processing time of an operation was exponentially dependent on the its processing sequence on the machine. Like the linear deterioration, this means the later the job being processed on the machine, it took longer time to complete. Additional notations are included in Table 4.3.

Table 4.3 Additional notations for Exponential D-JSSMH

M	Set of machines
I^m	The operations on Machine *m*.
q_{uv}	Indicator of sequence of operation *u* and *v* on the same machine. $q_{uv} = 1$ if *u* is processed before *v*.
r_i	Rank of operation *i* on the machine.
a	Parameter of deteriorating rate.

Note that variable q is redefined to form the operation sequence on machines. The exponential deterioration is described in Equation (4.39).

$$p_i = p_i^0(1 + a)^{r-1} \tag{4.39}$$

The optimization model of Exponential D-JSSMH is formulated in Equation (4.40) to (4.61).

$\min Z$ (4.40)

subject to:

$$Z \geq c_{N_j+n_j} \qquad \forall j \in J \qquad (4.41)$$

$$r_i \leq |I^m| \qquad \forall m \in M, i \in I^m \qquad (4.42)$$

$$r_v \geq r_u + 1 \; if \; q_{uv} = 1 \qquad \forall m \in M, u, v \in I^m \qquad (4.43)$$

$$q_{uv} + q_{vu} = 1 \qquad \forall m \in M, u, v \in I^m \qquad (4.44)$$

$$c_i - c_{i-1} \geq p_i(1+a)^{r_i-1} + t_i \qquad \forall i, i-1 \in I_j, j \in J \qquad (4.45)$$

$$c_{N_j+1} \geq p_{N_j+1}(1+a)^{r_{N_j+1}-1} + t_{N_j+1} \qquad \forall j \in J \qquad (4.46)$$

$$\begin{cases} c_u \geq c_v + p_u(1+a)^{r_u-1} - Hq_{uv} \\ c_v \geq c_u + p_v(1+a)^{r_v-1} - H(1-q_{uv}) \end{cases} \qquad \forall m \in M, u, v \in I^m \qquad (4.47)$$

$$x_{oi} + \sum_{h \in \bar{I}_i} x_{hi} = 1 \qquad \forall i \in I \qquad (4.48)$$

$$x_{ho} + \sum_{i \in \underline{I}_h} x_{hi} = 1 \qquad \forall h \in I \qquad (4.49)$$

$$\sum_{i \in I} x_{oi} \leq K \qquad (4.50)$$

$$\sum_{i \in I} x_{oi} - \sum_{h \in I} x_{ho} = 0 \qquad (4.51)$$

$$T_i \leq c_i - p_i(1+a)^{r_i-1} \qquad \forall i \in I \qquad (4.52)$$

$$T_i - t_i \geq c_{i-1} \qquad \forall i, i-1 \in I_j, j \in J \qquad (4.53)$$

$$D_{jih} = T_h + \tau_{h,i-1} \; if \; x_{h,i} = 1 \qquad \forall i, i-1 \in I_j, h \in \bar{I}_i, j \in J \qquad (4.54a)$$

$$D_{jih} = 0 \; if \; x_{h,i} = 0 \qquad \forall i, i-1 \in I_j, h \in \bar{I}_i, j \in J \qquad (4.54b)$$

$$T_i - t_i \geq x_{oi}\tau_{o,i-1} + \sum_{h \in \bar{I}_i} D_{jih} \qquad \forall i, i-1 \in I_j, j \in J \qquad (4.55)$$

$$S_{jh} = T_h + \tau_{ho} \; if \; x_{h,N_j+1} = 1 \qquad \forall h \in \bar{I}_{N_j+1}, j \in J \qquad (4.56a)$$

$$S_{jh} = 0 \; if \; x_{h,N_j+1} = 0 \qquad \forall h \in \bar{I}_{N_j+1}, j \in J \qquad (4.56b)$$

$$T_{N_j+1} - t_{N_j+1} \geq \sum_{h \in \bar{I}_{N_j+1}} S_{jh} \qquad \forall j \in J \qquad (4.57)$$

$$x_{oi}T_{\mathrm{i}} = x_{oi}\tau_{oi} \qquad \forall i \in I \qquad (4.58)$$

$$x, q \in \{0,1\} \qquad (4.59)$$

$r \in N^+$ (4.60)

$T, c, Z > 0$ (4.61)

Like all previous JSSMH models, Equation (4.40) minimizes the makespan defined by Equation (4.41). Equation (4.42) to (4.44) define the sequence of operations on the same machine with binary variable q, and regulate the rank of operation r_i as a unique positive integer between o and number of operation on the machine. Equation (4.45) to (4.47) represent the scheduling of job operations, under the realized processing time depending on operations ranking. Equation (4.45) to (4.47) as well as (4.52) are nonlinear constraints derived with Equation (4.39); however, for small D-JSSMH problems that number of operations on a machine is small, Equation (4.39) can be approximated with linear functions. Figure 4.1 shows the scatter plot of Equation (4.39) given deteriorating rate a=0.32 and basic processing time p=3 in X. Zhang et al. (2018), and Table 4.4 recorded the linear regression function and corresponding R^2 value with different maximum rank r (number of operations on a machine $|I^m|$) of operations.

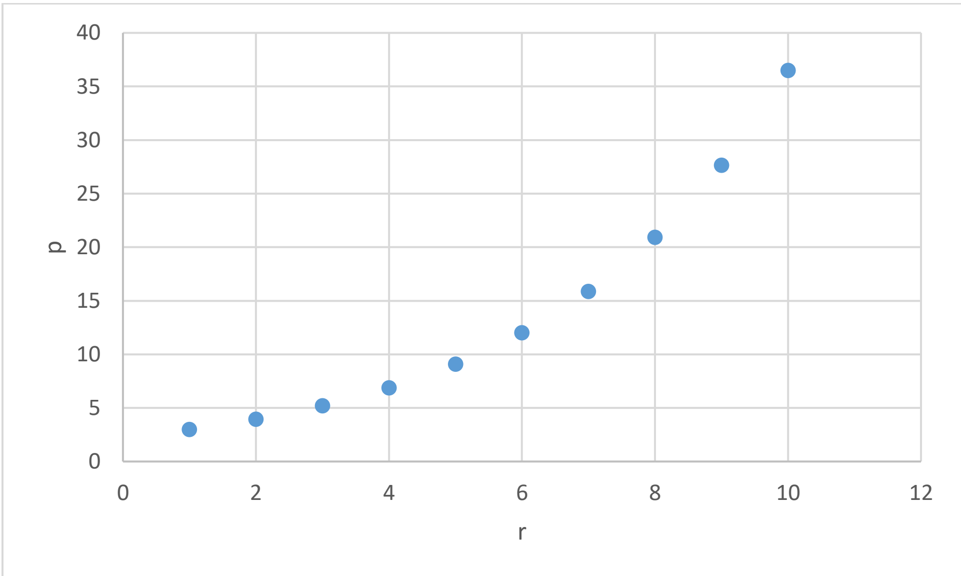


Figure 4.1 Scatter plot of Equation (4.39)

Table 4.4 Linear approximation of Exponential deteriorating function with p=3 and a=0.32

$\lvert I^m \rvert$	Linear regression function	R^2
3	$p = 0.371r + 0.612$	0.9937
4	$p = 0.432r + 0.510$	0.9851
5	$p = 0.505r + 0.364$	0.9743
6	$p = 0.593r + 0.160$	0.9615

In JSSMH application, number of operations on a machine is usually between 3 and 6, hence a linear approximation is accurate enough for scheduling. For large job sets, a piecewise regression can also retain the linearity in each short interval benefitted by the memorylessness of exponential functions. In other words, if the exponential deteriorating function in Equation (4.39) is divided evenly on the axis of r, the linear regression on all segments will have the same R^2.

With the linear approximation exponential deteriorating function, Exponential D-JSSMH model in Equation (4.40) to (4.61) can be solved with commercial solvers on cases in reasonable size.

4.3 Scheduling Example of SP-JSSMH and D-JSSMH

With stochastic job processing time, one option of job shop scheduling is adopting the average processing time (Y. Y. Xiao, Zhang, Zhao, & Kaku, 2012). In JSSMH, with average processing time the sequence of operations and route of vehicles can be determined, while the realization of variable processing time could result in different operation start and end time with the solution with average processing time. With deteriorating job processing time, the scheduling decision can be made without considering deterioration although the schedule will be influenced by realized deteriorating processing time. The models proposed in this study can be validated by better optimal solution of considering variable processing time (stochastic and deteriorating) in modeling compared to simply adopting average processing time or ignoring the deterioration. The job shop layout is Layout 1 in Bilge and Ulusoy (1995).

4.3.1 Job Shop Scheduling with SP-JSSMH

We use a small job set a simple example to demonstrate the validity of SP-JSSMH in solving the problem under uncertain job processing time. In this case there are 3 possible scenarios of job processing time for all operations, as shown in Table 4.5, and the probability of realizing each scenario is 1/3. The average scenario is calculated in Table 4.6, and this average scenario is used to make deterministic decisions with JSSMH model.

