THE IMPACT OF VEHICLE DISPATCHING ON THE DESIGN OF MULTIPLE-TRANSPORTERS MATERIAL HANDLING SYSTEMS

by

Yu-Cheng Shen

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirement for the degree of

DOCTOR OF PHILOSOPHY

in

Industrial and Systems Engineering

APPROVED:

Roderick J. Reasor, Ph.D., P.E., Co-chairman

John E. Kobza, Ph.D., Co-chairman

Ralph D. Badinelli, Ph.D.

Philip Y. Huang, Ph.D.

C. Patrick Koelling, Ph.D.

October 1995
Blacksburg, Virginia

Key words: Material Handling Systems, Automated Guided Vehicle Systems, Transportation, Vehicle Dispatching, Queueing Model

THE IMPACT OF VEHICLE DISPATCHING ON THE DESIGN OF MULTIPLE-TRANSPORTERS MATERIAL HANDLING SYSTEMS

by

Yu-Cheng Shen

Roderick J. Reasor, Ph.D., Co-chairman

John E. Kobza, Ph.D., Co-chairman

Industrial and Systems Engineering

(ABSTRACT)

Although Automated Guided Vehicle Systems (AGVS) have been around for more than thirty years, designing an AGVS is still a difficult process because of the interaction of numerous system decisions. The many important elements and variables that must be considered when designing AGVS include the number and location of pickup and delivery stations, the number of vehicles, the routes used by vehicles, the dispatching rules, and the guide path layout.

There are two basic categories of AGVS control: static and dynamic. A static control system requires the automated guided vehicle (AGV) to run the same route continuously with stops at each pickup/delivery station. On the other hand, vehicles in the dynamic control system can be routed to different stations using different paths. There are two types of dynamic vehicle control: workcenter-initiated and vehicle-initiated dispatching rules. The system invokes the workcenter-initiated dispatching rule when one workcenter has a pending request and more than one vehicle is available to pickup the request. Vehicle-initiated dispatching rule is employed when one vehicle is free and there is more than one outstanding request in the system. Most research to

date analyzes only the static aspect of AGVS.

This research attempts to find the minimum number of vehicles needed in an AGVS, using dynamic control of the vehicles, such that the chance of a vehicle-initiated situation occurring is less than a given very small threshold. Under these conditions, load requests will have the least chance of waiting to be picked up. Due to the stochastic behavior of the AGV systems, the proportion of time that the system is in either workcenter- or vehicle-initiated rules is unknown. In order to minimize the waiting time for the load requests while at the same time maintaining a minimum number of vehicles in the system, this research utilizes only workcenter-initiated dispatching rules. A model using queueing theory and Markovian processes is developed to investigate the relationships among empty vehicle travel time, number of vehicles, and the various types of workcenter-initiated dispatching rules. This model is then used to formulate a dispatching-rule based algorithm (DRBA) to determine the minimum number of vehicles. Also, given the number of vehicles required in the system, this research investigates the impact of the nearest-vehicle and farthest-vehicle dispatching rules on the steady-state system performance, e.g., the average waiting time of the load request.

The model and algorithm are able to (1) improve the estimate of the number of vehicles required in an AGVS in the early design phase; (2) provide a better understanding of the efficiency of using the different types of workcenter-initiated dispatching rules for various workload conditions; and (3) generate analytical results that can be used as initial estimates for more detailed simulation studies.

ACKNOWLEDGMENT

To hold a doctoral degree has been a dream of mine since my undergraduate years. This dream has been greatly influenced by my mother, Shen Chen Jing-Yueh, and my sister, Shen Li-Huei, who believe in the importance of education. The values of perseverance and hard work that my mother taught me are two major elements contributing to the realization of my dream. Also, my sister always solace me when things are not going in my way. I am very thankful for my mother's sacrifices over the past six years, and I am very grateful for both my mother and sister's encouragement, and their beliefs in my abilities to achieve the goals of my life.

The complete of this doctoral dissertation would not be possible without the intellectual contributions of several people. First, I would like to extend my sincere gratitude to my two research committee chairmen, Dr. Roderick Reasor and Dr. John Kobza. Dr. Reasor graciously arranged financial assistance and support for me at Virginia Tech, without which, my life here would be unbearable. Dr. Kobza, on the other hand, guided me through the tedious process of writing and fine-tuning my dissertation. In addition, I would like to thank both of them for their valuable feedback on my work. The members on my doctoral committee are Drs. Ralph Badinelli, Philip Huang, and Patrick Koelling. Many thanks to their helpful and insightful suggestions during the course of my work.

There are also many people, who, in one way or the other, contributed to the fulfillment of my dream. Dr. Linda Lau, a good friend of mine, always tells me that there is a light at the end of the tunnel. It was this invisible light that I hold on to throughout this ordeal of my life. I am very grateful for her endless encouragement over the past few years and the meticulous editing of my dissertation. My greatest appreciation to two of my fellow students, Rajiv Totlani and Zafar Ansari, for their generous help in developing the simulation programs used to test the analytical models proposed in my research.

Finally, I would also like to say thank you to all those people who continue to have faith in me, and who help to make my stay in this great country a pleasant one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Material flow is an integral part of the design of any manufacturing system. The designer of a material flow system is concerned not only with the specifications of individual system components, but also the associations between the components and the interactions of the material flow system within the manufacturing system itself. Basically, material handling is the moving of materials in the most economic and safe manner. Hence, the primary goal of material handling systems (MHS) is to reduce costs and waste, but at the same time, increase productivity and quality of service. In order to achieve this goal, MHS must be designed to provide the machining and production units with the needed material in a timely and economic fashion. The importance of this goal is further justified by the introduction of the *just-in-time* (JIT) concept.

An automated guided vehicle (AGV) is a driverless vehicle used for transporting goods and materials throughout a facility, usually by following either a wire guidepath embedded in the floor or a chemical guidepath painted on the floor (Hodgson, King, and Monteith, 1987). The applications of AGVs can be found in office environments, such as hospitals and post offices, as well as manufacturing areas such as the Volvo automobile organization. In the last thirty years, AGVs have progressed from simple driverless vehicle systems to sophisticated computer-controlled vehicle systems. Today, many AGVs have powerful on-board microcomputers that increase local guidance capabilities and the flexibility needed to accommodate any changes on the factory floor. Automated guided vehicle systems (AGVS) encompass not only the vehicle and its guidance system, but also the control of one or more such vehicles in the routine performance of the horizontal material handling tasks for which the system is designed. The AGVS may be controlled by a single host or by a hierarchy of computers. A computer system determines the vehicle's destinations and routes so that throughput is maximized, controls traffic to avoid collisions and minimize blocking, and schedules pickup and delivery of loads to optimize some appropriate economic or safety criteria. When properly designed and controlled, AGVS can provide significant savings in material handling costs and reductions in process inventories. However, while hardware guidance technology has seen extraordinary progress in the past twenty years, little progress has been made in the design and operation of AGVS.

The many important elements and variables that must be considered when designing AGVS include the number and location of pickup and delivery stations, the number of vehicles, the routes used by vehicles, the dispatching rules, the guide path layout, and the flow path. In addition, these decision variables are interrelated with each other, and such interdependencies further complicate the design problem.

There are two basic categories of AGVS control: static and dynamic. A static control system requires the AGV to run the same route continuously and to stop at each pickup/delivery station. On the other hand, vehicles in a dynamic control system can be dispatched to different stations using different paths. The dynamic AGV control systems are more applicable to job-shop type environments where demands change over time. So far, most research analyzes only the static aspect of AGVS, and factors such as the time at which loads become available and blocking or congestion in the system are not weighted (Blair, Charnsethikul, and Vasquez, 1987; Leung, Khator, and Kimbler, 1987; Maxwell and Muckstadt, 1982).

There are two types of vehicle dispatching: workcenter- and vehicle-initiated (Egbelu and Tanchoco, 1984). The workcenter-initiated dispatching problem exists when there is one workcenter with a request and there are more than one vehicle available to serve this request. The vehicle-initiated task assignment problem occurs in situations where there is more than one workcenter with requests pending, and there is only one vehicle available to serve these requests. When a system is in the vehicle-initiated situation all the time, this means that the number of vehicles is not sufficient for the load requests and the materials have to wait to be removed. On the other hand, when the system is always in the workcenter-initiated situation, then there are too many vehicles in the system. Although the dispatching rules used in a typical system are usually a combination of both the workcenter- and vehicle-initiated rules (Malmborg, 1990), an ideal environment is to have as few vehicles as possible while at the same time keeping the proportion of time in which the system uses the vehicle-initiated rules as small as possible. This can be accomplished by choosing an appropriate workcenter-initiated dispatching rule.

In most applications, the effects of dispatching are approximated using rules-of-thumb to characterize the volume of empty vehicle travel that will be expended in a system, e.g., assuming that the empty vehicle travel volume (time) is equal to the loaded vehicle travel volume (Egbelu, 1987). However, dispatching rule modifications have a major influence on the operating dynamics of a materials handling system. For example, they can be used to temporarily increase the handling capacity of a system by reducing the empty vehicle travel time and by imposing vehicle conserving transaction sequencing (Malmborg, 1991). This is a short term tool because resource-conserving rules tend to produce variations in service levels and vehicle utilization that are unacceptable for steady-state operation. Yet, it does provide a means for systems to deal with temporary peaks in the workload. This can produce significant savings in a system by avoiding the need to design for the worst-case operating scenario.

Based on the comprehensive review of AGVS conducted so far, there is a lack of discussion of the importance of dispatching rules and the impact of multiple vehicles in the AGVS. Subsequently, my research examines the impacts of the different dispatching rules and the number of vehicles on the steady-state system performance (e.g., the expected waiting time of a load request). However, due to the unpredictability of the dynamic behavior of the AGVS, the proportion of time that the system will be in the vehicle-initiated or workcenter-initiated situations is unknown. In order to minimize the waiting time for the load requests while at the same time maintain a minimum number of vehicles in the system. The research utilizes only two of the workcenter-initiated dispatching rules, namely, the nearest-vehicle and farthest-vehicle rules. This is because the empty vehicle travel incurred by the other workcenter-initiated dispatching rules will be a convex combination of both the shortest travel time rule and the longest travel time rule (Malmborg, 1991).

The objective of this research, then, is to develop an analytical model using queueing theory and the Markov process to examine the relationships among empty vehicle travel time, the number of vehicles, and the various types of workcenter-initiated dispatching rules. Given the material flow matrix, the travel time between stations, and the preferred dispatching rule, this research presents an algorithm which can be used by system designers to determine the number of vehicles to purchase in order for the AGVS to perform without any waiting time, or to determine the waiting time for a given number of vehicles.

Specifically, this research attempts to find the number of vehicles needed in the system, such that (1) the chance of a vehicle-initiated situation occurring is very small, so that the load requests will have the least chance of waiting to be picked up; and (2) the investment on vehicles purchased is kept to a minimum. Also, given the number of vehicles needed in the system, this research investigates the impact of the nearest-vehicle and farthest-vehicle dispatching rules on the steady-state system performance, e.g., the average waiting time in the queue. Finally, based on the developed analytical results, this research conducts a series of design experiments for the various frequencies of material flows among pairs of stations.

In light of the above objective, this research addresses three major topics of literature. First, a detailed description of AGVS is provided. Second, many decision variables that will affect the design and control of AGVS are discussed. Third, two categories of vehicle dispatching rules (workcenter- and vehicle-initiated) are reviewed. After the literature review, the research procedures and the discrete-time Markov Chains model are presented. In Chapter Six, simulation studies are conducted to test the validity of the model. Finally, this study concludes with the contributions of the research.

2. BASIC CONCEPTS OF AUTOMATED GUIDED VEHICLE SYSTEMS (AGVS)

Today, many manufacturing companies are utilizing flexible systems that are capable of rapidly and efficiently changing product mixes, demands, and designs (Ammons and McGinnis, 1987). A flexible manufacturing system (FMS) possesses the following distinct features:

- a set of machines or workstations which do not require significant setup or change-over time;
- a materials handling system (MHS) that is automated and flexible;
- a set of linked computers which (1) directs the routing of parts through the system and traces the status of all parts in progress, (2) contains and transfers instructions for the processing of all operations, and (3) monitors the performance of operations; and
- the storage of finished parts or work-in-process (WIP).

Due to the dynamic behavior of the product parts, the materials handling system (MHS) is a very important component in the FMS. An automated materials handling (AMH) system consists of conveyors, automated guided vehicles (AGVs), automated storage and retrieval systems (ASRS), and robots. The AGVs transport the parts between each pair of subcomponents.

Automated guided vehicle systems (AGVS) are a form of industrial automation that have proven their worth in installation. They increase the level of quality of transportation and the flexibility, while reducing working costs and materials handling inefficiency. However, many U.S. companies still shy away from investing in AGVS, despite the fact that this technology can help improve competition in the global marketplace. U.S. companies have reservations about investing in AGVS because (1) of financial justification and the analysis barrier; (2) they believe that AGVS may not work; (3) AGVS technology is perceived as a threat by lower- and middle-level managers; (4) corporate management is reluctant to risk their company's future with a technology that they don't understand; (5) custom vehicles and software packages increase costs, and the potential for problems retard the acceptance of the new technology; and (6) the perception that AGVS is ephemeral (Gould, 1990). The managers' misconceptions are partly due to the lack of communication in the design process between vendors (suppliers) and managers. Nonetheless, cost justification and analysis performed during the early design stage will definitely improve the acceptance of AGVS.

2.1 SIX TYPES OF AGVs

There are six types of automated guided vehicles: *towing vehicles*, *unit-load vehicles*, *pallet trucks*, *fork trucks*, *light-load vehicles*, and *assembly-line vehicles* (Koff, 1987).

(1) *Towing vehicles* were the first type introduced and are still a very popular type. *Towing vehicles* can pull a multitude of trailer types and have capacities ranging from 8,000 pounds to 50,000 pounds.

(2) *Unit-load vehicles* are equipped with decks which permit unit-load (i.e., the vehicles can carry only one load request at a time) transportation and automatic load transfer. The decks can be lift-and-lower type, powered or nonpowered, roller chain or belt decks.

(3) *Pallet trucks* are designed to transport palletized loads to and from the floor level without the need for fixed load stands.

(4) *Fork trucks* are a relatively new type of guided vehicles, which can be used to serve palletized loads both at the floor level and on stands.

(5) *Light-load AGVs* are used to transport small parts, baskets, or other light loads through a light manufacturing environment. Hence, the space requirement for the light-load AGVS is much less than other types of AGVs.

(6) *Assembly-line vehicles* are adapted from light-load AGVS for applications involving serial-assembly processes.

The type of vehicles used in an AGVS depends on the environment in which the vehicle will work, and the loads that the vehicle will deliver. For example, unit-load carriers are normally utilized in warehousing and distribution system where the guide path lengths are relatively short but volume is high.

2.2 FIVE FUNCTION OF AGVS

The basic functions of an AGVS are guidance, routing, traffic management, load transfer, and system management (Koff, 1987). Guidance allows the vehicle to follow a predetermined route, which is optimized for the material flow pattern of a given application. AGVS steering control allows AGVs to maneuver physically in different ways. There are two basic types of AGVs steering control, "differential-speed steer control" and "steered-wheel steer control". A differential-speed control uses two fixed wheel drives and varies the speeds between the two drives on either side of the guide path to allow the vehicle make a turn. The steered-wheel control uses an automotive-type steering control in which a front steered wheel turns to follow the guide path.

While vehicle guidance dictates the direction of the vehicle, routing determines the route that the vehicle should take. There are two types of routing methods, frequency select and path-switch select. In the frequency select method, the vehicle is presented with two or more frequencies at the decision point, representing the available directions. The selection of frequencies depends on the direction in which the vehicle wishes to go. The path-switch method employs a switch control that communicates with the vehicle and controls the activation device. The control box turns on one of the available paths while turning off the others.

Traffic management helps prevent collision and conflict between vehicles. Three traffic management techniques are zone control, forward sensing, and combination control. With zone control, the AGVS layout is segmented into separate zones. Each zone can be occupied by only one vehicle at any time. Zone control is accomplished by three methods: distributed, central, and on-board. Forward sensing control requires one sensing system on-board each vehicle to detect a vehicle in front of it. Three types of sensors are available: sonic, optical, and bumper. Combination control employs both zone and forward sensing control.

Load transfer can be accomplished either manually or by automatic couple and uncouple, power roller/belt/chain, power lift/lower, or power push/pull. The manual load transfer method includes uncoupling trailers and moving the trailers of the AGVs to the given work stations, or a simple roller-bed transfer from the AGVs to fixed roller stations by manually pushing the load off of the vehicles. Automatic couple and uncouple load-transfer methods are becoming more popular. Usually, the vehicle stops on a side spur and automatically uncouples trailers which the vehicle is pulling. Then, the vehicle proceeds on the main line to its next destination, where it stops and backs into another set of trailers and couples to them. Powered roller, belt, or chain transfer has been a standard for many years. Unit-load carriers are usually equipped with three power decks so that they may automatically transfer loads to and from fixed stations. The lift/lower fork truck is a relatively new concept that allows the guided vehicle to pick loads up off of the floor and deposit them on the load stands of in racks.

System management is divided into two areas: vehicle dispatching and system monitoring (Koff, 1987). The vehicle dispatch method includes on-board dispatch, off-board call systems, remote terminal, central computer, and combinations of these techniques. In an on-board dispatch system, each vehicle is equipped with a control panel which is used by the operator to dispatch the vehicle. A call box is dedicated to each vehicle in off-board call systems. The call box is used to communicate with and dispatch the vehicle. Remote terminal control allows the operator to control and dispatch the individual vehicle with one graphic display panel. The central computer control, the highest level of control, controls the movement of each individual vehicle. Although this method is similar to the remote terminal control in many ways, it eliminates the need of an operator. It is also possible that the control systems adopt a combination of the various control methods mentioned above.

System monitoring helps to prevent unnecessary loss or system breakdown. Three approaches to monitoring are locator panels, CRT color graphics displays, and central logging and reporting. While a locator panel provides the information about whether a vehicle is in a given area of the guide path, the CRT graphics reveal the location and status of each vehicle. Central logging and reporting keeps and develops chronological data on the system's performance.

2.3 SELECTION OF AN AGV

Although the technology of an AGV is acclaimed to be beneficial to the manufacturing organization, its growth rate has been relatively slow. This problem could be attributed to the lack of a set of objective criteria that will assist users in selecting AGVs for intended applications. Subsequently, Shelton and Jones (1987) have developed a three-step AGV selection model, and have presented an analytical framework in which it can be used. The three steps include (1) attribute identification, (2) attribute selection, and (3) attribute ranking.

2.3.1 Attribute Identification

The first step involved in the selection process is to identify the specifications, requirements, and preferences for the AGV attributes that are applicable to the user's production environment. Common attributes of AGVs suggested so far in the literature include return of investment (ROI), type of guide path, cost, load capacity, speed, and safety features (Kulwiec, 1984; Maxwell, 1981). Shelton and Jones (1987) conduct a more thorough search and formulate a comprehensive list of possible attributes related to the vehicle and vendor support and services.

2.3.1.1 The Vehicle

Significant attributes pertaining to the vehicle are given as follows:

1. cost;
2. guide path: wire, optical/chemical, others;
3. off-path travel capability;

4. vehicle width;

5. load capacity: weight, height, depth, etc.;

6. maximum speed unloaded and loaded;

7. run-time between charges;

8. charging time required;

9. on-line recharging capability;

10. bi-directional movement along a path;

11. maintenance aids: LED lights for self-diagnosis, low level power indicators, vehicle jack stands, modular components for each maintenance;

12. safety features: pressure-sensitive bumpers, emergency stop buttons, warning lights, warning beeps/messages;

13. vehicle re-start: manual, automatic, and both;

14. turning radius;

15. position sensors;

16. plane of movement: horizontal, horizontal and vertical;

17. lift height;

18. system capabilities: transport only (i.e., unit loads), pick-up, transport, etc.; and

19. the load: unit load, pallet, roll (i.e., carpet, paper), and special attachments' requirement.

2.3.1.2 Vendor Support and Services

Before making any major purchase decisions, users should expect the following types of support and services from the vendors:

1. customization of vehicle;

2. warranty coverage (length of time);

3. reputation of vendor; and

4. commitment to AGVs.

2.3.2 Attribute Selection

In the first step of the AGV selection procedure, the user is given a list of the above attributes and is asked to determine his/her specifications for any or all of the attributes. The selection model has two parts: attribute selection and ranking. Based on his/her preference value function for each continuous attribute, the user chooses not more than fifteen attributes to use as selection criteria. Each of these preference value functions is selected from a given list of fourteen preference curves, or the user can define his/her own curve.

2.3.3 Attribute Ranking

After the attributes are selected, scaling constants (weighting factors) are determined for these selection criteria. The indifference method, in which the user judges the trade-off between the values of different attribute levels, is frequently used. Next, the user ranks the attributes according to the degree of importance by assessing his/her preference and priorities with respect to these attributes.

Based on the information provided by the user (i.e., the preference value functions and trade-off), the selection model ranks the AGV models in the feasible set. Nonetheless, the user is advised also to conduct a sensitivity analysis of the decision model using slightly different preference value functions and variations in the trade-off.

2.4 CAPACITY REQUIREMENTS PLANNING

One of the significant functions in designing the AGVS is to determine the number, type, and assignment of AGVs in order to convey parts of different sizes between workstations. Given a system layout, capacity requirements planning deals with the determination of the types (unit-load or multiple-load, light-load AGVs or fork truck, or the speed of the vehicles) of AGVs required to meet specific material handling requirements. Capacity requirement planning depends on (1) the flow of parts between workstations; (2) the routes traveled by the parts; (3) the size of the unit load; and (4) the moves scheduled between workstations.

Kasilingam (1991) proposes an optimization cost model to perform such critical tasks while minimizing the overall cost. The overall cost is the sum of the AGVs annual operating cost and the parts' transportation cost between workstations. The general model provides the required number of AGVs of each type and the assignment of AGVs to transport parts between workstations. Unfortunately, the cost model has a tendency to underestimate vehicle requirements, because it is based on the net flow of vehicles (Egbelu, 1987b).

2.5 VEHICLE REQUIREMENTS

The number of vehicles required in an AGVS must be determined in the early stages of the decision-making process. Vehicle requirements have been examined in the past using the transportation formulation (Maxwell and Muckstadt, 1982), an analytical model (Mahadevan and Narendran, 1990), a simulation technique (Tanchoco, Egbelu, and Taghaboni, 1987), non-simulation-based calculation approaches (Egbelu, 1987a), and by determining the total empty vehicle travel time (Leung, Khator, and Kimbler, 1987). Traditionally, the detailed simulation technique is used to estimate vehicle requirements for complex systems, such as job shop operations, in which the order generation is random and the shop layout does not follow any recognized pattern. However, because it is costly and time consuming, simulation is not used until the advanced stages of system planning. During the early stages, vehicle planning is done using hand calculations, and these calculations are then used to perform the aggregate economic analysis.

Although the performance results from the non-simulation hand calculation or analytical procedures are not accurate, Egbelu (1987a) believes that such preliminary hand calculations are still valuable to system planners because they can be used obtain a more reliable estimate of the vehicle requirements at the early stages of system planning or equipment selection. The sharpened estimate yields substantial savings in planning and error costs, as well as better quality decisions.

In general, the number of vehicles required is influenced by the following six factors:

1. the designed system reliability;
2. the speed of travel;

3. the number of trip exchanges between workcenters per unit of time;
4. the system guide path layout;
5. the location of load transfer points; and
6. the vehicle dispatching strategy.

