



YAŞAR UNIVERSITY
GRADUATE SCHOOL

MASTER THESIS

**A NOVEL SCENARIO-BASED ROBUST SOLUTION
APPROACH TO THE PROBLEM OF SCHEDULING
AND ROUTING OF AUTOMATED GUIDED VEHICLES
(AGVS) IN A PRODUCTION ENVIRONMENT**

SELEN BURÇAK AKKAYA

THESIS ADVISOR: ASSIST. PROF.DR. MAHMUT ALİ GÖKÇE

INDUSTRIAL ENGINEERING

PRESENTATION DATE: 04.08.2022

BORNOVA / İZMİR
AUGUST 2022

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Jury Members:

Signature:

Assist. Prof. Dr. Mahmut Ali GÖKÇE
Yaşar University

.....

Prof. Dr. Mustafa Arslan ÖRNEK
Yaşar University

.....

Assist. Prof. Dr. Kamil Erkan KABAK
İzmir University of Economics

.....



Prof. Dr. Yücel Öztürkoğlu
Director of the Graduate School

ABSTRACT

A NOVEL SCENARIO-BASED ROBUST SOLUTION APPROACH TO THE PROBLEM OF SCHEDULING AND ROUTING OF AUTOMATED GUIDED VEHICLES (AGVS) IN A PRODUCTION ENVIRONMENT

AKKAYA, Selen Burçak

MSc, Industrial Engineering MSc (With Thesis/In English)

Advisor: Assist. Prof. Dr. Mahmut Ali GÖKÇE

August 2022

Use of Automated Guided Vehicles (AGVs) for material handling purposes has become increasingly popular. They introduce flexibility to the system by increasing speed, responsiveness, and freight capacity as well as enabling increased productivity, safety, efficient resource utilization and reducing costs. These advantages can be realized by intelligent assignment of AGVs to jobs, and routing of AGVs to help meet production plans. This thesis investigates the problem of scheduling and routing of a heterogenous fleet of AGVs, consisting of different types based on purpose of use, freight, and battery charge capacity used for handling transfer jobs in a job shop production environment. The objective is to optimize the schedules and routes of AGVs by minimizing total of the penalty cost for the late delivery of parts and energy consumption of the vehicles. In doing so, the thesis contributes to the literature by proposing a novel, scenario based robust mixed integer linear programming model for a heterogenous fleet of AGVs along with charging and energy consumption where partial recharging is allowed. Proposed model is first verified with a test case. Through design of experiment, performance of the model and how certain parameters affect the performance of the model is also investigated and results are reported.

keywords: AGV, Routing, Scheduling, Scenario Based Robust Optimization

ÖZ

ÜRETİM ORTAMINDAKİ OTOMATİK YÖNLENDİRİLMELİ ARAÇLARIN (AGVS) ÇİZELGELEME VE ROTALAMA PROBLEMİ İÇİN YENİ BİR SENARYO TEMELLİ GÜRBÜZ ÇÖZÜM YÖNTEMİ

AKKAYA, Selen Burçak

Yüksek Lisans Tezi/ Endüstri Mühendisliği Y. Lisans (Tezli/İngilizce)

Danışman: Dr. Öğrt. Üyesi Dr. Mahmut Ali GÖKÇE

Ağustos 2022

Otomatik Yönlendirmeli Araçların (AGV'ler) malzeme taşıma amacıyla kullanımı giderek daha popüler hale gelmiştir. Hızlı ulaşım sağlamaları, görevlere hızlıca atanabilmeleri ve nakliye kapasitesini arttırmalarının yanı sıra üretkenlik, güvenlik, kaynakların kullanım verimini arttırarak ve maliyetleri azaltarak, sisteme esneklik kazandırmaktadırlar. Bu faydaların elde edilebilmesi için AGV'lerin işlere sistematik bir şekilde atanması ve AGV'lerin üretim planlarını karşılayacak şekilde rotalanması ve çizelgelenmesi gerekir. Bu tez, bir atölye üretim ortamında istasyonlar arası taşıma işlerini sağlamak amacıyla kullanılan, kullanım amacına, taşıma ve pil şarj kapasitesine göre farklı tiplerden oluşan heterojen bir AGV filosunun çizelgeleme ve rotalama problemini incelemektedir. Amaç, siparişlere ait taşıma işlerinin geç teslimine atanan ceza maliyetini ve araçların enerji tüketimini en aza indireren gürbüz bir atama ve AGV'lere ait gürbüz rotaları ve çizelgeleri bulmaktır. Bunun amaçla bu tez, heterojen bir AGV filosu için bu kararların yanı sıra şarj planı ve enerji tüketimini de gözetken, senaryo tabanlı yeni bir gürbüz karma tamsayılı doğrusal programlama modeli önerilerek literatüre katkıda bulunmaktadır. Önerilen model öncelikle, IBM OPL CPLEX kullanılarak bir test senaryosu ile doğrulanmıştır. Yapılan deney tasarımı ile önerilen modelin performansı ve belirli parametrelerin model üzerindeki etkisi değerlendirilmiştir.

Anahtar Kelimeler: AGV, Rotalama, Çizelgeleme, Senaryo Tabanlı Gürbüz Optimizasyon

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis advisor Asst. Prof. Dr. Mahmut Ali GÖKÇE for his guidance and patience during this study. His motivation, knowledge, and enthusiasm helped me during the time of research and writing of this thesis. This thesis would not have been possible without his outstanding assistance at every stage of the process.

I would like to express my enduring love to my parents, who are always supportive, loving and caring to me in every possible way in my life.

I would like to express my gratitude to my fellow graduate student friends Aykut GÜL, Özge ÖZBİRİNCİ, Ali Mert CEYLAN and Ege DURAN for their help during my research.

Selen Burçak Akkaya
İzmir, 2022

TEXT OF OATH

I declare and honestly confirm that my study, titled “A NOVEL SCENARIO-BASED ROBUST SOLUTION APPROACH TO THE PROBLEM OF SCHEDULING AND ROUTING OF AUTOMATED GUIDED VEHICLES (AGVS) IN A PRODUCTION ENVIRONMENT” and presented as a Master’s Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Selen Burçak Akkaya

Signature

.....

August 17, 2022

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGEMENTS	ix
TEXT OF OATH	xi
TABLE OF CONTENTS	xiii
LIST OF FIGURES	xvii
LIST OF TABLES	xix
SYMBOLS AND ABBREVIATIONS	xxiii
CHAPTER 1 INTRODUCTION	1
1.1. Structure of AGV Systems	3
1.2. Uncertainties in AGV Operations	5
1.2.1. Robust Optimization Approach	6
CHAPTER 2 PROBLEM DEFINITION	9
2.1. Scope of the Problem	9
2.2. Assumptions	14
CHAPTER 3 LITERATURE REVIEW	17
3.1. AGV Problems in Literature	17
3.2. Robust Optimization and AGVs	21
CHAPTER 4 SOLUTION METHODOLOGY	27
4.1. Mathematical Model	27
4.2. Scenario-based Robust Linear Optimization Approach	32
4.2.1. Uncertainty and Scenario Set Creation	33
4.2.2. Scenario Selection and Weight Assumption	34
4.2.3. Scenario-based MILP	36
CHAPTER 5 COMPUTATIONAL RESULTS AND ANALYSIS	39
5.1. Model Verification	39
5.1.1. Verification of Initially Developed MILP with No Uncertainty	39

5.1.2. Scenario Based Model Verification	44
5.2. Experimental Design, Results and Analysis.....	48
CHAPTER 6 CONCLUSIONS AND FUTURE RESEARCH	55
REFERENCES.....	57
APPENDIX 1 – Computational Results of the Scenario BAsed Model Validation	61
APPENDIX 2 – Computational Results of the Experimental Design	63
APPENDIX 3 – Solution Method Implementation in CPLEX	65



LIST OF FIGURES

Figure 1. Figure of an AGV	2
Figure 2. Different types of AGVs.....	4
Figure 3. Network presentation of job orders	11
Figure 4. AGV trip scheme flow chart.....	13
Figure 5. Physical network representation of the test instance	41
Figure 6. Comparison of Order Size and Route Complexity vs CPU time (in seconds)	53
Figure 7. Comparison of Order Size and Route Complexity vs Optimality Gap %	53
Figure A1.1. Time Calculations for All Scenarios for Model Verification Example	62
Figure A2.1. Time Calculations for All Scenarios for 5 orders with low complexity in 10 Workstation Job Shop	64



LIST OF TABLES

Table 1. Job order routes and final due dates	10
Table 2. Structure of transportation jobs	11
Table 3. Summary of selected papers on AGV problems	19
Table 4. Summary of selected papers on robustness and robust AGV problems.....	22
Table 5. Scenario set creation example	34
Table 6. Likelihood Calculation of Scenarios	35
Table 7. Scenario Selection and Weight Assignment	36
Table 8. Production route and schedule of job orders of test instance	39
Table 9. Transfer jobs list for the test instance	40
Table 10. Network Information of the test instance	40
Table 11. AGV fleet in the test instance	42
Table 12. Job assignment and schedule of the test instance.....	42
Table 13. Routes of AGVs of the test instance	43
Table 14. Discrete scenarios generated for the test instance	44
Table 15. Optimal robust assignments using scenario based model	45
Table 16. Optimal robust routes using scenario based model	46
Table 17. Schedule of jobs for scenario 14 using scenario based approach.....	46
Table 18. Optimal AGV assignment and schedule for jobs of scenario 14 instances.....	47
Table 19. Optimal routes for AGV s for scenario 14 instances	47
Table 20. Input for the Design of Experiments	49
Table 21. Production Route Complexity	49
Table 22. Results of 10 Workstation Instances	50
Table 23. Results of 15 Workstation Instances	51
Table 24. Results of 20 Workstation Instances	52
Table A1.1. Robust AGV-Job Assignment of Test Instance	61



Table A1.2. Robust Routes and Charge Plan of AGVs of Test Instance	61
Table A2.1. Robust AGV-Job Assignment of 5 orders with low complexity in 10 Workstation Job Shop	63
Table A2.2. Robust Routes and Charge Plan of AGVs of 5 orders with low complexity in 10 Workstation Job Shop	63



SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS:

AGV	Automated Guided Vehicle
FMS	Flexible Manufacturing Systems
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows
RO	Robust Optimization
RDO	Robust Discrete Optimization
RC	Robust Counterpart
SO	Stochastic Optimization
IP	Integer Programming
MILP	Mixed Integer Linear Programming
SBMILP	Scenario Based Mixed Integer Linear Programming

SYMBOLS:

h, i, j	Location (node) indices
v	Vehicle index
w	Job index
sc	Scenario index
S	Set of discrete scenarios created using uncertain parameters ready and due times
D	Set of depot locations, $D = \{0\}$,
CS	Set of charging station locations,
PP	Set of pick-up points,
DP	Set of delivery points,
SP	Set of service (pickup/delivery) points, $SP = PP \cup DP$



N	Set of locations (nodes), $N = D \cup CS \cup SP$
$N0$	Set of locations (nodes) excluding the depot, $N0 = N/D$
A	Set of lanes (edges), $A = \{(i,j) : i, j \in N \ \& \ i \neq j\}$,
J	Set of jobs,
$Orj[J]$	Set of source locations (nodes (i)) of all jobs (w), $w \in J, i \in SP$,
$Des[J]$	Set of destination locations (nodes (i)) of all jobs (w), $w \in J, i \in SP$,
V	Set of AGVs, $V = V_{max} \cup V_{mid} \cup V_{min}$
V_{max}	Set of maximum sized AGVs
V_{mid}	Set of mid-sized AGVs
V_{min}	Set of minimum sized AGVs
$weight_{sc}$	Weight of scenario sc , $sc \in S$
d_{ij}	Distance between locations (nodes) i and j , $(i,j) \in A$
l_w	Load amount of job w , $w \in J$
r_w	Ready time of job w , $w \in J$
due_w	Due time of job w , $w \in J$
r_w^{sc}	Ready time of job w under scenario sc , $w \in J, sc \in S$
due_w^{sc}	Due time of job w under scenario sc , $w \in J, sc \in S$
s_w	Service time (loading/unloading) of job w , $w \in J$
C_v	Load capacities of the AGV fleet, $v \in V$
C_{max}	Load capacity of maximum sized AGVs, $v \in V_{max}$
C_{mid}	Load capacity mid-sized AGVs, $v \in V_{mid}$
C_{min}	Load capacity of minimum sized AGVs, $v \in V_{min}$
$InCharge_v$	Initial battery charge of AGV v , $v \in V$
B_v	Battery capacity of the AGV fleet, $v \in V$
B_{max}	Battery capacity of maximum sized AGVs, $v \in V_{max}$
B_{mid}	Battery capacity of mid-sized AGVs, $v \in V_{mid}$



B_{min}	Battery capacity of minimum sized AGVs, $v \in V_{min}$
$speed_v$	Speed (distance per minute) of the AGV fleet, $v \in V$
$speed_{max}$	Speed of maximum sized AGVs, $v \in V_{max}$
$speed_{mid}$	Speed of mid-sized AGVs, $v \in V_{mid}$
$speed_{min}$	Speed of minimum sized AGVs, $v \in V_{min}$
crl_v	Battery charge consumption of AGV v rate while travelling, units per distance, $v \in V$
crl_{max}	Unit battery charge consumption rate while travelling of maximum sized AGVs, $v \in V_{max}$
crl_{mid}	Unit battery charge consumption rate while travelling of mid-sized AGVs, $v \in V_{mid}$
crl_{min}	Unit battery charge consumption rate while travelling of minimum sized AGVs, $v \in V_{min}$
cro_v	Fixed battery charge consumption of AGV v (in units) while loading/unloading operations $v \in V$
cro_{max}	Unit battery charge consumption rate while loading/unloading operations of maximum sized AGVs, $v \in V_{max}$
cro_{mid}	Unit battery charge consumption rate while loading/unloading operations of mid-sized AGVs, $v \in V_{mid}$
cro_{min}	Unit battery charge consumption rate while loading/unloading operations of minimum sized AGVs, $v \in V_{min}$
CR_i	Indicator of charging stations, which is equal to 1 if of node i is a charging station, zero otherwise, $i \in N$
$chargeRate$	Charging rate (time per unit),
c_{late}	Penalty cost of delivering a job late, cost per unit time
c_{charge}	Cost of charging, cost per unit
c_{energy}	Penalty cost of energy utilization, cost per unit



x_{ijv}	Binary decision variable indicating if AGV v travels to node j immediately after visiting node i where $v \in V, (i,j) \in A$
y_{wv}	Binary decision variable indicating if job w , is assigned to AGV v , $w \in J, v \in V$
c_{vi}	units charged on node i , $v \in V, i \in N$
a_{vij}	arrival time of AGV v to node j travelling from node i , $v \in V, (i,j) \in A$
a_{vij}^{sc}	arrival time of AGV v to node j travelling from node i under scenario sc , $v \in V, (i,j) \in A$,
pt_w	pick-up time of job w , $w \in J$
pt_w^{sc}	pick-up time of job w under scenario sc , $w \in J, sc \in S$
dt_w	delivery time of job w , $w \in J$
dt_w^{sc}	delivery time of job w under scenario sc , $w \in J, sc \in S$
$tardiness_w$	tardiness duration of job w , $w \in J$
$tardiness_w^{sc}$	tardiness duration of job w under scenario sc , $w \in J, sc \in S$
b_{vij}	battery charge of AGV v when it arrives to node j travelling from node i , $v \in V, (i,j) \in A$
$load_{vij}$	load amount of AGV v when it arrives to node j travelling from node i , $v \in V, (i,j) \in A$

CHAPTER 1

INTRODUCTION

Technological advances over past decade have enabled many more smart systems to be used in production environment. Digitalization of manufacturing systems have become inevitable as a result of fourth industrial revolution, commonly referred as Industry 4.0. With Industry 4.0, the aim is to achieve a smart factory environment where advanced manufacturing technologies are used, and human involvement is as little as possible in decision making processes through cyber-physical systems (CPS). (Zhong et al.,2017) Many technological developments such as Internet of Things (IoT), cloud computing, additive manufacturing (3D printing) and RFID (Radio Frequency Identification) are currently being implemented to manufacturing systems. One of the important implementations of these technological developments is the increased usage of Automated Guided Vehicles (AGV) in production systems.

AGVs are unmanned and smart devices used for transportation of goods and materials in diverse environments. AGVs were first introduced in 1955 as cited by Müller (1983) (Vis ,2006) and has evolved immensely over time. Furthermore, the area of usage of AGVs expanded to include many environments and operations. AGVs are generally used for transportation of goods and materials related to manufacturing in a production environment, especially in Flexible Manufacturing Systems (FMS). However, they can be used for different purposes in various environments. Such environments may be indoors or outdoors, such as distribution, trans-shipment, and external transportation areas. (Vis ,2006) In a manufacturing environment, AGVs can be used for movement of raw materials, parts, tools, finished goods and work-in-process inventory. (MHI, n.d.) Figure 1. shows an example where an AGV is used for materials handling in a chemical agent destruction plant. Other examples of different application areas can be listed as; material sorting systems, automated warehouse systems, yard operations as a mean of secondary material handling agents (Yang et al., 2018) and service systems such as hospitals. One of the distinctive and intriguing applications of AGVs is an

underground automated transportation system at an airport where the transportation of cargo is provided through AGVs in tubes. (Van der Heijden et al.,2002).



Figure 1. Figure of an AGV

From Pueblo Chemical Agent-Destruction Pilot Plant Automated Guided Vehicles

[photograph] by PEO ACWA, 2013, Flickr

(<https://www.flickr.com/photos/39388148@N04/9366131108>) CC BY 2.0

According to Material Handling Industry Organization (MHI), AGV market is rapidly growing. (MHI) In fact, 2 billion USD worth 2021 market value has increased by 17 million USD in 2022 and is expected to grow up to 4.11 billion USD by 2029. (Fortune Business Insights, 2022). AGVs are easy to modify and can be integrated to work along with other equipment such as handling robots. In addition to this, these smart vehicles can collaborate with humans during operations. They are easy to use, monitor and control. They do not form a permanent obstacle within the production system like conveyor systems because they can operate continuously and move to perform the handling. They can navigate through obstacles changing directions. When a vehicle is down, another vehicle can take over and the production system can still operate during this down time, providing flexibility. Same can be said about the maintenance operations. Thus, AGVs can provide more reliability and durability to the operations in general in addition to flexibility. All of these benefits add value to the operations.