Table 4.5 Job Set Example with processing time in 3 scenarios and 1/3 probability for each scenario

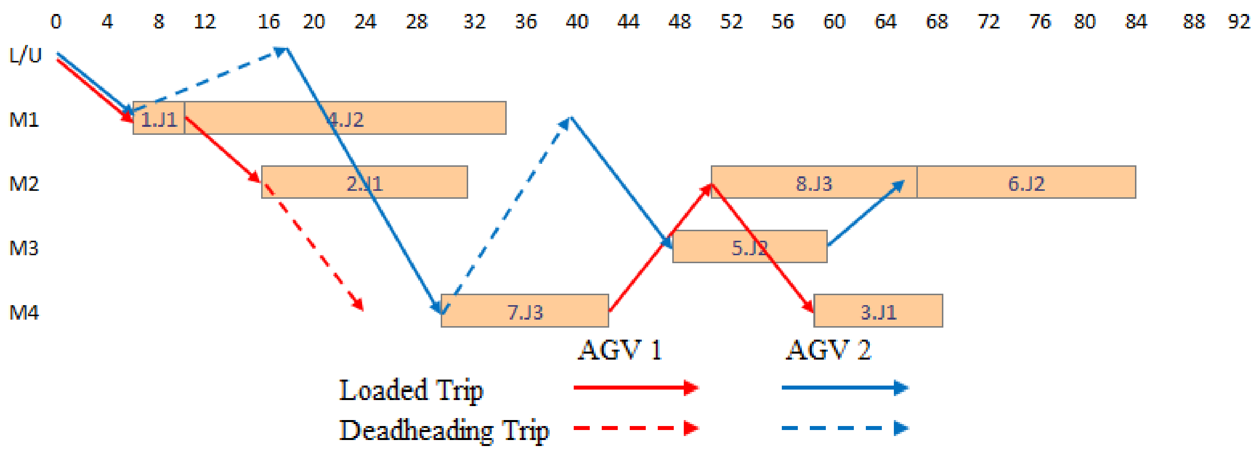
Job \ Operation		1	2	3
Job 1 (J1)	Scenario 1	M1(4)	M2(16)	M4(10)
	Scenario 2	M1(3)	M2(19)	M4(13)
	Scenario 3	M1(4)	M2(12)	M4(16)
Job 2 (J2)	Scenario 1	M1(25)	M3(12)	M2(17)
	Scenario 2	M1(15)	M3(10)	M2(16)
	Scenario 3	M1(22)	M3(14)	M2(18)
Job 3 (J3)	Scenario 1	M4(13)	M2(16)	
	Scenario 2	M4(16)	M2(17)	
	Scenario 3	M4(10)	M2(15)	

Table 4.6 Average scenario of the job set example

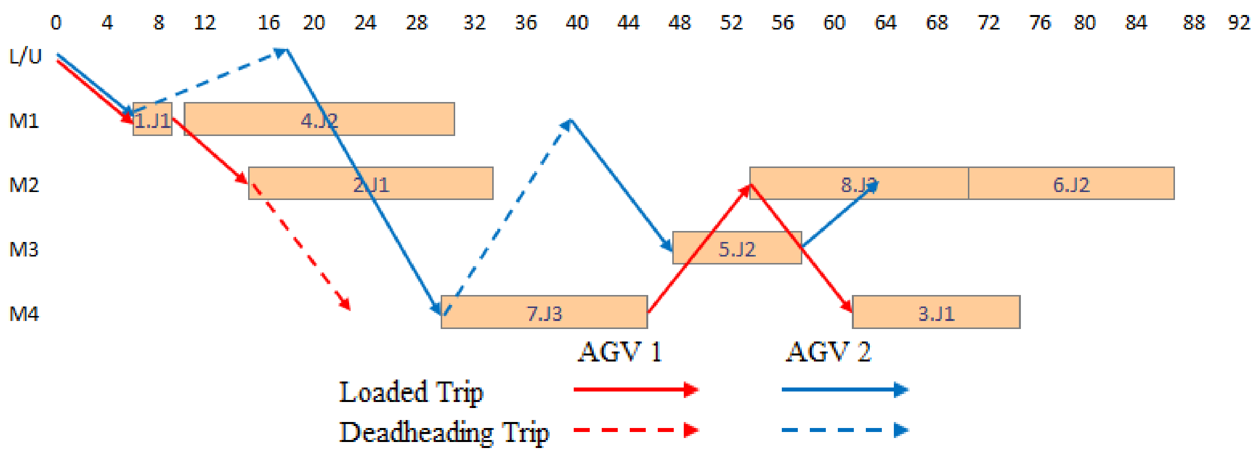
Job \ Operation	1	2	3
Job 1 (J1)	M1(3.67)	M2(15.7)	M4(13)
Job 2 (J2)	M1(20.7)	M3(12)	M2(17)
Job 3 (J3)	M4(13)	M2(16)	

The performance of SP-JSSMH is compared to the deterministic JSSMH model on average makespan in all scenarios with the AGV and job scheduling decision. It is expected that the two models will produce different scheduling results in terms of AGV routing and operations' start and end time.

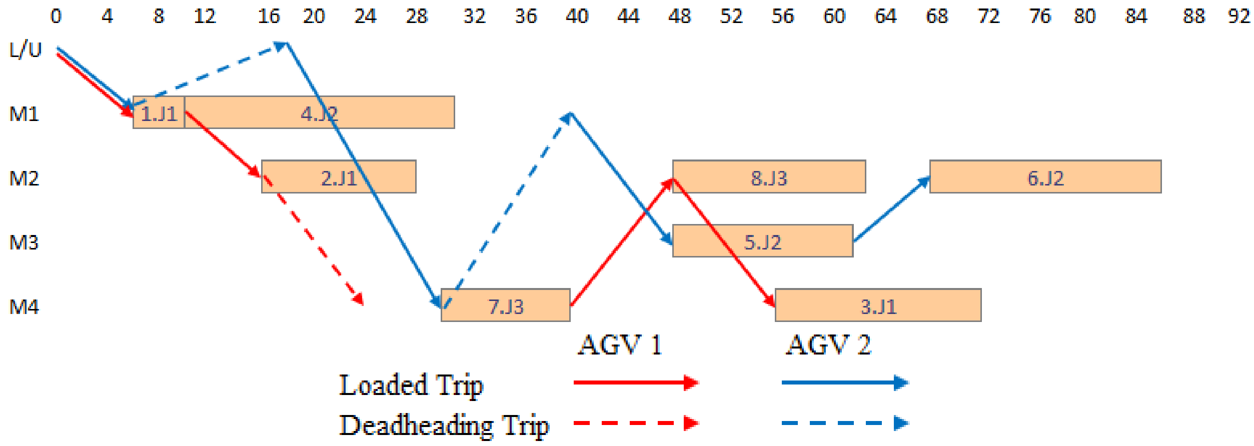
Figure 4.2(a) to (c) shows the job schedule and AGV routes solved by deterministic JSSMH based on average scenario in Table 4.6, which is known as the "Expected Value problem" (EV). Note that AGV routes and operations sequences on machines are fixed according to deterministic optimization results, but realized processing time in each scenario caused different operation start and end time. All operations are indexed to keep consistent with the model notations in Section 2.



(a) Schedule in Scenario 1



(b) Schedule in Scenario 2

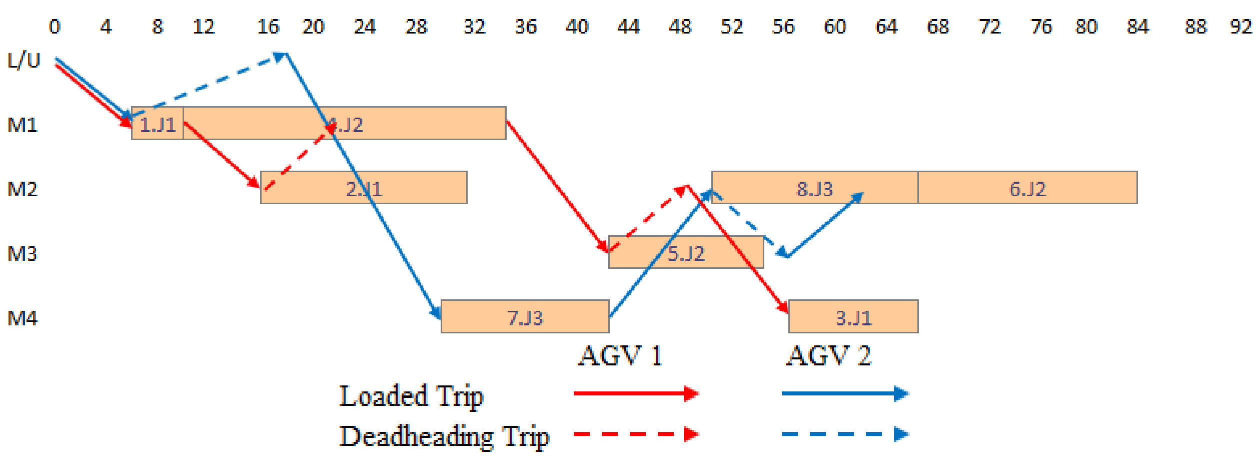


(c) Schedule in Scenario 3

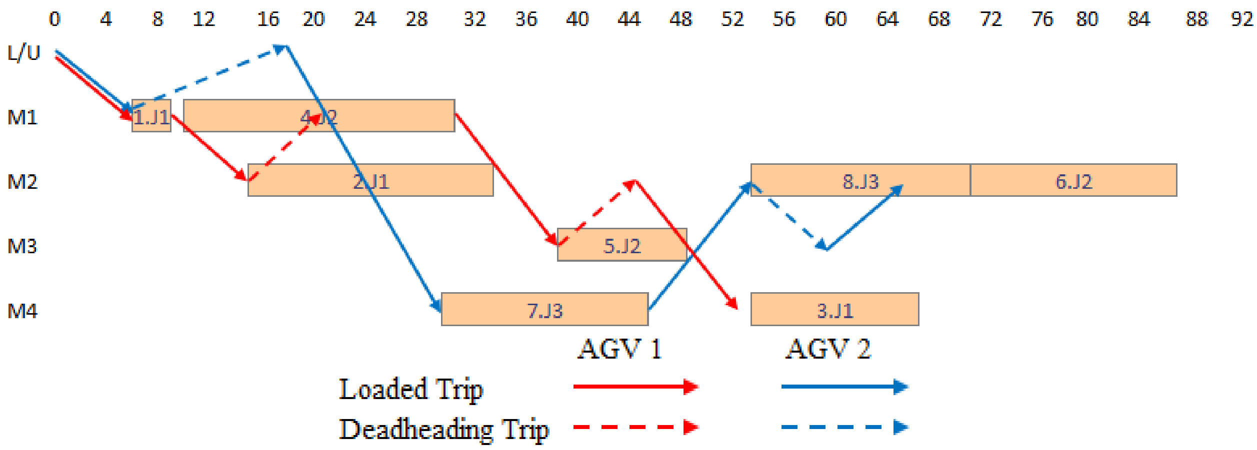
Figure 4.2 Schedule based on EV solution in all scenarios