Details concerning vehicle reliability and speed are available from the manufacturer. When load sizes are assumed to be fixed, the trip exchanges between workcenters have no association with the types or numbers of vehicles selected. Both the guide path layout and the location of load transfer points are layout-related problems, and currently, there is no systematic manner of selecting an optimal guide path layout for AGVS (Egbelu, 1987a). Hence, only the vehicle dispatching problem is addressed in the estimation of vehicle requirements.

The vehicle dispatching problem deals with assigning vehicles to pickup load and deliver orders. Thus, dispatching strategies determine the number of empty vehicle runs that occur within a specific work period. Vehicle dispatching and control procedures also account for the degree of interference experienced by the vehicles during any time interval. The vehicle dispatching component is the primary driving force for vehicle distribution throughout the AGVS. The dispatching rules used by the AGVS will determine the blocking/congestion time encountered by the vehicle during the shift. This, in turn, will affect the number of vehicles needed in the system.

3. AGVS DESIGN AND CONTROL

There are two types of AGVS issues: design and control problems (Johnson and Brandeau, 1993). Working towards the goal of meeting a minimum service level, design decisions determine the workstations that should be automated for service by the AGVS, the routes to be used by the vehicles, the number and location of the pickup and delivery stations in the system, and the number of vehicles needed in the AGVS. Generally, this service level refers to the mean waiting time for the load requests at the stations for the vehicle not to exceed a pre-specified level. On the other hand, control decisions determine when to respond to demands, how to group demands, as well as the order in which the stations are served. Yet, the design and control issues are closely related. Both issues affect the AGVS performance; as well, solutions for the AGVS design problems must depend on AGVS control and vice versa. Due to the interdependency between these two issues, researchers have to make some appropriate assumptions about the system in order to obtain reasonably meaningful and manageable results.

Although AGVS has been around for more than thirty years, designing an AGVS is by no means an easy task because of the need for numerous decision variables that interact within the system. Important decision variables that must be considered in designing an AGVS include the number and location of pickup and delivery stations, the number of vehicles, the routes used by vehicles, the dispatching rules, the guide path layout, and the flow path (Bozer and Srinivasan, 1991; Egbelu, 1987a and 1987b; Gaskins and Tanchoco, 1987; Gaskins, Tanchoco, and Taghaboni, 1989; Kaspi and Tanchoco, 1989; Leung, Khator, and Kimbler, 1987; Tanchoco, Egbelu, and Taghaboni, 1987; Usher, Evans, and Wilhelm, 1988). In addition, these decision variables are interrelated with each other, and such interdependencies further compound the complexity of the problems.

Approaches to developing control rules for an AGVS include integer programming for seeking the optimum solution (Maxwell, 1981; Maxwell and Muckstadt, 1982), optimization, heuristic algorithm (Leung, Khator, and Kimbler, 1990), queueing theory, and Markov Chains (Hodgson, King, and Monteith, 1987). However, most of the existing research on automated material handling systems (MHS) relates to operational control problems and is based on simulation studies. Nonetheless, there is very limited work on the design and control aspects of AGVS.

The following sections present a detailed analysis of these important decision variables: guide path layout, direction of the guide paths, and number of vehicles.

3.1 GUIDE PATH AND LAYOUT DESIGN

Some research has been conducted on facility layout, guide path, and pickup/delivery station location. However, due to the interdependent relations between guide path design and the location of pickup/delivery stations, researchers have to make certain assumptions about one when considering the other. Therefore, the results generated by these researchers usually do not provide an optimal solution because the location of pickup and delivery stations and guide path design have to be considered simultaneously. The following section focuses on these issues and examines the guide path and flow path design of an AGVS.

3.1.1 Guide Path Design of AGVS

The guide path configuration used in an AGVS will determine the routes used by the vehicle, which, in turn, will influence both the loaded and empty vehicle travel time of the vehicle. In a conventional AGVS, a segment of the guide path may be connected to many other segments of the same guide path. An AGVS layout, adapted from Wilhelm and Evans (1987), for job shop application is illustrated in Figure 3.1. The arrows in this figure represents the direction of the guide path. This design requires the use of fairly sophisticated vehicle routing and vehicle dispatching. Further, because the vehicles interact with one another, conventional AGVS's present some difficult design and analysis problems. Consequently, some researchers, efforts to simplify the guide path configuration design have yielded two new approaches, namely, Tandem AGVS and a Single-Loop guide path, to supplement the conventional guide path design.

3.1.1.1 Tandem Configuration

Bozer and Srinivasan (1991) propose the Tandem AGVS configuration concept and also an algorithm to implement it. The Tandem configuration is based on the partitioning of a given set of pickup/delivery points (the number and location of the pickup/delivery points are assumed to be known) into non-overlapping, closed-loop, single vehicle zones that are interconnected by transfer points. The partition algorithm is based on the Set Cover Problem. Each partitioned zone is served by a vehicle, and the control logic in each partition is implemented using Bartholdi and Platzman's (1989) decentralized vehicle control rule (also known as the FEFS dispatching rule) for closed-loop zones. Under the FEFS rule, a vehicle is never idle and is always traveling loaded or empty.

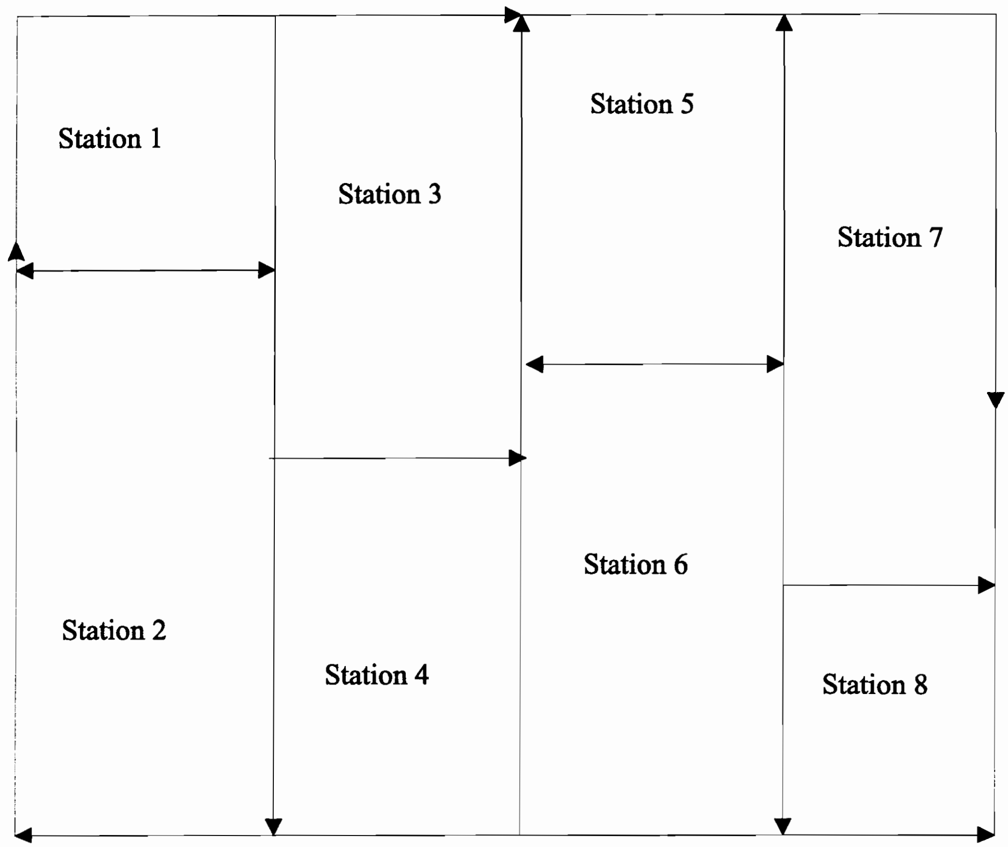

Figure 3.1 AGVS Layout for Job Shop Application

The Tandem configuration offers several advantages: (1) it simplifies the control system because there is only one vehicle in each zone, and the vehicles will never interfere with each other; (2) it offers flexibility because zones can be added, removed, or modified without affecting other zones; and (3) blocking and congestion can be eliminated. On the other hand, there are also some limitations of the Tandem configuration: (1) some loads may have to be handled by several vehicles before they reach their destination, causing additional delays at transfer points; (2) more floor space and investment are needed for the transfer points; and (3) the routing for the material might become more complicated.

Despite the promising aspects of the Tandem configuration, given that the performance and the control logic between zones are still unknown, this configuration can only be viewed as an alternative guide path design.

3.1.1.2 Single-Loop Guide Path

Unlike the conventional guide path layout, the Single-Loop layout design attempts to reduce the guide path to one single-loop connecting all the pickup and delivery stations. Tanchoco and Sinriech (1992) present an integer programming formulation procedure to find a "good" single loop guide path and to locate the pickup/delivery stations along the loop. Contrary to other researchers' assumptions, Tanchoco and Sinriech's (1992) construction procedure assumes that the location of the pickup/delivery stations are unknown. In comparison with the conventional layout, the Single-Loop guide path uses a simpler control logic but requires more vehicles. Performance results reveal that the Single-Loop configuration performs as well as the conventional configuration, but only under the light and average shop workload. As the workload increases, the Single Loop's performance decreases accordingly. This is because of the congestion and blocking that occurs during vehicle movement. Unfortunately, Tanchoco and Sinriech (1992) fail to identify the control logic used within the single-loop and to provide suggestions to avoid blocking and congestion.

3.1.2 Flow Path Design of AGVS

Flow path, in addition to vehicle and control logic, is one of the three major components of an AGVS that determines the location and direction of vehicle movement. There are two different types of flow path layout, namely, uni-directional and bi-directional. The vehicle in the uni-directional guide path design can only travel in one direction, while the vehicle in the bi-directional guide path design can travel in both directions.

Frequently, the material flow system adopted in a plant will influence the flow path configuration at any given time. Not only can the direction of the flow path be easily changed as the need arises, but the computational time required to obtain a new flow path should also be within a reasonable range. The flow path layout problem attempts to find the direction of flow for each segment in the flow path network. In the guide path layout design, an AGVS network is modeled as a graph consisting of nodes connected by a set of arcs. The nodes represent points on the network, such as pickup and/or delivery stations and intersections. At these nodes, the next immediate segment in the path is selected by an AGV traversing a route. Traffic conditions are also checked to evaluate the feasibility of allowing the AGV to proceed. The arcs represent the physical or virtual guide paths in which vehicles travel from node to node. Associated with each arc is a cost which denotes either the distance between the two end points of the segment, or the time required by an AGV to cross the arc.

3.1.2.1 Uni-Directional Flow Path for AGVS

Typically, the AGVS operates with uni-directional flow path layout. The advantages of the uni-directional flow design are the simplicity of design and control, as well as the reasonable cost of installing the guide path (Egbelu and Tanchoco, 1986).

The AGV guide path layout problem was first formulated using the zero-one integer linear programming technique (Gaskins and Tanchoco, 1987) to find the travel direction of the vehicle on the segments of the guide path. In their model, the facility layout is given, along with the locations of the pickup/delivery stations. The system is represented by a node-arc network. The binary variables in the formulation indicate the presence or absence of direct linkages between nodes. Given the layout and location of the pickup/delivery stations, the minimization of the loaded vehicle travel distance is subjected to two constraints: (1) connectivity constraints--which guarantee that there is at least one arc entering and leaving a node; and (2) reachability constraints--which assure that each node can be reached from any other node.

To apply the formulation, the flow intensity between each pairwise combination of load transfer points and the distance associated with alternative combinations of paths between nodes have to be determined. Each alternative path is represented by the product of a different combination of binary decision variables. The objective function value consists of the sum of these products for every pair of nodes multiplied by the corresponding flows on the network.

Gaskins and Tanchoco's (1987) formulated problem is solved by using a multi-purpose optimization package (Cohen and Stein, 1978). The formulation produces initial results much more quickly than those obtained using the simulation method. However, the number of zero-one variables expands drastically, and the computational efficiency becomes unacceptably low, as the size of the system increases.

Due to these impracticalities, Kaspi and Tanchoco (1990) propose an alternative formulation and present a computationally-efficient procedure based on the material flow from-to chart. The objective of the analytical model is to minimize the total travel time/distance that the vehicles have to traverse through the uni-directional flow path layout to satisfy the demand requirements subjected to numerous constraints. These constraints include (1) ensuring the feasibility of a path from the pick-up node to the delivery nodes; (2) uni-directionality; (3) at least one input and output arc for each node; (4) one output arc from the pick-up node using the path from the pick-up node to the delivery node; (5) one input arc to the delivery node using the path from pick-up node to the delivery node; and (6) a number of input arcs equal to the number of output arcs. The branch-and-bound algorithm, with an emphasis on the depth-search first and backtracking technique, is used so that a feasible solution can be obtained quickly.

Although Kaspi and Tanchoco's (1990) model is proposed with the consideration of the from-to flow charts and the branch-and-bound algorithm with an emphasis on tracking, unfortunately, it also suffers from the same problem of computational efficiency that the model is proposed to solve. In other words, when the problem size increases to a less manageable level, the proposed algorithm is computationally extensive and untrackable. The authors also recognize the pitfall of a vehicle's inability to leave a delivery station if there is no outgoing request in their analytical model. To remedy this setback, the authors propose another algorithm to accommodate this disadvantage in the flow path layout. Nonetheless, this pitfall arises because the authors focus on the deterministic behavior of the system. Simply put, if the authors would have considered the stochastic aspect of the system, the dispatching of vehicles would be considered in such situations. Hence, the results of the formulation can only be treated as references for the design of flow path layout.

Subsequently, in order to reduce the problem of computational intractability, Sinriech and Tanchoco (1991) propose an enhanced algorithm which branches only on the intersection nodes of the network. Based on the same analytical model proposed by Kaspi and Tanchoco (1990), the enhanced algorithm seeks to minimize the branching process, which, in turn, results in a shorter solution time. Only nodes located at intersection points will be considered when branching, because nodes that are not at the intersection points have only one incoming and outgoing arc, and the direction of the arcs are dependent on each other.

The algorithm implementation involves two stages: preparation, and branch-and-bound. The preparation stage eliminates paths that do not have flows between them, i.e., this stage considers only paths that include pickup and delivery stations that have flows between them. The flow length between the considered nodes is then recorded, and the arcs are arranged in increasing order of the length. The branch-and-bound stage involves setting the direction of the new arc at each step of the branch-and-bound procedure and calculating a lower bound.

Although the algorithm proposed by Sinriech and Tanchoco (1991) is more computationally efficient than Karps and Tanchoco's (1990), the material flow matrix in the numeric example is unbalanced, because the in-flows in some nodes are not equal to the out-flows of those nodes. This results in the accumulation of vehicles at those nodes. Furthermore, the assumption that the material flow matrix includes both the loaded and empty vehicle travel time are somewhat erroneous because the authors fail to mention how empty vehicles are dispatched, which will affect, in turn, the total vehicle travel distance/time.

Sharp and Liu (1990) propose a two-phase approach for determining where to include spurs and shortcuts in a uni-directional guidepath network. Phase I is an analytical procedure based on Markov processes for deciding on guidepath shortcuts, the existence of spurs, and the number of carriers required to satisfy the specified load movements, subject to some constraints. Phase II of the approach is a simulation procedure that is used to validate the results of Phase I. The results indicate that the two-phase approach is a promising avenue for the design of material handling systems; that is, the shortcuts and spurs can be a supplemental alternative for guide path design in AGVS.

3.1.2.2 Bi-Directional Flow Path for AGVS

The uni-directional guided path layout has had prevalence in the designing of automated guided vehicle systems for a long time. The main reason for the use of a uni-directional flow is its simplicity in design and control. Systems using bi-directional flow are more difficult to design because of additional control difficulties (Egbelu and Tanchoco, 1986). The advantages of bi-directional flows as opposed to uni-directional flows, have been well justified in other traffic flow systems, such as highways and railroads (Agent and Clark, 1982). Fuji and Sandoh (1987) consider the problems of simultaneously routing all vehicles on a bi-directional network. They solve a number of shortest path problems and then use an assignment scheme to pair the vehicles with the paths. Daniels (1988) develops a branch-and-bound algorithm to determine the shortest conflict-free path when adding a vehicle to an operating bi-directional network. The advanced technology makes it much easier to develop and design the control system. As a result, there is a need to investigate the use of the bi-directional guide path layout for an AGVS design.

3.1.2.2.1 Characteristics of a Bi-Directional Flow Network

Like the uni-directional flow path, the bi-directional guide path can also be modeled as a node-arc network. A segment/arc is uniquely defined by the nodes that bound it. Because two-way traffic is permitted, the traffic reference in a particular segment must specify the direction of flow. For instance, if the traffic flow is from node *a* to node *b*, then the traffic reference in that segment will be viewed as traveling in arc (*a*,*b*), and vice versa. The design issues involved in a bi-directional flow network are the *traffic flow model*, *routing through an intersection*, and the *design of buffering areas*, which are described below.

Traffic flow models: In the traffic flow model, the cartesian coordinate system is used for point location and vehicle tracking in which distances between nodes are assumed to be rectilinear for "L" shaped guide paths and Euclidean for straight line guide path segments. Also, the concepts of check zones and check points are introduced. Check zones are designed to ensure safe and collision-free intersections and to moderate traffic control. A check zone is a constrained region, where only one vehicle is allowed to be at any one time. These regions are usually located around the nodes. Check points, located at the intersection of check zones and segments, are decision points for vehicles in transit for holding, turning, etc.

Routing through an intersection: Due to the number of ramps around the intersections, more complicated algorithms are needed to control and route vehicles through intersections in a bi-directional network.

Design of buffering areas: Vehicular conflicts, an operational problem in designing the bi-directional network, generally occur when two vehicles traveling in opposite directions desire to use the same aisle. The conflict can be resolved by holding one vehicle while giving the other the right-of-way. Although vehicle buffering areas offer the ease of vehicle control, there is a trade-off for the space economy. The designs, locations, and sizes of the buffer are important considerations in the design of the bi-directional guide path. There are three alternative designs for the buffer areas: loop, siding, and spur. The loop design features two uni-directional loops per segment and the loops usually locate at the end of the aisle. Hence, for each node, the number of buffering facilities is equal to the number of directions that vehicles can enter the node. For the siding design, a uni-directional siding is at each end of an aisle, located close to the end nodes, and it serves only vehicles traveling in its direction of orientation. The spur design serves vehicles traveling in only one direction, with the sequence of the vehicle departure based on the last-in first-out approach.

3.1.2.2.2 Three Types of Bi-Directional Guide Path Systems

There are three ways to implement a bi-directional guide path. First, the *parallel tracks on each aisle design* is similar to that used for major highways. With sufficient clearance space between parallel tracks, there is virtually no interference between vehicles traveling in opposite directions. Its advantages include a reduced distance traveled by vehicles responding to the pending request. Also, blocking occurs only at the junctions. However, its disadvantages are a lack of space economy, and the need for traffic control at the intersections. Assuming that movements are permitted in each direction, the uni-directional flow layout requires only two interchange ramps at the intersections, while the bi-directional flow layout requires eight. Hence, this design has the highest initial investment cost.

The *single switchable track design* is similar to single railroad tracks between terminals (Petersen, 1977). For each segment or aisle, the traffic flow takes place in one of the two directions. The flow signal is transmitted to the segment, which operates as a switch. A signal in the reverse direction is automatically turned off. Thus, vehicles travel only in one direction at any one time. Due to the complex task of determining the direction of the flow at any given time, this type of bi-directional flow presents control problems at the intersections. The system controller has to handle the difficult intersection control problems, as well as interference between the vehicles within the aisles. Temporal vehicle buffering areas must be allocated in the guide path to hold blocked vehicles. The number and capacity of the buffer areas depend on fleet size, routing strategy, layout, and the size of the facility.

Finally, the *mixed design* combines the characteristics of both the uni-directional and bi-directional flow paths; some aisles of the guide path operate in a bi-directional mode, while others operate on a uni-directional mode. Frequently-used aisles are candidates for bi-directional flow in the mixed design.

3.1.2.3 Comparisons of Uni-Directional and Bi-Directional Flow Designs

The measurements used to compare the uni-directional and bi-directional flow layout designs are the throughput over a fixed amount of time, the number of vehicles needed in order to reach the desired throughput, and the length of time it takes to achieve a desired output rate (Egbelu and Tanchoco, 1986). Both designs are assumed to have the same number of vehicles.

In comparison, the bi-directional flow has better throughput over a fixed amount of time than the uni-directional layouts. The margin of difference decreases as the number of vehicles increases. This can be explained by the increased blocking time; as the number of vehicles increases, the chance of vehicle blocking also increases. However, the higher system throughput in a bi-directional layout is achieved at a slightly lower average vehicle utilization, which is computed based on the total time a vehicle is engaged in the mission. This high-performance low-vehicle utilization can be attributed to the reduction in empty vehicle travel time.

The second comparison suggests that the bi-directional layout will need fewer vehicles to achieve the desired throughput over a fixed time. This cost advantage over uni-directional layout should be considered in the design stage. Finally, the time span to achieve an output target for the bi-directional layout is shorter than that of the uni-directional layout. These time savings can be translated to savings in cost to facilitate adequate economic analysis. In conclusion, there is evidence to indicate that the bi-directional layout is better than the uni-directional layout in terms of shorter empty vehicle travel time and better performance of throughput based on the proposed single switchable track model (Egbelu and Tanchoco, 1986).

3.2 VIRTUAL FLOW PATHS FOR FREE-RANGING AGVS

Flow path, one of the three major components of an AGVS, determines the location and direction of vehicle movement. Traditionally, the flow path is characterized physically by cable lines buried underneath the ground of the facility. Examples of such arrangements are optical, magnetic guide paths, and embedded cables. However, recent advances in the technology permit vehicles to travel without any physical guide path; this is known as a free-ranging automated guided vehicle (Gaskins, Tanchoco, and Tanghaboni, 1989). Dead reckoning, inertial guiding, and ultrasonic imaging processes are examples of such systems. With virtual flow paths, the system controller can easily alter the guide path layout of the system to reflect the material flow demands.

The virtual flow path does indeed provide more flexibility for the system. The vehicles can travel to stations according to the material flow demands, as they do not have to follow any predetermined guide paths. However, potential problems associated with the use of virtual flow paths have not yet been investigated. For instance, the concept of the virtual flow path has no major contribution unless the plant facility has sufficient space to accommodate more than one possible flow path. Further, the traffic control problems at the intersections have not been addressed (Egbelu and Tanchoco, 1986). To illustrate, if all turns are permitted in a bi-directional flow network, then eight interchange ramps are required. As the number of ramps increases, so does the complexity of traffic control. For a virtual flow path, the number of ramps is usually more than eight; thus, the seriousness of the traffic control problem cannot be underestimated. Future research should focus on formulating heuristics to solve this traffic control problem. In addition, the virtual flow path model is also computationally difficult and time consuming.

Although the flexibility of the virtual flow path design is attractive, certain issues, such as space and traffic control problems, must be resolved before the potential of this design can be fully realized.

3.3 DETERMINING THE NUMBER OF VEHICLES

Using an analytical approach, Maxwell and Muchstadt (1982) find the minimum number of vehicles needed in an AGVS by assuming that the demand at each node is constant and known and that the exact timing of deliveries does not matter. First, they use the material flow information at each station (node) to perform an input/output flow analysis, where the empty vehicles are assumed to be assigned to the nearest station with a request. They then solve a transportation problem to find the minimum number of empty trips between each pair of stations. Using both the demand and empty trip information, they find the total required travel time, and thus the minimum number of vehicles necessary to meet the demand equals the total required travel time divided by the available vehicle time per shift. Tanchoco, Egbelu, and Taghaboni (1987) present a queueing theory-based procedure to estimate the number of vehicles needed in an AGVS. Egbelu and Tanchoco (1982) propose a simulation-based approach to determine the vehicle requirements. Tanchoco, Egbelu, and Taghaboni (1987) compare the effectiveness of CAN-Q to a simulation model for estimating the number of vehicles required in an AGVS. The comparison is done by shop throughput in unit loads versus the number of AGVs in the system.