Another benefit of these AGVs is that they can operate without the need for human intervention. Instead of doing the handling, operators/workers can perform other value-added jobs. Hence workforce can be effectively utilized. AGVs can be assigned to work in hazardous and undesirable environments, such as paint shops, cold-storage, and casting facilities. Unlike conventional means of material handling, they are quiet and eco-friendly in terms of how they are powered. They provide safety to the system by reducing noise, vibration, need to access hazardous place, product, and facility damage.

As a result, main benefits can be summarized as; reduced costs, increased productivity, decreased energy consumption and enhanced safety due to less human intervention in the system. (Bechtsis et al., 2017) There are some limitations of AGV systems. While the return could be desirable in long term, these advanced and complex systems may have a high initial cost. Even though they are easy to use and control, resources may need to be allocated on training of the maintenance personnel.

Many types of AGVs are equipped with sensors such as Radio Frequency Identification (RFID) or Laser Imaging Detection and Ranging (LIDAR) to recognize their surroundings and navigate within the physical system they are in. In addition to sensors, they are usually equipped with onboard Central Processing Unit (CPU) to make decisions regarding collision avoidance. (J. Mehmi et al., 2018)

1.1. Structure of AGV Systems

Systems where a fleet of AGVs are operating to perform transporting jobs are referred as Automated Guided Vehicle System (AGV System). (Vis ,2006) These systems generally consist of an AGV fleet and a network, where AGVs can travel along to perform the assigned transportation jobs.

Vehicles are one of the main components of AGV systems. Vehicle fleet can be homogeneous or heterogeneous depending on the usage. It may be more reasonable to consider a heterogenous fleet for a more adaptable and flexible system. Shape, size, and load capacity of vehicles change according to their purpose of use and area of use. Different types of AGVs can be listed as: towing, small or unit loader, masthead such as pallet truck or forklift truck and light load transporter. Figure 2. shows basic design of some types of AGVs; specifically masthead, unit loader and towing from left to right. Masthead vehicles are equipped with a part which allows them to interact with the

environment by lifting, stacking, or racking. Unlike the masted type, unit loaders and towing types require external means to be loaded and unloaded. However, some unit loaders may not need this because they are sometimes equipped with a partial lifting system on them. Like the unit loader, some types can carry only a single load, whereas others can carry multiple loads in the form of pallets. Moreover, according to their purpose of use, design, and configuration of AGVs differ. For example, some AGVs can have specific robotic mechanical parts allowing them to perform intended activities such as lifting or have monitors to allow users to control specific information about its current situation.

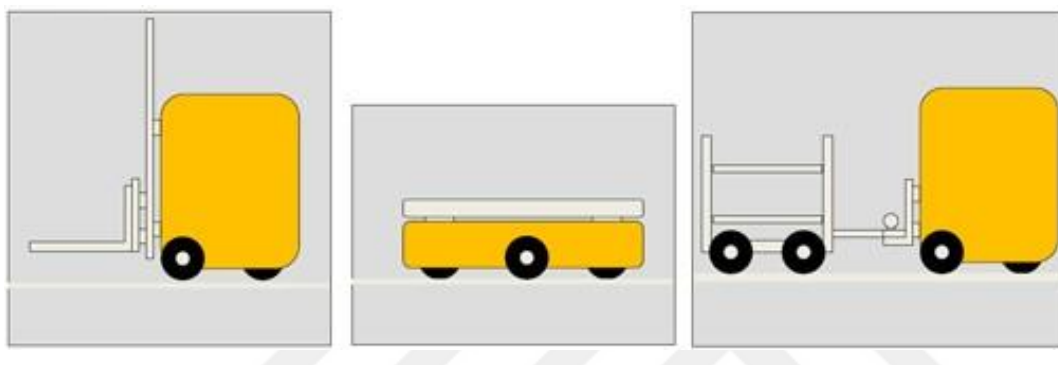


Figure 2. Different types of AGVs

From Applied Handling NW [photograph] by Applied Handling NW, n.d.,
 appliednw (<https://appliednw.com/automated-guided-vehicles/>)

There are various power sources considered for AGVs. Like traditional means, some AGVs can be fuel powered. Mainly, masted types or custom ones that are purposely designed to carry immense loads such as metal sheet coils, work-in-process or finished goods inventory like machines and vehicles are fuel-powered. However, majority of the vehicles are powered by electric batteries. Flood lead acid, NiCad, lithium ion and sealed batteries are some examples of different types of batteries used to power the vehicles. Some batteries are capable of being charged whereas some can only be swapped. For a more green and sustainable system chargeable batteries are a better option.

Vehicles navigate on a physical network within the facility. There are several ways an AGV can do this, either using a fixed route method such as inertial, electromagnetic, optical and tape guidance or a free route method such as laser, ultrasonic and image recognition. (Long and Zhang, 2012) One of the most popular free route methods is laser triangulation. Many types of AGVs are equipped with sensors such as Radio

Frequency Identification (RFID) or Laser Imaging Detection and Ranging (LIDAR) to recognize their surroundings and navigate within the physical system they are in. In addition to sensors, they are usually equipped with onboard Central Processing Unit (CPU) to make decisions regarding obstacle detection and collision avoidance. (J. Mehami et al., 2018) Another popular and simple method is magnetic grid navigation. As the name indicates, several magnet points are located on the floor of the physical system in the form of a grid. Location of these magnets are detected by AGV to help navigate. With the convenient placement of this grid system, it is possible to have vast application options at less cost. (Long and Zhang, 2012)

The AGV system is monitored and controlled through a PC based system controller which communicates with AGVs through a wireless network while they are navigating through the physical system. Through this system, commands can be given to AGVs such as travelling to a location and performing operations. Furthermore, AGVs can provide information on their location, working status, charge level, and maintenance schedule. An operating system ensures the information flow. Additionally, statistics and overview of the system can be accessed through an interface. For the system there are several decisions to be made. Selection of the AGV to perform handling, routing of the fleet, charging or swapping the batteries, and block and collision avoidance are the main decisions. For these decisions many optimization-based solution strategies, algorithms, heuristics, and meta-heuristics are used.

1.2. Uncertainties in AGV Operations

It is unreasonable to assume every input is known exactly in advance. Notwithstanding the general assumptions to make the examined system deterministic, to reflect real life as close as possible, uncertainties should be considered in the system. Even though AGV systems are beneficial in numerous aspects, they introduce additional complexities and uncertainties, to the environment. Within the production facilities uncertain events affecting the AGV operations are generally caused by a form of unavailability. AGVs may not be available due to many reasons. Operating system failure can prevent assigning job orders and learning about the current status and location of the vehicles. Similarly, vehicles may not be available due to a failure or maintenance activities. Components of the AGVs may be faulty and could prevent proper functioning, collaboration, and navigation. In the physical system some roads

could be unreachable due to collision, deadlock, or external blockage. Not only this creates an unavailability issue, but it can also provide uncertainty to the travelling time of the vehicles.

Some uncertainties can affect the system indirectly. Parts may not be available on workstations at the given time. Uncertainties in the processing times of the workstations can create uncertainties in the earliest pick-up time. In cases of job shop production where parts need to be processed following a specific route, delivery to other workstations can be affected. In addition to this, on time delivery of the final product could be greatly affected. Tardiness of orders can cause loss of customer and loss of available production time that could have been spent on production of another order. Therefore, ensuring on time delivery through efficient and effective assignment, scheduling, and routing of the AGV fleet with the consideration of uncertainties regarding the earliest pick-up times and the latest possible deliveries is important.

1.2.1. Robust Optimization Approach

To deal with uncertainty, two main approaches immediately come to mind: stochastic approach where the focus is to minimize an expected value of a performance measure, and robust approach, where constraints are imposed of probabilities. (Vidal et al., 2020) Stochastic approaches assume that the uncertain data is random and can be associated with a probability distribution with the use of historical data. However, in some cases, there may not be enough data to assume a probabilistic distribution. Furthermore, there could be errors during the data collection and measurement process. Therefore, it may not always be suitable. One of the critical issues is that stochastic methods evaluate decisions using only a single scenario, such as expected value as realized scenario, and cannot recognize that each decision has a whole distribution of outcomes based on the realized scenario. (Kouvelis and Yu, 1997). Lastly, finding solutions with stochastic approaches, especially for realistic sized problems, can be challenging or practically impossible. To overcome this and to evaluate all possible scenarios at the same time as if they are realized, robust approach where the aim is to find a solution that is immunized against uncertainty, is considered. Robust Optimization (RO) is a relatively new and a novel method which considers uncertainties. In robust approach, such a solution is found, that no matter which uncertain scenario is realized, the system

performs well. The idea of finding an immunized against uncertainty solution is preferable where the system is complex and on time delivery is important.

Considering the benefits of AGV systems and effects of uncertainties in AGV systems, the scope of this study is defined as Dispatching, Scheduling and Routing Problem of (Load and Battery) Capacitated AGVs with the consideration of an efficient charging plan. Moreover, the focus is on finding a solution approach which will produce robust solutions to the studied problem. In accordance with focus and scope of the problem, an analysis of existing literature consisting of journal articles, conference proceedings, book chapters and case studies is conducted. The results of the analysis are presented in the following chapters.

Within this context, this study is motivated by the remarks made as a result of the literature analysis and aims to answer the following research questions:

- How does the uncertainties in routing problem of chargeable battery powered AGVs used for materials handling in production systems, effect the performance of the production schedule and energy efficiency of AGVs?
- Can a scenario based deterministic method that gives robust solutions be proposed?
- How can we evaluate the robustness with this method?

Consequently, the objective of this study is determined as to find a solution method, which includes uncertainties in the AGV system used in a production environment and gives robust solutions to the problem of scheduling and routing of capacitated AGVs, with the consideration of charging and energy efficiency.

For this purpose, a MILP model with the objective of minimizing costs incurred due to delay of delivery for the jobs and energy utilization by AGV system is proposed. As output, the proposed method yields the assignment of transportation jobs to AGVs, the routes which AGVs use according to this assignment and energy consumption and charging plan of each AGV. To ensure the robustness of the solution, a scenario-based approach for the assumed uncertainties is applied along with the proposed method. Initially several scenarios based on predetermined cases due to occurrence of uncertainties in ready and due time of jobs are determined and solved using the proposed MILP model.

The rest of this thesis is structured as follows. In Chapter 2, the general structure of the studied problem and the problem specific assumptions, are explained. The analysis of the existing literature on Scheduling and Routing Problem of AGVs and method of the analysis are presented in Chapter 3. In order to provide a better understanding for the study, studied AGV problems, solution approaches to AGV problems, RO concepts and related terminology are also elaborated in this chapter. In Chapter 4, the proposed solution method is explained. Following this chapter, the verification of proposed robust mathematical model is presented in Chapter 5. Moreover, computational results of design of experiment and analysis are included in this chapter. Finally, concluding remarks regarding proposed solution approach and results are discussed and potential future research opportunities are addressed in Chapter 6.



CHAPTER 2

PROBLEM DEFINITION

In this chapter, detailed problem definition is given, system structure and assumptions are elaborated. The problem studied in this thesis is routing and scheduling of AGVs used for material handling operations in a job shop production system.

2.1. Scope of the Problem

In a job shop environment, there exist a number of machines (workstations) and unique jobs (orders) that have specific routes to follow through workstations. There could be recirculation, meaning that jobs may visit a specific machine more than once. Such systems allow high customization of products in terms of product variety. They are highly flexible since processes can be added, removed or sequences are changed easily. However, production planning and scheduling gets more complicated. Automating these systems might also be challenging due to lower lot sizes and/or investment costs required. Examples for job shop could be machine production, production of ship parts and repair shops.

Studied system is composed of job orders with predetermined routes and final due dates for these job orders according to a pre-determined delivery plan and schedule, a heterogenous fleet of AGVs to carry out the material handling operations and a magnetic grid network for AGVs' navigation, including locations such as depots, inventories, workstations, charging stations and junction points.

Job orders represent orders that have to be produced. Due to job shop production structure, each job order has a specific final due date and route to follow, determined as a result of production planning and scheduling. However, this production plan and schedule is assumed pre-determined and decisions regarding due dates and routes of job orders are out of scope of this study. It is assumed that each job order visits each workstation on its route at most once, hence no recirculation is allowed

For the AGVs, based on the route of the product, transfer jobs between workstations for all job orders are generated. These transfer jobs may represent movement of parts, work-in-process inventory, tools and equipment. Each of these transfer jobs have a source (origin) and destination. In addition, each transfer job has load amount, service time for loading/unloading, all of which are assumed known. The precedence relationship between transfer jobs of the same job order on a specific route is ensured indirectly through ready times and due times of transfer jobs.

On the workstations, job orders are processed. They will not be ready for handling until the processing is completed. Therefore, there is an associated ready time for each transfer job, representing the earliest time that can be picked-up from origin, according to the production plan and processing times. Moreover, each transfer job has an associated due time in accordance with the production plan. The due time represents the latest time of a transfer job to be picked up so that the final due date of the order can be met. Ready and due times are calculated before solving the AGV scheduling and routing problem, using a logic similar to project management theory of earliest start and latest start time based on the final due date of the job order, traveling time between workstations and processing time on the workstations. Through this logic a lower and upper bound for the pick-up and delivery time are calculated. Figure 3. and Tables 1. and 2. present an atomic example with job orders, routes, and schedules and how they are processed to become transfer jobs for AGVs to perform.

Table 1. Job order routes and final due dates

Job Orders	Route	Final Due Date
1	1-3-4	20
2	2-3-4	25

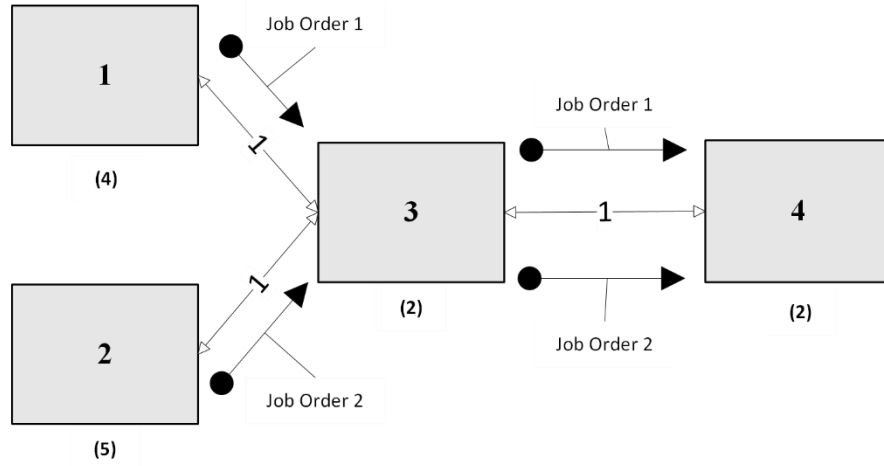


Figure 3. Network presentation of job orders for an atomic example

For this atomic example we assume distance between each workstation is 1 units and servicing time is also assumed 1 unit. As shown in Figure 3., order 1 needs to be finished at due time 20. In order for this to happen it should be delivered to workstation 4 at 18 since it needs to be unloaded and processed for 2 ($1+1$) time units. Hence the due time of transfer job 2 at workstation 4 is 18. For this job to be delivered to workstation 4 latest at 18 it should be picked up from workstation 3 at 15 at latest, considering the service time and travelling time between stations. This logic is used for calculation of due time of the first transfer job as well. To find the ready time of the first transfer job of this order we assume earliest start time logic. Since the order needs to be processed by 4 time units, the ready time of transfer job 1 is 4.

Table 2. Structure of transportation jobs

Transfer Jobs	From	To	Ready time	Due time
1	1	3	4	13
2	3	4	15	18
3	2	3	5	18
4	3	4	20	23

It is assumed that there exist a heterogeneous fleet of AGVs within the facility with different load and battery capacities according to type. To be more specific, for the purposes of this study, this heterogeneous fleet consists of three different types, namely maxi, midi and mini. Maxi AGVs can handle heavier items such as finished goods and sheet metal coils. In addition to this, both due to size and function, this type has large battery capacity to operate. Mini type vehicles can be thought as unit loader AGVs

where small items can be carried. Moreover, since there is no option to load or unload without external means, battery charge capacity is small.

Most AGVs used in the industry are battery powered and effective use of them depends heavily on their schedule as well. In the considered system, AGVs are assumed to be powered by rechargeable batteries. If necessary, AGVs can travel to charging stations to have their batteries recharged during their trip. Furthermore, AGVs consume energy from the battery while traversing and transporting jobs and servicing (loading and unloading operations) with different rates.

In the facility a magnetic grid system on the floor which consist of locations, representing workstations which are pickup and delivery points for jobs and charging stations, and bi-directional lanes, where AGVs can travel along, and transport jobs is assumed. Pick-up and delivery points consist of workstations and storage areas (storage areas for raw materials, work-in-process inventory, finished goods inventory, packing materials and maintenance/repair/operating supplies). It is assumed that buffer space at each workstation is not limited. Initially AGVs are located at AGV depot and are either fully or partially charged. Once a transfer job is assigned and the route is decided, AGVs will navigate using their sensors on this magnetic system. Conflict and congestion avoidance is assumed out of scope of this study since it adds to an already complex problem and is usually managed by modern AGVs using on-board computers. Therefore, it is assumed that lanes are wide enough to accommodate multiple AGVs at the same time and can travel successively without increasing the total travelling time of a single AGV.

Decisions that need to be made are two-fold. By processing job-orders, routings and due dates, transfer jobs are formed. Transfer jobs, representing movements of parts/subassemblies from source to destination must be assigned to individual AGVs. Also, based on this assignment, routing of AGV's need to be made, so that jobs can be completed. A job must be assigned to only a single AGV. If the AGV has enough battery charge and load capacity, more than one job can be assigned to it. Each job must be picked up from its source location by an AGV as soon as it becomes available if assigned to. Jobs can be delivered to destinations after being picked up. Deliveries can occur before or after due time. However, if delivered after, this job is considered tardy and is penalized.