Since the job sequence on each machine and the AGV path are defined as first-stage variables and do not change cross scenarios, it can be observed in Figure 4.2 that the sequences of jobs on each machine keep consistent over all scenarios, and the destinations of each AGV also keep the same, while there are only variances in time of starting and ending trips which are second-stage variables dependent on scenarios. Since each scenario has an equal likelihood of realization, the expectation of makespan with the EV solution is 85.67, which is known as "expected result of using the EV solution" (EEV).

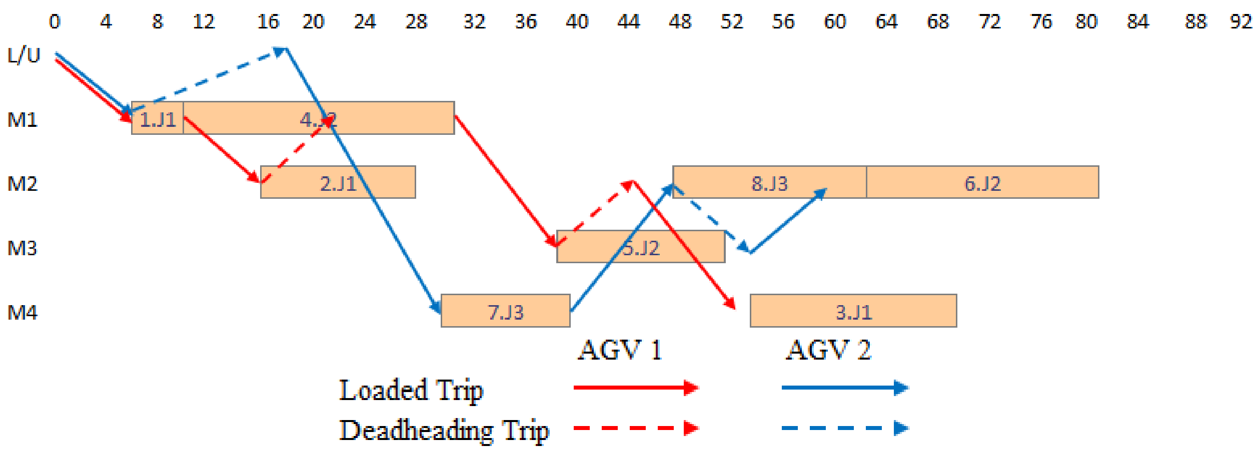
Figure 4.3 (a) to (c) shows the job schedule and AGV paths solved by SP-JSSMH, which is known as the "Recourse Problem" (RP).



(a) Schedule in Scenario 1



(b) Schedule in Scenario 1



(c) Schedule in Scenario 1

Figure 4.3 Schedule based on RP solution in all scenarios

For this small example, the sequence of jobs on machines does not change when stochastic programming is adopted comparing to the deterministic model; however, the AGV routes are different and lead to a shorter makespan in Scenario 3. Correspondingly, the makespan expectation becomes 84. Therefore, the Value of the Stochastic Solution (VSS) can be calculated as 1.67 according to Equation (4.62).

$$VSS = EEV - RP \tag{4.62}$$

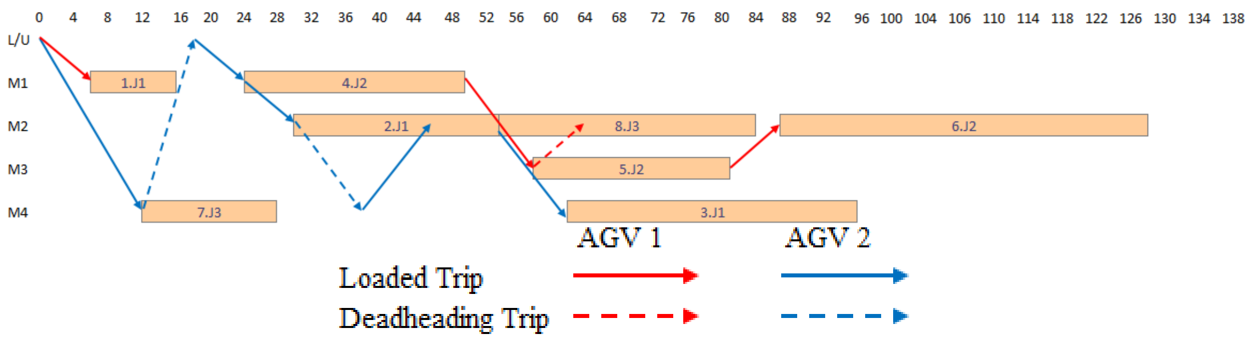
4.3.2 Job Shop Scheduling with D-JSSMH

Like SP-JSSMH, we validate the D-JSSMH model with an example job set. The basic processing time of the operations of the 4 jobs are included in Table 8, which is the input of D-JSSMH. For comparison, the processing time in Table 4.7 is directly adopted in deterministic JSSMH models, but the real processing time is deteriorating linearly or exponentially. In linear D-JSSMH, deteriorating rate λ=0.25. In exponential D-JSSMH, α=0.432, β=0.51, a=0.32.

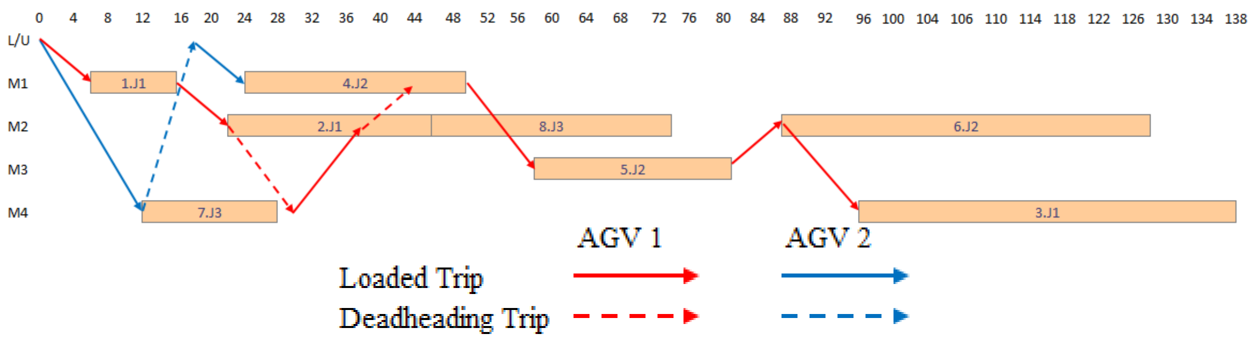
Table 4.7 Basic processing time of the job set example

Job \ Operation	1	2	3
Job 1 (J1)	M1(8)	M2(16)	M4(12)
Job 2 (J2)	M1(20)	M3(10)	M2(18)
Job 3 (J3)	M4(14)	M2(18)	

The performance of D-JSSMH is compared to the deterministic JSSMH model on makespan under two types of deterioration with the AGV and job scheduling decision. Figure 4 (a) and (b) present the resulting schedule of D-JSSMH and deterministic JSSMH under linear deterioration, and Figure 5 (a) and (b) present the schedule of two models under exponential deterioration.