The result shows that CAN-Q is a good tool for determining the lower bound on vehicle requirements, but it underestimates the actual number of vehicles required, as determined by the simulation. This conclusion indicates that CAN-Q can be used in the early design phases of an AGVS to set bounds on system parameters. Unfortunately, CAN-Q does not address the dynamic behavior of the system which will, in turn, affect the number of vehicles in the system.

Egbule (1987) proposes four analytical approaches for estimating the number of vehicles required in an AGVS. In the first method, empty travel vehicle distance is assumed to be equal to the loaded travel time. Therefore, the number of required vehicles, under this rule, is equal to the sum of loaded travel time, empty travel time, and the loading and unloading time of each operation, divided by the total time needed in the shift and vehicle efficiency.

The second estimation method considers blocking and idle time factors, both of which are facility-dependent and have to be estimated for each facility. The recommended ranges of the value for both factors is between 0.10 and 0.15 (Koff, 1985). Hence, the number of vehicles is given by:

$N = (\Sigma\Sigma f_{ij} / T) / (60/\underline{e})$, where f_{ij} is the flow intensity between stations i and j, T is the length of period/shift, and $\underline{e}$ is the mean travel and load and unload time per trip after accounting for blocking, idle, and vehicle efficiency.

The third method involves the computation of net traffic flowing into a workcenter. The net flow of a workcenter i, (f_i), is defined as the difference between the total number of in-flows from other workcenters and the total number of out-flows from workcenter i to other workcenters. If f_i is greater than zero, it implies that there are more deliveries into station i than pickups. Therefore, there will be empty runs from the delivery station of workcenter i to some pickup stations. On the other hand, if f_i is zero, then there will be no empty runs from station i.

While the workcenters with f_i smaller than zero are called net importers of empty vehicles, the workcenters with f_i greater than zero are called net exporters of empty vehicles. There are two assumptions underlying these estimation methods. First, for stations with f_i greater than zero, the empty vehicle will export to the nearest station with f_i smaller than zero. Second, if $\sum f_i = 0$, then there will be no loss or removal of parts at the stations or occasional manual transport of the parts. Hence, the total empty vehicle travel distance is defined as the product of the average distance traveled by the loaded vehicles, and the number of empty or loaded runs between stations. Therefore, the number of vehicles needed in the system is the sum of the mean loaded travel time, mean empty travel time, mean travel time between stations, and the average loading and unloading time divided by the total time needed in the shift.

The last method employs the concept of vehicle dispatching rules, because of the randomness of the load pickup requests. Vehicle dispatching from a station i to another station j depends on (1) the number of deliveries to station i; (2) the number of pickups from station j; and (3) the total number of pickups required during the period. Given that the dispatching of a vehicle from delivery station i to pickup station j may involve an empty vehicle run, the number of empty runs from station i to station j is given by g_{ij}, where g_{ij} is the expected number of deliveries to station i multiplied by the expected number of pickups from station j and then divided by the expected total number of pickups throughout the system. Therefore, the distance of the empty runs between the two stations i and j is equal to g_{ij} multiplied by the distance between these two stations. Accordingly, the number of vehicles required in the system is equal to the sum of the mean loaded travel time, mean empty travel time, and the mean loading and unloading time, divided by the available time per vehicle in the shift.

The four analytical models proposed by Egbelu (1987a) are overly optimistic when compared to the simulation studies employing combinations of the dispatching rules presented by Egbelu and Tanchoco (1984). However, Egbelu (1987a) also concludes that the adequacy of the estimating models should consider the dispatching rules in force. He also finds that the number of vehicles is under-estimated because the analytical techniques ignore, or only heuristically estimate, the effects of vehicle congestion.

Leung, Khator, and Kimbler (1987) propose a mixed integer linear programming model to find the number of vehicles in an AGVS where the carrying capacity and the operating speed of each vehicle type is different. Given the objective to minimize the total vehicle travel time (both loaded and empty), the model considers a static material flow in which the movement of materials between stations is known and fixed at the time when considering the number of vehicles.

4. VEHICLE DISPATCHING, ROUTING, AND SCHEDULING

The three major functions in automated vehicle system management are dispatching, routing, and scheduling. There are many similarities and differences among these three functions. Unfortunately, with the exception of Co and Tanchoco (1991), the literature on AGVS fails to arrive at formal and operational definitions for these functions. In light of this issue, the following subsections will describe the major activities of these functions and delineate their similarities and differences.

The three major functions in vehicle management are formally defined as follows (Co and Tanchoco, 1991):

- Dispatching: the process of selecting and assigning tasks to vehicles;
- Routing: the selection of the specific paths taken by vehicles to reach their destinations; and
- Scheduling: the determination of the arrival and departure times of vehicles at each segment along their prescribed paths to ensure collision-free journeys.

4.1 DISPATCHING

Traditional guide paths for AGVS's are uni-directional and are employed in most current applications for vehicle management in manufacturing environments. Advantages of employing bi-directional guide paths are well documented in traffic flow systems, for example, highways, streets, and railroads (Agent and Clark, 1982). The use of a bi-directional traffic flow network increases productivity in some AGV system installations, especially those that require fewer vehicles (Egbelu and Tanchoco, 1986). At the same time, the control aspect of vehicle management becomes more complicated, which increases the importance of dispatching in AGVS modeling.

Before making a dispatching decision, all available alternatives must be known because the impact of each alternative on the overall system performance is of vital importance to the system designers. The dispatching decision determines the proportion of time each dispatching rule needs to be invoked, as well as the situations which require use of the dispatching rules. The number of vehicles available in the system and the workcenter(s) with requests pending are the main concerns in this system.

Maxwell and Muckstadt (1982) examine vehicle dispatching in the AGVS by proposing a simple time-dependent analysis of vehicle dispatching sequences. The environment considered is assembly operations where the output and input rates are rather constant. The control and design of AGVS for assembly lines are simpler than job-shop or flow-shop circumstances. The dispatching rules proposed by Maxwell and Muckstadt (1982) attempt to assign vehicles to stations so that the time between visits (pickups or deliveries) at each station are distributed uniformly across the time period during the shift. This is done by spreading the trips (pickups or delivery) of the vehicle over the time period and assigning the vehicle to the request according to the time of occurrence. This method can only be done when the time of the requests for vehicles is pre-determined.

4.2 ROUTING

Vehicle routing also plays an important role in vehicle management. Given the location of an AGV and its predetermined destination, the sequence of nodes that specify the path of the vehicle must be determined by an operator. Alternative routes are evaluated based on the cost of traversing the sequence of arcs. Routing can either be static or dynamic (Hodgson, King, and Monteith, 1987). When routing is static, the path defined for an AGV between any two given nodes is fixed. The primary objective is to use the shortest path all the time. When the routing is dynamic, the AGV is permitted to use different paths.

Vehicle routing problems can be categorized as follows (Bodin and Golden, 1981):

1. the *traveling salesman* problem deals with the determination of a minimal cost cycle that passes through each node in the relevant graph exactly once;
2. the *Chinese postman* problem deals with the determination of the minimal cost cycle that passes through every arc of the graph at least one time;
3. the *M-traveling salesman* problem is a generalization of the traveling salesman problem when there is a need to account for more than one salesman;
4. the *single depot, multiple vehicle, node routing* problem asks for a set of delivery routes for vehicles housed at a central depot that serves all the nodes and minimizes total distance traveled;
5. the *multiple depots, multiple vehicles, node routing* problem is a generalization of the previous problem, in that the fleet of vehicles must now serve multiple depots instead of one;

6. the *single depot, multiple vehicle, node routing with stochastic demands problem* is identical to the classical vehicle routing problem, except that the demands are not known with certainty, but instead are derived from a specified probability distribution; and
7. the *capacitated arc routing* problem is one in which there is an undirected network with arc demands greater than 0 for each arc, which must be satisfied by one of a fleet of vehicles, each of capacity W.

Blair, Charnsethikul, and Vasques (1987) present a two-phase heuristic method for organizing a set of static material requests into tours with the objective function of minimizing the maximum tour length. In the first phase, the AGV routing problem is formulated as a multiple traveling salesman problem (MTSP), which is solved by a branch-and-bound algorithm. In the second phase, a tour improvement process is conducted by switching the nodes of the tours derived from Phase I to find a better tour for the vehicle. Because the proposed heuristic is based on the traveling salesman problem which is NP-complete, the performance of the heuristic degrades as the problem size increases.

4.3 SCHEDULING

The purpose of vehicle scheduling is to project traffic flow and congestion (Taghaboni and Tanchoco, 1988). Initially, the AGV is assumed to take the shortest path. A schedule of arrival and departure times at each node in the route is recorded for all current and future trips. No two vehicles should occupy the same node at the same time. When a conflict arises, all alternative routes that do not include the conflict segment are evaluated. If no alternative route is feasible, then the new vehicle is scheduled to slow down. Although the routing is dynamic, neither the route nor the schedule can be changed once a vehicle commences the trip.

4.4 RELATIONSHIPS BETWEEN DISPATCHING AND ROUTING

One major concern encountered in vehicle management is that many researchers tend to treat dispatching and routing synonymously, and they fail to provide operational definitions for dispatching and routing (Wilhelm and Evans, 1987; Zeng et al., 1991). The lack of differentiation between these two terms has created unnecessary confusion. Intuitively, one of the primary distinctions between dispatching and routing is that dispatching operates in a real-time mode, while routing operates in a static mode. In dispatching, the next location that the vehicle visits is determined by a set of dispatching strategies (e.g., maximum output queue, nearest, farthest, or random workstations). Dispatching problems become more significant when flexible but complicated vehicle guide paths are used (e.g., guide paths with bi-directional arcs connecting all pairs of workstations).

On the contrary, the routing problem generally has a unique route selected for any given vehicle based on the objective of minimizing the total cost objective (which can be the empty vehicle travel time or travel length) (Psaraftis, 1980). The next location of the vehicle is, therefore, predefined and known in advance, even though the system is operating in a dynamic environment. Furthermore, most variables (e.g., station demands and the sequence of the stations that vehicles have to visit) are also known and taken into consideration when defining the route path. In fact, it is plausible to regard routing as an element of dispatching, because the paths between each station are also predefined in dispatching.

4.5 CHARACTERIZATION OF AGV DISPATCHING DECISIONS

A dispatching strategy is a set of heuristic rules employed to select either the vehicles for a particular operation or the stations to be served by the available vehicle (i.e., to prioritize work stations requesting the services of a vehicle for material pickup). These rules determine the next station to which each vehicle should proceed. The performance of these dispatching rules depends mainly on the guide path layout, fleet size, and the transport patterns in the network. The selection of dispatching rules will have an important impact on issues such as material flow, buffer storage requirements in the workcenters, central facility buffer requirements, shop throughput, machine utilization, identification of poor layout designs, vehicle effectiveness, traffic control problems, load transfer mechanisms at load pickup and delivery points, and other operational issues (Egbelu and Tanchoco, 1984). The AGVs constantly receive load requests; therefore, the system must have the ability to instantaneously change the dispatching strategies.

Before making any dispatching decision, all alternatives, as well as the impacts of each alternative on the overall system performance, must be known. The choice of dispatching rules affects the proportion of time the system operates under these different rules and the situation appropriate for their use. The information required for this decision is the number of vehicles available in the system and the station(s) with requests at that time. The primary objective of the dispatching problem is to find the dispatching rules that will maximize the system's throughput (i.e., increase system performance). Simulation studies are performed to test different combinations of these dispatching rules.

So far, there are only two formal definitions and categories of vehicle dispatching decisions: workcenter- and vehicle-initiated (Egbelu and Tanchoco, 1984). These are detailed below.

4.5.1 The Workcenter-Initiated Task Assignment Problem

The workcenter-initiated dispatching problem exists when there is one workcenter with a request and there is more than one vehicle available to serve this request (Egbelu and Tanchoco, 1982). Hence, this decision involves selecting a vehicle from a set of idle vehicles and assigning it to a load request. The workcenter-initiated heuristic rules employed to assign priorities to vehicles for dispatching include the following:

- nearest vehicle (NV) rule: this rule dispatches the vehicle nearest to the workcenter with the pending request. This rule dispatches idle vehicle *j* such that $d_j = \max_{\forall i} \{d_i\}$ for all idle vehicles *i* where d_i is the distance of vehicle *i* from the workcenter. The vehicle can also be dispatched based on the shortest travel time. The decision is to dispatch vehicle *j* such that $d_j/s_j = \max_{\forall i} \{d_i/s_i\}$ for all idle vehicles *i* where s_i is the traveling speed of vehicle *i*;

- farthest vehicle (FV) rule: this rule dispatches the vehicle that is farthest from the workcenter with the pending request. This rule is not very useful, because of the longer empty vehicle travel time involved, and it dispatches idle vehicle j such that $d_j/s_j = \max_{\forall i} \{d_i/s_i\}$ for all idle vehicles i;

- random vehicle (RV) rule: this rule randomly assigns a vehicle to the workcenter with the pending request;

- longest idle vehicle (LIV) rule: this rule assigns the highest dispatching priority to the vehicle that has remained idle the longest among all the idle vehicles. This rule provides a workload balancing effect on all participating vehicles. This is equivalent to dispatching vehicle j such that:

  $t_j = \max_{\forall i} \{t_i\}$ where $t_i = T_c - T_i$,

  T_c = current time or the time that the vehicle dispatching decision is to be made,

  T_i = the time that vehicle i was last set idle, when

  $T_c >= T_i$ for all i; and

- least utilized vehicle (LUV) rule: this rule applies in systems in which time persistent statistics on vehicle utilization are recorded. Similar to the LIV rule, this rule also provides a workload balancing effect on the vehicles. This rule states that vehicle j is dispatched such that $U_j = \min_{\forall i} \{U_i\}$ for all idle vehicles i, where U_i denotes its mean utilization up to the time the vehicle dispatching decision is to be made.

Only the first three rules are commonly used in vehicle management.

4.5.2 The Vehicle-Initiated Task Assignment Problem

The vehicle-initiated task assignment problem occurs in situations where there is more than one workcenter with requests pending, and there is only one vehicle available to serve these requests (Egbelu and Tanchoco, 1984). Hence, heuristic rules are developed to determine the appropriate vehicle to be assigned to the load request. These heuristic rules prioritize workcenters with outstanding move requests based on parameters such as the distance of the workcenters from the free vehicle, the length of the outgoing load queue (i.e., the volume of orders to be transported from the workcenters), the outgoing buffer capacity, or the elapsed time since a move request was transmitted (Hodgson, King, and Monteith, 1987). Some of the vehicle-initiated task assignment heuristic rules are listed as follows:

- shortest travel time or distance (STT/D) rule: the vehicle is dispatched to the closest workcenter with a pending transport order. This rule is the most commonly used and is also known as the nearest-workcenter-first (NWF) rule (Co and Tanchoco, 1991), best-follow-up-order (BFUO) rule (Beisteiner and Moldaschl, 1983), and view-look-for-work (VLFW) rule in some research (Hodgson, King, and Monteith, 1987; Newton, 1985). The primary objective of this rule is to minimize empty travel time, and it is very appropriate for layouts where stations are arranged relatively close to each other in a single loop, and all transport orders are known in advance;

- first-encountered-first-served (FEFS) rule: this is a very simple rule that does not require centralized control. An unassigned vehicle picks up the first load request that it encounters in the loop (Co and Tanchoco, 1991);

- longest travel time or distance (LTT/D) rule: the vehicle is sent to the workcenter with a pending transport order that is farthest from the vehicle. This rule is not very useful and is seldom used; and

- random workstation (RW) rule: the vehicle is dispatched to a workcenter that is randomly selected from the list of workcenters requesting the service of this vehicle.

The dispatching rules used in a normal system are usually a combination of both the workcenter- and vehicle-initiated rules (Malmborg, 1990), but at any one time there can only be one dispatching rule in use. The selection of a dispatching discipline in a given system is a function of at least two factors: the volume of empty vehicle travel imposed and the service time distribution resulting from a dispatching rule combination. For example, the STT/NV dispatching rule combination will almost minimize the volume of empty vehicle travel in a system but can also cause unacceptable disparity among workcenters in the service level provided. In such cases, centrally located workcenters will receive a much higher service level than remotely located workcenters. The vehicle-initiated dispatching rule is significant in AGVS with high utilization rates, while the workcenter-initiated rule is significant in AGVS with low utilization rates (Egbelu and Tanchoco, 1982 and 1984). Two findings, shop locking and operation with finite buffer size, indicate that when the material flow volume is large, the vehicle-initiated dispatching rule will be invoked all the time. On the other hand, if the buffer size is infinite, then the performance depends on the workcenter-initiated dispatching rules only (Egbelu and Tanchoco, 1984).

Hodgson, et al. (1987) propose another vehicle dispatching control rule based on Markov decision processes in which the interarrival times of demands are exponentially distributed and the travel time between stations is constant. The model involves both uni- and multi-load (i.e., the vehicle can carry many loads at the same time) vehicles. This model also uses three parameters to determine the state of the AGVS at any time: (1) the present location of the vehicle; (2) the load(s) that the vehicle is carrying; and (3) the demands waiting in queues at each pickup/delivery station. Even in a simple AGVS, the number of states will grow exponentially with this modeling approach. In order to make this problem computationally tractable, the example used in the study is limited to a maximum of four pickup/delivery stations. The general dispatching control rules developed by the Markovian approach produce near-optimal performance. The dispatching rules proposed by Hodgson, et al. (1987) can be categorized as a vehicle-initiated dispatching rule, as proposed by Egbelu and Tanchoco (1984).

Srinivasan, Bozer, and Cho (1994) develop a general-purpose analytical model that can rapidly estimate the system throughput capacity of trip-based material handling systems in a manufacturing setting based on a new, simple, and efficient empty vehicle dispatching rule. The model can be used early in the design phase to fine tune the set of alternative handling systems and configurations prior to using the simulation technique. Basically, the proposed model tries to find the vehicle utilization rate using the *MOD FCFS* rule when there is empty vehicle dispatching involved. The *MOD FCFS* is stated as follows: "Upon delivering a load at a station, an empty device first inspects the output buffer of that station. If one or more unassigned move requests are found, the device is assigned to one of them." Otherwise, the device serves the oldest unassigned move request in the system, regardless of the location. The proposed model is one of the very few analytical models which explicitly incorporates an empty vehicle dispatching rule into the material handling systems. The statistical test results of the proposed model are as good as those obtained with the Shortest-Travel-Time-First (STTF) rule. However, there are two setbacks with this research. First, to simplify the model and to limit the scope of the research, it is understandable that the authors did not address the issue of material handling needs within a station. However, parts transportation within a cell is very important in material handling systems, and this issue cannot be ignored. Next, for ease of exposition, the authors assume that the material flow in the handling system is conserved at the processor stations. In other words, the authors assume that the manufacturing system in question is perfect, with no flaws in the production process. In practice, it is not possible to have a perfect material handling system. Although the authors claim that it is very straightforward to extend the model to handle situations where flow is not conserved at each processor station, they did not elaborate on the procedures.

Malmborg (1990) presents an analytical model to predict some system performance measures, including the maximum throughput capacity and risk factors associated with shop locking for a zone-controlled AGVS. The analytical approach is based on a Markov process. In order to find the desired service time for the server (AGVs), Malmborg (1990) argues that the actual amount of empty travel time resulting from a dispatching rule will be given by a convex combination of the volume of empty travel that corresponds to the minimum possible volume of vehicle recirculation, and the volume of empty travel that corresponds to the maximum possible volume (i.e., the vehicles are redirected to the most distant workstations). To find the proportion of time at which the vehicle-initiated and workcenter-initiated component of the rules are invoked, the author proposes an approximation procedure based on a **M/M/N** queueing model to find the parameters for the corresponding vehicle-initiated and workcenter-initiated rules, where N is the number of vehicles in the system.

The approximation procedure for estimating the proportion of time that the vehicle-initiated and the workcenter-initiated rules invoked in an AGVS proposed by Malmborg (1990) suffers some drawbacks. First, the author does not consider the impact of having multiple vehicles on the vehicle dispatching. Second, in order to find the service time for the **M/M/N** queue, the author estimates the service time for the vehicle, and based on this service time, he then estimates the ratios during which the system is in vehicle-initiated and workcenter-initiated conditions. The problem with this approach is that, while service time determines the ratio, this ratio also determines the service time.

In a subsequent paper, Malmborg (1991) proposes analytical procedures for estimating the expected empty vehicle travel time for several different dispatching rules, namely, random vehicle, shortest travel time, longest travel time, nearest vehicle, and farthest vehicle dispatching rules. These procedures find the expected empty vehicle travel for several dispatching rules, based on the assumptions that there is no blocking or congestion time involved and that the number of vehicles in the system does not influence the amount of empty travel time for different dispatching rules. The formulations are used in developing upper and lower bounds on the volume of empty vehicle travel associated with certain types of dispatching rules.

Malmborg and Shen (1991) present a simulation study based on the PC Model language to validate the impact of multiple vehicles on the AGVS. The study is constructed for different routing and dispatching rule combinations, including NV/STT, FV/LTT, and RV/RW. Their results show that the volume of empty vehicle travel time is substantially dependent on the system fleet size, and their analysis indicates that the analytical models proposed by Malmborg (1991) tend to underestimate empty travel for the NV/STT. This is because smaller fleet sizes could only afford fewer opportunities to fully realize the travel minimizing potential of the vehicle-initiated dispatching rule.

4.5.3 Critique of the Two Categories of Dispatching Rules

Although Egbelu and Tanchoco (1984) develop the two major categories of dispatching rules, they fail to present the applicability of some dispatching rules of these two categories. For instance, if the routes of the vehicle or parts are not defined in advance, then it would be difficult to determine the distance between the vehicle and the requesting workstation. Second, they also fail to specify the configurations of the guide path layout (i.e., unidirectional or bi-directional). If the guide path layout is uni-directional, then the significance of the dispatching rules becomes trivial, because the vehicle in a unidirectional configuration has few alternative paths. Finally, the simulation results might be questionable, because, in the simulation study, the guide path layout and the number of vehicles are regarded as fixed. Although the material flows between each workcenter are fixed and known in advance, the guide path layout and the number of vehicles are also important variables that must be considered in the system. Consequently, the simulation models should be treated as a case study rather than as an indication of the significance of the dispatching rule combinations.

Unfortunately, relatively few researchers have attempted to employ these two types of dispatching rules to capture the dynamic behaviors of the AGVS. For instance, Psaraftis (1980) suggests that a vehicle with a predefined route operating in a stochastic environment will take into consideration new requests received enroute, and it will redefine its route based on the new information received. However, his model redefines the route based solely on the new requests, and the vehicle will travel along this new path until further new requests are received. The proposed model did not consider the dispatching strategies as specified by Egbelu and Tanchoco (1984). Maxwell and Muckstadt (1982) propose an analytical method to find the minimum number of vehicles needed to satisfy a specific demand requirement which will also minimize the total empty vehicle travel time. Similarly, both authors fail to define clearly the dispatching rules used in their models.