Initially AGVs are located at AGV depot, ready to start their trip with no load and fully charged or partially charged battery (assuming start of workday and/or shift). AGVs must start their trips from the depot and travel to the pickup points, which are source locations of the transfer jobs assigned to them. After picking up the assigned transfer jobs they must travel to delivery points, destination locations of these transfer jobs. Additionally, if there are no transfer jobs left to be picked up, AGV must return to the depot. It is assumed an AGV must end its trip with no load. When arrived at each location, battery charge is controlled to prevent running out of power. Depending on the remaining work and residual battery, AGV must be either sent to recharge and charging amount must be decided or it can continue its trip by travelling to another point in the system. Furthermore, assuming enough remaining load capacity, another job can be picked-up if it is ready and assigned to that AGV. On the other hand, if there is not enough battery charge level and load capacity for the next job, AGV should not pick up other assigned jobs and must travel to a charging station to be charged. At the charging stations, AGVs need to wait as long as the charging time required to be charged as necessary amount given their current level of charge and remaining assigned jobs. At no time, AGVs are allowed to run out of battery. Routing decisions ensure AGVs to have a non-negative charge level. Detailed flow of assumed trip scheme of AGVs can be seen in Figure 4. below. Travelling times, pickup and unload times and charging times of the AGVs are considered for the calculation of completion time of each trip.

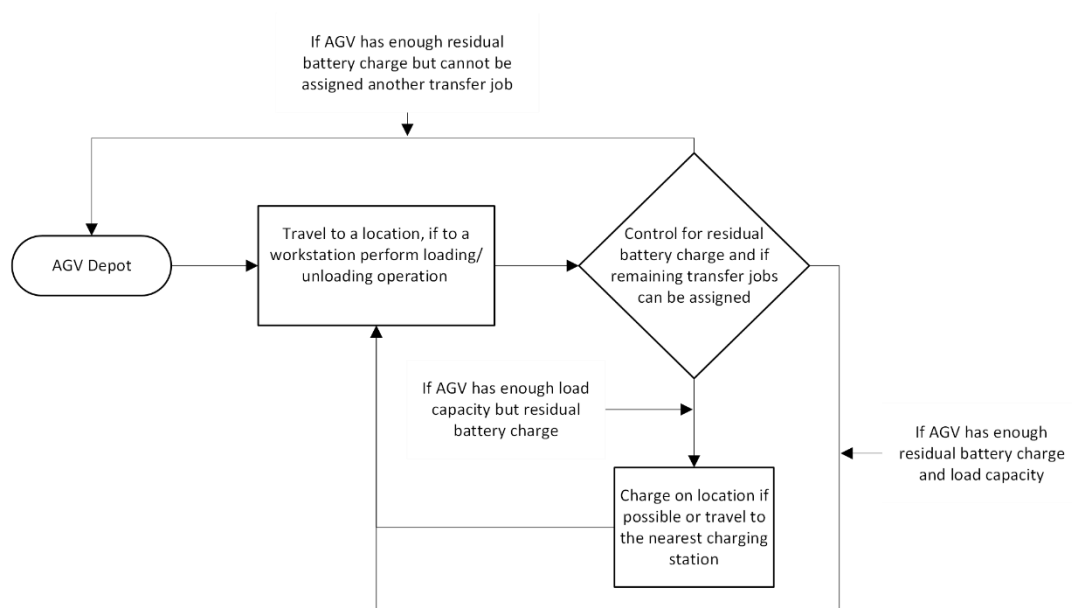


Figure 4. AGV trip scheme flow chart

Uncertainties may exist in processing times of jobs at different workstations. Since production plan and schedule is assumed pre-determined, uncertainties exist indirectly through ready times and due times of each transportation job, with the exception of last job's due time. The last job's due time represents the due date of the order and is pre-determined. No other form of uncertainty is included in this study.

Main objective is to find robust schedules and routes for capacitated (load and battery charge) AGVs used for materials handling operations in a job shop environment with the consideration of energy consumption and charging plan of AGVs to provide energy efficiency.

2.2. Assumptions

The main assumptions made are listed below.

- A graph network consisting of nodes representing locations, service points (pickup and delivery points, which are source and destination locations of transfer jobs) and charging stations and arcs formed in order to represent bi-directional lanes where AGVs can travel along and deliver jobs.
- Service points consist of workstations where orders need to be processed.
- It is assumed that there is unlimited input/output buffer at each service point.
- For the transportation of jobs, a heterogeneous fleet of AGVs consisting of three different types, namely maxi, midi and mini, with different load and battery capacities is considered.
- Initial battery charge of AGVs are assumed known.
- Transfer jobs in the system have a source (origin), destination, load amount, service time, ready time and due time, all of which are assumed known according to the pre-determined production plan and schedule.
- Decisions regarding production plan and schedule are out of scope of this study.
- Jobs may be available on service points initially. Therefore, ready time of such jobs are assumed to be zero.
- All jobs must be assigned and can only be assigned to a single AGV.
- An AGV must start and end its trip at the AGV depot with no load.

- Lanes in the system are assumed to provide conflict and congestion free routing scheme.
- AGV's are assumed to consume energy while traversing and transporting jobs per distance and servicing (loading and unloading operations) at different fixed rates based on type.
- Each AGV can return to the depot at most once during its trip.
- After arriving to a location, an AGV must first deliver assigned jobs that have this point as destination, charge itself if necessary and pick-up assigned jobs that have this point as source.
- An AGV can charge, either full or partially according to the need, on the same charging point at most once during its trip.
- A penalty cost is assigned to late jobs which are delivered after its due time. However, no penalty cost is applied to early jobs.
- AGVs can wait on a location without being penalized.
- Furthermore, there exist additional costs for energy utilization and charging.

CHAPTER 3

LITERATURE REVIEW

The preceding chapters provides insights into the general structure of the studied problem and focused problem environment. The detailed literature review for the AGV scheduling and routing problems and solution approaches are provided in this chapter. Studied problem is related to scheduling, dispatching, routing, charge planning problems of AGVs and RO. Therefore, literature review is divided into two parts as AGV problems in literature and RO. To the best knowledge of the authors', proposed model is the first to make scheduling and routing decisions for a heterogeneous fleet of AGVs, with energy efficiency consideration and partial recharge. The proposed model allows a more flexible, and eco-friendly approach. Furthermore, the model indirectly ensures the given production plan to be followed, by penalizing tardy deliveries. Contribution of this study is the novel model for dispatching & routing of (Load and Battery) Capacitated AGVs with the consideration of energy efficiency and a scenario based robust solution method. The objective of this model is to minimize costs incurred due to delay of delivery for the jobs and energy utilization by AGV system. As output, the model yields the assignment of transportation jobs to AGVs, the routes for AGVs and energy consumption and charging plan of each AGV. Robust model yields the same output under the consideration of uncertain scenarios.

For this review, search engines Google Scholar, Web of Science and ScienceDirect are used to gather publications. Journal articles, book chapters and conference papers between years 1995 and 2022 are analyzed. Following keywords and phrases are used for the research: "*AGV routing*", "*AGV scheduling and routing*", "*robust optimization*", "*scenario based robust approach*" and "*robust AGV scheduling and routing*". Results are summarized in the following parts of this chapter.

3.1. AGV Problems in Literature

Subject of AGVs have been a popular research topic of many researchers over the years. As mentioned in the previous chapter, Vis (2006) states AGVs were introduced

in 1955 as cited by Müller (1983). However, many concepts, definitions and solution approaches have changed since this introduction. Though they were mainly used for material handling operations in manufacturing systems, the usage expanded to many fields such as warehouse systems, container terminals and automated transportation systems. Table 3. summarizes the main studies reviewed and focused for AGV problems.

De Ryck, et al. (2020) categorizes the decisions to be made for AGVs under five fundamental tasks: task allocation, localization, path planning, motion planning, and vehicle management. In task allocation, tasks must be optimally assigned to AGVs. Localization deals with getting real-time imaging of the system such as current location of the AGVs and supports path planning process. Collision and deadlock avoidance is done in motion planning. Vehicle management task monitors the status of the system by checking battery levels and errors. Decisions regarding task allocation and path planning for AGVs are primarily Scheduling and Vehicle Routing Problem (VRP) related.

Scheduling and vehicle routing problems in their more generic form, are proven to be NP hard. Consequently, scheduling of FMS where AGVs are mainly used, which is complex due to additional considerations, also is NP hard. (Fazlollahtabar and Saidi-Mehrabad, 2013) Optimization models may be insufficient for solving large instances. For this reason, many researchers focus on heuristics, which are important for obtaining feasible solutions. Fazlollahtabar and Saidi-Mehrabad (2013) provides a detailed literature review on AGV scheduling and routing at manufacturing, distribution, transshipment, and transportation systems. Different methodologies to optimize AGV systems are reviewed and categorized mainly as exact and heuristic methods, simulation, meta-heuristic techniques, and artificial intelligent based approaches.

Table 3. Summary of selected papers on AGV problems

Paper	Year	Problem	Objectives	Contribution
Bilge & Ulusoy	1995	Simultaneous scheduling& routing of AGVs in FMS	Min. makespan	Non-linear MILP & Heuristic approach
Vis	2006	Design and control issues of AGV systems	Literature review	Literature paper
Nishi et al.	2011	Scheduling & conflict-free routing of AGVs	Min. total tardiness	MILP, Bi-level decomposition,
Singh, Sarngadharan, and Pal	2011	Scheduling of capacitated AGVs	System analysis on AGV utilization & throughput	Dispatching rules, simulation for performance evaluation
Fazlollahtabar and Saidi-Mehrabad	2013	AGV Scheduling & Routing Problems	Literature review	Literature paper
Miyamoto & Inoue	2016	Dispatch & conflict free routing of capacitated AGVs	Min. distance and min. max. delay	IP
Wang et al.	2019	Scheduling of AGVS	Min. energy consumption	MILP, genetic algorithm
De Ryck, et al.	2020	Control algorithms and techniques of AGVs	Literature review	Literature Paper
Singh, Dang, Akcay, et al.	2022	AGV Scheduling with battery constraints	Min. weighted total tardiness cost	MILP, Neighborhood Search

MILP: Mixed Integer Linear Programming, IP: Integer Programming

AGV scheduling and routing decisions can be made either simultaneously or separately. However, generally these problems are studied separately due to the fact that integrated scheduling and routing creates a very challenging NP-hard optimization problem. (Wang et al.,2019) One of the earlier studies by Bilge and Ulusoy (1995), address the problem of simultaneous scheduling of machines and material handling system where a number of identical AGVs are being used in a FMS. Objective is to minimize the makespan. Due to NP hard nature of the studied problem, authors propose an iterative procedure where at each iteration a new schedule is generated through a heuristic and completion times are used as time windows for the problem. Nishi et al. (2011) studies the problem of the simultaneous production scheduling and routing for vehicles with conflict-free path selection. Due to the complex nature of the problem, authors propose a bi-level decomposition algorithm with the overall objective of minimizing the total tardiness of set of tasks as solution methodology. Moreover, they provide a mixed integer linear formulation in decomposed two levels: master problem of task assignment and scheduling; routing subproblem.

Miyamoto and Inoue (2016) studies dispatching and routing problem (DRP) for AGVs with the consideration of deadlock and collision avoidance. Authors state that if the schedule is not properly done, the AGV system is prone to congestion. Therefore, they emphasize the importance of conflict free routing and scheduling and propose a multi objective nonlinear mixed integer programming model for this problem. Considering the additional constraints of conflict free routes, they also propose local and random search methods.

Singh, Sarngadharan, and Pal (2011) studies the scheduling problem of capacitated AGVs used for raw material handling in a machine shop of an automotive production facility. They allow some raw materials picked up from a source to be delivered to several different delivery points in different amounts. They propose a method to determine the number of required AGVs and several dispatching rules for AGVs . They analyze the performance of their methods through discrete time simulation with the consideration of deadlocks when more than one AGV is used. AGVs are identical and there is no mention of a heterogeneous fleet.

Specific types of AGVs use rechargeable batteries as power source. In order to avoid unwanted situations related to AGV availability, battery capacities and power consumptions of AGVs should be considered in the decision-making process. Wang et

al. (2019) emphasize on the lack of consideration of power consumption in task scheduling methods of AGVs. In order to fill this gap, they provide a model with the consideration of battery consumption and an algorithm. Singh, Dang, Akcay et al. (2022) studies the problem of AGV scheduling with the consideration of battery charge. Charging decision is vaguely included and considered as a task which can be assigned to AGVs. It is assumed that AGVs are only used for material handling and only discharge battery for transportation, they do not perform loading/unloading. They also point out that majority of the AGV problems are based on homogenous fleet of AGVs. For this reason, they consider a heterogeneous fleet. In fact, it is one of the first studies that consider heterogeneous fleet of AGVs. Inclusion of a heterogeneous fleet introduces more flexibility to the system.

3.2. Robust Optimization and AGVs

Many of the optimization problems involve a degree of uncertainty in some of the variables. (Garcia and Peña, 2018) Scheduling and routing problems are no exception. In a deterministic approach, uncertainties are not assumed in the modelling. To include uncertainty, stochastic solution methods, simulation models and fuzzy logic approaches are mainly used. In stochastic modelling uncertain data is usually assumed to follow a specific known probabilistic distribution. Yet, it may not always be possible to assume probabilistic distributions. Furthermore, there could be problems, where obtaining past data is simply not possible. There could be errors in the measurement and estimation process. To overcome these issues, a relatively new concept called robustness is being considered. A robust solution is often referred as a solution which is immune against uncertainty, meaning that no matter what disruptive event occurs, solution is not significantly affected. With the promise of immunity against uncertainty, robust solution approach has become a popular research area over the years. Table 4. summarizes the main studies reviewed and focused for robust optimization.

Table 4. Summary of selected papers on robustness and robust AGV problems

Paper	Year	Problem	Objectives	Contribution
Kouvelis & Yu	1997	Discrete RO	Providing comprehensive information about Discrete RO	Reference book for discrete RO concepts
Ben-Tal, El Ghaoui & Nemirovski	2009	RO	Providing comprehensive information about RO	Reference book for RO concepts
Goerigk & Schöbel	2011	RO	Minimize maximum regret	Scenario based robust algorithm
Agra et al.	2013	Robust VRPTW	Minimize cost of operating vehicles	Robust MIP, Branch and cut, column and row generation algorithms
Goerigk & Schöbel	2016	RO, Algorithm design	Providing comprehensive information about RO	Reference paper for RO concepts and algorithm design
Peters, Surana & Bullo	2018	Robust scheduling and routing of aerial vehicles with uncertainty	Minimize mission cost	Robust MILP, framework, simulation model
Wu & Hifi	2020	VRPTW with delay uncertainty	Minimize risk of delay	Scenario based MILP, Variable Neighborhood Search,

RO: Robust Optimization; VRP: Vehicle Routing Problem; VRPTW: Vehicle Routing Problem with Time Windows

Ben-Tal et al. (2009) and Kouvelis and Yu (1997) provide detailed information regarding the concept of robust optimization and its applications. Notwithstanding the popularity, there is no specific exact definition of robustness in the literature, there exists several fundamental definitions. Ben-Tal et al. (2009) explains the classical robust solution as a solution which minimizes the worst performance over all uncertain parameters. In this definition authors elaborate three key assumptions:

- The problem is solved prior to realization of the actual data,
- Responsibility of the outcome when the actual data is realized belongs to the decision maker,
- Violations in the constraints cannot be tolerated when the data is in the uncertainty set.

By using the classical robustness definition, an LP in the generic form: $\min_x \{c^T x : Ax \leq b\}$, takes the form of Robust Counterpart of this LP: $\min_{x,t} \{t : c^T x \leq t, Ax \leq b \forall (c, A, b) \in U\}$, where uncertain parameters are included in the model through uncertainty set U , a set which contains the possible values of uncertain parameters. (Ben-Tal et al., 2019) It should be noted here that uncertainty set U is defined by the decision maker. There are several approaches when it comes to forming uncertainty sets. Some of the approaches could be summarized as finite, interval-based, norm-based and constraint-wise uncertainty sets. (Goerigk and Schöbel, 2016)

Aside from classical robust definition, Goerigk and Schöbel (2016) also provide other robustness definitions with different advantages and disadvantages such as adjustable robustness, light robustness, recovery robustness and regret robustness. Authors explain that recovery robustness is a two-stage approach which allows the use of algorithms for recovery once a disturbance is observed. In this approach recovery cost between feasibility and optimality of a solution is aimed to be minimized. Goerigk and Schöbel (2011) state that the fundamental definition and formulation of classic robustness is too conservative due to the fact that the solution has to be feasible for all scenarios. Therefore, they explain that it is not always possible to find feasible robust solutions to problems using strict robustness. Within this context they propose a scenario-based algorithm, RecOpt, for finding robust solutions based on recovery robustness, which is less conservative.

Pinedo (2016) explains that schedules need to change frequently due to uncertainties such as machine breakdowns or addition of an urgent job that needs to be put into production. Robust scheduling enables a proactive approach and makes the rescheduling easier. Pinedo (2016), explicitly states that is not an easy task to define the robustness of a schedule. A generic robustness metric could be the relative difference between objective function values of after and before disruption. As another method, maximizing total weighted earliness or minimizing total weighted tardiness could provide a more robust schedule. Billaut, Moukrim, and Sanlaville (2010) also explain that it is difficult to define robustness of a schedule. For this reason, they provide few metrics to measure robustness. These metrics could be summarized as solution robustness (comparison of solutions), quality robustness (comparison in terms a performance measure such as objective function value) and deviation from optimality (worst case comparison).

Uncertainties also have great impact on routing problems, especially when uncertainties exist in the travelling times of vehicles and there is a specific time window. Just like scheduling, finding robust solutions can reduce the effects of uncertainty prior to the disruptive event and can make rerouting easier. Agra et al. (2013) is one of the earliest studies considering robust VRP with time windows (VRPTW) under travelling time uncertainty. They propose multiple solution strategies for the robust solution of this problem. They consider worst case and adjustable robustness in their approaches. Wu and Hifi (2020) propose a robust optimization strategy for VRPTW with the uncertain travelling time assumption as well. They define this uncertainty using discrete scenarios. They propose a MILP model with the objective of minimizing the total travelling time of the worst scenario over all defined scenarios. Due to the complexity of the problem, they also propose a Guided Neighborhood Search algorithm to obtain solutions in appropriate computational time. They evaluate the robustness of the solutions using risk of delay where they ratio the total number of violated constraints by total number of customers.

Stochastic and simulation models (Sabuncuoglu and Kizilisik, 2003 and Viharos and Németh, 2018) is commonly studied for AGV scheduling and routing problems with uncertainty as well as fuzzy logic applications (Mousavi, Yap, Musa and Dawal, 2017) Robust solution methods are hardly being used in AGV related studies. To the best of our knowledge, few studies consider robust approach and propose MILP models for

the solution of AGV scheduling and routing consider uncertainties in the system. Peters, Surana and Bullo (2018) is one of the examples where automated vehicles are considered. Authors propose a MILP and a simulation model for the robust scheduling and routing of unmanned, automated aerial vehicles using a scenario-based approach.