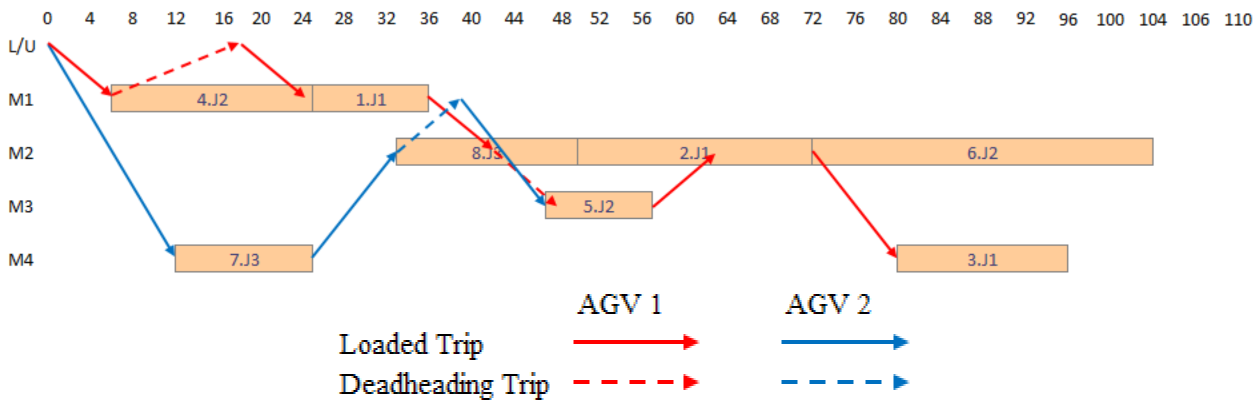


(a) Schedule produced by D-JSSMH

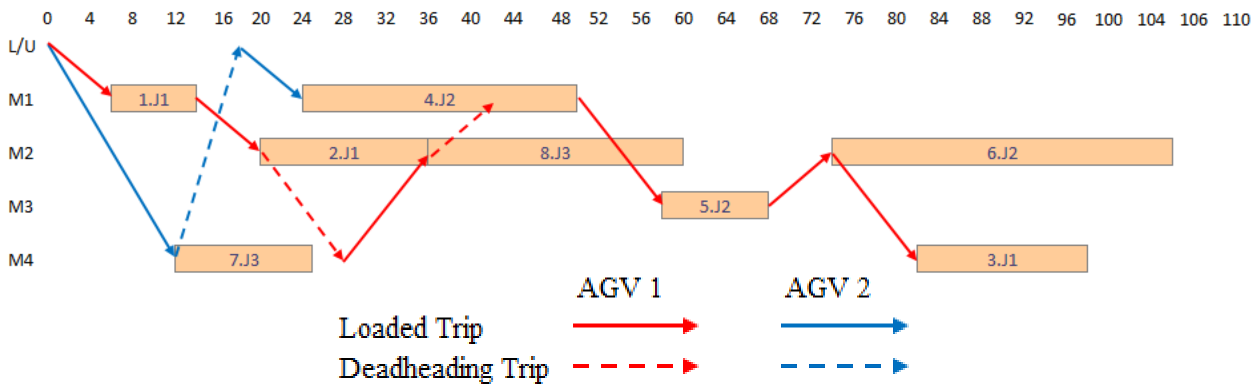


(b) Schedule produced by deterministic JSSMH

Figure 4.4 Schedule of example job set under linear deterioration



(a) Schedule produced by D-JSSMH



(b) Schedule produced by deterministic JSSMH

Figure 4.5 Schedule of example job set under exponential deterioration

It can be observed that ignoring deterioration in JSSMH would cause delay of makespan. The bottleneck in this case is always Job 2 consisting of Operation 4, 5 and 6; however, when linear deterioration exists, deterministic JSSMH model cannot foresee that delaying operation 3 could make its processing time so long that it becomes the last completed operation and enlarges the makespan, On the other hand the D-JSSMH model can deal with this by balancing the start time of all operations. Like the case of linear deterioration, D-JSSMH results in shorter makespan with realized exponential deteriorating processing time than adopting the original value of processing times in deterministic JSSMH model. Note that since in both cases the deterministic JSSMH model adopt the same original processing time data, the operation sequences and AGV routes are identical. Moreover, although there might be multiple optimal scheduling solution, like delivering Job 1 to Machine 4 after Operation 2 with AGV 2, the optimal makespan is always the same, determined by the bottleneck of processing Job 2.

We can conclude from Figure 4.4 and 4.5 that considering deterioration in modeling can achieve shorter makespan than simply making scheduling decision with average processing time. This means like the positive VSS of SP-JSSMH, we can also define that D-JSSMH has a positive impact on improving the optimal solution.

4.4 Case study

We tested SP-JSSMH and D-JSSMH formulation on the job shop layouts and job sets in Bilge and Ulusoy (1995) which has been widely adopted as a computation reference in the body of literature. There are 10 job sets and each includes 5 to 8 jobs. There are 4 different shop floor layouts, hence the case study consists of 40 cases. Each case is notated as EX*mn*, where *m* represents the index of job set and *n* is the index of shop floor layout. For example, EX41 means the combination of Job set 4 and Shop floor layout 1.

4.4.1 SP-JSSMH case study

To keep consistent with the body of existing literature, the processing time of operation *i* is assumed to follow triangular distribution *Triangular*($0.75p_i$, p_i , $1.25p_i$), where p_i is the original processing time. For each operation, 20 samples are generated following the triangular distribution, then in one scenario the processing time of operations is a combination of one sample of each operation. Hence there are 20 scenarios to reflect the stochasticity in the SP-JSSMH model. Figure 4.6 is the example of Job set 3 with 16 operations, in which the distributions of discretized processing time approximate the corresponding triangular distributions.

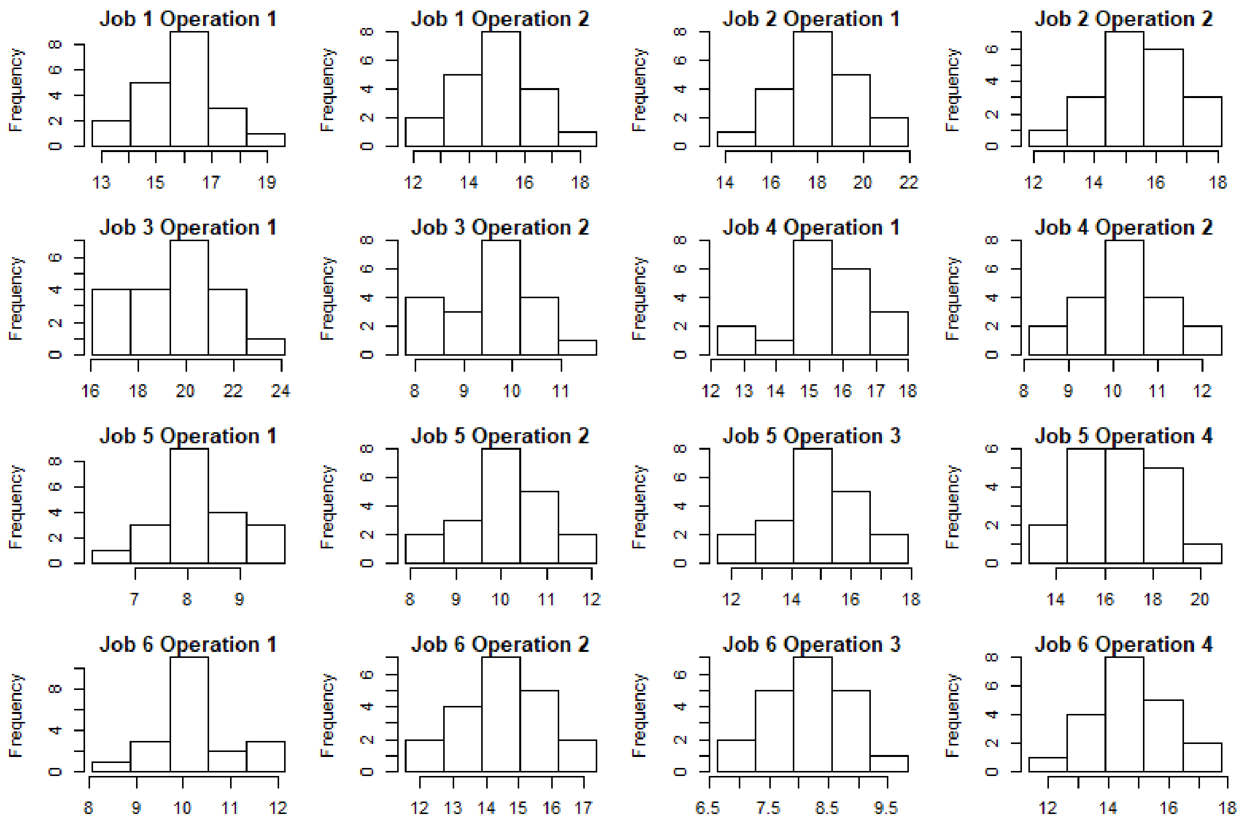


Figure 4.6 Distribution of discretized processing time of Job set 3 in Bilge and Ulusoy (1995)

The models are solved with Pyomo and CPLEX on a server with 252 GB memory and 40 CPUs, and 2 servers with 31 GB memory and 8 CPUs. Both RP and EV solutions are solved for each case, except for EX71 and EX74 on which Pyomo and CPLEX fail to solve. With realized stochastic processing time, Figure 4.7 presents the makespan resulting from implementing RP and EV solution with a scatter plot.