5. RESEARCH DESIGN AND METHODOLOGY

5.1 IMPORTANCE OF RESEARCH

Based on the comprehensive review of the AGVS conducted so far, there is a need to investigate the importance of dispatching rules and the impact of multiple vehicles in the AGVS. Subsequently, my research attempts to examine the effect of the different dispatching rules and the number of vehicles on the steady-state system performance (e.g., the expected waiting time of the load request). However, due to the unpredictability of the stochastic behavior of the AGVSs, the proportion of time that the system will be in the vehicle- or workcenter-initiated situations is unknown. In order to minimize the waiting time for the load requests, while at the same time maintaining a minimum number of vehicles in the system, this research will utilize only two of the workcenter-initiated dispatching rules, namely, the nearest-vehicle and farthest-vehicle rules. This is because the empty vehicle travel incurred by the other workcenter-initiated dispatching rules will be a convex combination of both the shortest-travel time rule and the longest travel time rule (Malmborg, 1991).

The objective of this research is to develop an analytical model using queueing theory and Markov chains, that simultaneously considers design (number of vehicles) and control (vehicle dispatching) issues. Specifically, this research will examine the relationships among empty-vehicle travel time, the number of vehicles, and the various types of workcenter-initiated dispatching rules. Given the material flow matrix, the travel time between stations, and the preferred dispatching rule, this research presents an algorithm that can be used by system designers in the early design phase to determine the number of vehicles required for an AGVS.

With the number of vehicles found by this research, the AGVS can operate with little probability of a waiting load request. In addition, the expected waiting time for a load request can be determined.

Because of the intricacy of the AGVS problem, most researchers focus their studies on the deterministic, rather than the stochastic, aspects of the AGVS. For example, Egbelu (1987) determines the number of vehicles needed in the system using four analytical models, and Kaspi and Tanchoco (1990) derive a guide path layout using integer programming. The load requests to be transported among the stations are known and given at the beginning of the shift, and there is no vehicle dispatching rule involved in their models. It is evident that the complexity of the AGVS problem increases dramatically when stochastic behavior is considered. For example, Hodgson, King, and Monteith (1987) propose a vehicle dispatching control rule based on Markov decision processes. The model uses three parameters to determine the state of the AGVS at any given time: (1) the present location of the vehicle; (2) the load(s) that the vehicle is carrying; and (3) the demands waiting in queues at each pickup/delivery station. Even in a simple AGVS, the number of states for this modeling approach grows exponentially as the number of stations and vehicles increase.

Due to the complex interdependencies among the decision variables (number of vehicles, guide path layout, dispatching rules, etc.), there is a shortage of literature dealing with stochastic models for designing and controlling an AGVS's. Although AGVS have been around for more than thirty years, simulation is the only methodology used to address the dynamic aspects of system operation. Unfortunately, this method can be very time consuming and expensive, particularly if the problem has many decision variables or variables with large ranges. Analytical techniques, such as queueing theory and Markov processes, are seldom used to analyze AGVS models due to the complexity of such problems. Nevertheless, a major benefit of such analytical models is that results can be used in the initial design stage to provide AGV designers with insights into the behavior of the AGVS.

The importance of empty vehicle travel time has not been adequately considered in the past, yet it constitutes a large proportion of vehicle travel time (Malmborg, 1990). Empty vehicle travel time influences blocking/congestion, the number of vehicles needed, and the required storage space at pickup/delivery stations. It is also directly affected by the dispatching rules used in the system. The amount of empty vehicle travel time has been assumed to be the same as the loaded vehicle travel time by both Egbelu (1987a) and Karps and Tanchoco (1990). But, this assumption inevitably undermines the accuracy of these models.

Further, none of the AGVS researchers used analytical methods to explore the interconnections between various vehicle dispatching rules and other decision variables. For example, the empty vehicle dispatching rule developed by Srinivasan, Bozer, and Cho (1994) considers only one vehicle, which is unrealistic in real-life situations. Also, Egbelu's (1987a) analytical models grossly underestimate the number of vehicles when they do not consider the dispatching rules in use. Hence, in order to have a better understanding of vehicle dispatching and empty vehicle travel time, the relationships among vehicle dispatching and other decision variables must be investigated. The analytical models presented later in this chapter specifically take stochastic behavior into account and explicitly incorporate an empty vehicle dispatching rule into the material handling systems.

Queueing theory and Markov processes have been used in the past to model and evaluate the performance of complex systems in a broad spectrum of fields such as telecommunications, inventory control, and air traffic control. Hodgson et al. (1987) and Johnson and Brandeau

(1993) have used stochastic processes to design an AGVS. In the study by Hodgson et al (1987), the states of the AGVS are represented by three parameters: (1) the present location of the vehicle; (2) the load(s) that the vehicle is carrying; and (3) the demands waiting in queues at each pickup/delivery station. Based on state transitions, the authors develop a heuristic dispatching rule for the vehicle. Unfortunately, the number of states increases exponentially according to the increase in the number of stations and vehicles. Johnson and Brandeau (1993) use queueing theory to help system designers determine which stations in the existing facility should be automated. A uni-directional guide path is used, and vehicles return to the depot once they finish delivering the loads. This assumption makes it easier to model the service time of the vehicle, which is equal to the sum of the loaded travel time, loading and unloading time, and the empty travel time. Because the vehicle returns directly to the depot, empty-vehicle travel time for a specific station is the travel time from the depot to that station. Hence, the average empty travel time is equal to the mean of the travel time from all the stations to the depot. Although it is important in AGVS research, none of these researchers have incorporated the concept of vehicle dispatching into their models.

Little work has been done for an AGVS with multiple vehicles. For instance, the dispatching rules extracted from the study by Hodgson et al (1987) consider the location of only one vehicle in the system. Srinivasan, Bozer, and Cho (1994) develop the **MOD FCFS** dispatching rule to find the vehicle utilization rate, which also assumes only one vehicle in the system. The authors state that one possible way to extend the single-device model to a multiple-device system is to use a "single faster device." That is, in order to model a system with ***K*** devices, they assume that there is a single device which travels ***K*** times faster. Unfortunately, this extension is oversimplified. When there are more devices in the material handling system, the interactions between devices become more significant as blocking and congestion occur, and the chance of invoking the workcenter-initiated dispatching rule increases. The dispatching rules for the idle devices also play an important role in the throughput of the system. Although some researchers (e.g., Egbelu (1987a) and Johnson and Brandeau (1993)) have developed several analytical models for estimating the number of vehicles needed in the system, none of them have specifically included the concept of empty-vehicle travel time in their models. The number of vehicles required in an AGVS can be estimated more accurately using the analytical model presented in this chapter, because the relationship between the empty vehicle travel time and the number of vehicles is estimated more accurately.

Finally, in most applications, the effects of dispatching are approximated using rules of thumb to characterize the volume of empty-vehicle travel time that will be expended in a system, assuming that the empty-vehicle travel volume will equal the loaded-vehicle travel volume (Egbelu, 1987). However, dispatching rule modifications have a major influence on the operating dynamics of a materials handling system. For example, they can be used to temporarily increase the handling capacity of a system by imposing vehicle conserving transaction sequencing to reduce empty vehicle travel (Malmborg, 1991). This is a short-term tool because resource-conserving rules tend to produce variations in service levels and vehicle utilization that are unacceptable for steady-state operation. Yet, it does allow systems to handle temporary peaks in workload. This can produce significant savings by avoiding the need to design the system for the worst-case operating scenario. Vehicle dispatching research seems to emphasize consideration of both vehicle- and workcenter-initiated dispatching rules in an operating AGVS. When a system is in the vehicle-initiated situation all the time, this means that the number of vehicles is not sufficient for the load requests, and the materials have to wait to be removed. On the other hand, when the system is always in the workcenter-initiated situation, then there are too many vehicles in the system. Although the dispatching rules used in a typical system are usually a combination of both the workcenter- and vehicle-initiated rules (Malmborg, 1990), an ideal environment is to have as few vehicles as possible while at the same time keeping the proportion of time in which the system uses the vehicle-initiated rules as small as possible. This can be accomplished by choosing an appropriate workcenter-initiated dispatching rule.

In short, the number of vehicles needed in an AGVS is usually found using either deterministic environments or simulation studies. The deterministic studies ignore the stochastic behavior of the AGVS, thus underestimating the number of vehicles. On the other hand, simulation studies are time consuming and expensive. Hence there is a need to develop an analytical model that considers the stochastic aspects of AGVS in order to estimate the number of required vehicles.

5.2 PROBLEM SETTING

As indicated in the previous section, none of the previous researchers have attempted to investigate the relationship among the number of vehicles, vehicle dispatching rules, and the empty vehicle travel time, simultaneously. This research will find the number of vehicles required in the system, while considering the effect of vehicle dispatching rules and empty vehicle travel time. However, due to the interdependencies among the various decision variables (e.g., the number of vehicles and the dispatching rules), only workcenter-initiated behavior is studied. This research finds the number of vehicles needed in the system, such that the probability of invoking the vehicle-initiated dispatching rule is less than a value α, which is specified by the system designer. Here, α is the probability that the system has more load requests than the number of vehicles at any arbitrary point in time. When there are more load requests than vehicles in the system, then some of the load requests will have to wait. Hence, the smaller the value of α, the lower the chance that a load request has to wait and a vehicle-initiated condition will exist. Once the number of vehicles has been determined, the expected waiting time for a load request is found.

5.2.1 Assumptions Underlying the Research

The following assumptions are made.

1. The guide paths are given and the location of each pickup/drop-off (P/D) station is known.

Because the benefits of Single-Loop and Tandem configuration layouts are not well-documented in the literature, a conventional guide path layout will be used in this research. A typical guide path layout for an AGVS with four workcenters is shown in Figure 5.1. Stations 1, 2, 3, and 4 are flexible manufacturing cells. This research estimates the number of vehicles for a given layout. Hence, guide path design is beyond its scope. Once the topology of the layout is determined, the location of pickup/drop-off must also be defined to decide the travel time for the vehicles among the stations.

2. Vehicle congestion/blocking in the guide paths is insignificant.

Usually, there are spurs to accommodate two or more vehicles at the same section of the guide path in the AGVS. Therefore, the congestion/blocking time are insignificant. Also, this research focuses on low vehicle utilization (i.e., only the workcenter-initiated rule is considered); for this case, the congestion or blocking time should be very small.

3. The travel times between the segments are fixed and constant.

The speed of the vehicles is constant throughout the operation, and there is no time lost in accelerating or slowing the vehicles. Based on the previous assumptions, the distances among stations are known and fixed, because the speed of the transporters is given. Therefore, it is reasonable to assume that the travel time between the segments of the guide path are fixed and constant.

4. The vehicles are unit-loaded (i.e., each vehicle can carry only one load/tote at a time).

The carrying capacity of the transporters is also a system decision variable. Vehicle dispatching is extremely difficult when a vehicle contains loads with different destinations. Typically, the AGVs in the market are unit-loaded.

5. After dropping off a load, the vehicle remains at that station if there are no load requests waiting in the system.

The vehicle can either go to the central depot, or stay at the station where it drops off a load, when there are no load requests waiting in the system. However, if the vehicle goes back to the central depot, the average empty vehicle travel time will increase. Except for the purpose of recharging, it is desirable to keep the vehicle at the station where it drops off the load in order to wait for the next call to pickup a request.

6. The routes between stations used by the vehicles are determined before the system begins operation.

To determine the sequence of stations the system searches for a free vehicle when a load request is pending in the AGVS, the routes used by the vehicles to reach the stations should be known and determined in order to decide the travel time between stations. Without the knowledge of the travel times for the vehicles among stations, the vehicle dispatching rules can not be utilized.

7. Material requests are assumed to arrive according to a Poisson process.

In contrast to other studies, the arrival of load requests in this research is not known in advance (i.e., the arrival is random). Use of the Poisson process may not capture the actual system behavior, but it does provide a simple analytical tool that can be used to model the randomness of the actual arrival process. The number of stations to be visited by the different products is predefined. Hence, the material flows between the stations/cells can be characterized by the Material Flow Matrix in Table 5.1. There are an average of ***a*** units of materials going from Station 1 to Station 2 per hour, and an average of ***b*** units of materials going from Station 1 to Station 3, and so on.

Table 5.1 Material Flow Matrix

Source workstations	*Destination workstation* 1	2	3	4
1	0	a	b	c
2	d	0	e	f
3	g	h	0	i
4	j	k	l	0

8. There is infinite waiting space at each pickup/delivery station.

This research focuses on light system traffics; therefore, the probability of having two or more load requests at the same station is very small. It is reasonable to assume that the waiting space at each pickup/delivery station can accommodate the load requests at that station. Also analytically, without this assumption, there could be no results available for the queueing network considered in this research.

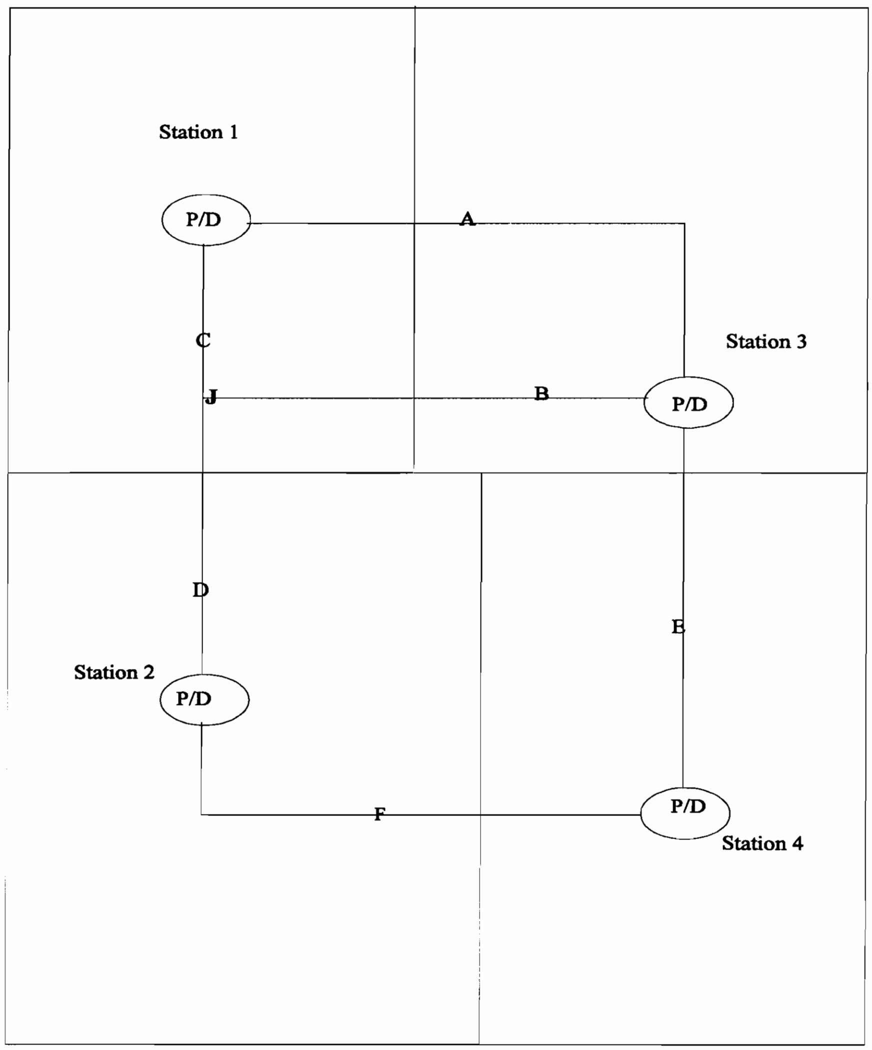

Figure 5.1 Conventional Automated Guided Vehicle System Guide Path Layout

5.3 THE DISPATCHING-RULE BASED ALGORITHM (DRBA)

This section introduces an algorithm that determines the minimum number of vehicles required in an AGVS, using dynamic control of the vehicles, such that the chance of a vehicle-initiated situation occurring is less than a given small threshold. Under these conditions, the load requests will have the least chance of waiting to be picked up. Due to the stochastic behavior of the system, and to minimize the waiting time for the load requests, while at the same time maintaining a minimum number of vehicles in the system, this research focuses on the workcenter-initiated vehicle dispatching rule only.

5.3.1 Notations Definition

The following notations are used:

W : total number of workstations to be served by the vehicles;

M_{ij} : rate of material flow between stations i and j in terms of the average number of loads per shift, for $i = 1, \ldots, W$, and $j = 1, \ldots, W$;

T_{ij} : travel time for the vehicle between stations i and j (in minutes);

PD : the time needed to perform the actual transfer of a unit load from a vehicle to a workstation or from a workstation to a vehicle (in minutes);

N : the number of vehicles in the system; and

e : the efficiency rate for vehicles (i.e., the proportion of each hour that a vehicle is actually available to service transactions based on maintenance, battery charging, etc.).

The dispatching-rule based algorithm (**DRBA**) starts by finding an upper bound for the required number of vehicles. In order to assure that the upper bound is larger than the minimal number of vehicles needed, the empty-vehicle travel time and the blocking and congestion times are estimated to be twice the loaded travel time. The number of vehicles required, derived from this estimation, is compared with the number of stations in the station. The maximum of these two numbers is selected as the upper bound on the number of vehicles required. Based on this number, the algorithm uses a queueing model to determine the probability that the system has more load requests than the number of vehicles at any arbitrary time. If this stationary probability is smaller than the specified value α, the algorithm decreases the number of the vehicles in the system by one and repeats the process. Otherwise, the algorithm proceeds using the last acceptable value for the number of vehicles to find the expected waiting time for the load request.

The second step of the algorithm updates the number of required vehicles if the projected probability of the vehicle-initiated situation occurring is smaller than the specified value α. Load requests will not wait in the system as long as their number is less than or equal to the number of vehicles. Let P' be the probability that the number of vehicles is less than or equal than the number of load requests in the system at any given time. Then, $P' = \sum_{i=0}^{N} P_i$, where P_i is the probability that there are i load requests in the system. The P_i's can use an M/M/s queueing model with s, the number of servers, equal to N, the number of vehicles in the system. If $(1-P') < \alpha$, the algorithm decreases the number of vehicles and repeats Step 2; otherwise, the minimum number of vehicles has been determined and is 1 more than the value just used. And the algorithm proceeds to Step 3.

Once the number of vehicles has been determined, the expected waiting time for the load requests can be calculated using an approximation for the **M/G/N** system, originally proposed by Lee and Longton (1957).

The algorithm is given as follows:

Step 1: ***Initialization: Determine an upper bound on the number of required vehicles.***

$$N = \mathbf{max}\,(W, \lceil \{3 * \sum_{i=1}^{W}\sum_{j=1}^{W} M_{ij} * T_{ij} + 2 * \sum_{i=1}^{W}\sum_{j=1}^{W} M_{ij} * PD\} / (60*\boldsymbol{e}) \rceil), \qquad \mathbf{(1)}$$

where $\lceil\,\rceil$ denotes rounding up to the next integer.

Step 2: ***Determine if α criteria is satisfied.***

For N vehicles, use a Markov chain to determine the first and second moments of service time for a load request. Use an *M/M/N* queueing model with this service time to compute $\boldsymbol{P_i}$, the probability that there are i load requests in the system.

Let $\boldsymbol{P}' = \sum_{i=0}^{N} \boldsymbol{P_i}$, if $(1-\boldsymbol{P}') < \alpha$, set $\boldsymbol{N} = \boldsymbol{N} - 1$ and repeat Step 2; otherwise, set $\boldsymbol{C_{min}}$ (minimum number of required vehicles) = $\boldsymbol{N} + 1$ and go to Step 3.

Step 3: ***Calculate the Expected Waiting Time.***

$E(\text{wait}(\boldsymbol{C_{min}}, \boldsymbol{D}))_{Lee}$

$= (1/2) * \left[E(S(\boldsymbol{C_{min}}, \boldsymbol{D})^2) \;/\; (E(S(\boldsymbol{C_{min}}, \boldsymbol{D}))^2\right] * E(\text{wait}(\boldsymbol{C_{min}}, \boldsymbol{D}))_{M/M/C_{min}}.$

5.3.2 Step One: Find An Upper Bound for the Number of Vehicles

This first step of the algorithm finds an upper bound for the number of vehicles needed in the system. In order to assure that the N is larger than the minimal number of vehicles needed, the empty-vehicle travel and the blocking and congestion times, due to the presence of multiple vehicles, are estimated to be twice the loaded travel time. This results in the value of 3 in Equation (1). The value of 2 in front of the term $\sum_{i=1}^{W}\sum_{j=1}^{W} M_{ij} * PD$ is due to the loading and unloading activities. This estimate should provide a starting point for the number of vehicles required, such that the system will remain in the workcenter-initiated condition most of the time.

5.3.3 Step Two: Determine If α Criteria Is Satisfied

Step two of the algorithm models the AGVS as a queueing system. In order to find the performance of this queueing system, the arrival rate, the service rate, and the waiting space in the queue have to be determined first. Because the arrival rates of the load requests to the AGVS are known and the size of the waiting space is assumed to be infinite, the only parameter needed in order to find the system performance is the average service time of the load request. The service time of the load request includes the time that the vehicle travels empty to pick up the load request, the time that the load request is loaded and unloaded from the vehicle, and the time that the vehicle transports the load request. The expected loaded travel time per load request is simply the summation of the travel time of the vehicle between each two stations, multiplied by the proportion of the total number of trips divided by the total number of load requests to the AGVS per hour. Because the loading and unloading time of the load requests is given in advance, the empty vehicle travel time is the only information that is needed to find the expected service time of each load request.

5.3.3.1 Empty Vehicle Travel Time

The purpose of the first step of the proposed algorithm is to ensure that the probability of invoking vehicle-initiated rules is very small. This results in more vehicles than are needed to satisfy system demands. Given that the number of vehicles is N, the empty vehicle travel time may not be the same as estimated in the first step of the algorithm. In fact, based on the simulation studies in the literature (Egbelu, 1988; Malmborg, 1991), the empty vehicle travel should be smaller than the one used in determining the upper bound on the required vehicles.

In order to find the expected empty vehicle travel time of a load request, the location of the vehicles and the origination and destination of a load request must be known. If a load request arrives at a station with at least one vehicle, then there will be no empty vehicle travel time. However, if a load request arrives at a station with no vehicle, then the amount of empty travel time will depend on the travel time between workcenters and the workcenter-initiated dispatching rule utilized by the system.

Because the load requests arrive at the system according to a Poisson process, the probability that a load request arrives at a given station can be found using the average arrival rates given in the material flow matrix. Also, the Poisson Arrivals See Time Average (**PASTA**) property implies that the incoming load request to the AGVS will see the stationary distribution of vehicles in the system. To determine the vehicle location distribution (note that because the probability that a load request has to wait is so small), the vehicles are usually waiting at stations when load requests arrive. By embedding Markov chains at the points of load request arrival, the stationary distribution of the number of vehicles at each station can be found. However, this distribution may not be accurate when the frequency of incoming load requests to the AGVS is relatively high. As the frequency of the load requests increases, the probability of having two simultaneous load requests in the AGVS increases accordingly. Nonetheless, the assumption that vehicles are waiting when load requests arrive can be used with the proposed algorithm because of the light traffic of the modeled system.

The accuracy of the Markov Chains model requires that, when a load request is in service (i.e., being loaded, unloaded), or transported, the probability of having another load request coming into the AGVS is very small. This assumption is consistent with the first step of the proposed algorithm. Because the number of vehicles in the system is far more than needed, this results in a light utilization system. Hence, the probability of having two load requests existing at the same time is reduced.