Results of the literature review shows there is a gap in the robust routing and scheduling problem of a heterogenous AGV fleet with the consideration of charge plan and energy efficiency. Our study contributes to literature by proposing a novel robust MILP which solves routing and scheduling of AGVs with energy efficiency. To the best of our knowledge this is the first study done within this context



CHAPTER 4

SOLUTION METHODOLOGY

In previous chapter, relevant literature review is presented and scenario based robust optimization approach as a modeling and computational tool is discussed. It is stated that this approach is found to be useful in such cases where it is not possible to use probabilistic distributions for uncertain parameters.

In this chapter, proposed solution method is elaborated. To solve the robust routing and scheduling problem of AGVs using the structure and assumptions provided in Chapter 2, we initially develop a MILP for the deterministic case solution. Based on this proposed MILP, we develop the proposed scenario-based robust model.

4.1. Mathematical Model

The objective of the MILP model is minimizing costs incurred due to tardiness of delivery for the jobs and energy usage by AGV fleet. As output, the proposed method yields the assignment of transportation jobs to AGVs, the individual routes for AGVs and their charging plan. The parameters and decision variables required for the model and the model are given in below.

Sets and Indices:

h, i, j : Location (node) indices

v : Vehicle index

w : Job index

D : Set of depot locations, $D = \{0\}$,

CS : Set of charging station locations,

PP : Set of pick-up points,

DP : Set of delivery points,

SP : Set of service (pickup/delivery) points, $SP = PP \cup DP$

N : Set of locations (nodes), $N = D \cup CS \cup SP$

$N0$: Set of locations (nodes) excluding the depot, $N0 = N/D$

A : Set of lanes (edges), $A = \{(i, j) : i, j \in N \ \& \ i \neq j\}$,

J : Set of jobs,

$Orj[J]$: Set of source locations (nodes (i)) of all jobs (w), $w \in J$, $i \in SP$,

$Des[J]$: Set of destination locations (nodes (i)) of all jobs (w), $w \in J$, $i \in SP$,

V : Set of AGVs, $V = V_{max} \cup V_{mid} \cup V_{min}$

V_{max} : Set of maximum sized AGVs

V_{mid} : Set of mid-sized AGVs

V_{min} : Set of minimum sized AGVs

Parameters:

d_{ij} : Distance between locations (nodes) i and j , $(i,j) \in A$

l_w : Load amount of job w , $w \in J$

r_w : Ready time of job w , $w \in J$

due_w : Due time of job w , $w \in J$

s_w : Service time (loading/unloading) of job w , $w \in J$

C_v : Load capacities of the AGV fleet, $v \in V$

- C_{max} : Load capacity of maximum sized AGVs, $v \in V_{max}$
- C_{mid} : Load capacity mid-sized AGVs, $v \in V_{mid}$
- C_{min} : Load capacity of minimum sized AGVs, $v \in V_{min}$

$InCharge_v$: Initial battery charge of AGV v , $v \in V$

B_v : Battery capacity of the AGV fleet, $v \in V$

- B_{max} : Battery capacity of maximum sized AGVs, $v \in V_{max}$
- B_{mid} : Battery capacity of mid-sized AGVs, $v \in V_{mid}$
- B_{min} : Battery capacity of minimum sized AGVs, $v \in V_{min}$

$speed_v$: Speed (distance per minute) of the AGV fleet, $v \in V$

- $speed_{max}$: Speed of maximum sized AGVs, $v \in V_{max}$
- $speed_{mid}$: Speed of mid-sized AGVs, $v \in V_{mid}$
- $speed_{min}$: Speed of minimum sized AGVs, $v \in V_{min}$

crl_v : Battery charge consumption of AGV v rate while travelling, units per distance, $v \in V$

- crl_{max} : Unit battery charge consumption rate while travelling of maximum sized AGVs, $v \in V_{max}$
- crl_{mid} : Unit battery charge consumption rate while travelling of mid-sized AGVs, $v \in V_{mid}$
- crl_{min} : Unit battery charge consumption rate while travelling of minimum sized AGVs, $v \in V_{min}$

$crov$: Fixed battery charge consumption of AGV v (in units) while loading/unloading operations $v \in V$

- $crov_{max}$: Unit battery charge consumption rate while loading/unloading operations of maximum sized AGVs, $v \in V_{max}$
- $crov_{mid}$: Unit battery charge consumption rate while loading/unloading operations of mid-sized AGVs, $v \in V_{mid}$
- $crov_{min}$: Unit battery charge consumption rate while loading/unloading operations of minimum sized AGVs, $v \in V_{min}$

CR_i : Indicator of charging stations, which is equal to 1 if of node i is a charging station, zero otherwise, $i \in N$

$chargeRate$: Charging rate (time per unit),

c_{late} : Penalty cost of delivering a job late, cost per unit time

c_{charge} : Cost of charging, cost per unit

c_{energy} : Penalty cost of energy utilization, cost per unit

Decision Variables:

x_{ijv} : Binary decision variable indicating if AGV v travels to node j immediately after visiting node i where $v \in V$, $(i,j) \in A$

y_{wv} : Binary decision variable indicating if job w , is assigned to AGV v , $w \in J$, $v \in V$

c_{vi} : units charged on node i , $v \in V$, $i \in N$

a_{vij} : arrival time of AGV v to node j travelling from node i , $v \in V$, $(i,j) \in A$,

pt_w : pick-up time of job w , $w \in J$

dt_w : delivery time of job w , $w \in J$

$tardiness_w$: tardiness duration of job w , $w \in J$

b_{vij} : battery charge of AGV v when it arrives to node j travelling from node i , $v \in V$, $(i,j) \in A$

$load_{vij}$: load amount of AGV v when it arrives to node j travelling from node i , $v \in V$, $(i,j) \in A$

Mathematical Model (MILP):

Minimize

$$\sum_{w \in J} c_{late} tardiness_w + \sum_{v \in V} \sum_{i \in N} c_{charge} c_{vi} + \sum_{v \in V} \sum_{ij \in A} c_{energy} d_{ij} c_{rl_v} x_{ijv} \quad (1)$$

s.t

$$\sum_{(0,j) \in A} x_{0jv} \geq y_{wv} \quad \forall (0,j) \in A, v \in V, w \in J \quad (2)$$

$$\sum_{(h,i) \in A} x_{hiv} = \sum_{(i,j) \in A} x_{ijv} \quad \forall v \in V, i \in N \quad (3)$$

$$\sum_{(i,0) \in A} x_{i0v} \leq 1 \quad \forall (0,j) \in A, v \in V \quad (4)$$

$$\sum_{v \in V} y_{wv} = 1 \quad \forall w \in J \quad (5)$$

$$\sum_{(i,j) \in A: j \neq 0} x_{ijv} \geq y_{wv} \quad \forall v \in V, w \in J, i \in PP \text{ \& } i = Orj[J] \quad (6)$$

$$\sum_{(i,j) \in A: i \neq 0} x_{ijv} \geq y_{wv} \quad \forall v \in V, w \in J, j \in DP \text{ \& } j = Des[J] \quad (7)$$

$$0 + \frac{d_{0j}}{speed_v} \leq a_{v0j} + M(1 - x_{0jv}) \quad \forall v \in V, (0,j) \in A \quad (8)$$

$$a_{vhi} + \sum_{w \in J: Des[J]=i} s_w y_{wv} + CR_i c_{vi} chargeRate + \sum_{w \in J: Orj[J]=i} s_w y_{wv} + \frac{d_{ij}}{speed_v} \leq a_{vij} + M(2 - x_{hiv} + x_{ijv}) \quad \forall v \in V, (h,i), (i,j) \in A, i \in N0 \quad (9)$$

$$pt_w \geq a_{vhi} + \sum_{w \in J: Des[J]=i} s_w y_{wv} + CR_i c_{vi} chargeRate + s_w - M(2 - y_{wv} - x_{ijv}) \quad \forall v \in V, (i,j) \in A, i \in PP, w \in J: Orj[J] = i \quad (10)$$

$$dt_w \geq a_{vhi} + s_w - M(2 - y_{wv} - x_{hiv}) \quad \forall v \in V, (h,i) \in A: i \in DP, w \in J: Des[J] = i \quad (11)$$

$$a_{vij} \geq pt_w + \frac{d_{ij}}{speed_v} - M(2 - y_{wv} - x_{ijv}) \forall v \in V, (i, j) \in A, i \in PP, w \in J :$$

$$Orj[J] = i \quad (12)$$

$$dt_w \leq a_{vhi} + s_w + M(2 - y_{wv} - x_{hiv}) \forall v \in V, (h, i) \in A: i \in DP, w \in J: Des[J] = i \quad (13)$$

$$pt_w \geq r_w + s_w \forall w \in J \quad (14)$$

$$pt_w \leq dt_w \forall w \in J \quad (15)$$

$$tardiness_w \geq dt_w - due_w \forall w \in J \quad (16)$$

$$load_{v0j} = 0 \forall v \in V, (0, j) \in A \quad (17)$$

$$load_{vi0} = 0 \forall v \in V, (i, 0) \in A \quad (18)$$

$$load_{vij} \leq C_v x_{ijv} \forall v \in V, (i, j) \in A \quad (19)$$

$$load_{vhi} - \sum_{w \in J: Des[J]=i} s_w y_{wv} + \sum_{w \in J: Orj[J]=i} s_w y_{wv} \leq load_{vij} + M(2 - x_{hiv} - x_{ijv}) \forall v \in V, (h, i), (i, j) \in A \quad (20)$$

$$b_{v0j} = (InCharge_v - d_{0j} crl_v) x_{0jv} \forall v \in V, (0, j) \in A \quad (21)$$

$$b_{vij} \leq b_{vhi} - \sum_{w \in J: Des[J]=i} y_{wv} cro_v + CR_i c_{vi} - \sum_{w \in J: Orj[J]=i} y_{wv} cro_v - d_{ij} crl_v x_{ijv} + M(2 - x_{hiv} - x_{ijv}) \forall v \in V, (h, i), (i, j) \in A, i \in N0 \quad (22)$$

$$b_{vhi} - \sum_{w \in J: Des[J]=i} y_{wv} cro_v + CR_i c_{vi} \leq B_v + M(1 - x_{ijv}) \quad \forall v \in V, (i, j) \in A, i \in N0 \quad (23)$$

$$b_{vij} \leq B_v x_{ijv} \forall v \in V, (i, j) \in A \quad (24)$$

$$x_{ijv} \in \{0,1\} \quad v \in V, (i, j) \in A \quad (25)$$

$$y_{wv} \in \{0,1\} \quad w \in J, v \in V \quad (25)$$

$$c_{vi} \geq 0 \quad \forall i \in N, v \in V \quad (27)$$

$$a_{vij} \geq 0 \quad \forall v \in V, (i, j) \in A \quad (28)$$

$$dt_w \geq 0 \quad \forall w \in J \quad (29)$$

$$pt_w \geq 0 \quad \forall w \in J \quad (30)$$

$$b_{vij} \geq 0 \quad \forall v \in V, (i, j) \in A \quad (31)$$

$$load_{vij} \geq 0 \quad \forall v \in V, (i, j) \in A \quad (32)$$

$$tardiness_w \geq 0 \quad \forall w \in J \quad (33)$$

The objective function (1) minimizes costs incurred due to tardiness of delivery for the jobs and charging AGVs. (2)-(7) show main routing constraints. Specifically,

constraint (2) ensures AGVs that are assigned jobs start their trip from the AGV depot, (3) provides flow equity, and (4) makes sure that each AGV returns to the depot at most once. Constraints (5)-(13) are concerned with dispatching in addition to routing decisions. (5) explains that a job can be assigned to a single AGV. Constraints (7) and (8) ensure AGVs travel to a job's source and destination nodes during their trip if that job is assigned to them. Time-related constraints are shown in constraints (8)-(16). (8)-(13) show arrival time calculations of AGVs to locations. Pick-up and delivery time calculations of jobs are provided in (10)-(16). Constraints (14) and (15) explain that the pick-up time of a job cannot be lower than its ready time and greater than its delivery time respectively. (16) provides the tardiness calculation of a job. Constraints (17)-(20) represent load constraints. (17) and (18) ensure that the initial and final load of AGVs is zero. Constraint 19 show load capacity constraints of AGVs according to their type. Load calculations during the trip of AGVs are provided through constraint (20). Constraints (21)-(24) are concerned with the energy consumption and battery charge of AGVs. Finally, constraints (25)-(33) explain the domains of decision variables and ensure non-negativity.

4.2. Scenario-based Robust Linear Optimization Approach

We have orders with routes and a specific final due date for each order. Based on the route of the product associated with the order and due time, we create transfer jobs with ready and due times to ensure that due date of the order is met. As it was explained in Chapter 2, we assume these times are calculated based on a logic similar to project management theory of earliest start and latest start times. They form lower and upper bound for the pick-up and delivery time.

In reality, these times may change according to uncertainties in travelling time between locations and uncertain processing times of stations on the production route. Through the uncertainties in processing times, ready and due times of transfer jobs have uncertainties (since they are calculated with a logic similar to project management theory of earliest start and latest start time based on the final due date of the job order, traveling time between workstations and processing time on the workstations as explained in Chapter 2) with the exception of last job's due time. The last transfer job's due time represent the due date associated with the order. To ensure the inclusion of uncertainties within the production plan, we propose scenario-based robust counterpart

of the proposed MILP model. In this scenario-based approach, a finite discrete uncertainty set is defined as scenarios. To decrease computational effort, a scenario selection rule is proposed based on assigned weights using likelihoods of scenarios. Robust counterpart is developed to include scenarios based on the initial MILP model. Instead of a worst case approach used in strict robustness we aim to minimize the total weighted cost over all scenarios, aiming to get an overall robust solution. In the proposed approach, the robust solution will not be infeasible under any of the possible scenarios, but the overall performance may change.

4.2.1. Uncertainty and Scenario Set Creation

To include uncertainties in the ready and due times, we define a finite discrete uncertainty set. From this uncertainty set, we create scenarios. Uncertainties are therefore defined through a set of discrete scenarios. We assume that there is a base case for orders, which is represented through the most likely times without any delay. For the base case of an order, we assume ready and due times for transfer jobs on its production route are most likely to occur. We perturbate this base case by either increasing or decreasing the ready and due time of transfer jobs of the order. Because the due date of orders are dependent on the customer, we do not perturbate the due time of the last transfer job on the route.

For the creation of scenarios, we determine a level of change according to the uncertain cases we would like to consider. The change in the base case of an order can either occur independently from the other job orders or all the orders can change at the same time. To obtain scenarios we list the permutations with repetition using level of change and job orders. ($nPr = n^r$, where $n = (\text{level of change} + 1)$ and $r = \text{number of job orders}$) As a result, a finite discrete uncertain scenario set is created.

To give an example, for 3 job orders and an increase of 20 minutes hence 20 minutes of increase in the ready and due times of transfer jobs of each order, through permutation with repetition, a total of eight scenarios are formed. Table 5. shows the scenario list for this example.

Table 5. Scenario set creation example

Scenarios	Job Orders		
	Job Order 1	Job Order 2	Job Order 3
Scenario 1	B	B	B
Scenario 2	B + 20	B	B
Scenario 3	B	B + 20	B
Scenario 4	B	B	B + 20
Scenario 5	B + 20	B + 20	B
Scenario 6	B + 20	B	B + 20
Scenario 7	B	B + 20	B + 20
Scenario 8	B + 20	B + 20	B + 20
B: base case instances			

4.2.2. Scenario Selection and Weight Assumption

Using all of the defined scenarios for robust optimization is usually not feasible and preferred for large instances (depending on the level of change and the number of job orders.) due to the large number of scenarios and resulting required computational effort. For this reason, a scenario selection rule to represent enough uncertainty for the case on hand might be considered so current computational complexity stays manageable. Random selection of scenarios is one of the common methods. Instead of a completely random selection, we propose a selection rule based on weights calculated by the likelihood of occurrence of each scenario. This selection rule makes sure that each likelihood is represented.

Realistically not all scenarios are equally likely, therefore less emphasis could be put on (highly) unlikely scenarios. Based on past data, likelihood of occurrence of scenarios may be determined. Probabilistic distributions are usually considered for this purpose. However, as it was explained in chapter 3, it may not always be possible to use probability theory. In such cases, robustness and robust optimization definitions are used. In the generic robust approach, all elements in an uncertainty set are assumed to have equal chance of occurrence and worst scenario is optimized as a result. However, worst scenario may not be very likely in reality. Therefore, to consider a more realistic approach and follow an intelligent method for scenario selection, we propose assigning weights to each scenario, representing likelihoods of the change, not probability.

For weight assignment, we propose a method similar to rank order clustering method used in machine part clustering. We first assign three likelihoods like in a triangular distribution to each uncertain case: 1, 2 and 4 indicating least likely, less likely, and most likely categories respectively. We calculate the likelihood of scenarios by multiplying all the individual likelihoods assigned to job orders in this scenario. Then we calculate the frequency of the scenario likelihoods. We divide the likelihoods by the total of the likelihoods to calculate the weights. We eliminate the scenarios with a weight below 1%, since they represent the scenarios which are highly unlikely and the total scenarios selected will represent the majority of all uncertainties. After the elimination, from the remaining scenarios, we make sure to select at least one scenario with each likelihood. For additional selection, we select scenarios proportional to its both likelihood and likelihood frequency. Likelihood calculation, weight assignment and scenario selection methods are shown in Tables 6. and 7. for the previous example.

Table 6. Likelihood Calculation of Scenarios

Scenarios	Job Orders			Scenario
	Job Order 1	Job Order 2	Job Order 3	Likelihoods
Scenario 1	B (4)	B (4)	B (4)	64
Scenario 2	B + 20 (2)	B (4)	B (4)	32
Scenario 3	B (4)	B + 20 (2)	B (4)	32
Scenario 4	B (4)	B (4)	B + 20 (2)	32
Scenario 5	B + 20 (2)	B + 20 (2)	B (4)	16
Scenario 6	B + 20 (2)	B (4)	B + 20 (2)	16
Scenario 7	B (4)	B + 20 (2)	B + 20 (2)	16
Scenario 8	B + 20 (2)	B + 20 (2)	B + 20 (2)	8
B: base case instances, B +20 : Base case increased by 20 units				
(2): less likely; (4): more likely				

Table 7. Scenario Selection and Weight Assignment

Likelihoods	Count	Count Freq.	Weight	# of Selected
8	1	0.125	0.0667	1
16	3	0.375	0.1333	2
32	3	0.375	0.2667	2
64	1	0.125	0.5333	1
120	8	1	1	6

Weights are assigned to scenarios through their likelihoods. For example, in this example (Table 7) we select one scenario with a likelihood of 8 which has an assigned weight of 0.0667.