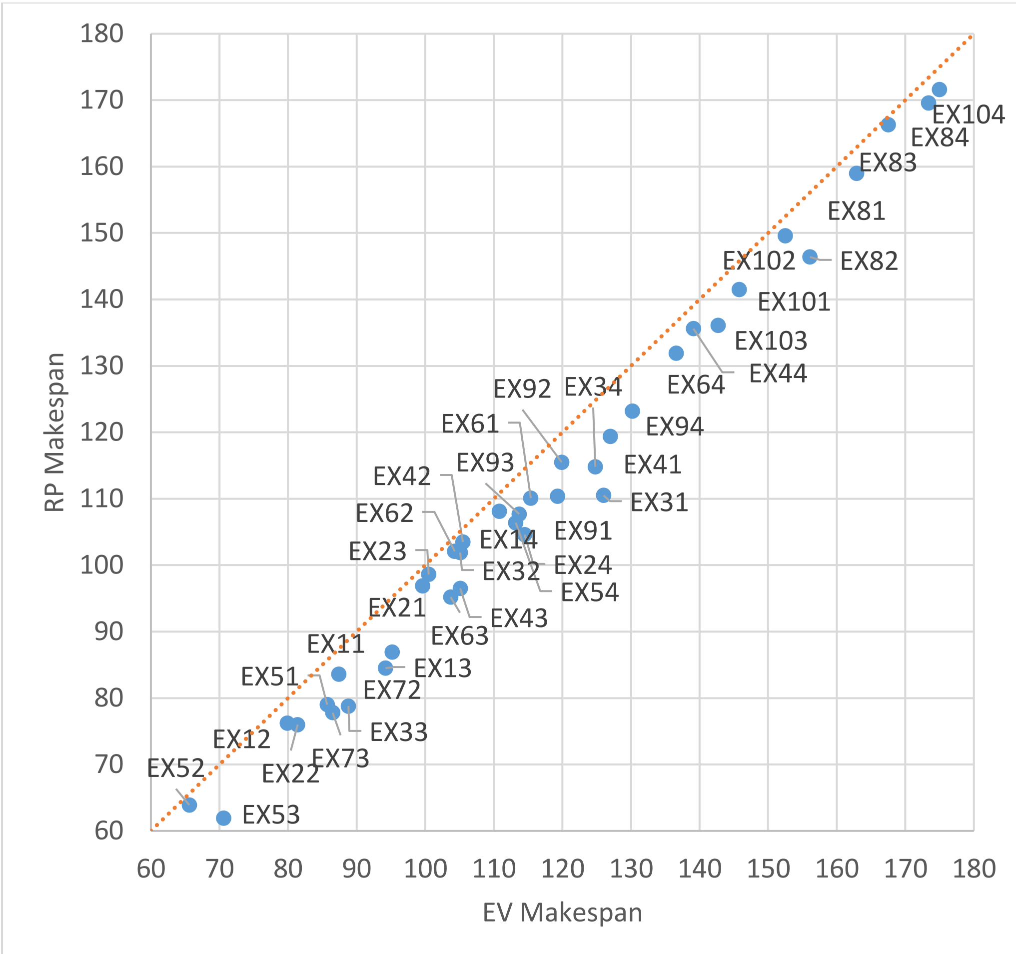


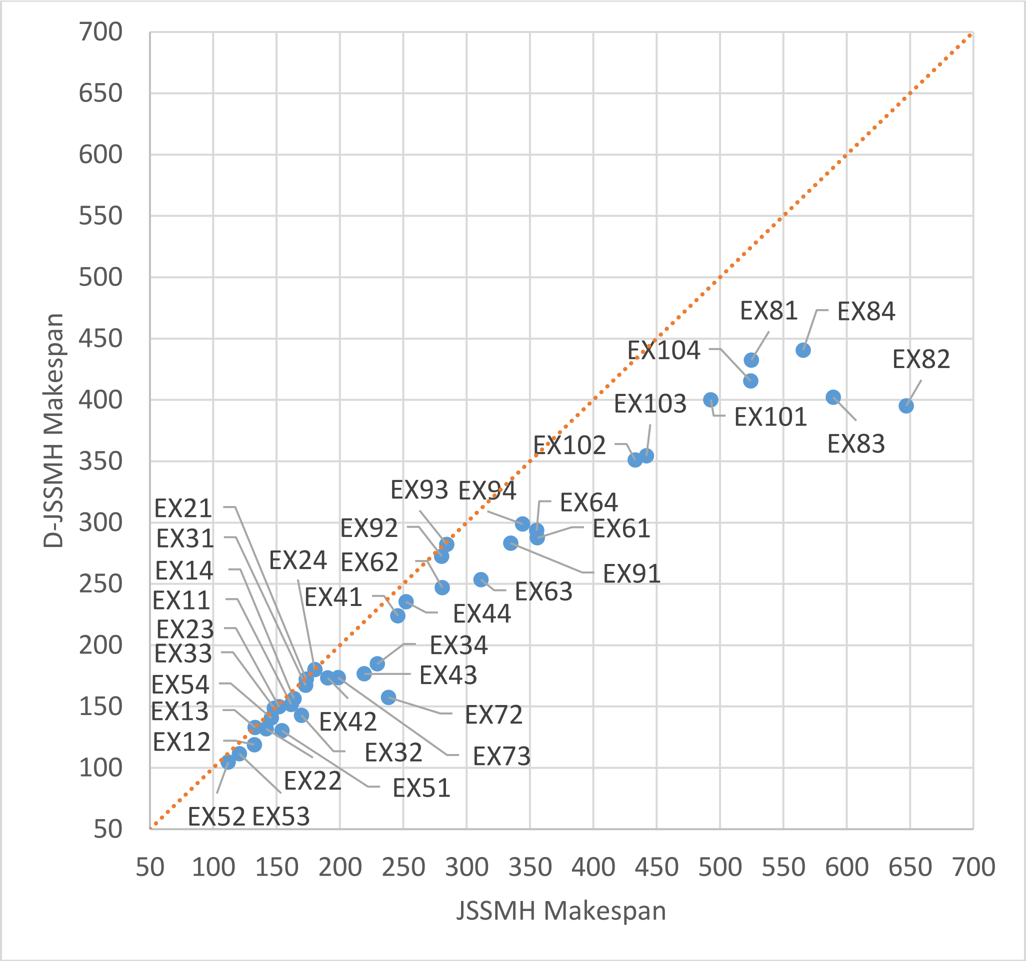
Figure 4.7 Comparison of makespan under stochastic processing time with RP and EV solution

If a point lies on the dash line in Figure 4.7, it means EV and RP lead to the same makespan with realized stochastic processing time of operations of the job set. It can be observed that all points are below the dash line, which means the RP solutions always result in shorter makespan than EV solutions. Averagely, with the RP solution of SP-JSSMH model, the makespan can be reduced for 5.4%. The most significant makespan reductions happen to EX31 and EX53, where 15.5 minutes or 12.3% of makespan was reduced compared to the corresponding EV solution.

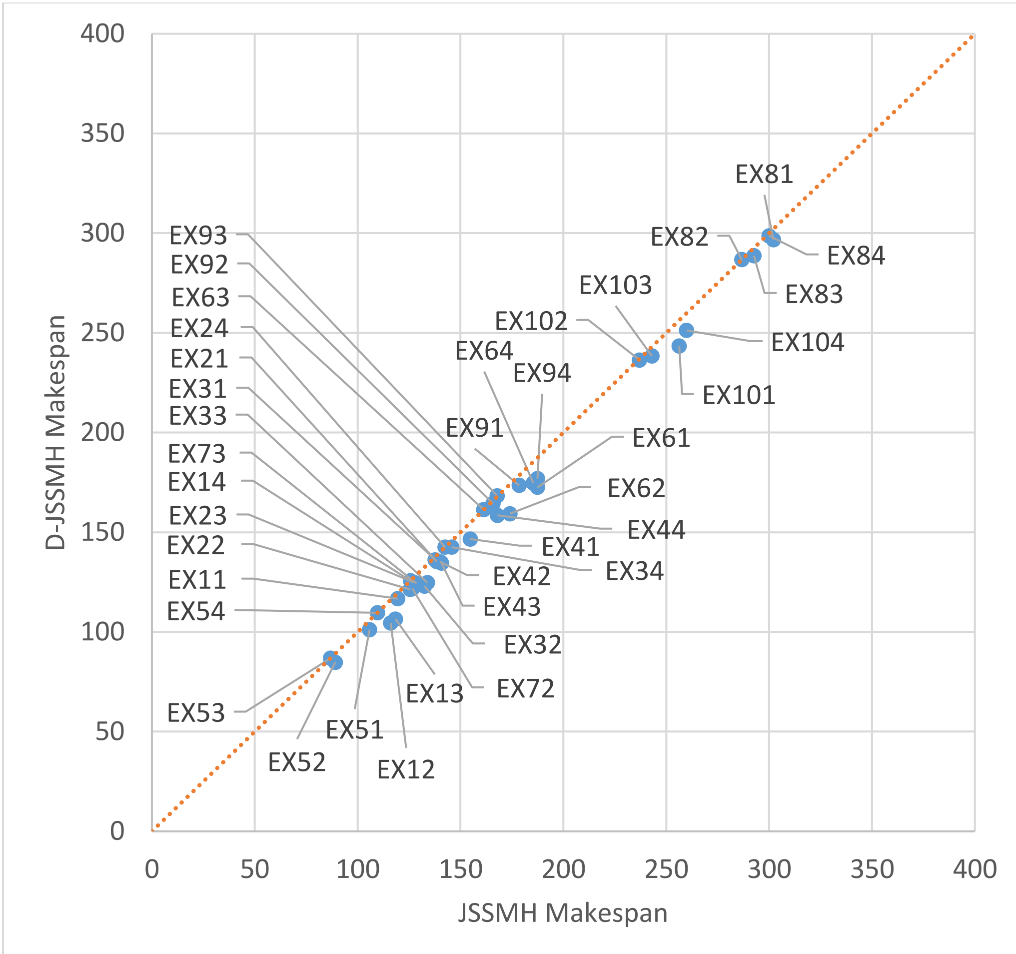
4.4.2 D-JSSMH case study

Same with the validation example of D-JSSMH models, parameters associated with deteriorating rates applied to Bilge cases are λ=0.25 for linear deterioration and α=0.432, β=0.51, a=0.32 for exponential deterioration. The solution of operation sequences and AGV routes solved by original JSSMH model and

D-JSSMH are implemented respectively, and the resulting makespans affected by deterioration are compared with each other. Models are formulated in AMPL and solved with CPLEX on NEOS public server, on which conditional constraints based on binary variables can be directly input without complicated linearization, including Equation (4.32), (4.34), (4.43), (4.54) and (4.56). Figure 4.8(a) and (b) present the scatter plot of makespan under different job scheduling and AGV routing solutions facing with deteriorating operations.