This model views vehicle movement among stations as depending only on the origination of the vehicle and the destination of the load requests. For example, consider an AGVS with three stations and one vehicle located at station 3. Suppose a load request wanted to go from station 1 to station 2. The free vehicle located at station 3 is dispatched to station 1 to pick up the pending request. When the vehicle arrives at station 1 to pick up the request, this temporary change in vehicle location should not be considered as a state transition because (1) the vehicle status (busy or idle) at the intermittent station will complicate the state space, and (2) the probability of two load requests existing in the system at the same time is very small. Therefore, it is assumed that the free vehicle will deliver the load request to its destination before another request arrives at the system. Given the vehicle location and the dispatching rule, it is possible to determine the empty vehicle travel time for an arriving load request. Using the law of total probability, it is possible to find the expected empty vehicle travel for a load request.

The estimation of the empty vehicle travel time based on the Markov chains model can be justified by the following explanation. Step 1 of the proposed algorithm gives an upper bound on the number of vehicles by assuming that the empty vehicle travel and the blocking times are the same as the loaded vehicle travel. This estimated number of vehicles will result in a system that has a very small chance of more than one load request in the AGVS, because the service rate will be higher due to the lesser empty vehicle travel time (i.e., the AGVS has zero or one load request in the system most of the time). Hence, it is reasonable to assume that the load request entering into the system will see all the vehicles idle and waiting. As the algorithm continues to reduce the number of vehicles needed in the system, the probability that the system has more than one load request increases. However, Step 2 of the proposed algorithm will limit the summation of the probability of having more load requests than the number of vehicles to a specified α.

5.3.3.2 Distribution of the Vehicle Location

The AGVS with N vehicles is modeled with discrete-time Markov chains, imbedded at the points in time just before a load request enters the AGVS.

Based on the material flow matrix and the assumption that the requests arrive at the AGVS according to the Poisson arrival, the probability of any load request going from station i to station j will be

$$P_{ij} = M_{ij} / \sum_{i=1}^{W} \sum_{j=1}^{W} M_{ij}, \quad \text{where } i = 1, \ldots, W, \text{ and } j = 1, \ldots, W. \qquad \textbf{(2)}$$

Over the long run, these probabilities and the dispatching rule employed in the system will dictate the equilibrium distribution of the location of the vehicles in the AGVS.

The state of the system is defined as the number of vehicles at each station. The distribution of the vehicles among the workcenters is characterized by a W-tuple element, where W is the number of workcenters in the AGVS. The state $(\boldsymbol{E_1}, \boldsymbol{E_2}, \ldots, \boldsymbol{E_i}, \ldots, \boldsymbol{E_j}, \ldots, \boldsymbol{E_W})$ indicates that there are $\boldsymbol{E_1}$ vehicles at Station 1, $\boldsymbol{E_2}$ vehicles at Station 2, etc. The summation of $\boldsymbol{E_i}$ is equal to N.

Vehicle movement among workcenters is described by the transition matrix $\boldsymbol{R}$, which is a K x K matrix, where K refers to the total number of possibilities for allocating N vehicles among W workcenters. The state space of the discrete-time Markov chains may be large, but transitions are not possible between all the states, because a transition involves only the movement of one vehicle.

Definitions:

- Two states, l and k, are adjacent if R_{lk} (the one-step transition probability from state l to state k) is greater than zero;
- For a particular state transition, a station is called a "*vehicle exporter*" if a vehicle is sent out from that station; and
- A station is called a "*vehicle importer*" if a vehicle delivers a load request to that station. For a state, the same station may be an exporter for one transition and an importer for another.

Theorem: States l and k are adjacent if and only if $(E_i(l) - E_i(k)) + (E_j(l) - E_j(k)) = 0$ for a pair of stations i and j, and the remaining corresponding elements $E_n(l) = E_n(k)$, for all $n=1, ..., W$, and $n \neq i, j$, where $E_i(l)$ is the element E_i of state l and $E_j(k)$ is the element E_j of state k.

Proof. If states l and k are adjacent, then $R_{lk} > 0$ and it is possible for the system to go from states l and k in one transition. Because each transition involves the movement of one vehicle, only one station can be an exporter (call it station i) and only one station can be an importer (call it station j). The number of vehicles at each of the remaining stations will not change, so $E_n(l) = E_n(k)$, for all $n \neq i, j$. If $i \neq j$, then $E_i(l) - E_i(k) = +1$ and $E_j(l) - E_j(k) = -1$. If $i = j$, then $E_i(l) - E_i(k) = 0$ and $E_j(l) - E_j(k) = 0$. Hence, $(E_i(l) - E_i(k)) + (E_j(l) - E_j(k)) = 0$ and $E_n(l) = E_n(k)$, for all $n \neq i, j$.

Suppose $(E_i(l) - E_i(k)) + (E_j(l) - E_j(k)) = 0$, and the remaining corresponding elements $E_n(l) = E_n(k)$, for all $n=1, ..., W$, and $n \neq i, j$. If $i \neq j$, then a vehicle has moved between station i and station j, and states l and k are adjacent. If $i = j$, then a vehicle started at station i, traveled to pickup a load request, and delivered it to station i, so states l and k are adjacent (in fact, $l = k$).

Consequently, this Theorem must be true.

For all other states that do not satisfy the Theorem, $R_{lk} = 0$. The following sets will be used to compute the transition probabilities. Let

U_l^i : be the set of station(s) of state l that do not have any vehicles and would be served by a vehicle at station i under the nearest-vehicle rule; and

V_l^i : be the set of station(s) of state l that do not have any vehicles and would be served by a vehicle at station i under the farthest-vehicle rule.

As indicated, the transition probabilities of the discrete-time Markov chains are determined by the origin and destination of the load request, the dispatching rule, and the location of the available vehicles. Given a specific dispatching rule, the transition probability of any two adjacent states depends on the location of the vehicle exporter of one state and the vehicle importer of the other. For the nearest-vehicle rule, the transition probability includes the probability of a load request going directly from the vehicle exporter station to the vehicle importer station as defined in Equation (2), and the probability of the load request arriving at a station without a vehicle that is closest to station i compared to every other station (i.e., the station is in U_l^i).

As for the farthest-vehicle rule, the transition probability includes the probability of a load request going directly from the vehicle exporter station to the vehicle importer station as defined in Equation (2), and the probability of the load request arriving at a station without a vehicle that is farthest from station i compared to every other station (i.e., the station is in U_l^i). We will now determine the one-step transition probability from state l to state k of the discrete-time Markov chains (i.e., R_{lk}) as follows:

- ***The Nearest Vehicle Rule***

Denote the vehicle exporter for state l as station i and the vehicle importer for state k as station j. The probability of the system going from state l to state k is equal to

$$R_{lk} = P_{ij} + \sum_{s \in U_l^i} P_{sj} . \tag{3}$$

When the system is in state l, a transition to state k occurs whenever a vehicle at station i delivers a load to station j. This happens whenever a load request with station j as its destination arrives at station i or at a station without a vehicle, which is served by the vehicle at station i. Note that U_l^i contains all the stations without vehicles for which station i has the shortest vehicle travel time.

For the diagonal elements of the transition probability matrix, the vehicle exporter and the vehicle importer are the same station. Note that the material flow matrix implies $P_{ii} = 0$ for all i. That is, there is no possibility that a load request arrives at station i with station i as its destination. As a result, the transition probability for the diagonal element includes only the probability that the load request arrives at a station served by the vehicle at station i with station i as its destination. Hence, the state transition probability matrix is of the following form:

R(Nearest Rule) =

State	***1***	***2***	***3***	.	.	.	***K***
1	$\boldsymbol{R_{11}}$	.	.	.	.	.	$\boldsymbol{R_{1K}}$
2	$\boldsymbol{R_{21}}$	.	.	.	.	.	.
3	$\boldsymbol{R_{31}}$	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
K	$\boldsymbol{R_{K1}}$	$\boldsymbol{R_{K2}}$	$\boldsymbol{R_{K3}}$	.	.	.	$\boldsymbol{R_{KK}}$

- ***The Farthest Vehicle Rule***

The probability of the system going from state l to state k is equal to

$$\boldsymbol{R}'_{lk} = P_{ij} + \sum_{t \in V_l^i} P_{tj} \, . \tag{4}$$

This transition from state l to state k happens whenever a load request with station j as its destination arrives at station i or at a station to be served by the vehicle at station i under the farthest vehicle rule. Note that V_l^i contains all the stations for which station i has the longest vehicle travel time.

Hence, the transition probability is of the following form:

$$R'(\textit{Farthest Rule}) = \begin{array}{ccccccc} \textit{State} & 1 & 2 & 3 & \cdot\;\cdot\;\cdot & K \\ 1 & R'_{11} & \cdot & \cdot & \cdot\;\cdot\;\cdot & R'_{1K} \\ 2 & R'_{21} & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\ 3 & R'_{31} & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot & \cdot \\ K & R'_{K1} & R'_{K2} & R'_{K3} & \cdot\;\cdot\;\cdot & R'_{KK} \end{array}$$

Note that the discrete-time Markov chains are irreducible, i.e., all the states communicate. The difference in the number of vehicles between the corresponding stations of any two states is the result of a combination of dispatching rules and the material flow matrix.

As a result, any state can reach any other state through a series of vehicle dispatches. Because some states of the Markov chains can transition back to themselves, the discrete-time Markov chains are also aperiodic. Hence, they are ergodic and there exists a unique non-negative solution of

$\boldsymbol{\Pi}^{NV} = \boldsymbol{\Pi}^{NV} * \boldsymbol{R}$ (for the Nearest Rule) or $\boldsymbol{\Pi}^{FV} = \boldsymbol{\Pi}^{FV} * \boldsymbol{R}'$ (for the Farthest Rule), **(5)**

where $\Pi = [\Pi_1, \Pi_2, \ldots, \Pi_K]$

$$\sum_{j=1}^{K} \Pi_j = 1 \qquad \textbf{(6)}$$

and Π_j is the stationary probability that the discrete-time Markov chains are in State j, $j = 1, 2, \ldots$ and K (Ross, 1993). These stationary probabilities represent the distribution of vehicle location, i.e., the number of vehicles found at the different locations by an arriving request.

5.3.3.3 Calculation of the Empty Vehicle Travel Time

Given the vehicle locations, the dispatching rule, and the location of the arriving load request, the empty vehicle travel time is known. The expected value of the empty vehicle travel time can then be computed using conditioning. The stationary distribution of the discrete-time Markov chains is used to condition on the vehicle locations for the dispatching rule. The material flow matrix is used to condition on the arrival location of the load request. The expected empty vehicle travel time (EVT) for each load request in the system is

$$\mathbf{E[EVT(NR)]} = \sum_{i=1}^{W} \{(\sum_{j=1}^{W} P_{ij}) * (\sum_{k=1}^{K} (\Pi^{NV}_k * TN^k_i)))\} \text{ [for the Nearest Rule], or}$$

$$\mathbf{E[EVT(FR)]} = \sum_{i=1}^{W} \{(\sum_{j=1}^{W} P_{ij}) * (\sum_{k=1}^{K} (\Pi^{FV}_k * TF^k_i)))\} \text{ [for the Farthest Rule]} \quad \mathbf{(7)}$$

where TN^k_i is the empty vehicle travel time involved when the system is in state k and a request arrives at station i for the nearest rule, and TF^k_i is the empty vehicle travel time involved when the system is in state k and a request arrives at station i for the farthest-vehicle rule. Π^{NV}_k and Π^{FV}_k are the stationary probabilities for state k found in the previous section for the nearest-vehicle and farthest-vehicle rules, respectively.

5.3.3.4 Distribution of the Number of Load Requests in the System

The average service time of a load request is the sum of the expected empty vehicle travel time, loaded vehicle travel time, and the loading/unloading time. The average loaded vehicle travel time for each load request is

$$\underline{T} = \sum_{i=1}^{W} \sum_{j=1}^{W} P_{ij} * T_{ij} \quad \mathbf{(8)}$$

where T_{ij} is the vehicle travel time from workcenter i to workcenter j. The empty vehicle travel time can be derived from Equation (7) for both the *Nearest Vehicle* and *Farthest Vehicle Rules*. Hence, $E[S(N,D)]$, the expected service time of the load request, given that there are N vehicles in the system and the workcenter-initiated rule employed is D, is given as follows:

(1) if **D** is the nearest vehicle (NV) rule, then

$$E[S(N,NV)], = \sum_{i=1}^{W}\sum_{j=1}^{W} P_{ij} * T_{ij} + \sum_{i=1}^{W} \{(\sum_{j=1}^{W} P_{ij}) * (\sum_{k=1}^{K} (\Pi^{NV}{}_{k} * TN^{k}{}_{i})))\} + 2*PD \qquad \textbf{(9)}$$

(2) if **D** is the farthest vehicle (FV) rule, then

$$E[S(N,FV)] = \sum_{i=1}^{W}\sum_{j=1}^{W} P_{ij} * T_{ij} + \sum_{i=1}^{W} \{(\sum_{j=1}^{W} P_{ij}) * (\sum_{k=1}^{K} (\Pi^{FV}{}_{k} * TF^{k}{}_{i})))\} + 2*PD \qquad \textbf{(10)}$$

Given an estimated system service rate (that includes both the loaded and empty vehicle travel) and N vehicles in the AGVS, the probability distribution of the number of transactions pending in the system (denoted as P_i for $i = 0, 1, \ldots, \infty$) can be approximated using the $M/M/N$ queueing model as follows:

$$P_0 = [\sum_{n=0}^{N-1}(1/n!)*(\lambda/\mu)^n + (1/n!)*(\lambda/\mu)^n * (\frac{n\mu}{n\mu-\lambda})]^{-1} \qquad \textbf{(11)}$$

$$P_n = (\lambda^n/(n!*\mu^n)) * P_0 \qquad \{1 \leq n \leq N)$$

$$= (\lambda^n/(\mu^n *N!*N^{n-N})) * P_0 \quad \{n>N\} \qquad \textbf{(12)}$$

where

$$\lambda = \sum_{i=1}^{W}\sum_{j=1}^{W} M_{ij}$$

and

$$\mu_{NV} = 1/\{\sum_{i=1}^{W}\sum_{j=1}^{W} P_{ij} * T_{ij} + \sum_{i=1}^{W} \{(\sum_{j=1}^{W} P_{ij}) * (\sum_{k=1}^{K} (\Pi^{NV}{}_{k} * TN^{k}{}_{i})))\} + 2*PD\ \}$$

for the Nearest-Vehicle Rule, and

$$\mu_{FV} = 1/\{ \sum_{i=1}^{W} \sum_{j=1}^{W} P_{ij} * T_{ij} + \sum_{i=1}^{W} \{(\sum_{j=1}^{W} P_{ij}) * (\sum_{k=1}^{K} (\Pi^{FV}_{k} * TF^{k}_{i})))\} + 2*PD \}$$

for the Farthest-Vehicle Rule.

5.3.4 Step Three: Calculate the Expected Waiting Time

In Step 3 of the proposed algorithm, the AGVS is modeled as an **M/G/N** queueing system, where **M** represents the inter-arrival time of the load requests as having a memoryless distribution, i.e., exponential distribution; **G** represents the service time of the vehicle as having a general service time distribution; and **N** is the number of vehicles (servers) in the AGVS. The service time of the vehicles (servers) is generally distributed, because the empty vehicle travel time of the vehicles between stations cannot be predicted with certainty and is dependent on the dispatching rule employed. The expected service time for the server will also depend on the dispatching rule employed, which is either the nearest or the farthest vehicle rule. This service time is the sum of the expected loaded travel and the empty vehicle travel time plus loading and unloading time. A discrete-time Markov process was used to find the expected empty vehicle travel.

5.3.4.1 The M/G/N Approximation

Due to the lack of a closed-form solution, an approximation is used for the expected delay of the **M/G/N** system. The **M/G/N** approximation used in this research was originally proposed by Lee and Longton (1957) and was later independently proposed by several other authors, such as Maaloe (1973), Stoyan (1976), and Nozaki and Ross (1978). The approximation attempts to adjust the **M/M/N** formula to account for larger or smaller service time variations. It is simple to evaluate and has been proven to be fairly accurate (Whitt, 1983). Although Lee and Longton (1957) have one of the earliest approximations in the literature, this formula is still used as a benchmark for new research in this field and has been shown to be robust for a wide range of service distributions (Whitt, 1983).

This approximation equation is given as follows:

$$\boldsymbol{E}[\text{wait}(\boldsymbol{C_{min}}, \boldsymbol{D})]_{\text{Lee}} = (1/2) * \{\boldsymbol{E}[S(C_{min}, D)^2] / \boldsymbol{E}[S(C_{min}, D)]^2\} * \boldsymbol{E}(\text{wait}(\boldsymbol{C_{min}}, \boldsymbol{D}))_{\text{M/M/}\boldsymbol{Cmin}},$$

where $\text{E}[\text{wait}(C_{min}, D)]_{\text{Lee}}$ is the approximation of the expected waiting time for an $M/G/C_{min}$ queue, given that there are C_{min} vehicles in the system and the workcenter-initiated dispatching rule employed is D, which could be either the nearest vehicle (NV) or the farthest vehicle (FV) dispatching rule. The second and first moments of the service time for a load request are $\boldsymbol{E}[S(C_{min}, D)^2]$ and $\boldsymbol{E}[S(C_{min}, D)]$, respectively, given C_{min} and D. Finally, $\text{E}[\text{wait}(C_{min}, D)]_{\text{M/M/}Cmin}$ is the expected waiting time of an $M/M/C_{min}$ queue, assuming that the service time of the servers is exponentially distributed with the mean of $E[S(C_{min}, \boldsymbol{D})]$. The second moment of the service time will now be determined.

5.3.4.2 Second Moment of the Service Time

Based on the discrete-time Markov chains model of the AGVS, the second moment of the service time can be found by conditioning on (1) the state of the system and (2) the source and destination of the load requests. The second moment of the service time, given that there are N vehicles in the AGVS and the dispatching rule employed is $\boldsymbol{D}$, is derived as follows:

$$\boldsymbol{E}[S(C_{min}, D)^2] =$$

$$\sum_{i=1}^{W}\sum_{j=1}^{W}\sum_{k=1}^{K} \boldsymbol{E}[S_{ijk}(N, \boldsymbol{D})^2 \mid \text{request goes from stations } i \text{ to } j\text{, and the system at state } k] * P_{ij} * \Pi_k$$

Under the nearest rule, given that the load request arrives at station i with destination station j, and the system is in state k, then $S_{ijk}(N, NV) = TN^k_i + 2*\boldsymbol{PD} + T_{ij}$. Hence,

$$\boldsymbol{E}[S(N, NV)^2] = \sum_{i=1}^{W}\sum_{j=1}^{W}\sum_{k=1}^{K} [(TN^k_i + 2*\boldsymbol{PD} + T_{ij})^2] * P_{ij} * \Pi^{NV}_k \quad \textbf{(13)}$$

where TN^k_i is the empty vehicle travel time involved when the system is in state k and a request arrives at station i, $\boldsymbol{PD}$ is the loading/unloading time, P_{ij} is the probability that an incoming load request goes from station i to station j, and finally, Π^{NV}_k is the probability that the discrete-time Markov chains are at state k when the AGVS uses the nearest-vehicle rule. The outer summation sums up the probability that the load request arrives at station i, the second summation considers the destination of the load request, and the inner summation enumerates all the possible states in which an arriving load request may find the system.

Under the farthest-vehicle rule, given that the load request arrives at station i with destination station j and the system is in state k, then $S_{ijk}(N, FV) = TF^k_i + 2*\boldsymbol{PD} + T_{ij}$. Hence,

$$E(S(N, FV)^2) = \sum_{i=1}^{W} \sum_{j=1}^{W} \sum_{k=1}^{K} [(\boldsymbol{TF^k_i} + 2*\boldsymbol{PD} + T_{ij})^2] * P_{ij} * \Pi^{FV}_k \quad \textbf{(14)}$$

where TF^k_i is the empty vehicle travel time involved when the system is in state k and a request arrives at station i, $\boldsymbol{PD}$ is the loading/unloading time, P_{ij} is the probability that an incoming load request goes from station i to station j, and finally, Π^{FV}_k is the probability that the system is in state k.

6. MODEL TESTING

In this chapter, the validity of the analytical model and the accuracy of the algorithm are examined using a simulation model. The AGVS depicted in Figure 5.1 will be used to demonstrate the procedure and serve as the context for simulation comparison. The layout has four workstations and a load transfer station. The intersection point of the guide path is **J**, and **A**, **B**, **C**, **D**, **E**, and **F** are segments of the layout. Segment A is the length extending from the P/D of Station 1 to the P/D of Station 3, and Segment B is the length extending from the P/D of Station 3 to the intersection point J. Similarly, Segment C extends from the intersection point J to the P/D of Station 1, and Segment D extends from the intersection point J to the P/D of Station 2. Finally, Segment E extends from the P/D of Station 3 to the P/D of Station 4, and Segment F extends from the P/D of Station 4 to the P/D of Station 2. For simplicity, the pickup and delivery operations are assumed to occur at the same location for each workstation.

The travel times from Stations 2, 3, and 4 to Station 1 are 140, 160, and 340 seconds, respectively. Table 6.1 summarizes the parameters for the sample problem. The material flow between workstations is measured in terms of the average number of load requests per hour, and the travel time for the vehicle between stations is measured in seconds. The time for the vehicle to load or unload a request is 12 seconds. The vehicle is assumed to be available for the whole shift and is charged at the end of the shift. The probability (α) of the load request having to wait for a vehicle is specified by the system designer as 0.10.

The dispatching rule will dictate the dispatching of a free vehicle to the load request. The vehicle dispatching sequence determines the sequence the system searches the stations for a free vehicle when a load request is pending in the AGVS. The dispatching sequences for each station under the different dispatching rules are also given in Table 6.1. Based on the distance between stations, the dispatching sequence for the nearest vehicle rule for station 1 is Station 2 first, then Station 3, and finally, Station 4. The system will first dispatch a free vehicle at Station 2 to pickup a load request at Station 1. If there is no free vehicle available at Station 2, then the system will look for a free vehicle at Station 3. Finally, if there is no free vehicle available at Station 3, then the system will search for a free vehicle located at Station 4. If all vehicles are busy at the time that the load request enters the system, this load request is put into the waiting queue of Station 1. Similarly, the dispatching sequence for other stations are determined according to the travel time between other stations.

For this sample problem, the number of required vehicles for the system using the four analytical models of Egbelu (1987) are:

Rule 1: $N = 2.7$;

Rule 2: $N = 1.8$;

Rule 3: $N = 1.5$; and

Rule 4: $N = 2.4$.

As indicated by Egbelu (1987), these four analytical models tend to underestimate the number of vehicles needed in the system because vehicle dispatching rules were not considered in the study. The results of following section illustrate the important fact that the dispatching rule employed in the control of the vehicles greatly affects the number of vehicles required by the system.

6.1 PROBLEM DEMONSTRATION

The algorithm of Chapter 5 is applied to the sample problem in the following two sections for both the nearest- and farthest-vehicle dispatching rules.

6.1.1 Demonstration for the Nearest-Vehicle Dispatching Rule

The algorithm is applied to the sample problem operating under the nearest-vehicle dispatching rule.

6.1.1.1 First Iteration of DRBA for Nearest-Vehicle Dispatching Rule

First, the algorithm determines an upper bound on the number of vehicles required by the system to process the arriving load requests. After this, it finds the probability distribution of the number of load requests in the system, given the number of vehicles, and checks to determine if the α criteria is satisfied.

Step 1*:** ***Find the Upper Bound on the Number of Required Vehicles

The total loaded travel time based on the sample data is 4,785 seconds, and there are an average of 23 (= $\sum_{i=1}^{w}\sum_{j=1}^{w} M_{ij}$) load requests per hour that need to be transported.