4.2.3. Scenario-based MILP

We propose a Scenario-Based MILP model, namely SBMILP with the aim of minimizing total weighted cost over scenarios by introducing additional parameters to the initially developed MILP. The SBMILP includes all selected scenarios in to the problem.

In order to include scenarios, the following index and set are added to the proposed MILP model:

sc : Scenario index,

S : Set of discrete scenarios created using uncertain parameters ready and due times.

$weight_{sc}$: Weight of scenario sc , $sc \in S$

Following parameters and decision variables are updated in order to include scenarios.

r_w^{sc} : Ready time of job w under scenario sc , $w \in J$, $sc \in S$

due_w^{sc} : Due time of job w under scenario sc , $w \in J$, $sc \in S$

a_{vij}^{sc} : arrival time of AGV v to node j travelling from node i under scenario sc , $v \in V$, $(i,j) \in A$,

pt_w^{sc} : pick-up time of job w under scenario sc , $w \in J$, $sc \in S$,

dt_w^{sc} : delivery time of job w under scenario sc , $w \in J$, $sc \in S$,

$tardiness_w^{sc}$: tardiness duration of job w under scenario sc , $w \in J$, $sc \in S$,

For the SBMILP, objective function is to minimize the total weighted cost over

scenarios, including penalty for tardiness, charging and energy spent during the routing, and is updated as (34).

Minimize

$$\sum_{sc \in S} (\sum_{w \in J} c_{late} tardiness_w^{sc} + \sum_{v \in V} \sum_{i \in N} c_{charge} c_{vi} + \sum_{v \in V} \sum_{ij \in A} c_{energy} d_{ij} c_{rl_v} x_{ijv}) weight_{sc} \quad (34)$$

Constraints (8)-(16) and (28)-(33) are changed as below. Other constraints in the initial MILP remain the same for this model.

$$0 + \frac{d_{0j}}{speed_v} \leq a_{v0j}^{sc} + M(1 - x_{0jv}) \forall sc \text{ in } S, v \in V, (0, j) \in A \quad (35)$$

$$a_{vhi}^{sc} + \sum_{w \in J: Des[J]=i} s_w y_{wv} + CR_i c_{vi} chargeRate + \sum_{w \in J: Orj[J]=i} s_w y_{wv} + \frac{d_{ij}}{speed_v} \leq a_{vij}^{sc} + M(2 - x_{hiv} + x_{ijv}) \forall sc \text{ in } S, v \in V, (h, i), (i, j) \in A, i \in N0 \quad (36)$$

$$pt_w^{sc} \geq a_{vhi}^{sc} + \sum_{w \in J: Des[J]=i} s_w y_{wv} + CR_i c_{vi} chargeRate + s_w - M(2 - y_{wv} - x_{ijv}) \forall sc \text{ in } S, v \in V, (i, j) \in A, i \in PP, w \in J: Orj[J] = i \quad (37)$$

$$dt_w^{sc} \geq a_{vhi}^{sc} + s_w - M(2 - y_{wv} - x_{hiv}) \forall sc \text{ in } S, v \in V, (h, i) \in A: i \in DP, w \in J: Des[J] = i \quad (38)$$

$$a_{vij}^{sc} \geq pt_w^{sc} + \frac{d_{ij}}{speed_v} - M(2 - y_{wv} - x_{ijv}) \forall sc \text{ in } S, v \in V, (i, j) \in A, i \in PP, w \in J: Orj[J] = i \quad (39)$$

$$dt_w^{sc} \leq a_{vhi}^{sc} + s_w + M(2 - y_{wv} - x_{hiv}) \forall sc \text{ in } S, v \in V, (h, i) \in A: i \in DP, w \in J: Des[J] = i \quad (40)$$

$$pt_w^{sc} \geq r_w^{sc} + s_w \forall sc \text{ in } S, w \in J \quad (41)$$

$$pt_w^{sc} \leq dt_w^{sc} \forall sc \text{ in } S, w \in J \quad (42)$$

$$tardiness_w^{sc} \geq dt_w^{sc} - due_w^{sc} \forall sc \text{ in } S, w \in J \quad (43)$$

$$a_{vij}^{sc} \geq 0 \forall sc \text{ in } S, v \in V, (i, j) \in A \quad (44)$$

$$dt_w^{sc} \geq 0 \forall sc \text{ in } S, w \in J \quad (45)$$

$$pt_w^{sc} \geq 0 \forall sc \text{ in } S, w \in J \quad (46)$$

$$tardiness_w^{sc} \geq 0 \forall sc \text{ in } S, w \in J \quad (47)$$

As output this model gives a robust AGV-job assignment and routes for AGVs. With this plan the model also calculates tardiness, pick-up and ready times for each scenario.

CHAPTER 5

COMPUTATIONAL RESULTS AND ANALYSIS

In this chapter test data and results obtained by solving the test data by using the proposed model are explained. First, initial and robust models were tested for verification and accuracy in 5.1. Then in 5.2, the performance and the limitations of the methods are tested through experimental design.

5.1. Model Verification

The model is verified using a small instance. Problem inputs are prepared as a combination of generated data and Solomon's (1987) data instances. Some of the parameters such as locations, distances and time windows (ready and due times) are based on Solomon (1987), because majority of the papers studying VRPTW test their proposed solution methodologies using these instances.

5.1.1. Verification of Initially Developed MILP with No Uncertainty

A test instance without the inclusion of uncertainty is defined below. For this purpose, we envision a small hypothetical job shop facility. In the system there are 3 job orders with specific routes and final due dates. Table 8. summarizes the production route and schedule of job orders.

Table 8. Production route and schedule of job orders of test instance

#	Route & Processing Times (min)				Due Date (min)
Order 1	5	8	3	1	775
	50	40	50	75	
Order 2	2	9	10		750
	50	50	50		
Order 3	7	6	4		725
	25	50	25		

Orders are assumed to be ready at the first workstation on their routes, therefore their ready times are assumed zero while the transfer job conversion process. Transfer job lists formed from job order due dates, processing times and routings are shown in Table 9. For the calculation of ready and due times we use the approach explained in Chapter 2.

Table 9. Transfer jobs list for the test instance

#	Source	Destination	Load (kg)	Ready Time (min)	Due Time (min)	Service Time (min)
Job 1	5	8	260	0	100	10
Job 2	8	3	90	140	300	10
Job 3	3	1	130	350	700	10
Job 4	2	9	700	0	300	10
Job 5	9	10	160	350	700	10
Job 6	7	6	50	0	500	10
Job 7	6	4	30	550	700	10

The production facility includes 11 locations (nodes), an AGV depot and 10 workstations. All of the workstations include a charging area therefore they are also considered as charging stations. Charge rate (time per unit) on each node and assumed 1 minute per energy unit. All nodes are not directly connected, however, all nodes are reachable from each other. In total there are 22 bi-directional lanes, which corresponds to 44 arcs, where AGVs can navigate. Table 10. and Figure 5. shows the network information in detail. Distances between nodes in meters are shown on Figure 5.

Table 10. Network Information of the test instance

Network Information		
Total number of points (nodes) = 11		
Total number of lanes (arcs) = 44		
Depot	0	
Service Points (Work Stations)	Source	Destination
	2,3,5,6,7,8,9	1,3,4,6,8,9,10
Charging Stations	2,3,4,5,6,7,8,9,10	

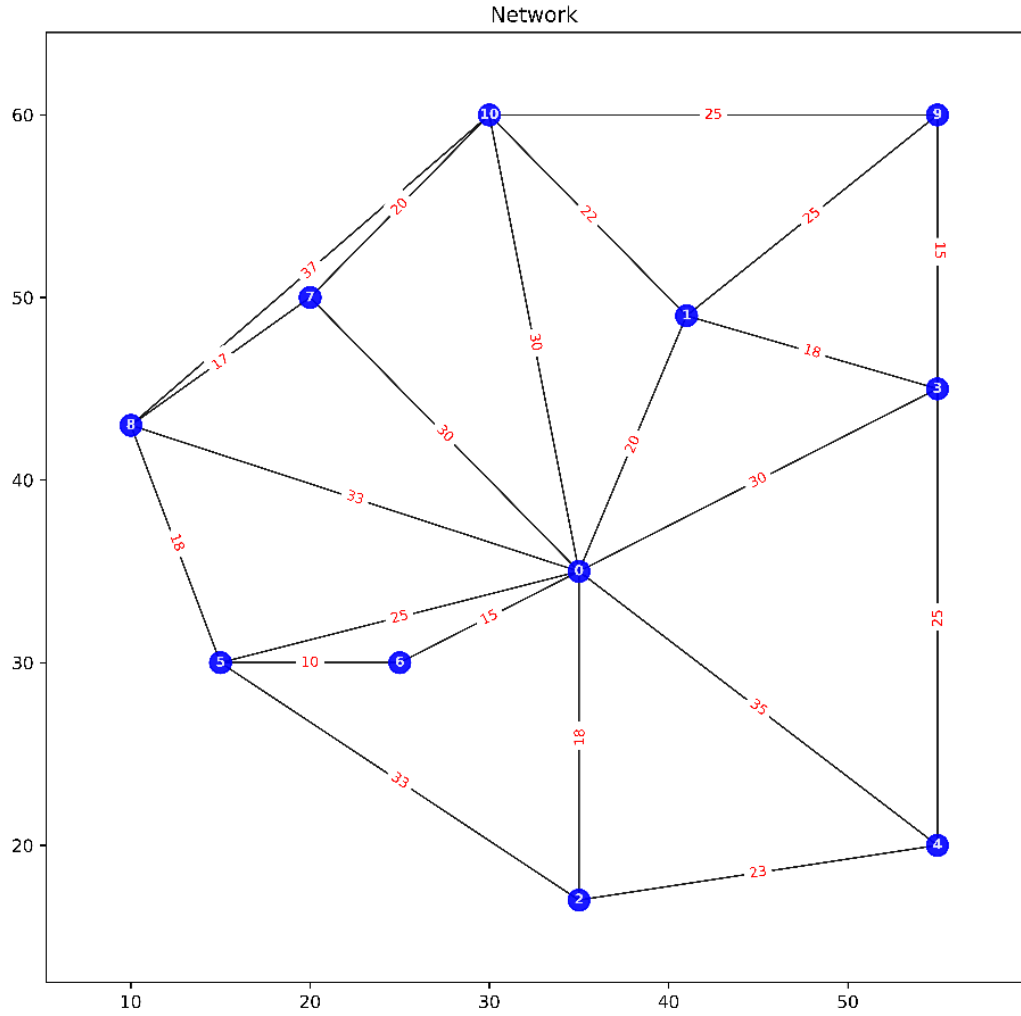


Figure 5. Physical network representation of the test instance

For the transfer jobs, there are 6 AGVs, 3 of each type (mini, midi and maxi). Information regarding the fleet is provided in Table 11. Speed is assumed one meter per minute, fixed for all AGVs. This assumption is made since in an actual facility there usually are specific speed limit for AGVs to ensure occupational safety. Furthermore, unit energy consumption rate while travelling is assumed 10Ah and while loading/unloading operations is assumed 20Ah for all AGVs.

Table 11. AGV fleet in the test instance

AGV Fleet			
AGVs	V_{max}	V_{mid}	V_{min}
	1,2	3,4	5,6
Load Capacity	500kg	200kg	100kg
Battery Capacity	2000Ah	1000Ah	500Ah
Initial Charge	2000Ah	1000Ah	500Ah

Unit penalty cost of delivering a job late is assumed 10\$ per minute and unit cost of charging and unit penalty cost of energy utilization is assumed 10\$ per energy unit.

This test instance is solved using IBM ILOG CPLEX Optimization Studio 12.8 on Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz 2.00 GHz with 16.0 GB RAM and default thread count. Optimal solution with objective function value 4700 was obtained in 27.09 seconds. According to the solution jobs were assigned to three AGVs and no job was delivered tardy. Along its tour medium sized AGV 3 required charging of 200Ah. The solution is shown in Table 12. and Table 13.

Table 12. Job assignment and schedule of the test instance

Job	Source	Destination	Assigned AGV	Pick-up Time (min)	Delivery Time (min)	Tardiness (min)
1	5	8	2	72	100	0
2	8	3	2	150	300	0
3	3	1	2	672	700	0
4	2	9	2	28	275	0
5	9	10	1	360	415	0
6	7	6	1	445	500	0
7	6	4	3	560	636	0

Table 13. Routes of AGVs of the test instance

AGV	From	To	Arrival Time	Arrival Battery Level	Arrival Load
1	0	3	30	1700	0
	3	9	45	1130	0
	9	10	405	860	160
	10	7	435	640	0
	7	8	462	450	50
	8	5	480	270	50
	5	6	490	170	50
	6	0	515	0	0
AGV	From	To	Arrival Time	Arrival Battery Level	Arrival Load
2	0	2	18	1820	0
	2	5	62	1470	70
	5	8	90	1270	330
	8	10	187	860	160
	10	9	265	610	160
	9	3	290	440	90
	3	1	690	220	130
	1	0	720	0	0
AGV	From	To	Arrival Time	Arrival Battery Level	Arrival Load
3	0	6	15	850	0
	6	5	570	730	30
	5	2	603	400	30
	2	4*	626	170	30
	4*	0	691	0	0

* AGV 3 was charged 200 Ah on node 4.

The sample implementation shows a correct and logical solution. Hence the model is verified. Furthermore, the optimal solution is obtained in a reasonable computational time using the proposed method. This is expected due the size of the problem. As the number of locations, lanes, transfer jobs and AGVs increase the computational time will increase due the complexity of the problem.

5.1.2. Scenario Based Model Verification

To verify the proposed scenario based model, we created scenarios on the previous test instance by perturbing the job orders 2 levels. We increased each ready and due time of transfer jobs of the orders by 20 and 40 minutes. A total of 27 scenarios were created using the proposed scenario generation rule explained in Chapter 4. Since this was a very small instance problem all scenarios are selected. Selected scenarios are listed in Table 14. Though the model is verified through an example, it can solve any input set. We hope to validate the model through real life implementation in the future.

This test instance is solved using IBM ILOG CPLEX Optimization Studio 12.8 on Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz 2.00 GHz with 16,0 GB RAM and default thread count.

Table 14. Discrete scenarios generated for the test instance

Scenarios for 3 job orders; 3 levels (base; base+20; base+40)					
(1: least likely; 2: less likely; 4: most likely)					
	Order 1	Order 2	Order 3	Likelihood	Weight
BBB	4	4	4	64	0.503
20BB	2	4	4	32	0.252
B20B	4	2	4	32	0.252
BB20	4	4	2	32	0.252
2020B	2	2	4	16	0.126
20B20	2	4	2	16	0.126
B2020	4	2	2	16	0.126
202020	2	2	2	8	0.063
40BB	1	4	4	16	0.126
B40B	4	1	4	16	0.126
BB40	4	4	1	16	0.126
4040B	1	1	4	4	0.031
40B40	1	4	1	4	0.031
B4040	4	1	1	4	0.031
4020B	1	2	4	8	0.063
40B20	1	4	2	8	0.063
2040B	2	1	4	8	0.063

Table 15. (cont'd) Discrete scenarios generated for the test instance

B4020	4	1	2	8	0.063
20B40	2	4	1	8	0.063
B2040	4	2	1	8	0.063
404020	1	1	2	2	0.015
402040	1	2	1	2	0.015
204040	2	1	1	2	0.015
204020	2	1	2	4	0.031
202040	2	2	1	4	0.031
4040B	1	1	4	4	0.031
404040	1	1	1	1	0.007

Optimal solution for the problem was obtained in 712.14 seconds. According to the solution jobs were assigned to three AGVS. Robust assignments and routes are provided in Tables 15. and 16.

The sample implementation of the robust model shows a correct and logical solution. Hence the model is verified. Furthermore, the optimal solution is obtained in a reasonable computational time using the proposed method. Since the proposed scenario based MILP is based on initially developed MILP model, we again expect the computational time to increase as the size of the problem instances gets larger.

Table 16. Optimal robust assignments using scenario based model

Job No	Source	Destination	Assigned AGV
1	5	8	1
2	8	3	1
3	3	1	1
4	2	9	1
5	9	10	2
6	7	6	2
7	6	4	4

Table 17. Optimal robust routes using scenario based model

AGV1		AGV2		AGV 4	
From	To	From	To	From	To
0	2	0	3	0	6
2	5	3	9	6	5
5	8	9	10	5	2
8	10	10	7	2	4*
10	9	7	8	4*	0
9	3	8	5		
3	1	5	6		
1	0	6	0		

* AGV 4 was charged 200 units on node 4.

The proposed scenario based approach aims to find a robust solution with a weighted total cost over all scenarios, not based on the worst scenario. Therefore, for the solution, arrival, pick-up and delivery times and tardiness values may be different for scenarios. The robust solution may not be optimal for each individual scenario, yet the solution is feasible and recovery to optimality can be done easily. To give context, the following tables (Table 17., 18. and 19.) show the results for a specific scenario of the verified sample data.

Based on the robust assignment and routing, pick-up and delivery times and tardiness of transfer jobs are presented in Table 17. for scenario 14. Objective function value is calculated as 4910 for this scenario since there is a 21 minutes tardiness in the system.

Table 18. Schedule of jobs for scenario 14 using scenario based approach

Scenario 14 (B, B+40, B+40), Robust Objective Value : 4910			
Job No	Pick-up Time	Delivery Time	Tardiness
1	93	121	21
2	150	300	0
3	360	388	0
4	50	222	0
5	400	435	0
6	465	520	0
7	600	676	0

If we solve the model individually for the instances of scenario 14, we obtain the optimal assignment, schedule and routes for AGVs shown in Tables 18. and 19. In this solution the optimal objective function value is 4800 and no transfer job is tardy.