(a) Makespan comparison with linearly deteriorating processing time



(b) Makespan comparison with exponentially deteriorating processing time

Figure 4.8 Comparison of makespan under deterioration with solution of JSSMH and D-JSSMH

Same with Figure 4.7, points below the dash line indicate solutions resulting in shorter makespan with D-JSSMH solutions, which happen to most of the cases. In some cases, JSSMH and D-JSSMH produce the same job scheduling and AGV routing solutions, thus revealing identical makespan represented by the corresponding points lying on the dash line in Figure 8. In some cases, there are significant difference on makespan, for example EX82 under linear deterioration, reducing the makespan for approximately 40%. It can be observed that under current parameter setting, the influence of deterioration is more significant with the linear deteriorating function than that with exponential deteriorating function. Figure 4.9 provides a clearer comparison for the effectiveness of modeling with deterioration.

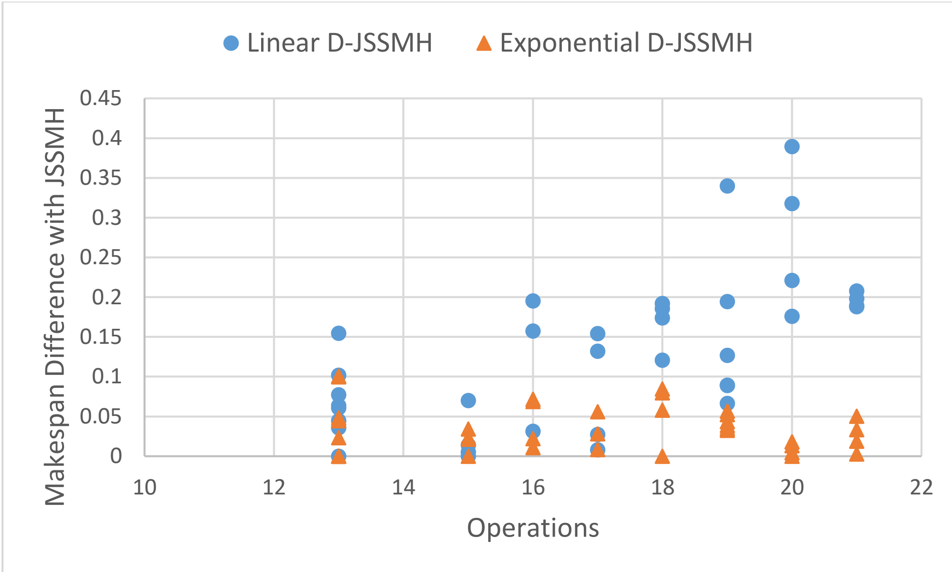


Figure 4.9 Makespan difference with original JSSMH with D-JSSMH under linear and exponential deterioration

In Figure 4.9, the *x*-axis is the number of operations in each job set. D-JSSMH with linearly deteriorating processing time is more effective on reducing the makespan compared to original JSSMH model. Furthermore, for more complicated job sets with relatively large number of operations, solutions of D-JSSMH have obvious advantage.

CHAPTER 5. SUMMARY AND DISCUSSION

In this dissertation, three studies in Job Shop Scheduling with Material Handling (JSSMH) are included, and each of the studies focuses on a specific aspect of the JSSMH problem, including AGV assignment in the first study, a heuristic algorithm for JSSMH in the second study, and JSSMH with variable processing time in the third study. With three independent but correlated chapters beginning from Chapter 2, this dissertation aims to provide a systematic approach for JSSMH. The contribution includes innovations in mathematical modeling as well as solution techniques.

In the first study, JSSMH is regarded as the combination of a series of AGV assignment problems. Classic AGV assignment rules make decision when transportation requests are generated, while in our study AGVs are assigned with optimization models that account for current as well as future requests. Two AGV dispatching strategies based on combinatorial optimization of assignment problems were developed with different decision making horizons. In the first strategy, AGV assignment decisions are iteratively made for two consecutive requests, and in the second strategy the assignment decisions are made for all current jobs in each work station. The results of the case study show that the proposed AGV assignment strategies result in shorter job waiting time than classic AGV assignment rules, which is critical in many production scenarios, such as steel and food industries that jobs cannot be exposed to room temperature or natural environments for too long.

The first study suggests two research extensions. First, besides optimization based on the assignment problems in network optimization, additional optimization of AGV dispatching, such as models of vehicle routing problems in network optimization, should have better performance in improving the job shop efficiency. In fact, classic JSSMH models considering job scheduling and AGV routing can be formulated to achieve the global optimal makespan of a job shop. The second extension direction is to focus on taking variability in production systems into consideration to make the optimization on the system more robust, and JSSMH models can be modified to account for processing time with multiple kinds of variability. Both extensions are realized in the following two studies.

In the second study, the linearized optimization model of JSSMH is formulated, and a heuristic algorithm is proposed to solve the problem instead of exact solution to achieve a good quality solution in a reasonable amount of time. Given that available AGV fleet size is smaller than the job set size, the proposed algorithm starts with the scenario that AGV fleet size is same as the number of jobs. With this assumption, the job schedule solved by the Job Shop Scheduling model without material handling would be a feasible solution and can be found relatively easily. In each iteration, AGV fleet size is reduced by 1, and whenever an AGV is removed from the system, the operations served by the removed AGV are reassigned to remaining AGVs according to a series of heuristic rules, while the incumbent schedule may also be adjusted. The algorithm ends when all the operations are handled by AGVs, and the remaining AGV fleet size matches with the original AGV availability. Overall the proposed algorithm can provide an optimal or near-optimal solution very efficiently, and this would enable real time scheduling and reactive scheduling on the shop floor when decisions must be made in a short time. To illustrate the algorithm, a new visualization method extending traditional Gantt charts is proposed to reflect the interaction between AGV movements and job operations. Same with the first study in this dissertation, in this second study, the processing time is assumed to be known with certainty, however, the uncertain processing time is very common in real industrial applications. This consideration of variabilities of processing time serves as the major motivation for the third study.

In the third study, three models are formulated to incorporate variable processing time in job shop scheduling problems with material handling. Based on literature review and anecdotal information, the two common types of variabilities in processing time are uncertainty (randomness) and deterioration. Random processing time in production scheduling problems usually results from inaccurate data collection or uncontrollable operations, and deterioration describes the phenomenon that processing becomes less efficient as production moves on, resulting in longer processing time. When processing times are random and follow specific distributions, a two-stage stochastic programming model is formulated to minimize the expectation of makespan across a series of scenarios discretized form the distribution of processing time. Deteriorating processing time can be linear to operation start time or exponential to operations' sequence on a machine, hence when deterioration is considered in modeling, two models are formulated to incorporate the deterioration functions respectively. Modeling techniques are proposed to linearize the nonlinear model and ensure the model solvability. The necessity of this study is supported by comparing the makespan based proposed models and solutions of original models without considering the variable processing times. Based on the case studies, the proposed models considering variable processing time outperform the original models in minimizing the makespan under random or deteriorating processing time.

To summarize, this dissertation focuses on the JSSMH problem, which has been addressed comprehensively with AGV assignment, classic modeling and corresponding solution techniques, and extensions for variable production parameters. Multiple theories are covered, including classic JSSMH modeling, AGV assignment problems as a simplification of JSSMH, an extension beyond JSSMH considering randomness and deterioration. Optimization models in different types are formulated, including linear programming, mix-integer linear programming and nonlinear programming. Various tools are utilized in the series of studies to validate and implement the proposed models and solution techniques. Simulation models are constructed to study the AGV movement and shop floor workflow, and they are utilized as the platform to test existing AGV assignment rules and proposed strategies based on optimization. The simulation platform also contributes to a good reference in validating the models and solution techniques in the following studies at early stages. The mathematical models are coded with various programming languages including. In the first study, models are coded with JAVA to iteratively solve optimization models in simulation platform. In the second study, the proposed algorithm is realized with R, while original models are coded with AMPL and solved on NEOS public solvers for the algorithm validation. In the third study, the stochastic programming model is coded with Python to call

Pyomo with specific algorithms for stochastic programming, while models considering deterioration are coded in AMPL.

This dissertation is subject to a few limitations which suggest future research directions. First, in job shop scheduling studies, besides random and deteriorating processing time, shortening processing time is also sometimes reported, mainly due to the learning effect of workers, who become increasingly proficient in the with production moving on. The learning process could be described with much more complicated models than random distributions and deterioration functions, and the scenario can be even more complicated if the combinations of them are considered. Therefore, the modeling of JSSMH could be expanded to incorporated workers' learning effects and the mixed effects of learning, deterioration and randomness. Second, with specific shop floor configuration, JSSMH should meet many additional requirements in application, such as avoiding AGV collision, reducing AGV congestion, and instant response to jobs with preemption; therefore the JSSMH model ask for further modification and this might bring more challenge to computation. Third, there are some JSSMH cases that are extremely difficult for commercial solvers and need special attention, hence better solution techniques could be developed in future research to ensure solvability of JSSMH models and its extensions.