Hence, Equation (1) indicates the number of vehicles (N) required is 5 (maximum of 4 and 5).

Step 2*:** ***Determine if α criteria is satisfied with five vehicles

In order to find the distribution of the number of load requests, the empty-vehicle travel time based on the nearest-vehicle dispatching rule must first be determined. The state space of the discrete-time Markov chains and the transition probabilities (in terms of the material flow matrix) are given in Appendix A. In the appendices, P_{12} represents the probability that the load request goes from station *1* to station *2*. Table 6.2, extracted from the data in Appendix A, shows the P_{ij} (in brackets) that are added to determine the transition probabilities for the discrete-time Markov chains. For example, the probability of the discrete-time Markov chains going from state *(5,0,0,0)* to state *(4,1,0,0)* is the summation of P_{12}, P_{32}, and P_{42}.

Table 6.1 Parameters for the Sample AGVS

Material Flow Matrix
(loaded trips per hour between workstations)

Source workstations	*Destination workstation* **1**	**2**	**3**	**4**
1	0	4	2	2
2	2	0	1	2
3	2	2	0	1
4	3	1	1	0

Travel Time Matrix
(vehicle travel time, in seconds, among workstations)

Source workstations	*Destination workstation* **1**	**2**	**3**	**4**
1	0	140	160	150
2	140	0	255	240
3	160	255	0	250
4	340	240	250	0

Vehicle Dispatching Sequence

Nearest Rule:	**1**	2	3	4
	2	1	4	3
	3	1	4	2
	4	1	2	3
Farthest Rule:	**1**	4	3	2
	2	3	4	1
	3	2	4	1
	4	3	2	1

Loading/Unloading time (time to transfer material from vehicle to workstation or vice versa): 12s
Efficiency factor: 1.00
α (the probability that the system designer is willing to let the load request wait): 0.10

Table 6.2 State Space and Transition Probabilities for Four Stations and Five Vehicles Using the Nearest Rule

State	Possible Transitions (Nearest Rule)			
(5,0,0,0)	(4,1,0,0) [p12,p32,p42]	(4,0,1,0) [p13,p23,p43]	(4,0,0,1) [p14,p24,p34]	(5,0,0,0) [p21,p31,p41]
(0,5,0,0)	(1,4,0,0) [p21,p31,p41]	(0,4,1,0) [p23,p13,p43]	(0,4,0,1) [p24,p14,p34]	(0,5,0,0) [p12,p32,p42]
(0,0,5,0)	(1,0,4,0) [p31,p21,p41]	(0,1,4,0) [p32,p12,p42]	(0,0,4,1) [p34,p14,p24]	(0,0,5,0) [p13,p23,p43]
(0,0,0,5)	(1,0,0,4) [p41,p21,p31]	(0,1,0,4) [p42,p12,p32]	(0,0,1,4) [p43,p13,p23]	(0,0,0,5) [p14,p24,p34]
(4,1,0,0)	(4,0,1,0) [p23]	(4,0,0,1) [p24]	(4,1,0,0) [p31,p41]	(3,1,0,1) [p14,p34]
(4,0,1,0)	(4,1,0,0) [p32]	(4,0,1,0) [p21,p41]	(4,0,0,1) [p34]	(3,1,1,0) [p12,p42]

The stationary probability distributions for the discrete-time Markov chains are found by solving the limiting probability equations (i.e., Equations (5) and (6)). This was done using Lp-Solve, a linear programming based tool. The equations are given in Appendix B, and the stationary probability distributions of this discrete-time Markov chains (i.e., with four stations and five vehicles using the nearest rule) are given in Table 6.3. The states of the discrete-time Markov chains S_i are arranged in sequence corresponding to Appendix A.

The stationary probability distribution of the discrete-time Markov chains given in Table 6.3 suggests that the chance of having at least one vehicle in Station 2 is higher than for other stations. This is because, over the long run, there will be an average of 7 load requests per hour arriving at Station 2, and only an average of 5 load requests per hour will leave that station. While for other stations, the average number of load requests coming into a station is almost the same as the average number leaving that station. This makes Station 2 a vehicle surplus station (i.e., more vehicles entering into the station than leaving the station). These distributions also reveal that the probability of having vehicles at 3 or 4 of the four stations tends to be greater than the average probability. This implies the nearest-vehicle rule tends to underestimate the real empty vehicle travel time, because the probability of having more than one load request in the system is much greater than that of having zero or one load request in the system.

Using Equation (7), the expected empty vehicle travel per load request is 63.17 seconds. Based on Equation (9), the service rate of the vehicles (which includes both the loaded and empty vehicle travel) is 12.61 load requests per hour, and the probability distribution for the M/M/N queueing model is given in Table 6.4. Since $(1\text{-}\boldsymbol{P}') = 0.0156 < \alpha\,(=0.10)$, set $N = \boldsymbol{5}$ -1 and repeat Step 2.

Table 6.3 Stationary Probabilities of the Markov Chains for Four Stations and Five Vehicles Using the Nearest Dispatching Rule

State	S1	S2	S3	S4	S5	S6	S7
Probability	0.00178	0.01888	0.00930	0.01906	0.00476	0.00362	0.00407

State	S8	S9	S10	S11	S12	S13	S14
Probability	0.02646	0.01904	0.02612	0.01757	0.01741	0.01842	0.02142

State	S15	S16	S17	S18	S19	S20	S21
Probability	0.03745	0.02876	0.00917	0.00567	0.00744	0.01538	0.01729

State	S22	S23	S24	S25	S26	S27	S28
Probability	0.02791	0.00931	0.01643	0.02125	0.01279	0.03103	0.02473

State	S29	S30	S31	S32	S33	S34	S35
Probability	0.00806	0.00921	0.00723	0.02119	0.02476	0.02310	0.01795

State	S36	S37	S38	S39	S40	S41	S42
Probability	0.01675	0.02348	0.02751	0.02182	0.03288	0.01378	0.01602

State	S43	S44	S45	S46	S47	S48	S49
Probability	0.01192	0.01100	0.01582	0.01255	0.02134	0.01897	0.02594

State	S50	S51	S52	S53	S54	S55	S56
Probability	0.02592	0.01907	0.02776	0.01362	0.01975	0.01832	0.02181

6.1.1.2 Second Iteration of DRBA for Nearest-Vehicle Dispatching Rule

Step 2: ***Determine if α criteria is satisfied with four vehicles***

The state space of the discrete-time Markov chains and the transition probabilities (in terms of the material flow matrix) are given in Appendix C. The stationary probability distribution of these discrete-time Markov chains (i.e., with four stations and four vehicles using the nearest rule) is given in Table 6.5, and the linear probability equations are given in Appendix D.

Using Equation (7), the expected empty vehicle travel per load request is 70.91 seconds. Based on Equation (9), the service rate (which includes both the loaded and empty vehicle travel) of the vehicles is 12.22 load requests per hour, and the probability distribution for the corresponding queueing network is given in Table 6.6. Since $(1-\boldsymbol{P}') = 0.057 < \alpha\ (=0.10)$, set $\boldsymbol{N} = \boldsymbol{4} - 1$ and repeat Step 2.

6.1.1.3 Third Iteration of DRBA for Nearest-Vehicle Dispatching Rule

Step 2: ***Determine if α criteria is satisfied with three vehicles***

The state space of the discrete-time Markov chains and the transition probabilities (in terms of the material flow matrix) are given in Appendix E. The stationary probability distribution of these discrete-time Markov chains (i.e., with four stations and three vehicles using the nearest rule) are given in Appendix F, and the linear probability equations are given in Appendix G.

Using Equation (7), the expected empty vehicle travel per load request is 96.85 seconds. Based on Equation (9), the service rate (which includes both the loaded and empty vehicle travel) of the vehicles is 11.26 load requests per hour, and the probability distribution for the corresponding queueing network is given in Appendix H. Since (1-***P***') = 0.3217 > α (=0.10), set C_{min} = 3 + 1 and go to Step 3.

Step 3: ***Calculate the Expected Waiting Time Using Lee and Longton's (1957) approximation procedure***

The expected waiting time for the M/M/N queueing system is 0.361 minutes, and the coefficient for Lee and Longton's algorithm is 0.801 (using Equations (9) and (13)). Hence, the expected waiting time for the load request is 0.289 minutes.

Table 6.4 Probability Distribution of the Load Requests for Four Stations and Five Vehicles Using the Nearest Dispatching Rule

n (# of Load Requests)	0	1	2	3	4	5
P(n)	0.1604	0.2928	0.2673	0.1626	0.0742	0.00270
n (# of Load Requests)	**6**	**7**	**8**	**9**	**10**	**11**
P(n)	0.0098	0.0036	0.0013	0.0004	0.0001	0.0000

Table 6.5 Stationary Probabilities of the Markov Chains for Four Stations and Four Vehicles Using the Nearest Dispatching Rule

State	S1	S2	S3	S4	S5	S6	S7
Probability	0.004406	0.028493	0.015527	0.027077	0.011526	0.008933	0.009857

State	S8	S9	S10	S11	S12	S13	S14
Probability	0.039643	0.029153	0.040024	0.029246	0.029153	0.030708	0.030366

State	S15	S16	S17	S18	S19	S20	S21
Probability	0.053220	0.040894	0.022043	0.015219	0.018011	0.027563	0.044234

State	S22	S23	S24	S25	S26	S27	S28
Probability	0.035553	0.019332	0.022054	0.017561	0.032029	0.037343	0.036566

State	S29	S30	S31	S32	S33	S34	S35
Probability	0.029847	0.027709	0.039329	0.039047	0.031028	0.046788	0.030518

Table 6.6 Probability Distribution of the Load Requests for Four Stations and Four Vehicles Using the Nearest Dispatching Rule

n **(# of load requests)**	**0**	**1**	**2**	**3**	**4**	**5**	**6**
P(n)	0.1476	0.2782	0.2623	0.1648	0.0776	0.0366	0.0172
n **(# of load requests)**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
P(n)	0.0081	0.0038	0.0018	0.0009	0.0004	0.0002	0.0001

6.1.2 Demonstration for the Farthest-Vehicle Dispatching Rule

The farthest-vehicle dispatching rule is demonstrated in this section using the sample problem designed for four stations. The demonstration commences with finding an upper bound on the number of vehicles required by the system.

6.1.2.1 First Iteration of DRBA for Farthest-Vehicle Dispatching Rule

Step 1: ***Find the Upper Bound on the Number of Required Vehicles***

The total loaded travel time based on the sample data is 4,785 seconds, and there are 23 load requests needed to be transported. Hence, the number of vehicles (*N*) needed is 5 (maximum of 4 and 5 (Equation (1)).

Step 2: ***Determine if α criteria is satisfied with five vehicles***

The state space of the discrete-time Markov chains and the transition probabilities (in terms of the material flow matrix) are given in Appendix I. The stationary probability distribution of these discrete-time Markov chains (i.e., with four stations and five vehicles using the farthest rule) is given in Table 6.6, and the limiting probability equations are given in Appendix J.

Using Equation (7), the expected empty-vehicle travel time per load request is 91.38 seconds. Based on Equation (10), the service rate (which includes both the loaded and empty vehicle travel) is 11.51 load requests per hour, and the probability distribution of the queueing system is given in Table 6.8. Because (1-***P'***) = 0.0239 < α (=0.10), set ***N*** = ***5*** -1 and repeat Step 2.

Table 6.7 Stationary Probabilities of the Markov Chains for Four Stations and Five Vehicles Using the Farthest Dispatching Rule

State	S1	S2	S3	S4	S5	S6	S7
Probability	0.05702	0.03175	0.000491	0.00281	0.054075	0.028348	0.047909

State	S8	S9	S10	S11	S12	S13	S14
Probability	0.03578	0.03041	0.030386	0.00120	0.002415	0.001499	0.008571

State	S15	S16	S17	S18	S19	S20	S21
Probability	0.00733	0.00406	0.045538	0.00782	0.027108	0.037140	0.015420

State	S22	S23	S24	S25	S26	S27	S28
Probability	0.01848	0.00293	0.007104	0.00267	0.016411	0.012172	0.003738

State	S29	S30	S31	S32	S33	S34	S35
Probability	0.02498	0.03935	0.019514	0.02407	0.029375	0.022826	0.004720

State	S36	S37	S38	S39	S40	S41	S42
Probability	0.00334	0.00556	0.015713	0.00843	0.009290	0.023253	0.031921

State	S43	S44	S45	S46	S47	S48	S49
Probability	0.00985	0.00761	0.025829	0.01330	0.013053	0.011768	0.021929

State	S50	S51	S52	S53	S54	S55	S56
Probability	0.01511	0.00605	0.008184	0.01983	0.019697	0.009927	0.014788

Table 6.8 Probability Distribution of the Load Requests for Four Stations and Five Vehicles Using the Farthest Dispatching Rule

n (# of Load Requests)	0	1	2	3	4	5
P(n)	0.1343	0.2687	0.2686	0.1791	0.0896	0.0358
n (# of Load Requests)	**6**	**7**	**8**	**9**	**10**	**11**
P(n)	0.0143	0.0057	0.0023	0.0010	0.0004	0.0002

6.1.2.2 Second Iteration of DRBA for Farthest-Vehicle Dispatching Rule

Step 2: ***Determine if α criteria is satisfied with four vehicles***

The state space of the discrete-time Markov chains and the transition probabilities (in terms of the material flow matrix) are given in Appendix K. The stationary probability distribution of this discrete-time Markov chains (i.e., with four stations and four vehicles using the farthest-vehicle rule) is given in Table 6.9, and the limiting probability equations are given in Appendix L.

Using Equation (7), the expected empty vehicle travel per load request is 99.02 seconds. Based on Equation (10), the service rate is 11.21 load requests per hour and the probability distribution for the queueing network is given in Table 6.10. Because $(1\text{-}\boldsymbol{P}') = 0.0962 < \alpha$ (=0.10), set $N = \boldsymbol{4} - 1$ and repeat Step 2.

6.1.2.3 Third Iteration of DRBA for Farthest-Vehicle Dispatching Rule

Step 2: ***Determine if α criteria is satisfied with four vehicles***

The state space of the discrete-time Markov chains and the transition probabilities (in terms of the material flow matrix) are given in Appendix M. The stationary probability distribution of these discrete-time Markov chains (i.e., with four stations and three vehicles using the farthest-vehicle rule) is given in Appendix N, and the limiting probability equations are given in Appendix O.

Using Equation (7), the expected empty vehicle travel per load request is 118.25 seconds. Based on Equation (10), the service rate is 10.58 load requests per hour, and the probability distribution for the queueing network is given in Appendix P. Because (1-***P'***) = 0.3833 > α (=0.10), set $C_{min} = 3 + 1$ and go to Step 3.

Step 3: ***Calculate the Expected Waiting Time Using Lee and Longton's (1957) approximation procedure***

The expected waiting time for the M/M/N queueing network is 0.528 minutes, and the coefficient for Lee and Longton's algorithm is 0.85 (using Equations (10) and (14)). Hence, the expected waiting time for the load request is 0.4488 minutes.

6.1.3 Summary of the Algorithm Testing

From the examples illustrated above, it is evident that the empty-vehicle travel time for the farthest-vehicle dispatching rule is larger than that of the nearest-vehicle rule. This is consistent with the expectation that vehicles will travel farther under the farthest-vehicle rule than under the nearest-vehicle rule. It also makes sense that, under the same rule, when the number of vehicles in the system decreases, the empty vehicle travel time increases accordingly. This is because a smaller number of vehicles in the system increases the chance of invoking the empty vehicle travel. It is reasonable to state that this research can represent the AGVS appropriately when the *just-in-time* concept is considered (i.e., when the α value is small.)

Consistent with the simulation studies reported by Egbelu (1987), the rules-of-thumb appear to show an underestimation bias. This is because (1) these analytical procedures did not account for the dispatching rules utilized by the system and (2) the arrival of the load requests are assumed to be deterministic. This research supports Egbelu's (1987) conclusion that the number of vehicles required in the system will be larger when vehicle dispatching effects are considered in the system modeling process. The four analytical models proposed by Egbelu (1987) provide the system designer with only an estimation of the number of vehicles. Unlike these four analytical models, this research is able to provide the system designer with more information regarding the steady-state system performance, e.g., the expected waiting time for the load requests, the vehicle utilization rates, and the expected empty vehicle travel time.

The research results from the algorithm testing suggest that when α is set to be 0.10, four vehicles can meet the waiting time requirement for the sample problem, for both the nearest and farthest rules. However, the farthest rule will incur a longer waiting time than the nearest rule because of the larger empty vehicle travel time for the farthest-vehicle rule. To minimize the chance of waiting by the load requests, the system designer can reduce the size of α, increase the number of vehicles in the system, implement a different dispatching rule, or use a combination of these approaches. The reduction in α will definitely force the system to require more vehicles.

The sample problem used by Egbelu (1987) to determine the required number of vehicles for the four analytical models is also used to validate the proposed model. The number of vehicles needed for the four models are 8, 7, 9, and 13, respectively. Depending on the α value specified by the system designer, the number of vehicles required will be 14, 13, or 12 (the smaller the α, the larger the number of vehicles). The simulation study conducted by Egbelu (1987) indicates that the system needs 13 vehicles. This result shows that the proposed model can estimate the number of vehicles accurately. In addition to the number of vehicles, this research can also provide information regarding the steady-state system performance, e.g., the expected waiting time.

Table 6.9 Stationary Probabilities of the Markov Chains for Four Stations and Four Vehicles Using the Farthest Dispatching Rule

State	S1	S2	S3	S4	S5	S6	S7
Probability	0.081332	0.042843	0.001408	0.005763	0.080166	0.039958	0.065778

State	S8	S9	S10	S11	S12	S13	S14
Probability	0.049367	0.040176	0.041358	0.003857	0.00676	0.004092	0.017121

State	S15	S16	S17	S18	S19	S20	S21
Probability	0.014769	0.008393	0.055559	0.010721	0.033745	0.018495	0.024876

State	S22	S23	S24	S25	S26	S27	S28
Probability	0.006401	0.034357	0.052029	0.026048	0.032718	0.042085	0.029802

State	S29	S30	S31	S32	S33	S34	S35
Probability	0.01392	0.009992	0.014376	0.031196	0.016119	0.017984	0.026435

Table 6.10 Probability Distribution of the Load Requests for Four Vehicles Using the Farthest Dispatching Rule

n (# of load requests)	0	1	2	3	4	5	6
P(n)	0.1230	0.2526	0.2594	0.1776	0.0912	0.0468	0.0240
n (# of load requests)	**7**	**8**	**9**	**10**	**11**	**12**	**13**
P(n)	0.0123	0.0063	0.0033	0.0017	0.0009	0.0004	0.0002

6.2 PERFORMANCE EVALUATION

To validate the accuracy of the research, a series of simulation models programmed in the "C" language were used to evaluate the system's performance. The simulation models were verified using the "extreme case scenario." If we increase the number of AGVs to a very large number and distribute them equally (as far as possible) during the initialization stage, then the expected waiting time for a load request would be zero according to these models. Simulation runs were made using 24 vehicles (six in each location initially), and the waiting times for all the loads were determined to be zero. As a logical continuation of this, the system was simulated with just one vehicle, which lead to a very rapid buildup on all the queue lengths.

For validation and verification, the empty vehicle travel times and the loading/unloading time were set to zero, and the travel times between the workcenters during loaded travel were exponentially set, all with the same mean. The arrival rates at all the stations were set to the same constant rate. The system now essentially reduces to an M/M/c queueing system, analytical results for which are known. The simulation was run with the above distributions, and the simulation results compared with the analytically computed results.

Transactions from each workcenter in the AGVS were randomly generated using the Poisson distribution for the parameter values as taken from the material flow matrix given in Table 6.1. In order to evaluate the performance of the analytical model, different dispatching rule combinations were used in the simulation runs based on the same data given in the previous section for the analytical model. A separate simulation model was constructed for each dispatching rule combination, which includes the *NV/NW*, *NV/FW*, *FV/NW* and *FV/FW* dispatching rules, which are defined as follows:

- *NV/NW*: When there is one workcenter pending request(s) and there is more than one vehicle available, the system dispatches the nearest vehicle to the pending station. When there is one free vehicle and more than one station with a load request, this free vehicle is dispatched to the closest station with a pending request;

- *NV/FW*: When there is one workcenter pending request(s) and there is more than one vehicle available, the system dispatches the nearest vehicle to the pending station. When there is one free vehicle and more than one station requesting vehicle, this free vehicle is dispatched to the farthest station with a pending request for the free vehicle;

- *FV/NW*: When there is one workcenter pending request(s) and there is more than one vehicle available, the system dispatches the farthest vehicle to the pending station. When there is one free vehicle and more than one station requesting vehicle, this free vehicle is dispatched to the closest station with a pending request for the free vehicle; and

- *FV/FW*: When there is one workcenter pending request(s) and there is more than one vehicle available, the system dispatches the farthest vehicle to the pending station. When there is one free vehicle and more than one station requesting vehicle, this free vehicle is dispatched to the farthest station with a pending request for the free vehicle.

The *NV/NW* and *NV/FW* dispatching rules were used to test the results derived from the nearest vehicle dispatching rule, while the *FV/NW* and *FV/FW* dispatching rules were used to test the results derived from the farthest vehicle dispatching rule. The program for NV/NW is selected as representative of these four rule combinations, and is given in Appendix I.

6.2.1 Simulation Experiments

Each simulation model was executed for material flow intensities varying from 23 to 27 loads per hour. This is because when the material flow intensity is 28, the number of vehicles required for the Dispatching-Rule Based Algorithm will be 6. The state space for the discrete-time Markov chains for the second step of the algorithm will be 86 in this case, which poses as a difficulty in finding the stationary probability. When the flow intensity is 22, only 3 vehicles are needed, which is smaller than the number of stations. The DRBA requires that the number of vehicles be larger than or equal to the number of stations in the system. Hence, the simulation study's focus on the system will be five vehicles. Twenty different random seeds were used for each rule combination with a particular flow. Each simulation run has a ten-hour, warm-up period. The data for the load request waiting time is collected after one hundred hours. The comparisons of the simulation studies and the analytical model are based on the average expected waiting time of the load requests. Different material flows among stations were used to investigate the behavior of the analytical model. The outcome of these simulation runs for all the rule combinations are given in Appendices R through Y. Graphical charts of those averages against the analytical results are shown in Figures 6.1 through 6.8, and the 95% confidence intervals for these simulation runs are given in Tables 6.11 through 6.18.

Figures 6.1 through 6.4 show that the nearest dispatching rule is insensitive to the α value (i.e., the number of vehicles needed to satisfy the α criterion for different material flow intensities tends to be the same). In the simulation studies, the number of vehicles required for material flows intensities ranging from 23 to 27 for α is equal to 0.14, and 0.12 is the same (which is 4). When $\alpha = 0.10$, the number of required vehicles changes. Figure 6.4 suggests that, when α is 0.08, 5 vehicles are needed for flow intensities 25, 26 or 27. Note that the sharp dip in the curves is the result of adding a vehicle to the system in order to satisfy the α criterion for the given flow in testing.