Table 19. Optimal AGV assignment and schedule for jobs of scenario 14 instances

Scenario 14 (B, B+40, B+40), Optimal Objective Value: 4800						
Job	Source	Destination	Assigned AGV	Pick-up Time (min)	Delivery Time (min)	Tardiness (min)
1	5	8	2	35	100	0
2	8	3	2	150	237	0
3	3	1	2	360	700	0
4	2	9	2	50	340	0
5	9	10	1	400	435	0
6	7	6	1	465	520	0
7	6	4	3	600	676	0

Table 20. Optimal routes for AGV s for scenario 14 instances

AGV1		AGV2		AGV 4	
From	To	From	To	From	To
0	5	0	2	0	6
5	8	2	4	6	5
8	10	4	3	5	2
10	9	3	9	2	4*
9	3	9	10	4*	0
3	1	10	7		
1	0	7	8		
		8	5		
		5	6		
		6	0		

* AGV 4 was charged 200 units on node 4.

Results show that optimal solution for scenario 14 and the robust model are slightly different. This difference is expected since the weight we assigned to scenario 0.031, therefore we prioritize other scenarios over this one. As a result, we verify that the

proposed model provides robust solutions which is not significantly affected by uncertainties.

5.2. Experimental Design, Results and Analysis

After the verification of the model, generated data instances are tested to evaluate the performance of the proposed model. Considering the computational complexity of the model, obtaining solution for large instances may not be possible in reasonable computational time. Therefore, solution time and optimality gap (in case an optimal solution cannot be found in given time limit) are recorded at each test.

Instances are generated on three hypothetical job shop environments with different number of workstations. For this design of experiment, the inputs that are given in Table 20. is used. The scenario based mathematical model is solved using IBM ILOGCPLEX 12.8 solver on Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz 2.00 GHz with 16,0 GB RAM and Intel(R) Core (TM) i9-9900U CPU @ 3.10GHz 3.00 GHz with 32,0 GB RAM for different job shop environments.

For the hypothetical job shop environments, we consider 3 cases: 10, 15 and 20 workstation cases. These cases also have AGV depot in the system, therefore they have a total of 11,16 and 21 nodes respectively. To reflect a real-life job shop as much as possible, we assume that stations are reachable through defined lanes. We consider the number of lanes fixed since the facility layout plan is predetermined and may not be easily changed. For 10 workstations we assume 22, for 15 workstations we assume 38 and for 20 workstations we assume 48 lanes where AGVs can navigate at a fixed speed of 1 meter per minute under occupational safety rule. We made sure that number of lanes for each case is proportional to the number of workstations to have homogenous system structure. Since these lanes are bi-directional there are a total of 44, 76 and 96 arcs as input.

Due to low volume structure of job shop production, a total of 6 AGVs, 2 of each type is used. Same instances of AGVs considered in verified test instance in 5.1.1 is used. Cost and energy instances are also the same with that case.

Table 21. Input for the Design of Experiments

Total number of Workstations	Total number of Job Orders	Route Complexity of Orders
10	3	Low complexity
15	5	Medium complexity
20	7	High complexity

An important measure in this structure is total number of transfer jobs in the system. As the number increases, we expect the computational time to increase as well. Number of transfer jobs are correlated with the existing job orders and the complexity of their production route. Job orders have multi-level routes. As the complexity of the route increases, the number of transfer jobs also increases. To include this phenomenon into the model, we use 3 categories of complexity in routes of orders: high, medium, and low. Maximum of 3 levels is allowed in high complexity routes. Meaning that at most 3 transportation jobs will be required for a job order. An example explaining route complexity for a 3 job order system is provided in Table 21.

Table 22. Production Route Complexity

Complexity Level	Route Orders	Production Routes	Tot. Number of Transfer Jobs
Low Complexity	2-2-1	O1: 5-8-3 O2: 2-9-10 O3: 7-6	5
Medium Complexity	3-2-1	O1: 5-8-3-1 O2: 2-9-10 O3: 7-6	6
High Complexity	3-3-2	O1: 5-8-3-1 O2: 2-9-10-1 O3: 7-6-4	8

For orders, we generated scenario sets using the scenario creation rules explained in 4.2. We assumed that each ready and due time of a transfer job belonging to an order,

can increase two levels by 15 minutes and 30 minutes. Using these levels of increase and the scenario selection rule we used 8, 10 and 10 scenarios out of 27, 243 and 2187 for 3, 5 and 7 job orders respectively. Weights are also assigned to these scenarios based on weight determination rule explained in 4.2.2.

For the experiment, the number of job orders, complexity of the routes of these orders and consequently the number of transfer jobs in the system varies in each instance while other parameters are constant. Tables 22., 23. and 24. summarizes the performance of each workstation instances.

Table 23. Results of 10 Workstation Instances

No	Orders	Complexity of Order	Objective Function Value	Dec. Var.	Constr.	CPU (in sec)	Gap (%)
1	3	Low	4124.5	3120	15815	48.80	%0.00
2	3	Medium	4214.5	3150	17400	16.45	%0.00
3	3	High	5205.5	3210	20378	279.11	%0.00
4	5	Low	5161.3	3750	21787	329.44	%0.00
5	5	Medium	5915.1	3786	23570	1418.97	%0.00
6	5	High	7112.	3822	24993	769	%0.00
7	7	Low	4376.7	3822	25233	329.11	%0.00
8	7	Medium	5761.6	3558	26656	1245.83	%0.00
9	7	High	5776.7	3930	30102	5194.36	%0.00

Table 22. shows the results of the first job shop environment instance with 10 workstations and 44 arcs. For all instances, the proposed SBMILP model was solved optimally within reasonable time. As the number of job orders and the complexity increases, though the number of decision variables do not increase considerably,

number of constraints and computational time increases significantly. Especially for cases of 7 orders with medium and high level of route complexity.

Table 24. Results of 15 Workstation Instances

No	Orders	Complexity of Order	Objective Value	Dec. Var.	Constr.	CPU (in sec)	Gap (%)
1	3	Low	3870	5130	30701	184.13	%0.00
2	3	Medium	4102.0	5160	33336	747.20	%0.00
3	3	High	4867.7	5220	38510	6389.49	%0.00
4	5	Low	5036.6	6120	41089	2556.03	%0.00
5	5	Medium	5790.5	6156	43946	7200*	%14.30
6	5	High	6417.7	6192	46683	7200*	%32.07
7	7	Low	4276.3	6192	46803	7200*	%18.74
8	7	Medium	5934.4	6228	49540	7200*	%43.43
9	7	High	7912.5	6300	55374	7200*	%59.40
*Solution time is limited by 2 hours.							

Results of second job shop environment instances with 15 workstations and 76 arcs are shown in Table 23. For the cases with 3 orders the model was solved optimally within 2 hours. For the case with 5 orders and low complexity, the model was also solved optimally. However, for the rest of the instances optimal solutions were not found within the time limit. Results indicate that the optimality gap increases as the number of orders and the complexity increases. On the other hand, within the 2 hour limit the proposed model was able to find many solutions with reasonable gaps. Therefore, the model can be used as an initial step to obtain feasible solutions for an improvement heuristic.

Table 25. Results of 20 Workstation Instances

No	Orders	Complexity of Order	Objective Value	Dec. Var.	Constr.	CPU (in sec)	Gap (%)
1	3	Low	3937.7	6678	41861	2690.38	%0.00
2	3	Medium	4259.6	6708	45348	7200*	%7.70
3	3	High	6868.5	6768	52418	7200*	%43.48
4	5	Low	5316.4	7944	55549	7200*	%19.08
5	5	Medium	6089.7	7980	59282	7200*	%30.78
6	5	High	7704.6	8016	63015	7200*	%49.60
7	7	Low	4295.8	8016	63015	7200*	%29.06
8	7	Medium	9879.8	8052	66748	7200*	%68.90
9	7	High	17372.5	8124	7214	7200*	%82.19
*Solution time is limited by 2 hours.							

Table 24. summarizes the results of the third job shop environment instances with 20 workstations and 96 arcs. Optimal solution was only found for the low route complexity case of 3 job order case. Just like the 15 workstation instances, as the number of orders and complexity increases, computational time and optimality gap also increases. This is expected due to the complexity of the studied problem. Therefore, results of the experiments show that for large instances there is a need for a faster solution method. However, solving the problem optimally may not be possible, instead near optimal solutions can be found.

Results of the computational experiment are visualized in Figures 6. and 7. in a line chart. In Figure 6., CPU time is plotted against different order sizes and route complexities for all workstation cases to evaluate their effect on performance of the model. For 10 workstation case all instances are solved within short time. As the number of workstations, number of orders and route complexity increased the

computational time also increased. For large instances run time was limited by 2 hours. For 20 workstation cases all instances were solved within 2 hours except for 3 job orders with low route complexity.

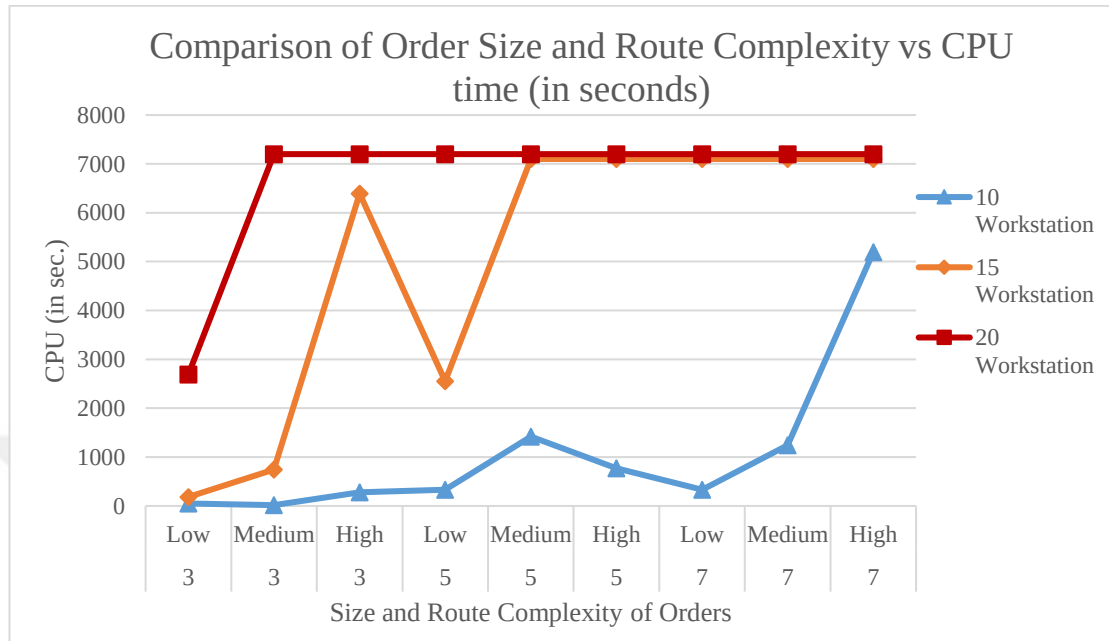


Figure 6. Comparison of Order Size and Route Complexity vs CPU time (in seconds)

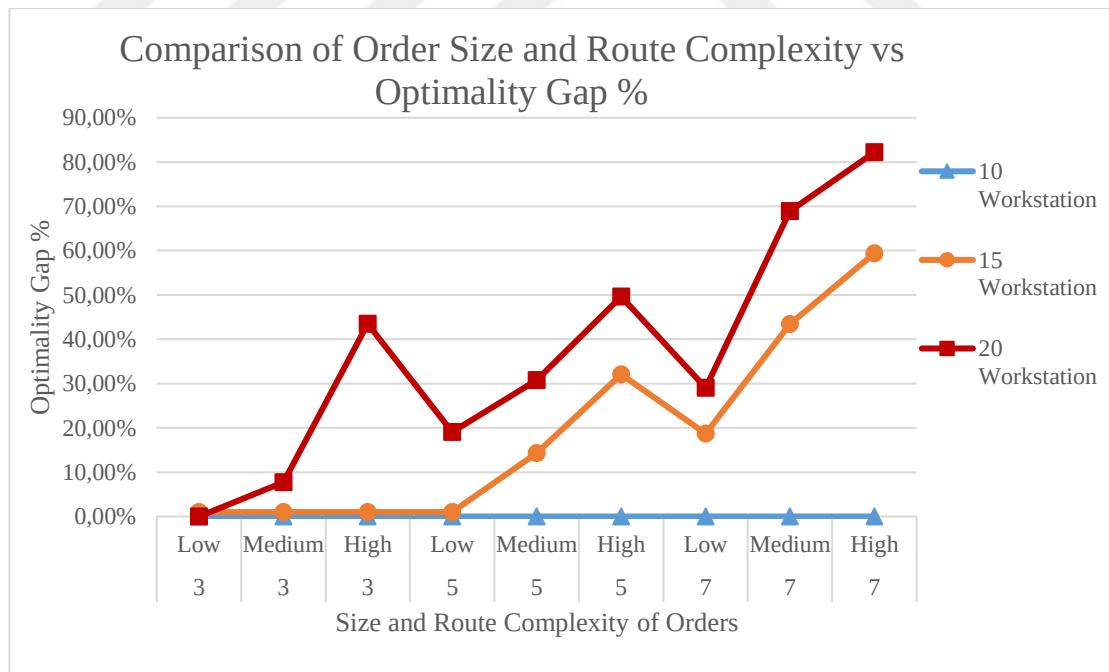


Figure 7. Comparison of Order Size and Route Complexity vs Optimality Gap %

In Figure 7. optimality gap is plotted against different order sizes and route complexities for all workstation cases to evaluate their effect on performance of the model. It can be clearly seen that optimal solutions were obtained for small instances. For the 15 workstation case only the first 4 instances were solved optimally. In case of 20 workstation only 3 job orders with low complexity case is solved optimally. As the size and the complexity increases optimality gap significantly increases. Results showed that larger instance will require more than 2 hours to be solved optimally.

This design of experiment was conducted to evaluate the performance of the proposed method and the effects of the changes in parameters number of workstations, number of job orders and route complexity on robust optimal solutions. Small and large data instances are solved and analyzed in order to assess the complexity of the studied problem. The results of the experiment show that total number of number of workstations, number of job orders and route complexity have significant effect on performance of the model. For small size problems the model was solved quickly. However, when the problem size has increased, the solution time of the model has also increased.

Even though we designed our experiments with the consideration of a real-life job shop environment, real life cases may include a greater number of workstations and complex production routes. As a result, we estimate for larger problem sizes this problem is hard to solve and is in need for a faster solution methodology. For large instances the model can find feasible solutions. These solutions can be used as initial solutions for an improvement heuristic.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

AGVs are increasingly used in more industries in versatile roles. In this thesis we study the problem of Dispatching, Scheduling and Routing Problem of (Load and Battery) Capacitated AGVs with the consideration of an efficient charging plan. Moreover, to include the uncertainties into the problem, we look at a solution approach which gives robust solutions, that is immune to uncertainties. Implementation of the study is given in a job-shop environment, in which routing of parts and a production plan is pre-determined. To the best knowledge of our knowledge, proposed model is the first to make robust scheduling and routing decisions using a scenario based robust approach for a heterogeneous fleet of AGVs, with energy efficiency consideration and partial recharge. The proposed model allows a more flexible, and eco-friendly approach. This data can be processed to come up with a transfer job list for the AGVs with ready and due times. Problem description is elaborated in Chapter 2. According to the literature review presented in Chapter 3, there is no study that looks at robust scheduling and routing decisions for a heterogeneous fleet of AGVs, with energy efficiency consideration and partial recharge. To the best of our knowledge, we present a novel mathematical model for routing and scheduling of capacitated AGVs, with partial recharging and a robust model which ensures that uncertainties in the processing times effecting ready and due times of jobs in a production environment is included. Detailed explanation of the approaches is given in Chapter 4. The proposed models are verified at a test problem. An experimental study evaluating the effects of workstations and orders with different production routing structures is performed. Performance of the proposed methods are also evaluated. The results are shared in Chapter 5. The results indicate for small instances the method yields solutions in reasonably short time. As the size of the instances increase, solution time inevitably increases. This is expected considering the complexity of the problem. Therefore, we hope to develop a heuristic method for solving really large practical instances.

Although the implementation of the model is presented for a job-shop environment, the model can easily be implemented in other usage areas AGVs.

Computational results indicate that for more than 15 transportation job and high complexity orders requires more than 2 hours and more advanced computational power. In the literature it is also stated that the studied problem is considered NP hard. Therefore, there is a need for more efficient solution methodology for the solution of this problem. The results of the literature review shows that variants of Neighborhood Search algorithms such as Guided Neighborhood Search and Adaptive Large Neighborhood Search is being used for this purpose. These methods require an initial solution. Our proposed method yields feasible solutions to the problem in short time; hence it can be used to develop a heuristic method for efficiently solving larger instances.

For the proposed scenario-based model, a solution method, where we can use to minimize the cost of recovery to optimal solution of the specific realized scenario could be considered. This method would provide an efficient robust method which can enable a faster rescheduling and rerouting.

Aforementioned RecOpt algorithm by Goerigk and Schöbel (2011) could be implemented along with the proposed MILP to consider a robustness definition different than classical strict robustness. In the initial step of the algorithm, several scenarios based on predetermined cases due to occurrence of uncertainties in ready time of jobs are determined and solved using the proposed MILP model. Then in the second step, a weighted location problem, where weights are expected probabilities of occurrence of respective cases, could be solved using the solutions obtained from the model with the objective of minimizing regret.