BIBLIOGRAPHY

Abbas, A. S., Mohamed, T. A., & Hafez, M. G. (2014). Improving Localization Accuracy of Autonomous Guided Vehicles for Material Handling Applications. *Journal of Management & Engineering Integration*, *7*(1), 82.

Abdelmaguid, T. F., Nassef, A. O., Kamal, B. a., & Hassan, M. F. (2004). A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, *42*(2), 267–281. http://doi.org/10.1080/0020754032000123579

Ahmadi-Javid, A., & Hooshangi-Tabrizi, P. (2015). A mathematical formulation and anarchic society optimisation algorithms for integrated scheduling of processing and transportation operations in a flow-shop environment. *International Journal of Production Research*, *53*(19), 5988–6006. http://doi.org/10.1080/00207543.2015.1035812

Ahmadi-Javid, A., & Hooshangi-Tabrizi, P. (2017). Integrating employee timetabling with scheduling of machines and transporters in a job-shop environment: A mathematical formulation and an Anarchic Society Optimization algorithm. *Computers and Operations Research*, *84*, 73–91. http://doi.org/10.1016/j.cor.2016.11.017

Albert Jones, & C.Rabelo, L. (1999). Survey of Job Shop Scheduling Techniques. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 17. http://doi.org/10.1002/047134608X.W3352

Almeder, C., Preusser, M., & Hartl, R. F. (2009). Simulation and optimization of supply chains: Alternative or complementary approaches? *OR Spectrum*, *31*(1), 95–119. http://doi.org/10.1007/s00291-007-0118-z

Ashour, S., & Hiremath, S. R. (1973). A branch-and-bound approach to the job-shop scheduling problem. *International Journal of Production Research*, *11*(1), 47–58.

Baños, R., Ortega, J., Gil, C., Márquez, A. L., & De Toro, F. (2013). A hybrid meta-heuristic for multi-objective Vehicle Routing Problems with Time Windows. *Computers and Industrial Engineering*, *65*(2), 286–296. http://doi.org/10.1016/j.cie.2013.01.007

Baruwa, O. T., & Piera, M. A. (2016). A coloured Petri net-based hybrid heuristic search approach to simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, *54*(16), 4773–4792. http://doi.org/10.1080/00207543.2015.1087656

Beamon, B. M. (1998). Performance, reliability, and performability of material handling systems. *International Journal of Production Research*, *36*(2), 377–393. http://doi.org/10.1080/002075498193796

Bertsekas, D. P. (1998). *Network Optimization : Continuous and Discrete Models*. Belmont: Athena Scientific.

Bilge, Ü., & Ulusoy, G. (1995). A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS. *Operations Research*, *43*(6), 1058–1070. Retrieved from http://www.jstor.org/stable/171645

Bonfill, A., Espuna, A., & Puigjaner, L. (2005). Addressing robustness in scheduling batch processes with uncertain operation times. *Industrial and Engineering Chemistry Research*, *44*(5), 1524. http://doi.org/doi:10.1021/ie049732g

Browne, J., Dubois, D., Rathmill, K., Sethi, S. P., & Stecke, K. E. (1984). Classification of flexible manufacturing systems. *FMS Magazine*, *2*, 114–117. Retrieved from https://s3.amazonaws.com/academia.edu.documents/31443546/BrowneDuboisRathSethiSteckeFMS1984.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1505313723&Signature=1gyAZHpFBh7SJWo%2FVly9aJ2X81E%3D&response-content-disposition=inline%3B filename%3DClassificat

Bürgy, R., & Gröflin, H. (2016). The blocking job shop with rail-bound transportation. *Journal of*

Combinatorial Optimization, *31*(1), 152–181.

Carlo, H. J., Vis, I. F. A., & Roodbergen, K. J. (2014). Transport operations in container terminals : Literature overview , trends , research directions and classification scheme. *Eropean Journal of Operational Research*, *236*, 1–13. http://doi.org/10.1016/j.ejor.2013.11.023

Chaudhry, I. A., & Khan, A. A. (2016). A research survey: Review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, *23*(3), 551–591. http://doi.org/10.1111/itor.12199

Chiang, T.-C., & Lin, H.-J. (2013). A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling. *International Journal of Production Economics*, *141*(1), 87–98. http://doi.org/10.1016/j.ijpe.2012.03.034

Confessore, G., Fabiano, M., & Liotta, G. (2013). A network flow based heuristic approach for optimising AGV movements. *Journal of Intelligent Manufacturing*, *24*(2), 405–419. http://doi.org/10.1007/s10845-011-0612-7

Culler, D., & Long, J. (2016). A Prototype Smart Materials Warehouse Application Implemented Using Custom Mobile Robots and Open Source Vision Technology Developed Using EmguCV. *Procedia Manufacturing*, *5*, 1092–1106. http://doi.org/10.1016/j.promfg.2016.08.080

Czyzyk, J., Mesnier, M. P., & Moré, J. J. (1998). The NEOS Server. *IEEE Journal on Computational Science and Engineering*, *5*(3), 68–75.

Dang, Q.-V., & Nguyen, L. (2017). Makespan minimisation for pre-emptive scheduling of mobile robots in FMS environment. *International Journal of Metaheuristics*, *6*(1–2), 3–36.

Deroussi, L., Gourgand, M., & Tchernev, N. (2008). A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Research*, *46*(8), 2143–2164. http://doi.org/10.1080/00207540600818286

Doh, H., Yu, J., & Kim, J. (2013). A priority scheduling approach for flexible job shops with multiple process plans. *International Journal of Production Research*, *51*(February 2015), 3748–3764. http://doi.org/10.1080/00207543.2013.765074

Dolan, E. (2001). *The NEOS Server 4.0 Administrative Guide*.

Egbelu, P. (1987). The use of non-simulation approaches in estimating vehicle requirements in an automated guided vehicle based transport system. *Material Flow*, *4*(1), 17–32.

Egbelu, P., & Tanchoco, J. M. A. (1984). Characterization of automatic guided vehicle dispatching rules. *International Journal of Production Research*, *22*(3), 359–374.

El Maraghy, H. A. (2006). Flexible and reconfigurable manufacturing systems paradigms. *Flexible Services and Manufacturing Journal*, *17*(4 SPECIAL ISSUE), 261–276. http://doi.org/10.1007/s10696-006-9028-7

Fauadi, M. H. F. bin M., Yahaya, S. H., & Murata, T. (2013). Intelligent Combinatorial Auctions of Decentralized Task Assignment for AGV With Multiple Loading Capacity. *IEEJ Transactions on Electrical and Electronic Engineering*, *8*, 371–379. http://doi.org/10.1002/tee.21868

Fazlollahtabar, H. (2016). Parallel autonomous guided vehicle assembly line for a semi-continuous manufacturing system. *Assembly Automation*, *36*(3), 262–273. http://doi.org/10.1108/AA-08-2015-065

Fink, A., Kliewer, N., Mattfeld, D., Mönch, L., Rothlauf, F., Schryen, G., … Voß, S. (2014). Model-Based Decision Support in Manufacturing and Service Networks. *Business & Information Systems Engineering*, *6*(1), 17–24. http://doi.org/10.1007/s12599-013-0310-4

Gawiejnowicz, S., Lee, W. C., Lin, C. L., & Wu, C. C. (2011). Single-machine scheduling of proportionally deteriorating jobs by two agents. *Journal of the Operational Research Society*, *62*(11), 1983–1991. http://doi.org/10.1057/jors.2010.171

Gropp, W., & Moré, J. J. (1997). Optimization Environments and the NEOS Server. *Approximation Theory and Optimization*, 167–182.

Gupta, J. N. D., & Gupta, S. K. (1988). Single facility scheduling with nonlinear processing times. *Computers and Industrial Engineering*, *14*(4), 387–393. http://doi.org/10.1016/0360-8352(88)90041-1

Han, L., Xing, K., Chen, X., Lei, H., & Wang, F. (2014). Deadlock-free genetic scheduling for flexible manufacturing systems using Petri nets and deadlock controllers. *International Journal of Production Research*, *52*(5), 1557–1572. http://doi.org/10.1080/00207543.2013.849012

Huang, S., Brown, C., & Hu, G. (2017). Shop Floor AGV Assignment Optimization with Uncertain Request Arrival. In K. Coperich, E. Cudney, & H. Nembhard (Eds.), *Proceedings of the 2017 IIE Annual Conference*. Pittsburgh. Retrieved from https://www.xcdsystem.com/iise/2017_proceedings/prof2693.html

Huang, S., & Hu, G. (2017a). A Vehicle Reducing Algorithm for Job Shop Scheduling with Material Handling. *Computers & Operations Research, In Review*.

Huang, S., & Hu, G. (2017b). Automated Guided Vehicle Dispatching Based on Network Optimization in Shop Floors. *International Journal of Production Research, In Review*.

Huang, S., & Hu, G. (2018). A Two-stage Stochastic Programming Model for Job Shop Scheduling with Material Handling under Production Uncertainty, *In Progres*.

Huang, X., Wang, M. Z., & Ji, P. (2014). Parallel machines scheduling with deteriorating and learning effects. *Optimization Letters*, *8*(2), 493–500. http://doi.org/10.1007/s11590-012-0490-8

Izabela Nielsen, Dang, Q.-V., Nielsen, P., & Pawlewski, P. (2014). Scheduling of Mobile Robots with Preemptive Tasks. In *Distributed Computing and Artificial Intelligence, 11th International Conference*.