Figures 6.5 through 6.8 show that the farthest-vehicle rule is more sensitive to the α value. For a given value of α, the farthest-vehicle rule requires the system to add a vehicle sooner than the corresponding system under the nearest-vehicle rule. This is because the expected empty vehicle travel time for the nearest-vehicle dispatching rule is far less than that for the farthest-vehicle rule when the same number of vehicles is used. (Based on the sample data, the empty vehicle travel time for the nearest rule with four vehicles is 73 seconds, and for the farthest rule is 101 seconds.) The empty vehicle travel time greatly affects the service rate of the vehicle for the different dispatching rules. When the service rate of the system is large, the probability of having more customers than servers in the system becomes smaller. This, illustrated in the second step of the algorithm, will make it more likely that the α criterion is satisfied. On the contrary, when the service rate is small, the probability of having more customers than servers in the system increases, and the system will be prone to violate the α criterion. The following section discusses the comparisons between the NV, NV/FW and NV/NW rule combinations, and the FV, FV/FW and FV/NW rule combinations.

6.2.2 The Nearest Rule Comparisons

As illustrated in Figures 6.1 through 6.4, the analytical model tends to underestimate the waiting time for the load requests. The reason for this is that the analytical model underestimates the empty-vehicle travel time. This is due to the discrete-time Markov chains model of the vehicle location used in the second step of the proposed algorithm. In order to find the location of the vehicles, the discrete-time Markov chains assume that each time a load request comes into the system, all the vehicles are available and are spread across the stations. This may not be true during some portion of the operating time of the shift (i.e., there is more than one load request in the system). Some of the fleet will be busy traveling along the guide path or loading and unloading the load request(s). The vehicle that would have to pickup the load request as specified by the discrete-time Markov chains is busy. The nearest rule will dispatch a free vehicle at the nearest location to pickup the load request. However, some of these free vehicles dispatched by the system may not be the nearest vehicle identified by the discrete-time Markov chains. Hence, the empty-vehicle travel time of the load request is underestimated for the nearest rule when this assumption is employed. However, this underestimation can be reduced by decreasing the α value (i.e., increasing the number of vehicles). When the number of vehicles is increased, the chance of having two load requests in the system at the same time will be lower, and the discrete-time Markov chains will capture the system behavior more accurately.

The simulation studies suggest that the NV/FW rule combination will result in a longer waiting time for the load requests than the NV/NW rule combination. Although the workcenter-initiated dispatching rule employed for both the rule combinations are the same (i.e., NV [nearest vehicle]), the vehicle-initiated dispatching rule used by the system will affect the waiting time.

When the FW vehicle-initiated rule is used, the system will dispatch the free vehicle to the station that is farthest from the station where the vehicle is located. This will result in more empty vehicle travel time than the NW rule. However, because of the unpredictability of the system's behavior, this research focuses on the workcenter-initiated dispatching rule (i.e., light traffic intensity). That means the probability of invoking the vehicle-initiated condition will be very small.

The statistical results show that the waiting time for the load requests is longer for the NV/FW rule combination than the NV/NW rule combination. Nevertheless, the small probability of invoking the vehicle-initiated rule implies there should not be a big difference in the waiting time for the NV/FW and NV/NW rule combinations. The simulation results confirm this and show that the biggest difference in the waiting time is less than three seconds between these two rules under the same conditions (i.e., the same number of vehicles and the same material flow intensity). Further, the difference will become smaller when the number of vehicles increases (or when the α value decreases). This is because the larger the number of vehicles in the system, the smaller is the chance of invoking the vehicle-initiated dispatching rule.

In this research, the estimation of the empty vehicle travel time is based purely on the workcenter-initiated rule. Ignoring the vehicle-initiated dispatching rule results in the underestimation of the expected waiting time for the analytical model. However, the underestimation of the expected waiting time does not undermine the accuracy of the analytical model because of the small chance of invoking the vehicle-initiated rule. As the material flow intensities increase, the underestimation decreases.

6.2.3 The Farthest Rule Comparisons

As indicated in Figures 6.5 through 6.8, the analytical model tends to overestimate the waiting time for the load requests. The reason for this is that the model overestimates the empty-vehicle travel time for the farthest-vehicle rule. As before, this is due to the assumption of the discrete-time Markov chains model that vehicles are always waiting when a load request arrives. Again, this overestimation can also be reduced by decreasing the α value.

The simulation studies also suggest that the FV/FW rule combination will result in a longer waiting time for the load requests than the FV/NW rule combination. Overall, the analytical model tends to overestimate the empty vehicle travel time (i.e., the servers will have a smaller service rate), which in turn will result in the overestimation of the waiting time for the load requests. The results show that the waiting time for the load requests is longer for the FV/FW rule combination than the FV/NW rule combination. Nevertheless, the small probability of invoking the vehicle-initiated rule will not generate a big difference on the waiting time for both the FV/FW and FV/NW rule combinations. As before, the small chance of the vehicle-initiated case will create little difference in the waiting time for the two rule combination. In this research, the estimation of the empty vehicle travel time is purely based on the workcenter-initiated rule. Ignorance of the vehicle-initiated dispatching rule results in the underestimation of the expected waiting time for the analytical model. However, the farthest rule tends to overestimate the expected waiting time. This overestimation of the vehicle travel time is greater than the underestimation of the expected waiting time because of a small chance of invoking the vehicle-initiated dispatching rule. As shown in Figures 6.5 through 6.8, the final results for the waiting time is determined by the farthest rule instead of by ignorance of the vehicle-initiated rule.

6.2.4 Data Analysis

To investigate the effectiveness of the analytical model, the expected waiting time from the analytical model is compared to the simulation data using t-test confidence intervals. The t-statistic is evaluated at a 95% level of significance with 19 degrees of freedom (i.e., 20 random seeds replications). With the tabulated value of the t-statistic equal to 2.093, the results are summarized in Tables 6.11 through 6.18. The numbers in the brackets after the material flow are the numbers of vehicles required, and the numbers in the parenthesis under the material flow in Tables 6.11 through 6.18 are the expected waiting times derived from the analytical models. The upper and lower bounds in Tables 6.11 through 6.18 are for a 95% confidence interval with 19 degrees of freedom. When the expected waiting time derived from the analytical model falls within the range of the lower bound to the upper bound, the analytical model is accepted (i.e., the analytical model can represent the real-life operations). Otherwise, the analytical model is rejected (i.e., the analytical model can not accurately estimate the expected waiting time). As the results of the tests suggest, the analytical model is more consistent with the simulation models when the number of vehicles in the system increases (i.e., the α value is smaller). The results also indicate that the predictions of the expected waiting time from the analytical model are consistent with the simulation studies with a small α and the nearest-workcenter rule in the vehicle-initiated condition. These conclusions suggest that the analytical model of this report is consistent with the real-life operation of the system.

The accuracy of the analytical models is also tested by finding the minimum number of vehicles needed to satisfy the α criterion (ranging from 0.04 to 0.20) for several different material flow intensities, ranging from 23 to 27, against the number of vehicles needed under the same working conditions for the simulation studies. The number of vehicles needed for the analytical model using the nearest rule is compared with the number required for the NV/NW and NV/FW rule combinations simulation studies. The results are given in Table 19. In Table 20, the number of vehicles required for the analytical model using the farthest rule is compared with that for the FV/NW and FV/FW simulation studies.

In this situation, when $\alpha = 0.08$ and material flow is 24 or $\alpha = 0.12$ and material flow is 26, the analytical model using the nearest rule underestimates the number of vehicles. As indicated in previous section, the nearest rule tends to underestimate the empty vehicle travel time, thus results in overestimating the probability of invoking the vehicle-initiated rule (i.e., α). This inevitably underestimates the required number of vehicles. Overall, the results show that the analytical models will require the same number of vehicles that are needed for the simulation studies to satisfy the same α criterion, given the same material flow.

Table 6.11 T-Test Results for the Nearest Dispatching Rule, with $\alpha = 0.14$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
NV/NW	23 [4]	(16.71,18.53)	Yes
NV/FW	(15.46)	(18.75,21.33)	Yes
NV/NW	24 [4]	(20.85,22.82)	Yes
NV/FW	(18.08)	(23.69,25.31)	Yes
NV/NW	25 [4]	(22.82,24.42)	Yes
NV/FW	(21.01)	(25.91,27.04)	Yes
NV/NW	26 [4]	(24.85,26.74)	Yes
NV/FW	(24.51)	(26.39,28.16)	Yes
NV/NW	27 [4]	(25.77,27.37)	Yes
NV/FW	(28.58)	(29.81,31.37)	Yes

Table 6.12 T-Test Results for the Nearest Dispatching Rule, with $\alpha = 0.12$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
NV/NW	23 [4]	(17.05,19.01)	Yes
NV/FW	(15.46)	(18.88,21.54)	Yes
NV/NW	24 [4]	(20.96,22.79)	Yes
NV/FW	(18.08)	(24.13,25.43)	Yes
NV/NW	25 [4]	(23.49,25.20)	Yes
NV/FW	(21.01)	(25.89,27.36)	Yes
NV/NW	26 [4]	(24.87,26.79)	Yes
NV/FW	(24.51)	(26.37,28.18)	Yes
NV/NW	27 [4]	(26.18,27.47)	Yes
NV/FW	(28.58)	(29.91,31.51)	Yes

Table 6.13 T-Test Results for the Nearest Dispatching Rule, with $\alpha = 0.10$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
NV/NW	23 [4]	(17.54,19.40)	Yes
NV/FW	(15.46)	(19.44,21.94)	Yes
NV/NW	24 [4]	(20.95,23.02)	Yes
NV/FW	(18.08)	(23.97,25.95)	Yes
NV/NW	25 [4]	(23.46,24.76)	Yes
NV/FW	(21.01)	(26.07,27.15)	Yes
NV/NW	26 [5]	(4.13,6.08)	No
NV/FW	(5.25)	(4.91,6.12)	No
NV/NW	27 [5]	(5.31,6.13)	No
NV/FW	(6.13)	(5.73,6.55)	No

Table 6.14 T-Test Results for the Nearest Dispatching Rule, with $\alpha = 0.08$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
NV/NW	23 [4]	(16.84,18.83)	Yes
NV/FW	(15.46)	(18.84,21.52)	Yes
NV/NW	24 [4]	(20.95,22.98)	Yes
NV/FW	(18.08)	(23.69,25.38)	Yes
NV/NW	25 [5]	(23.02,24.52)	Yes
NV/FW	(4.38)	(26.02,27.16)	Yes
NV/NW	26 [5]	(4.32,6.01)	No
NV/FW	(5.25)	(5.24,7.35)	No
NV/NW	27 [5]	(5.47,7.17)	No
NV/FW	(6.13)	(6.17,7.47)	No

Table 6.15 T-Test Results for the Farthest Dispatching Rule, with $\alpha = 0.14$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
FV/NW	23 [4]	(21.91,23.41)	Yes
FV/FW	(26.62)	(23.46,26.30)	Yes
FV/NW	24 [4]	(24.90,26.91)	Yes
FV/FW	(31.27)	(27.45,29.13)	Yes
FV/NW	25 [4]	(28.31,29.97)	Yes
FV/FW	(36.85)	(32.08,33.80)	Yes
FV/NW	26 [5]	(9.03,9.98)	No
FV/FW	(9.60)	(9.22,10.73)	No
FV/NW	27 [5]	(9.04,10.69)	Yes
FV/FW	(11.15)	(9.89,11.59)	No

Table 6.16 T-Test Results for the Farthest Dispatching Rule, with $\alpha = 0.12$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
FV/NW	23 [4]	(21.28,23.89)	Yes
FV/FW	(26.62)	(22.25,24.32)	Yes
FV/NW	24 [4]	(24.95,26.97)	Yes
FV/FW	(31.27)	(27.68,29.36)	Yes
FV/NW	25 [5]	(7.17,8.59)	No
FV/FW	(8.05)	(7.71,8.64)	No
FV/NW	26 [5]	(8.53,10.34)	No
FV/FW	(9.60)	(9.42,11.14)	No
FV/NW	27 [5]	(9.32,10.49)	Yes
FV/FW	(11.15)	(10.10,11.37)	No

Table 6.17 T-Test Results for the Farthest Dispatching Rule, with $\alpha = 0.10$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
FV/NW	23 [4]	(22.49,23.61)	Yes
FV/FW	(26.63)	(23.60,26.26)	Yes
FV/NW	24 [5]	(6.29, 7.15)	No
FV/FW	(6.82)	(6.40, 7.47)	No
FV/NW	25 [5]	(7.31,8.97)	No
FV/FW	(8.05)	(8.01,9.74)	No
FV/NW	26 [5]	(8.47,10.30)	No
FV/FW	(9.60)	(9.50,11.19)	No
FV/NW	27 [5]	(9.10,10.71)	Yes
FV/FW	(11.15)	(9.87,11.61)	No

Table 6.18 T-Test Results for the Farthest Dispatching Rule, with $\alpha = 0.08$

Rule Combination	Material Flow	(Upper Bound, Lower Bound)	Reject
FV/NW	23 [5]	(5.20,7.05)	No
FV/FW	(5.58)	(5.63,7.98)	Yes
FV/NW	24 [5]	(6.43,7.63)	No
FV/FW	(6.82)	(6.90,8.09)	Yes
FV/NW	25 [5]	(7.37,8.53)	No
FV/FW	(8.05)	(7.49,9.06)	No
FV/NW	26 [5]	(8.44,10.11)	No
FV/FW	(9.60)	(9.20,10.93)	No
FV/NW	27 [5]	(9.70,10.80)	Yes
FV/FW	(11.15)	(10.43,12.27)	No

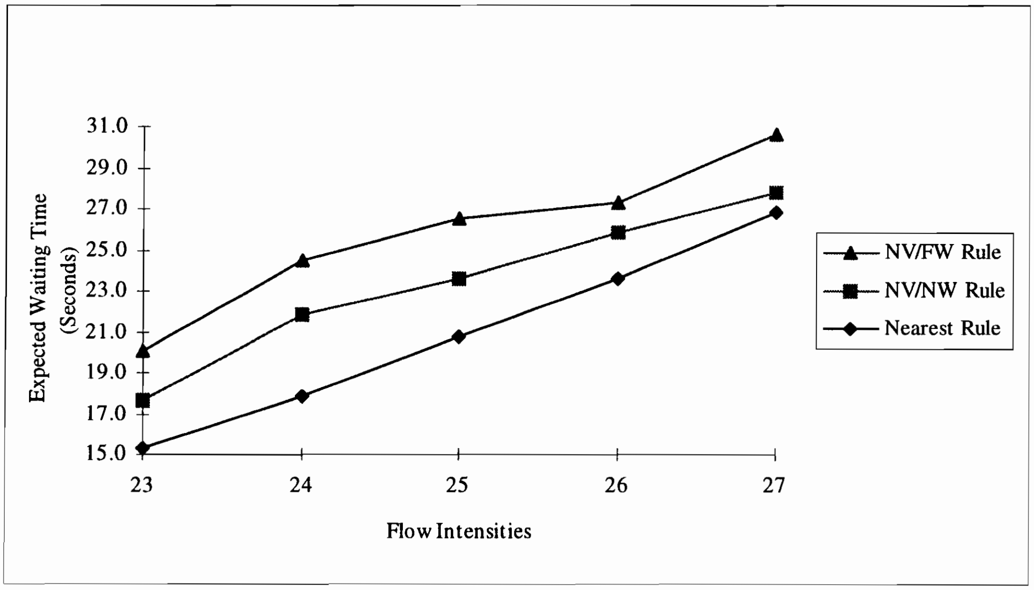

Figure 6.1 Comparisons of the Expected Load Requests Waiting Time from the Nearest Dispatching Rule and Simulation Studies ($\alpha = 0.14$)

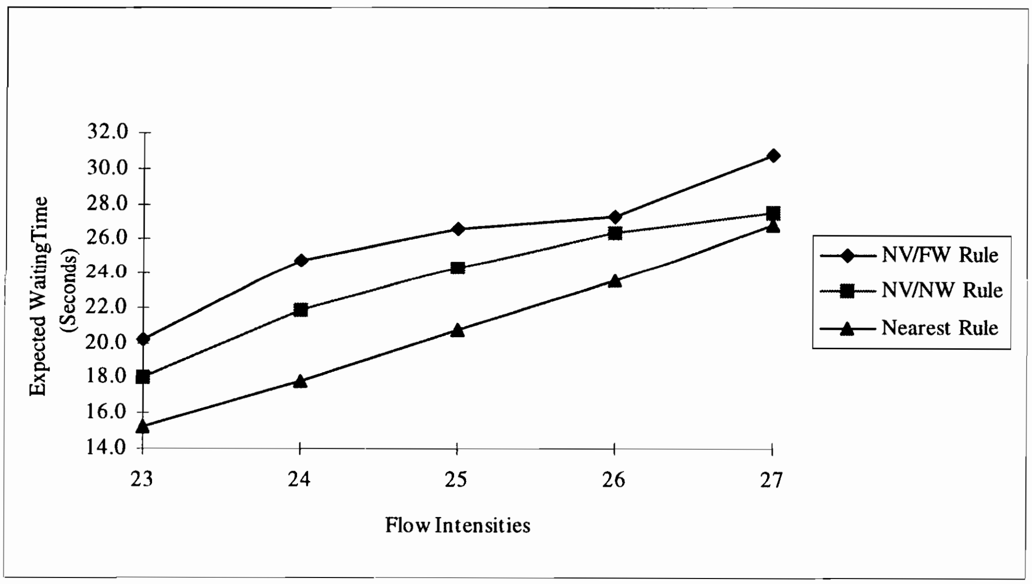

Figure 6.2 Comparisons of the Expected Load Requests Waiting Time from the Nearest Dispatching Rule and Simulation Studies ($\alpha = 0.12$)

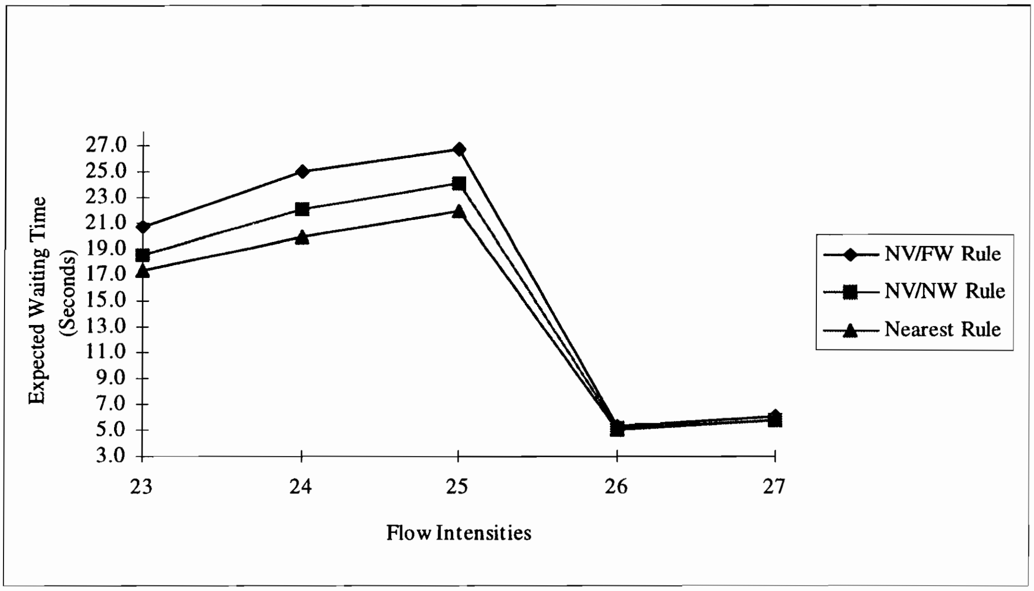

Figure 6.3 Comparisons of the Expected Load Requests Waiting Time from the Nearest Dispatching Rule and Simulation Studies ($\alpha = 0.10$)

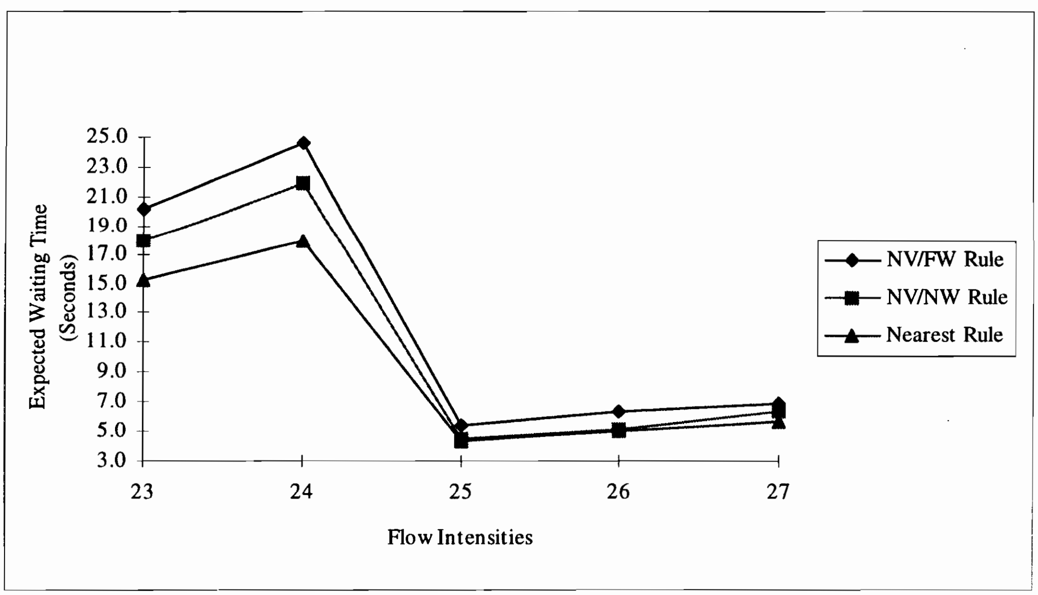

Figure 6.4 Comparisons of the Expected Load Requests Waiting Time from the Nearest Dispatching Rule and Simulation Studies ($\alpha = 0.08$)

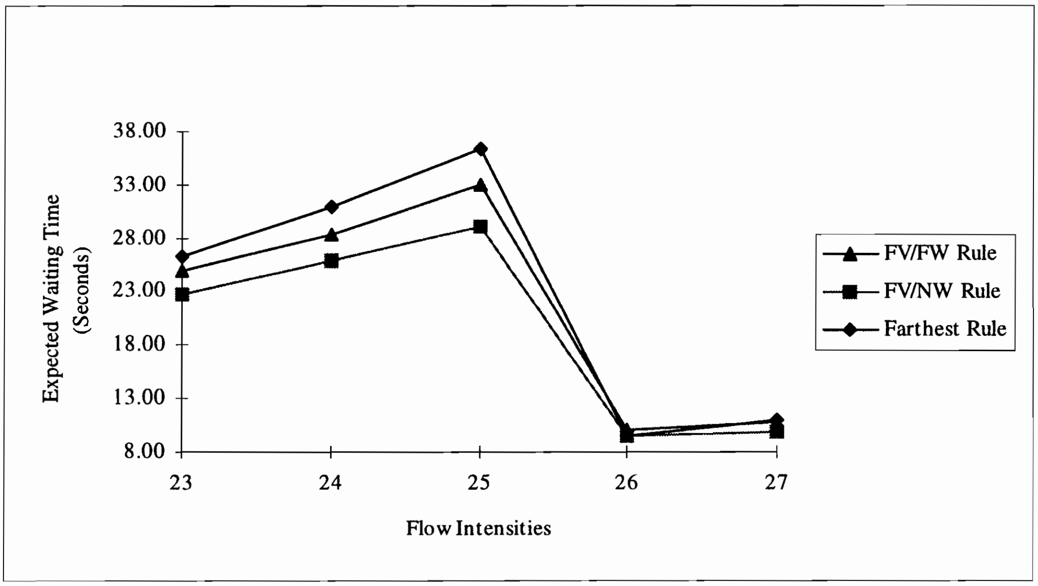

Figure 6.5 Comparisons of the Expected Load Requests Waiting Time from the Farthest Dispatching Rule and Simulation Studies ($\alpha = 0.14$)