REFERENCES

- Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L. M., Poss, M., & Requejo, C. (2013). The robust vehicle routing problem with time windows. *Computers and Operations Research*, 40(3), 856–866. <https://doi.org/10.1016/j.cor.2012.10.002>
- Applied Handling NW. (n.d.). Applied Handling NW [Photograph]. appliednw. <https://appliednw.com/automated-guided-vehicles/>
- Bechtsis, D., Tsolakis, N., Vlachos, D., & Iakovou, E. (2017). Sustainable supply chain management in the digitalisation era: The impact of Automated Guided Vehicles. *Journal of Cleaner Production*, 142, 3970–3984. <https://doi.org/10.1016/j.jclepro.2016.10.057>
- Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust optimization. Princeton series in applied mathematics. In *Daubechies, E. Weinan, JK Lenstra, and E. Suli, Editors*. Springer New York LLC.
- Bilge, U., & Ulusoy, G. (1995). Time window approach to simultaneous scheduling of machines and material handling system in an FMS. *Operations Research*, 43(6), 1058–1070. <https://doi.org/10.1287/opre.43.6.1058>
- Billaut, J. C., Moukrim, A., & Sanlaville, E. (2010). Introduction to Flexibility and Robustness in Scheduling. *Flexibility and Robustness in Scheduling*, 15–33. <https://doi.org/10.1002/9780470611432.ch1>
- De Ryck, M., Versteijhe, M., & Debruwere, F. (2020). Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, 54(December 2018), 152–173. <https://doi.org/10.1016/j.jmsy.2019.12.002>
- Fazlollahtabar, H., & Saidi-Mehrabad, M. (2013). Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 77(3–4), 525–545. <https://doi.org/10.1007/s10846-013-0003-8>
- Fortune Business Insights (2022). Industrial Equipment / Automated Guided Vehicle (AGV) Market. Retrieved March 7, 2022, from <https://www.fortunebusinessinsights.com/automated-guided-vehicle-agv-market-101966>
- Goerigk, M., & Schöbel, A. (2011). A scenario-based approach for robust linear optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6595 LNCS, 139–150. https://doi.org/10.1007/978-3-642-19754-3_15
- Goerigk, M., & Schöbel, A. (2016). Algorithm engineering in robust optimization. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics), 9220 LNCS(246647), 245–279. https://doi.org/10.1007/978-3-319-49487-6_8

Kouvelis, P., & Yu, G. (1997). Robust Discrete Optimization and Its Applications. In Springer US. <https://doi.org/10.1007/978-1-4757-2620-6>

Long, J., & Zhang, C. L. (2012). The summary of AGV guidance technology. *Advanced Materials Research*, 591–593, 1625–1628. <https://doi.org/10.4028/www.scientific.net/AMR.591-593.1625>

Mehami, J., Nawi, M., Zhong, R. Y., Wernke, R. (2018). Smart automated guided vehicles for manufacturing in the context Smart automated guided vehicles for manufacturing in the context Industry of Industry Costing models for of capacity between used capacity. *Procedia Manufacturing*, 26, 1077–1086. <https://doi.org/10.1016/j.promfg.2018.07.144>

MHI (n.d). AGV101. Retrieved January 9, 2020, from <https://www.mhi.org/mag/Knowledge>

Miyamoto, T., & Inoue, K. (2016). Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems. *Computers and Industrial Engineering*, 91, 1–9. <https://doi.org/10.1016/j.cie.2015.10.017>

Mousavi, M., Yap, H. J., Musa, S. N., & Dawal, S. Z. M. (2017). A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS. *International Journal of Simulation Modelling*, 16(1), 58–71. [https://doi.org/10.2507/IJSIMM16\(1\)5.368](https://doi.org/10.2507/IJSIMM16(1)5.368)

Nishi, T., Hiranaka, Y., & Grossmann, I. E. (2011). A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Computers and Operations Research*, 38(5), 876–888. <https://doi.org/10.1016/j.cor.2010.08.012>

Peters, J. R., Surana, A., & Bullo, F. (2018). Robust scheduling and routing for collaborative human/Unmanned aerial vehicle surveillance missions. *Journal of Aerospace Information Systems*, 15(10), 585–603. <https://doi.org/10.2514/1.I010560>

PEO ACWA. (2013). Pueblo Chemical Agent-Destruction Pilot Plant Automated Guided Vehicles [Photograph]. Flickr. <https://www.flickr.com/photos/39388148@N04/9366131108/>

Pinedo, M. L., (2016). Design and Implementation of Scheduling Systems: More Advanced Concepts. In Springer (Ed.), *Scheduling* (4th ed., pp. 485–508). https://doi.org/10.1007/978-3-319-26580-3_18

Sabuncuoglu, I., & Kizilisik, O. B. (2003). Reactive scheduling in a dynamic and stochastic FMS environment. *International Journal of Production Research*, 41(17), 4211–4231. <https://doi.org/10.1080/0020754031000149202>

- Singh, N., Dang, Q. V., Akcay, A., Adan, I., & Martagan, T. (2022). A matheuristic for AGV scheduling with battery constraints. *European Journal of Operational Research*, 298(3), 855–873. <https://doi.org/10.1016/j.ejor.2021.08.008>
- Singh, N., Sarngadharan, P. V., & Pal, P. K. (2011). AGV scheduling for automated material distribution: A case study. *Journal of Intelligent Manufacturing*, 22(2), 219–228. <https://doi.org/10.1007/s10845-009-0283-9>
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35, 254–265
- Van der Heijden, M. C., Van Harten, A., Ebben, M. J. R., Saanen, Y. A., Valentin, E. C., & Verbraeck, A. (2002). Using simulation to design an automated underground system for transporting freight around Schiphol Airport. *Interfaces*, 32(4), 1–19. <https://doi.org/10.1287/inte.32.4.1.49>
- Viharos, A. B., & Németh, I. (2018). Simulation and scheduling of AGV based robotic assembly systems. *IFAC-PapersOnLine*, 51(11), 1415–1420. <https://doi.org/10.1016/j.ifacol.2018.08.317>
- Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170, 677–709. <https://doi.org/10.1016/j.ejor.2004.09.020>
- Wang, F., Zhang, Y., & Su, Z. (2019). A novel scheduling method for automated guided vehicles in workshop environments. *International Journal of Advanced Robotic Systems*, 16(3), 1–13. <https://doi.org/10.1177/1729881419844152>
- Yang, Y., Zhong, M., Dessouky, Y., & Postolache, O. (2018). An integrated scheduling method for AGV routing in automated container terminals. *Computers and Industrial Engineering*, 126(July), 482–493. <https://doi.org/10.1016/j.cie.2018.10.007>
- Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent Manufacturing in the Context of Industry 4 . 0: A Review. *Engineering*, 3(5), 616–630. <https://doi.org/10.1016/J.ENG.2017.05.015>



APPENDIX 1 – COMPUTATIONAL RESULTS OF THE SCENARIO BASED MODEL VALIDATION

Tables and figure below show (Table A1.1.-A1.2. and Figure A1.1.) the computational results of the proposed scenario based model verification using test instance explained in Chapter 5.

Table A1.1. Robust AGV-Job Assignment of Test Instance

Job Assignment List				
Job No	Source	Destination	AGV	
1	5	8	1	1
2	8	3	1	1
3	3	1	1	1
4	2	9	1	1
5	9	10	2	2
6	7	6	2	2
7	6	4	4	4

Table A1.2. Robust Routes and Charge Plan of AGVs of Test Instance

Routes and Charge Plan Of AGVs

AGV 1		AGV 2		AGV 4	
From	To	From	To	From	To
0	2	0	3	0	6
2	5	3	9	6	5
5	8	9	10	5	2
8	10	10	7	2	4
10	9	7	8	4	0
9	3	8	5		
3	1	5	6		
1	0	6	0		

Charge Plan

AGV	Station	Units
4	4	200

Job Schedule According to Scenario															
Scenario 1	BBB	Objective	4700	Scenario 2	20BB	Objective	4700	Scenario 3	B20B	Objective	4710	Scenario 4	BB20	Objective	4700
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	71	100	0	1	92	120	0	1	73	101	1	1	72	100	0
2	150	247	0	2	170	267	0	2	150	267	0	2	150	247	0
3	672	700	0	3	380	408	0	3	360	388	0	3	360	388	0
4	28	222	0	4	28	242	0	4	30	222	0	4	29	222	0
5	360	395	0	5	360	395	0	5	380	415	0	5	360	395	0
6	425	480	0	6	425	480	0	6	445	500	0	6	425	480	0
7	560	636	0	7	560	636	0	7	560	636	0	7	580	656	0
Scenario 5	2020B	Objective	4700	Scenario 6	20B20	Objective	4700	Scenario 7	B2020	Objective	4710	Scenario 8	202020	Objective	4700
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	92	120	0	1	71	120	0	1	112	140	0	1	93	121	21
2	170	267	0	2	170	267	0	2	190	287	0	2	150	247	0
3	380	408	0	3	380	408	0	3	400	428	0	3	360	388	0
4	30	242	0	4	28	242	0	4	69	262	0	4	50	222	0
5	380	415	0	5	360	395	0	5	360	395	0	5	400	435	0
6	445	500	0	6	425	480	0	6	425	480	0	6	465	520	20
7	560	636	0	7	580	656	0	7	560	636	0	7	560	636	0
Scenario 9	40BB	Objective	4700	Scenario 10	B40B	Objective	4710	Scenario 11	4040B	Objective	4910	Scenario 12	4020B	Objective	4700
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	112	140	0	1	73	101	1	1	112	140	0	1	93	121	21
2	190	287	0	2	150	300	0	2	190	340	0	2	150	247	0
3	400	428	0	3	360	388	0	3	400	428	0	3	360	388	0
4	69	262	0	4	30	275	0	4	50	262	0	4	50	222	0
5	360	435	0	5	380	415	0	5	400	435	0	5	400	435	0
6	425	500	0	6	445	500	0	6	465	520	20	6	465	520	20
7	560	636	0	7	600	676	0	7	560	636	0	7	560	636	0
Scenario 13	40B40	Objective	4700	Scenario 14	B4040	Objective	4910	Scenario 15	4020B	Objective	4700	Scenario 16	40B20	Objective	4700
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	71	140	0	1	93	121	21	1	73	140	0	1	112	140	0
2	190	287	0	2	150	247	0	2	190	287	0	2	190	287	0
3	400	428	0	3	360	388	0	3	400	428	0	3	400	428	0
4	28	262	0	4	50	222	0	4	30	262	0	4	69	262	0
5	360	395	0	5	400	435	0	5	380	415	0	5	360	395	0
6	425	480	0	6	465	520	0	6	445	500	0	6	425	480	0
7	600	676	0	7	600	676	0	7	560	636	0	7	580	656	0
Scenario 17	2040B	Objective	4910	Scenario 18	B4020	Objective	4910	Scenario 19	20B40	Objective	4700	Scenario 20	B2040	Objective	4710
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	93	121	1	1	93	121	1	1	93	121	21	1	92	120	0
2	170	267	0	2	170	267	0	2	150	300	0	2	170	267	0
3	380	408	0	3	380	408	0	3	360	388	0	3	380	408	0
4	50	242	0	4	50	242	0	4	50	222	0	4	49	242	0
5	400	435	0	5	400	435	0	5	400	435	0	5	360	395	0
6	465	520	20	6	465	520	0	6	465	520	0	6	425	480	0
7	560	636	0	7	560	636	0	7	580	656	0	7	600	676	0
Scenario 21	404020	Objective	4700	Scenario 22	402040	Objective	4710	Scenario 23	204040	Objective	4710	Scenario 24	204020	Objective	4710
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness
1	112	140	0	1	93	121	1	1	93	121	1	1	93	121	1
2	190	287	0	2	170	267	0	2	170	267	0	2	170	267	0
3	400	428	0	3	380	408	0	3	380	408	0	3	380	408	0
4	69	262	0	4	50	242	0	4	50	242	0	4	50	242	0
5	380	415	0	5	400	435	0	5	400	435	0	5	400	435	0
6	445	500	0	6	465	520	0	6	465	520	0	6	465	520	0
7	600	676	0	7	600	676	0	7	580	656	0	7	600	676	0
Scenario 25	202040	Objective	4700	Scenario 26	4040B	Objective	4900	Scenario 27	404040	Objective	4700	Scenario 28	204040	Objective	4700
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Ti	Tardiness	Job No	Pick-up Tir	Delivery Ti	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	73	101	0	1	112	140	0	1	112	140	0	1	73	101	0
2	170	267	0	2	190	287	0	2	190	287	0	2	150	247	0
3	380	408	0	3	400	428	0	3	400	428	0	3	360	388	0
4	30	242	0	4	50	262	0	4	50	262	0	4	28	222	0
5	380	415	0	5	400	435	0	5	400	435	0	5	360	395	0
6	445	500	0	6	465	520	20	6	465	520	0	6	425	480	0
7	600	676	0	7	560	636	0	7	600	676	0	7	580	656	0

Figure A1. 1. Time Calculations for All Scenarios for Model Verification Example

APPENDIX 2 – COMPUTATIONAL RESULTS OF THE EXPERIMENTAL DESIGN

Tables and figure below show an example (Table A2.1.-A2.2. and Figure A2.1.) the computational results of the experimental design. Results of the 10 Workstation case with 5 job orders and low complexity case is provided in the following tables. Data instances and solutions of the experimental design are stored in a google drive and can be accessed with the following link:

<https://drive.google.com/drive/folders/1baeI4qgBzit8fbA5E-YZtH1K8ku1nB5G?usp=sharing>.

Table A2. 1. Robust AGV-Job Assignment of 5 orders with low complexity in 10 Workstation Job Shop

Job Assignment List			
Job No	Source	Destination	AGV
1	5	8	2
2	8	3	2
3	2	9	1
4	9	10	1
5	7	6	3
6	3	2	2
7	8	1	2

Table A2. 2. Robust Routes and Charge Plan of AGVs of 5 orders with low complexity in 10 Workstation Job Shop

Routes Of AGVs

AGV 1		AGV 2		AGV 3	
From	To	From	To	From	To
0	2	0	6	0	7
2	4	6	5	7	8
4	3	5	8	8	5
3	9	8	7	5	6
9	10	7	10	6	0
10	0	10	1		
		1	3		
		3	4		
		4	2		
		2	0		

Charge Plan of AGVs

AGV	Station	Units
2	8	20

Job Schedule According to Scenario											
Scenario 1				Scenario 2				Scenario 3			
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	35	63	0	1	35	63	0	1	35	63	0
2	150	247	0	2	150	247	0	2	150	247	0
3	28	300	0	3	28	101	0	3	28	101	0
4	360	700	0	4	360	395	0	4	360	395	0
5	75	130	0	5	90	145	0	5	75	130	0
6	257	315	0	6	257	315	0	6	257	315	0
7	150	219	0	7	150	219	0	7	75	219	0
Scenario 4				Scenario 5				Scenario 6			
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	35	63	0	1	35	63	0	1	35	100	0
2	150	247	0	2	165	262	0	2	150	247	0
3	28	101	0	3	28	101	0	3	28	101	0
4	375	410	0	4	360	395	0	4	375	410	0
5	90	145	0	5	75	130	0	5	75	130	0
6	257	315	0	6	272	330	0	6	257	315	0
7	75	219	0	7	75	234	0	7	112	219	0
Scenario 7				Scenario 8				Scenario 9			
Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness	Job No	Pick-up Time	Delivery Time	Tardiness
1	35	63	0	1	35	63	0	1	35	63	0
2	165	262	0	2	165	262	0	2	165	262	0
3	28	101	0	3	40	113	0	3	40	113	0
4	360	395	0	4	390	425	0	4	390	425	0
5	90	145	0	5	90	145	0	5	105	160	0
6	272	330	0	6	272	330	0	6	272	330	0
7	165	234	0	7	75	234	0	7	165	234	0
Scenario 10											
Job No	Pick-up Time	Delivery Time	Tardiness								
1	35	100	0								
2	150	247	0								
3	40	113	0								
4	390	425	0								
5	105	160	0								
6	257	315	0								
7	150	219	0								

Figure A2. 1. Time Calculations for All Scenarios for 5 orders with low complexity in 10 Workstation Job Shop

APPENDIX 3 – Solution Method Implementation in CPLEX

Proposed models are coded in IBM ILOG CPLEX OPL optimization software and verified using generated sample data. The codes are provided below.

MILP Model .mod file:

```

/*****
 * OPL 12.8.0.0 Model
 * Author: Selen Burçak AKKAYA
 *
 *****/

//Time Limit
    execute timeTermination {
        cplex.tilim = 2*60*60;
    }

//SETS & PARAMETERS

    int M=5000;

    //Node & Arc Information
    //Nodes
    {int} D={0};
    {int} CS=...;
    {int} PP=...;
    {int} DP=...;
    {int} SP=PP union DP;
    {int} N=D union CS union SP;
    {int} N0=N diff D;
    //Node Information
    float CR[N]=...;

    //Arcs
    tuple arc{
        int FromNode;
        int ToNode;
    }
    {arc} A =...;
    //Arc Information
    float d[A]=...;

    //AGV Information
    //AGVs
    {int} Vmax =...;
    {int} Vmid=...;
    {int} Vmin=...;
    {int} V = Vmax union Vmid union Vmin;
    //Load Capacities
    int C[V]=...;
    //Battery Charge
    float InCharge[V]=...;
    float B[V]=...;
    //Speed Values
```

```

float Speed[V]=...;

//Job Information
tuple jobs{
    int jobNo;
    int OrjJ;
    int DesJ;
}
{jobs} J=...;
int s[J]=...;
int l[J]=...;
int r[J]=...;
int due[J]=...;

//Energy Information
float crl[V]=...;
float cro[V]=...;
float chargeRate=...;

//Cost Information
int clate=...;
int ccharge=...;
int cenergy=...;

// DECISION VARIABLES

//Routing and Assignment Decision Variables
dvar boolean x[A][V];
dvar boolean y[J][V];
//Time Decision Variables
dvar int+ a[V][A];
dvar int+ dt[J];
dvar int+ tardiness[J];
//Energy & Battery Charge Decision Variables
dvar int+ c[V][N];
dvar int+ b[V][A];
//Load Decision Variable
dvar int+ load[V][A];

//OBJECTIVE FUNCTION

minimize (sum(w in J)clate*tardiness[w])+sum(v in V,i in
N)(c[v][i]*ccharge)+sum(v in V,ij in A)cenergy*d[ij]*crl[v]*x[ij][v];

//CONSTRAINTS
subject to{

    //Routing constraints
    forall(v in V,w in J)
    c2:
    sum(ij in A:(ij.FromNode==0))x[ij][v]>=y[w][v];

    forall(v in V,i in N)
    c3:
    sum(hi in A:(hi.ToNode==i))x[hi][v]==sum(ij in
A:(ij.FromNode==i))x[ij][v];

    forall (v in V)
    c4:

```

```

sum(ij in A:(ij.ToNode==0))x[ij][v]<=1;

forall(w in J)
c5:
sum(v in V)y[w][v]==1;

forall(i in PP,w in J:w.OrjJ==i,v in V)
c6:
sum(ij in A:(ij.FromNode==i)&&(ij.ToNode!=0))x[ij][v]>=y[w][v];

forall(j in DP,w in J:w.DesJ==j,v in V)
c7:
sum(ij in A:(ij.FromNode!=0)&&(ij.ToNode==j))x[ij][v]>=y[w][v];

//Time constraints
forall(v in V,ij in A:(ij.FromNode==0))
c8:
0+d[ij]/Speed[v]<=a[v][ij]+M*((1-x[ij][v]));

forall(i in N0,hi in A:(hi.ToNode==i),ij in A:(ij.FromNode==i),v in
V)
c9:
a[v][hi]+sum(w in
J:(w.OrjJ==i)|| (w.DesJ==i))s[w]*y[w][v]+CR[i]*c[v][i]*chargeRate+d[ij]/Spe
ed[v]<=a[v][ij]+M*((2-x[hi][v]-x[ij][v]));

forall(i in PP,hi in A:(hi.ToNode==i),v in V,w in J:w.OrjJ==i)
c10:
pt[w]>=a[v][hi]+sum(w in
J:(w.DesJ==i))s[w]*y[w][v]+CR[i]*c[v][i]*chargeRate+s[w]-M*(2-y[w][v]-
x[hi][v]);

forall(i in DP,hi in A:(hi.ToNode==i),v in V,w in J:w.DesJ==i)
c11:
dt[w]>=a[v][hi]+s[w]-M*(2-y[w][v]-x[hi][v]);

forall(i in PP,ij in A:(ij.FromNode==i),v in V,w in J:w.OrjJ==i)
c12:
a[v][ij]>=pt[w]+d[ij]/Speed[v]-M*(2-y[w][v]-x[ij][v]);

forall(i in DP,hi in A:(hi.ToNode==i),v in V,w in J:w.DesJ==i)
c13:
dt[w]<=a[v][hi]+s[w]+M*(2-y[w][v]-x[hi][v]);

forall(w in J)
c14:

pt[w]>=r[w]+s[w];

forall(w in J)
c15:
pt[w]<=dt[w];

forall(w in J)
c16:
tardiness[w]>=dt[w]-due[w];

//Load Constraints
forall(v in V, ij in A:(ij.FromNode==0))

```

```

c17:
load[v][ij]==0;

forall(v in V, ij in A:(ij.ToNode==0))
c18:
load[v][ij]==0;
forall(v in V, ij in A)
c19:
load[v][ij]<=C[v]*x[ij][v];

forall(v in V,w in J,i in N0,hi in A:(hi.ToNode==i),ij in
A:(ij.FromNode==i))
c20:
load[v][hi]+sum(w in J:w.OrjJ==i)l[w]*y[w][v]-sum(w in
J:w.DesJ==i)l[w]*y[w][v]<=load[v][ij]+M*(2-x[hi][v]-x[ij][v]);

//Energy Constraints

forall(v in V,ij in A:(ij.FromNode==0))
c21:
b[v][ij]==(InCharge[v]-d[ij]*crl[v])*x[ij][v];

forall(v in V,i in N0,hi in A:(hi.ToNode==i), ij in
A:(ij.FromNode==i))
c22:
b[v][ij]<=b[v][hi]-sum(w in
J:(w.DesJ==i))y[w][v]*cro[v]+c[v][i]*CR[i]-sum(w in
J:(w.OrjJ==i))y[w][v]*cro[v]-d[ij]*crl[v]+M*(2-x[hi][v]-x[ij][v]);

forall(v in V,i in N0,hi in A:(hi.ToNode==i))
c23:
b[v][hi]-sum(w in
J:(w.DesJ==i))y[w][v]*cro[v]+c[v][i]*CR[i]<=B[v]+M*(1-x[hi][v]);

forall(ij in A, v in V)
c24:
b[v][ij]<=B[v]*x[ij][v];
}

//Listing AGV Routes and Job-AGV Assignments
tuple SOLRoute
{
arc ij;
int v;
int value;
int a;
float b;
int load;
}

tuple SOLAssignment
{
jobs w;
int v;
int value;
int pt;
int dt;
int lateness;
}

```

```

tuple SOLCharge
{
int v;
int i;
int value;
}
{SOLRoute} RoutesOfAGVs = {<ij,v,x
[ij][v],a[v][ij],b[v][ij],load[v][ij]> | ij in A, v in V :x [ij][v]==1};
{SOLAssignment} JobAGVAssignments = {<w,v,y
[w][v],pt[w],dt[w],lateness[w]> | w in J, v in V :y [w][v]==1};
{SOLCharge} ChargePlanOfAGVs = {<v,i,c[v][i]>| i in N, v in
V :c[v][i]>=1};

//Solution Representation
execute{
writeln(JobAGVAssignments)
writeln(RoutesOfAGVs)
writeln(ChargePlanOfAGVs)
}

string outputRange1;
string outputRange2;
string outputRange3;
int numRows1;
int numRows2;
int numRows3;

execute {
numRows1 = JobAGVAssignments.size;
outputRange1 = "'JobAssignments'!A2:" + "H" + (numRows1+1);
writeln(outputRange1);

numRows2 = RoutesOfAGVs.size;
outputRange2 = "'RoutesOfAGVs'!A2:" + "G" + (numRows2+1);
writeln(outputRange2);

numRows3 = ChargePlanOfAGVs.size;
outputRange3 = "'ChargePlan'!A2:" + "C" + (numRows3+1);
writeln(outputRange3);
}

```

MILP Model .dat file:

```

/*****
* OPL 12.8.0.0 Model
* Author: Selen Burçak AKKAYA
*
*****/

//Sheet Connection For Reading the Input Data File
SheetConnection InputData ("Input Data-Example.xlsx");
//Reading Inputs
//Node & Edge Information
//Nodes
CS from SheetRead (InputData, "'LocationInfo'!CS");
PP from SheetRead (InputData, "'LocationInfo'!PP");
DP from SheetRead (InputData, "'LocationInfo'!DP");
//Node Information

```

```

CR from SheetRead (InputData, "'LocationInfo'!CR");
//Arcs
A from SheetRead (InputData, "'LocationInfo'!A");
//Arc Information
d from SheetRead (InputData, "'LocationInfo'!d");

//AGV Information
//AGV Fleet
Vmax from SheetRead (InputData, "'AGVInfo'!Vmax");
Vmid from SheetRead (InputData, "'AGVInfo'!Vmid");
Vmin from SheetRead (InputData, "'AGVInfo'!Vmin");
//Fleet Load Capacities
C from SheetRead (InputData, "'AGVInfo'!LoadCap");
//Fleet Battery Charge Capacities and Initial Charge Information
InCharge from SheetRead (InputData, "'AGVInfo'!Incharge");
B from SheetRead (InputData, "'AGVInfo'!BatteryCap");
//Fleet Speed Information
Speed from SheetRead (InputData, "'AGVInfo'!Speed");

//Job Information
//Job Set
J from SheetRead (InputData, "'TransferJobs'!J");
//Job Information
l from SheetRead (InputData, "'TransferJobs'!load");
r from SheetRead (InputData, "'TransferJobs'!readyBase");
due from SheetRead (InputData, "'TransferJobs'!dueBase");
s from SheetRead (InputData, "'TransferJobs'!service");
//Energy Information
crl from SheetRead (InputData, "'AGVInfo'!CRL");
cro from SheetRead (InputData, "'AGVInfo'!CRO");
//Cost Information
clate=1;
cenergy=1;
ccharge=1;

//Sheet Connection For Exporting the Solution
SheetConnection OutputData ("Solution-Example.xlsx");

//Exporting/writing the solution in the form of tuples
JobAGVAssignments to SheetWrite(OutputData,outputRange1);
RoutesOfAGVs to SheetWrite(OutputData,outputRange2);
ChargePlanOfAGVs to SheetWrite(OutputData,outputRange3);

```

SBMILP Model .mod file:

```

/*****
 * OPL 12.8.0.0 Model
 * Author: Selen Burçak AKKAYA
 *
 *****/
//Time Limit
execute timeTermination {
    cplex.tilim = 45*60; /
}

//SETS & PARAMETERS

int M=5000;

//Scenario Information
{int} S=...;
float w[S]=...;

//Node & Arc Information
//Nodes
{int} D={0};
{int} CS=...;
{int} PP=...;
{int} DP=...;
{int} SP=PP union DP;
{int} N=D union CS union SP;
{int} N0=N diff D;
//Node Information
float CR[N]=...;

//Arcs
tuple arc{
    int FromNode;
    int ToNode;
}
{arc} A =...;
//Arc Information
float d[A]=...;

//AGV Information
//AGVs
{int} Vmax =...;
{int} Vmid=...;
{int} Vmin=...;
{int} V = Vmax union Vmid union Vmin;
//Load Capacities
int C[V]=...;
//Battery Charge
float InCharge[V]=...;
float B[V]=...;
//Speed Values
float Speed[V]=...;

//Job Information
tuple jobs{
    int jobNo;
    int OrjJ;
}
```

```

        int DesJ;
    }
    {jobs} J=...;
    int s[J]=...;
    int l[J]=...;
    int r[J][S]=...;
    int due[J][S]=...;

    //Energy Information
    float crl[V]=...;
    float cro[V]=...;
    float chargeRate=...;

    //Cost Information
    int clate=...;
    int ccharge=...;
    int cenergy=...;

// DECISION VARIABLES
    //Routing and Assignment Decision Variables
    dvar boolean x[A][V];
    dvar boolean y[J][V];
    //Time Decision Variables
    dvar int+ a[V][A][S];
    dvar int+ pt[J][S];
    dvar int+ dt[J][S];
    dvar int+ lateness[J][S];
    //Energy & Battery Charge Decision Variables
    dvar int+ c[V][N];
    dvar int+ b[V][A];
    //Load Decision Variable
    dvar int+ load[V][A];

//OBJECTIVE FUNCTION

    minimize sum(sc in S)(sum(w in J)clate*lateness[w][sc]+sum(v in V,i
in N)(c[v][i]*ccharge)+sum(v in V,ij in
A)cenergy*d[ij]*crl[v]*x[ij][v])*w[sc];
//CONSTRAINTS
    subject to{
        //Routing constraints

        forall(v in V,w in J)
        c2:
        sum(ij in A:(ij.FromNode==0))x[ij][v]>=y[w][v];

        forall(v in V,j in N)
        c3:
        sum(arcs in A:(arcs.ToNode==j))x[arcs][v]==sum(arcs in
A:(arcs.FromNode==j))x[arcs][v];

        forall (v in V)
        c4:
        sum(ij in A:(ij.ToNode==0))x[ij][v]<=1;

        forall(w in J)
        c5:
        sum(v in V)y[w][v]==1;

```

```

forall(i in PP,w in J:w.OrjJ==i,v in V)
c6:
sum(ij in A:(ij.FromNode==i)&&(ij.ToNode!=0))x[ij][v]>=y[w][v];

forall(j in DP,w in J:w.DesJ==j,v in V)
c7:
sum(ij in A:(ij.FromNode!=0)&&(ij.ToNode==j))x[ij][v]>=y[w][v];

//Time constraints

forall(v in V,ij in A:(ij.FromNode==0),sc in S)
c8:
0+d[ij]/Speed[v]<=a[v][ij][sc]+M*((1-x[ij][v]));

forall(i in N0,hi in A:(hi.ToNode==i),ij in A:(ij.FromNode==i),v in
V,sc in S)
c9:
a[v][hi][sc]+sum(w in
J:(w.OrjJ==i)|| (w.DesJ==i))s[w]*y[w][v]+CR[i]*c[v][i]*chargeRate+d[ij]/Spe
ed[v]<=a[v][ij][sc]+M*((2-x[hi][v]-x[ij][v]));

forall(i in PP,hi in A:(hi.ToNode==i),v in V,w in J:w.OrjJ==i,sc in
S)
c10:
pt[w][sc]>=a[v][hi][sc]+sum(w in
J:(w.DesJ==i))s[w]*y[w][v]+CR[i]*c[v][i]*chargeRate+s[w]-M*(2-y[w][v]-
x[hi][v]);

forall(i in DP,hi in A:(hi.ToNode==i),v in V,w in J:w.DesJ==i,sc in
S)
c11:
dt[w][sc]>=a[v][hi][sc]+s[w]-M*(2-y[w][v]-x[hi][v]);

forall(i in PP,ij in A:(ij.FromNode==i),v in V,w in J:w.OrjJ==i,sc
in S)
c12:
a[v][ij][sc]>=pt[w][sc]+d[ij]/Speed[v]-M*(2-y[w][v]-x[ij][v]);

forall(i in DP,hi in A:(hi.ToNode==i),v in V,w in J:w.DesJ==i,sc in
S)
c13:
dt[w][sc]<=a[v][hi][sc]+s[w]+M*(2-y[w][v]-x[hi][v]);

forall(w in J, sc in S)
c14:
pt[w][sc]>=r[w][sc]+s[w];

forall(w in J, sc in S)
c15:
pt[w][sc]<=dt[w][sc];

forall(w in J,sc in S)
c16:
lateness[w][sc]>=dt[w][sc]-due[w][sc];

//Load Constraints

```

```

forall(v in V, ij in A:(ij.FromNode==0))
c17:
load[v][ij]==0;

forall(v in V, ij in A:(ij.ToNode==0))
c18:
load[v][ij]==0;

forall(v in V, ij in A)
c19:
load[v][ij]<=C[v]*x[ij][v];

forall(v in V,w in J,i in N0,hi in A:(hi.ToNode==i),ij in
A:(ij.FromNode==i))
c20:
load[v][hi]+sum(w in J:w.OrjJ==i)l[w]*y[w][v]-sum(w in
J:w.DesJ==i)l[w]*y[w][v]<=load[v][ij]+M*(2-x[hi][v]-x[ij][v]);

//Energy Constraints
forall(v in V,ij in A:(ij.FromNode==0))
c21:
b[v][ij]==(InCharge[v]-d[ij]*crl[v])*x[ij][v];

forall(v in V,i in N0,hi in A:(hi.ToNode==i), ij in
A:(ij.FromNode==i))
c22:
b[v][ij]<=b[v][hi]-sum(w in
J:(w.DesJ==i))y[w][v]*cro[v]+c[v][i]*CR[i]-sum(w in
J:(w.OrjJ==i))y[w][v]*cro[v]-d[ij]*crl[v]+M*(2-x[hi][v]-x[ij][v]);

forall(v in V,i in N0,hi in A:(hi.ToNode==i))
c23:
b[v][hi]-sum(w in
J:(w.DesJ==i))y[w][v]*cro[v]+c[v][i]*CR[i]<=B[v]+M*(1-x[hi][v]);

forall(ij in A, v in V)
c24:
b[v][ij]<=B[v]*x[ij][v];
}

//Listing AGV Routes and Job-AGV Assignments
tuple SOLRoute
{
arc ij;
int v;
int value;
int sc;
int a;
float b;
int load;
}

tuple SOLAssignment
{
jobs w;
int v;
int sc;
int value;
}

```

```

int pt;
int dt;
int lateness;
}
tuple SOLCharge
{
int v;
int i;
int value;
}

{SOLRoute} RoutesOfAGVs = {<ij,v,x
[ij][v],sc,a[v][ij][sc],b[v][ij],load[v][ij]> | ij in A, v in V, sc in
S :x [ij][v]==1};
{SOLAssignment} JobAGVAssignments = {<w,v,y
[w][v],sc,pt[w][sc],dt[w][sc],lateness[w][sc]> | w in J, v in V,sc in S :y
[w][v]==1};
{SOLCharge} ChargePlanOfAGVs = {<v,i,c[v][i]>| i in N, v in
V :c[v][i]>=1};

//Solution Representation
execute{
writeln(JobAGVAssignments)
writeln(RoutesOfAGVs)
writeln(ChargePlanOfAGVs)
}
string outputRange1;
string outputRange2;
string outputRange3;
int numRows1;
int numRows2;
int numRows3;

execute {
numRows1 = JobAGVAssignments.size;
outputRange1 = "'JobAssignments'!A2:" + "I" + (numRows1+1);
writeln(outputRange1);

numRows2 = RoutesOfAGVs.size;
outputRange2 = "'RoutesOfAGVs'!A2:" + "H" + (numRows2+1);
writeln(outputRange2);

numRows3 = ChargePlanOfAGVs.size;
outputRange3 = "'ChargePlan'!A2:" + "C" + (numRows3+1);
writeln(outputRange3);
}

```

SBMILP Model .dat file:

```

/*****
* OPL 12.8.0.0 Model
* Author: Selen Burçak AKKAYA
*
*****/
//Sheet Connection For Reading the Input Data File
SheetConnection InputData ("Input Data-Example.xlsx");
//Reading Inputs

//Scenarios

```

```

S from SheetRead (InputData, "'Scenarios'!scenarios");
w from SheetRead (InputData, "'Scenarios'!weightsSC");
//Node & Edge Information
//Nodes
CS from SheetRead (InputData, "'LocationInfo'!CS");
PP from SheetRead (InputData, "'LocationInfo'!PP");
DP from SheetRead (InputData, "'LocationInfo'!DP");
//Node Information
CR from SheetRead (InputData, "'LocationInfo'!CR");
//Arcs
A from SheetRead (InputData, "'LocationInfo'!A");
//Arc Information
d from SheetRead (InputData, "'LocationInfo'!d");

//AGV Information
//AGV Fleet
Vmax from SheetRead (InputData, "'AGVInfo'!Vmax");
Vmid from SheetRead (InputData, "'AGVInfo'!Vmid");
Vmin from SheetRead (InputData, "'AGVInfo'!Vmin");
//Fleet Load Capacities
C from SheetRead (InputData, "'AGVInfo'!LoadCap");
//Fleet Battery Charge Capacities and Initial Charge Information
InCharge from SheetRead (InputData, "'AGVInfo'!Incharge");
B from SheetRead (InputData, "'AGVInfo'!BatteryCap");
//Fleet Speed Information
Speed from SheetRead (InputData, "'AGVInfo'!Speed");

//Job Information
//Job Set
J from SheetRead (InputData, "'TransferJobs'!J");
//Job Information
l from SheetRead (InputData, "'TransferJobs'!load");
r from SheetRead (InputData, "'TransferJobs'!readySC");
due from SheetRead (InputData, "'TransferJobs'!dueSC");
s from SheetRead (InputData, "'TransferJobs'!service");
//Energy Information
crl from SheetRead (InputData, "'AGVInfo'!CRL");
cro from SheetRead (InputData, "'AGVInfo'!CRO"); //Battery charge
consumption while loading/unloading operations (fixed)
chargeRate=1;
//Cost Information
clate=1;
cenergy=1;
ccharge=1;

//Sheet Connection For Exporting the Solution
SheetConnection OutputData ("Solution-Example.xlsx");

//Exporting/writing the solution in the form of tuples
JobAGVAssignments to SheetWrite(OutputData,outputRange1);
RoutesOfAGVs to SheetWrite(OutputData,outputRange2);
ChargePlanOfAGVs to SheetWrite(OutputData,outputRange3);

```