Joe, Y. Y., Gan, O. P., & Lewis, F. L. (2014). Multi-commodity flow dynamic resource assignment and matrix-based job dispatching for multi-relay transfer in complex material handling systems (MHS). *Journal of Intelligent Manufacturing*, *25*(4), 681–697. http://doi.org/10.1007/s10845-012-0713-y

Khayat, G. El, Langevin, A., & Riopel, D. (2006). Integrated production and material handling scheduling using mathematical programming and constraint programming. *European Journal of Operational Research*, *175*(3), 1818–1832. http://doi.org/10.1016/j.ejor.2005.02.077

Kim, J., Choe, R., & Ryu, K. R. (2013). Multi-objective optimization of dispatching strategies for situation-adaptive AGV operation in an automated container terminal. *Proceedings of the 2013 Research in Adaptive and Convergent Systems on - RACS '13*, 1–6. http://doi.org/10.1145/2513228.2513277

Kim, K. S., Chung, B. Do, & Jae, M. (2003). A design for a tandem AGVS with multi-load AGVs. *International Journal of Advanced Manufacturing Technology*, *22*(9–10), 744–752. http://doi.org/10.1007/s00170-003-1614-1

Kumar, R., Haleem, A., Garg, S. K., & Singh, R. K. (2015). Automated guided vehicle configurations in flexible manufacturing systems : A comparative study. *International Journal of Industrial and Systems Engineering*, *21*(2), 207–226. http://doi.org/10.1504/IJISE.2015.071510

Kuo, W. H., Hsu, C. J., & Yang, D. L. (2012). Worst-case and numerical analysis of heuristic algorithms for flowshop scheduling problems with a time-dependent learning effect. *Information Sciences*, *184*(1), 282–297. http://doi.org/10.1016/j.ins.2011.08.018

Lee, L. H., Chew, E. P., Tan, K. C., & Wang, Y. (2010). Vehicle dispatching algorithms for container transshipment hubs. *OR Spectrum*, *32*(3), 663–685. http://doi.org/10.1007/s00291-010-0199-y

Lee, W. C., Shiau, Y. R., Chen, S. K., & Wu, C. C. (2010). A two-machine flowshop scheduling problem with deteriorating jobs and blocking. *International Journal of Production Economics*, *124*(1), 188–

197. http://doi.org/10.1016/j.ijpe.2009.11.001

Lenstra, J. K., & Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, *11*(2), 221–227.

Liu, C., Fan, Y., Zhao, C., & Wang, J. (2017). Multiple common due-dates assignment and optimal maintenance activity scheduling with linear deteriorating jobs. *Journal of Industrial and Management Optimization*, *13*(2), 713–720. http://doi.org/10.3934/jimo.2016042

Luo, J., & Wu, Y. (2015). Modelling of dual-cycle strategy for container storage and vehicle scheduling problems at automated container terminals. *Transportation Research Part E: Logistics and Transportation Review*, *79*, 49–64. http://doi.org/10.1016/j.tre.2015.03.006

Luo, J., Wu, Y., & Mendes, A. B. (2016). Modelling of integrated vehicle scheduling and container storage problems in unloading process at an automated container terminal. *COMPUTERS & INDUSTRIAL ENGINEERING*, *94*, 32–44. http://doi.org/10.1016/j.cie.2016.01.010

Na, B., Woo, J. E., & Lee, J. (2016). Lifter assignment problem for inter-line transfers in semiconductor manufacturing facilities. *International Journal of Advanced Manufacturing Technology*, *86*(5–8), 1615–1626. http://doi.org/10.1007/s00170-015-8327-0

Nageswararao, M., Rao, K. N., & Rangajanardhana, G. (2012). INTEGRATION OF STRATEGIC TACTICAL AND OPERATIONAL LEVEL PLANNING OF SCHEDULING IN FMS. *International Journal of Advanced Engineering Research and Studies*, *1*(2).

Ng, T. L., Eheart, J. W., Cai, X., & Braden, J. B. (2011). An agent-based model of farmer decision-making and water quality impacts at the watershed scale under markets for carbon allowances and a second-generation biofuel crop. *Water Resources Research*. http://doi.org/10.1029/2011WR010399

Novas, J. M., & Henning, G. P. (2014). Integrated scheduling of resource-constrained flexible manufacturing systems using constraint programming. *Expert Systems with Applications*, *41*(5), 2286–2299. http://doi.org/10.1016/j.eswa.2013.09.026

Pezzella, F., Morganti, G., & Ciaschetti, G. (2008). A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers and Operations Research*, *35*(10), 3202–3212. http://doi.org/10.1016/j.cor.2007.02.014

Pinedo, M. (2009). *Planning and Scheduling in Manufacturing and Services*. Springer. http://doi.org/10.1007/978-1-4419-0910-7

Pisuchpen, R. (2012). A simulation of AGV dispatching rules for supplying parts to assembly lines. *International Journal of Modelling in Operations Management*, *2*(1), 1–25.

R Core Team. (2015). R: A language and environment for statistical computing. Austria. Retrieved from http://www.r-project.org/

Reddy, B. S. P., & Rao, C. S. P. (2006). A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS. *International Journal of Advanced Manufacturing Technology*, *31*(5–6), 602–613. http://doi.org/10.1007/s00170-005-0223-6

Rustogi, K., & Strusevich, V. A. (2012). Simple matching vs linear assignment in scheduling models with positional effects: A critical review. *European Journal of Operational Research*, *222*(3), 393–407. http://doi.org/10.1016/j.ejor.2012.04.037

Sacone, S., & Siri, S. (2009). An integrated simulation-optimization framework for the operational planning of seaport container terminals. *Mathematical and Computer Modelling of Dynamical Systems*, *15*(3), 275–293.

Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of Operational Research*, *120*(2), 393–407. http://doi.org/10.1016/S0377-2217(99)00094-6

Tanaka, Y., Nishi, T., & Inuiguchi, M. (2010). Systems , and Dynamic Optimization of Simultaneous

Dispatching and Conflict-free Routing for Automated Guided Vehicles - Petri Net Decomposition Approach - ∗. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *4*(3), 701–715. http://doi.org/10.1299/jamdsm.4.701

Ulusoy, G., Sivrikaya-Şerifoğlu, F., & Bilge, Ü. (1997). A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. *Computers & Operations Research*, *24*(4), 335–351. http://doi.org/10.1016/S0305-0548(96)00061-5

Umar, U. A., Ariffin, M. K. A., Ismail, N., & Tang, S. H. (2013). Conflict-Free Automated Guided Vehicles Routing Using Multi-Objective Genetic Algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, *6*(14), 2681–2684.

Umar, U. A., Ariffin, M. K. A., Ismail, N., & Tang, S. H. (2015). Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. *International Journal of Advanced Manufacturing Technology*, *81*(9–12), 2123–2141. http://doi.org/10.1007/s00170-015-7329-2

Villarreal, B., Garza-Reyes, J. A., & Kumar, V. (2016). A lean thinking and simulation-based approach for the improvement of routing operations. *Industrial Management & Data Systems*, *116*(5), 903–925. http://doi.org/10.1108/IMDS-09-2015-0385

Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, *170*(3), 677–709. http://doi.org/10.1016/j.ejor.2004.09.020

Wang, C., Guan, Z., Shao, X., & Ullah, S. (2014). Simulation-based optimisation of logistics distribution system for an assembly line with path constraints. *International Journal of Production Research*, *52*(12), 3540–3553. http://doi.org/10.1080/00207543.2013.874602

Xiao, Y. J., Zheng, Y., & Jia, Q. (2014). An iterated greedy algorithm for the integrated scheduling problem with setup times. In *Information, Computer and Application Engineering: Proceedings of the International Conference on Information Technology and Computer Application Engineering*. Hong Kong.

Xiao, Y. Y., Zhang, R. Q., Zhao, Q. H., & Kaku, I. (2012). Permutation flow shop scheduling with order acceptance and weighted tardiness. *Applied Mathematics and Computation*, *218*(15), 7911–7926. http://doi.org/10.1016/j.amc.2012.01.073

Xie, C., & Allen, T. T. (2015). Simulation and experimental design methods for job shop scheduling with material handling : a survey. *International Journal of Advanced Manufacturing Technology*, *80*(1–4), 233–243. http://doi.org/10.1007/s00170-015-6981-x

Yang, Y., Chen, W., Wei, L., & Chen, X. (2018). Robust optimization for integrated scrap steel charge considering uncertain metal elements concentrations and production scheduling under time-of-use electricity tariff. *Journal of Cleaner Production*, *176*, 800–812. http://doi.org/10.1016/j.jclepro.2017.12.094

Zhang, X., Wu, W. H., Lin, W. C., & Wu, C. C. (2018). Machine scheduling problems under deteriorating effects and deteriorating rate-modifying activities. *Journal of the Operational Research Society*, *5682*, 1–10. http://doi.org/10.1057/s41274-017-0200-0

Zhang, Y., Huang, G. Q., Sun, S., & Yang, T. (2014). Multi-agent based real-time production scheduling method for radio frequency identification enabled ubiquitous shopfloor environment. *COMPUTERS & INDUSTRIAL ENGINEERING*, *76*, 89–97. http://doi.org/10.1016/j.cie.2014.07.011

Zheng, Y., Xiao, Y., & Seo, Y. (2016). A tabu search algorithm for simultaneous machine / AGV scheduling problem. *International Journal of Production Research*, *52*(19), 5749–5764. http://doi.org/10.1080/00207543.2014.910628