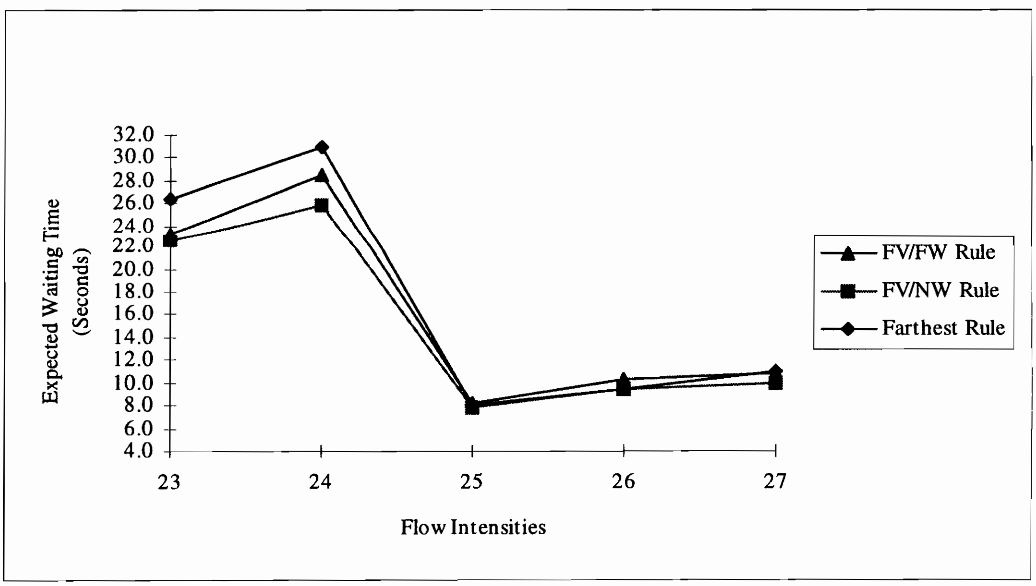

Figure 6.6 Comparisons of the Expected Load Requests Waiting Time from the Farthest Dispatching Rule and Simulation Studies ($\alpha = 0.12$)

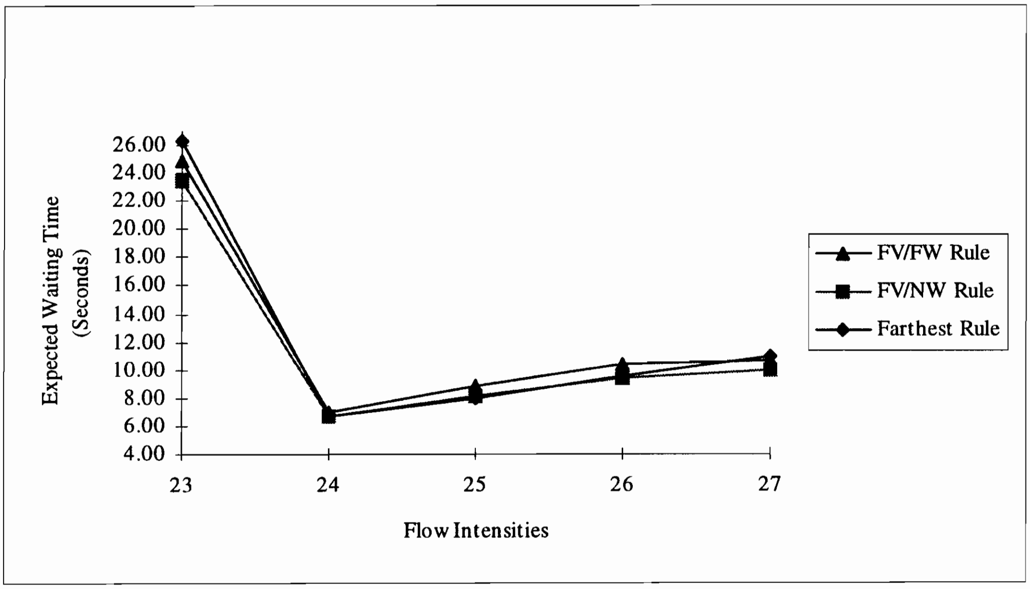

Figure 6.7 Comparisons of the Expected Load Requests Waiting Time from the Farthest Dispatching Rule and Simulation Studies ($\alpha = 0.10$)

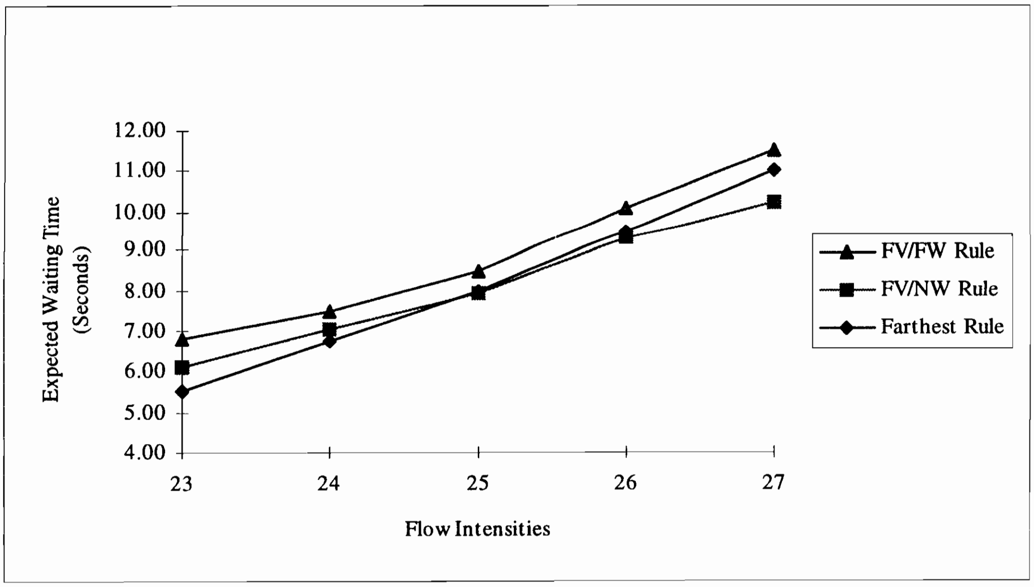

Figure 6.8 Comparisons of the Expected Load Requests Waiting Time from the Farthest Dispatching Rule and Simulation Studies ($\alpha = 0.08$)

Table 6.19 Number of Vehicles Required for the Analytical Model Using the Nearest Rule, NV/NW and NV/FW Simulation Studies

Flows	23			24			25			26			27		
α value	NV	NV/ NW	NV/ FW	NV	NV/ NW	NV/ FW	NV	NV/ NW	NV/ FW	NV	NV/ NW	NV/ FW	NV	NV/ NW	NV/ FW
0.04	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
0.08	4	4	5	4	5	5	5	5	5	5	5	5	5	5	5
0.12	4	4	4	4	4	4	4	4	5	4	5	5	5	5	5
0.16	4	4	4	4	4	4	4	4	4	4	4	4	5	5	5
0.20	4	4	4	4	4	4	4	4	4	4	4	4	5	4	5

Table 6.20 Number of Vehicles Required for the Analytical Model Using the Farthest Rule, FV/NW and FV/FW Simulation Studies

Flows	23			24			25			26			27		
α value	FV	FV/ NW	FV/ FW	FV	FV/ NW	FV/ FW	FV	FV/ NW	FV/ FW	FV	FV/ NW	FV/ FW	FV	FV/ NW	FV/ FW
0.04	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6
0.08	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
0.12	4	4	4	4	4	5	5	4	5	5	5	5	5	5	5
0.16	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5
0.20	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

7. SUMMARY AND CONCLUSIONS

AGVSs are indeed an important topic in manufacturing systems. Applications of AGVSs can be found in office environments, such as hospitals and post offices, as well as manufacturing areas such as the Volvo automobile organization. AGVSs encompass not only the vehicle and its guidance system, but also the control of one or more vehicles in the routine performance of the horizontal material handling tasks, for which the system is designed. When properly designed and controlled, AGVSs can provide significant savings in material handling costs as well as reductions in process inventories. The decision variables that must be considered in the design of AGVSs are the number and location of pickup and delivery stations, the number of vehicles, the routes used by the vehicles, the dispatching rules, the guide path layout, and the flow path. These decision variables are interdependent; in other words, changing the value of one variable can significantly affect the optimal values for other variables. In addition, the relationships among these variables are not well understood. This, in part, explains why there are no comprehensive models for the AGVS design problem. Instead, given the individual models, assumptions must be made regarding the value of several variables, while trying to find the optimal value for one or two of the remaining decision variables.

In Chapter Two, the functions of an AGVS and the vehicle and capacity requirements are described, followed by a three-step selection model for an AGV--attribute identification, attribute selection, and attribute ranking. In the same chapter, issues concerned with AGVS design and control (such as guide path layout, number of vehicles, vehicle dispatching, routing, and scheduling) are also discussed. Dispatching is one of the three major functions in AGV management. Vehicle dispatching deals with empty vehicle travel time/distance. The empty vehicle travel time/distance determines vehicle utilization because the loaded vehicle travel time is fixed. The distribution of the empty vehicle travel time over a specific operating time period, in order to minimize the blocking/congestion, the number of the vehicles needed and the required storage space of the pick-up/delivery stations are important issues in the design of AGVS.

Based on a comprehensive review of past AGVS research, this dissertation investigates the relationships among empty vehicle travel time, the number of vehicles, and the vehicle-initiated dispatching rules. However, due to the unpredictability of the dynamic behavior of the AGVSs, the proportion of time that the system is in either workcenter- or vehicle-initiated rules is unknown. In order to minimize the waiting time for the load requests, while at the same time maintaining a minimum number of vehicles in the system, this research utilizes only the workcenter-initiated dispatching rule.

Chapter Five presents a three-step algorithm to find the minimum number of vehicles, such that the probability of invoking the vehicle-initiated dispatching rule is smaller than a value specified by the system designer (α). The algorithm begins by finding an upper bound on the number of required vehicles. The first step of the algorithm assumes that the blocking and congestion time of the vehicles is equal to twice the loaded travel time. The second step of the algorithm uses a queueing model of the AGVS to determine if the α criterion is satisfied. In order to find α, discrete-time Markov chains are used to find the expected empty-vehicle travel time, which is a part of the service time. This service time is then used in a queueing model to approximate the distribution of the number of load requests in the system, which is used to estimate the value of α. As discussed in Chapter Five, the empty-vehicle travel time is a function of the number of vehicles in the system and the dispatching rule in force. The last step of the algorithm finds the expected waiting time for each load request in the system by using the approximation method proposed by Lee and Longton (1957).

A series of simulation studies, with different material flow intensities (i.e., arrival rates to the system) and α values, is conducted in Chapter Six to validate the proposed analytical model. The simulation studies show that the nearest-vehicle rule is less sensitive to changes in the α value than the farthest-vehicle rule. The simulation results also reveal that when α is large, the nearest-vehicle rule tends to underestimate the expected waiting time, while the farthest-vehicle rule tends to overestimate the expected waiting time. This is because the discrete-time Markov chains used in Chapter Six assume that all the vehicles are available when a load request enters the system. This results in an underestimation of the empty vehicle travel time for the nearest rule and an overestimation of the empty vehicle travel time for the farthest rule. The simulation studies also suggest that the model can predict more accurately when α is small. This is because when α is decreased, the number of vehicles needed in the system is increased accordingly and the probability of the vehicle-initiated situation decreases. When the material flow to the system is fixed, the more vehicles in the system, the more accurate is the representation of the discrete-time Markov chains. The simulation studies conclude that the model performs better as the material flows increase for the farthest-vehicle rule, and as the material flows decrease for the nearest-vehicle rule, when the number of vehicles is fixed.

Due to the approximation used in developing the algorithm, it can only be used in the early phase of designing an AGVS. It cannot replace the use of simulation as a design tool for an AGVS. However, it does provide a screening device that can be used by the system designer to narrow the range of system alternatives that warrant further investigation using the simulation technique. The following section states some of the limitations encountered in this research.

7.1 RESEARCH LIMITATIONS

1. Due to the complexity of transporting materials in an AGVS, only material flows between flexible manufacturing cells are considered. In other words, the material flows in the cells are ignored. However, in an actual flexible material system (FMS), the transportation of materials within the cells is as important as between the cells;
2. The storage space for each station is assumed to be infinite, but this assumption is somewhat impractical due to the space limitations in an AGVS;
3. The analytical model considers only two types of workcenter-initiated dispatching rules. In the AGV environment, the dispatching rules invoked are usually a combination of both the vehicle- and workcenter-initiated rules. (To some extent, the proportion of time that the system is in the vehicle-initiated condition can be relatively small if the number of vehicles in the system is large. Similarly, the proportion of time the system is in the workcenter-initiated condition is very large if the number of vehicles in the system is large.) Further, due to the dynamic behavior of the modeled system, the proportion of time during which each dispatching rule is used in the system is uncertain;
4. The algorithm depends on the given value of α. The value of α should be kept very small in order to ensure that the system will be in a workcenter-initiated condition. However, the value of α in real-life situations is not the only parameter that determines system behavior. A "good" dispatching rule or routing can achieve the same result in some cases; and

5. The model is limited by the number of vehicles and the number of workstations in the system. When the number of workstations and vehicles increases, the state space of the discrete-time Markov chains utilized to find the expected empty vehicle travel time increases dramatically. This results in an impractical computation.

7.2 RESEARCH CONTRIBUTIONS

This research results in five contributions to the AGVS literature. First, given that there is a lack of literature in material handling systems that deals with mathematical models, particularly for controlling empty-vehicle dispatching, this research is one of the very few analytical models that specifically takes stochastic behavior into account and explicitly incorporates an empty-vehicle dispatching rule into the material handling systems.

Second, the number of vehicles required in an AGVS can be estimated more accurately using the presented analytical model because of the relationship between the empty-vehicle travel time and the number of vehicles. Given that organizations now have a better understanding of the number of vehicles required to perform an operation, they can significantly reduce operating costs by permitting fewer vehicles to remain idle.

Third, the proposed analytical model is able to provide a better understanding of the efficiency of using different types of workcenter-initiated dispatching rules to deal with temporary peaks in system workloads. For instance, the derived equation would allow system designers to make necessary changes with the workcenter-initiated dispatching rules to accommodate vehicle resources lost due to congestion or blocking.

Fourth, the presented algorithm requires only the material flow matrix, travel distance, and a predetermined value to derive the number of vehicles needed, such that the load requests can have the shortest waiting time, and the system can employ the smallest possible number of vehicles without trading off the waiting time of the load requests.

Finally, the analytical results derived from this research can be used as initial estimates for more detailed simulation studies. With a more accurate estimate of the number of vehicles required and a better understanding of the workcenter-initiated dispatching rules, the range of the parameters of the simulation model can be greatly reduced. Also, unlike these four analytical models proposed by Egbelu (1987), this research can provide the system designer with more information regarding the steady-state system performance, e.g., the expected waiting time for the load requests, the vehicle utilization rates, and the expected empty vehicle travel time.

7.3 PROPOSED FUTURE RESEARCH

Extensions of this work include the following three proposed research areas:

1. ***Decision variables***: This research considers only the expected waiting time as the sole decision variable used to find the number of vehicles needed in the system. In real life situations, additional decision variables, such as the cost of the vehicles, should be considered in the design process;
2. ***The AGVS layout***: The simulation results show that with four vehicles in the system, the expected waiting time for a load request is around 30 seconds, and the vehicle utilization rate

is less than 50%. Increasing the vehicle utilization rate will automatically increase the waiting time of the load requests, resulting in a dilemma for the system designer. One potential research area would be to design a more efficient layout or guide path for the AGVS, in conjunction with the concept of vehicle dispatching discussed in this research, such that, on one hand, the vehicle utilization rate can be increased, and on the other hand, the expected waiting time of the load requests will remain the same; and

3. ***Workcenter-initiated dispatching rules***: This research utilizes only the nearest and farthest vehicle rules. However, there are many other vehicle dispatching rules that are more practical than the farthest rule. Therefore, future research should focus on dispatching vehicles that are least used in the system in order to balance the work loads.

WORKS CONSULTED

Agent, K., and Clark, J., 1982. "Evaluation of Reversible Lanes," ***Traffic Engineering and Control***, 23: 551-555.

Ammons, J., and McGinnis, L., 1987. "FMSs: An Overview for Management," ***The Production Handbook***, Chapter 8 (pp. 8-74 to 8-88), edited by J. White. New York, NY: John Wiley and Sons.

Bartholdi, J., and Platzman, L., 1989. "Decentralized Control of Automated Guided Vehicles on a Simple Loop," ***IIE Transactions***, 21.1: 76-81.

Beisteiner, F., and Moldaschl, J., 1983. "Strategies for the Employment of Vehicles in an Automated Guided Transportation System," ***Proceedings of the Second International Conference AGVS***, IFS Conferences Ltd., Stuttgart, West Germany, pp. 119-128.

Blair, E., Charnsethikul, P., and Vasques, 1987. "Optimal Routing of Driverless Vehicles in a Flexible Material Handling System," ***Material Flow***, 4: 73-83.

Bodin, L., and Golden, B., 1981. "Classifications in Vehicle Routing and Scheduling," ***Networks***, 11: 97.

Bozer, Yavuz, and Srinivasan, Mandyam, 1991. "Tandem Configurations for Automated Guided Vehicle Systems and the Analysis of Single Vehicle Loops," ***IIE Transactions***, 23.1: 72-82.

Co, Christine, and Tanchoco, J., 1991. "A Review of Research on AGVS Vehicle Management," ***Engineering Costs and Production Economics***, 21: 35-42.

Cohen, C., and Stein, J., 1978. Multi Purpose Optimization System User's Guide. Manual No. 320, Vogelback Computing Center, Northwestern University, Evaston, Illinois, U.S.A.

Daniels, S., 1988. “Real Time Conflict Resolution in Automated Guided Vehicle Scheduling,” Doctoral dissertation, Department of Industrial and Management Systems Engineering, Penn State University.

Egbelu, Pius, 1987a. "The Use of Non-Simulation Approaches to Estimating Vehicle Requirements in an Automated Guided Vehicle Based Transport System," ***Material Flow***, 4.1-2: 33-51.

---, 1987b. "Integration of Unit Load Selection in the Design of AGV Based Automated Manufacturing Shop," ***Proceedings of the IXth Conference Production Research***, Cincinnati, OH, pp. 2159-2166.

---, and Tanchoco, J., 1982. "Characterization of Automatic Guided Vehicle Dispatching Rules," ***International Production Research***, 22.3: 359-374.

---, 1986. "Potentials for Bi-Directional Guide-Path for Automated Guided Vehicle Based Systems," ***International Journal Production Research***, 24.5: 1075-1097.

Fujii, S., and Sandoh, H., 1987. “A Routing Algorithm for Automated Guided Vehicles in FMS,” ***Proceeding of the Ninth International Conference on Production Research***, 2: 2261-2267.

Gaskins, R., and Tanchoco, J., 1988. "Flow Path Design for Automated Guided Vehicle Systems," ***International Journal of Production Research***, 25.5: 667-676.

---, and Taghaboni, F., 1989. "Virtual Flow Path for Free-ranging Automated Guided Vehicle Systems," ***International Journal of Production Research***, 27.1: 91-100.

Hodgson, T. , King, R., and Monteith, S., 1987. "Developing Control Rules for an AGVS using Markov Decision Processes," ***Material Flow***, 4.1: 85-96.

Johnson, Eric, and Brandeau, M., 1993. "Analytic Model for Design of a Multi-Vehicle Automated Guided Vehicle System," ***Management Science***, 39.12: 1477-1489.

Kasilingam, R. , 1991. “Mathematical Modeling of the AGVS Capacity Requirements Planning Problem,” ***Engineering Costs and Production Economics***, 21: 171-175.

Kaspi, M. and Tanchoco, J., 1990. "Optimal Flow Path Design of Uni-Directional AGV Systems," ***International Journal of Production Research***, 28.6: 1023-1030.

Koff, G., 1987. "Automatic Guided Vehicle Systems: Applications, Controls, and Planning," ***Material Flow***, 4.1-2: 3-16.

Kulwiec, R., 1984. "Trends in Automatic Guided Vehicles Systems," ***Plant Engineering***, 38.23: 66-73.

Lee, A., and Longton, P., 1957. “Queueing Processes Associated with Airline Passenger Check-in,” ***Operations Research Quarterly***, 10: 56-71.

Leung, L., Khator, S., and Kimbler, D., 1987. "Assignment of AGVS with Different Vehicles Types," ***Material Flow***, 4.1-2: 65-72.

Maaloe, E., 1973. “Approximation Formula for Estimation of Waiting-time in Multiple-Channel Queueing Systems,” ***Management Science***, 19: 703-710.

Mahadevan, B., and Narendran, T., 1990. "Design of an Automated Guided Vehicle-Based Material Handling System for a Flexible Manufacturing System," ***International Journal of Product Research***, 28.9: 1611-1622.

Malmborg, Charles, 1990. “A Model for the Design of Zone Control Automated Guided Vehicle Systems,” ***International Journal of Product Research***, 28.10: 1741-1758.

---, 1991. “Tightened Analytical Bounds on the Impact of Dispatching Rules in Automated Guided Vehicle Systems,” ***Applied Mathematical Modelling***, 15 (June): 305-311.

---, and Shen, Yu-Cheng, 1991. “Simulation Based Validation of Real Time Control Rules in Discrete Materials Handling Systems,” ***Proceedings of the 1991 IEEE***

International Conference on Systems, Man and Cybernetics, Charlottesville, Virginia, October, 1991.

Maxwell, W., 1981. "Solving Material Handling Design Problems with OR," ***Industrial Engineering***, 13.4: 58-69.

---, and Muckstadt, J., 1982. "Design of Automatic Guided Vehicle Systems," ***IIE Transactions***, 14.2: 114-124.

Nozaki, S., and Ross, S., 1978. "Approximations in Finite-Capacity Multi-Server Queueing with Poisson Arrivals," ***Journal of Applied Probability***, 15: 826-834.

Petersen, E., 1977. "Capacity of Single Track Rail Line," Technical Report, Canadian Institute of Guided Ground Transport, Queen's University at Kingston, Ontario.

Psaraftis, Harilaos, 1980. "A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem," ***Transportation Science***, 14.2: 130-154.

Ross, S., 1993, ***Introduction to Probability Models***, Harcourt Brace & Company, NY: Academic Press, Inc.

Sharp, G., and Liu, F., 1990. "An analytical Method for Configuring Fixed-Path Closed-Loop Material Handling Systems," ***International Journal of Production Research***, 28.4: 757-783.

Shelton, Debra, and Jones, Marilyn, 1987. "A Selection Method for Automated Guided Vehicles," ***Material Flow***, 4: 97-107.

Sinriech, D., and Tanchoco, J., 1991. "Intersection Graph Method for AGV Flow Path Design," ***International Journal of Production Research,*** 29.9: 1725-1732.

Srinivasan, M., Bozer, Y., and Cho, M., 1994. "Trip-Based Material Handling Systems: Throughput Capacity Analysis," ***IIE Transaction***, 26.1: 70-89.

Stoyan, D., 1976. "Approximations for M/G/s Queues," ***Math. Operational. Stat.***, 7: 587-594.

Taghaboni, F., and Tanchoco, J., 1988. "A LISP-Based Controller for Free-Ranging Automated Guided Vehicle Systems," ***International Journal of Production Research***, 26.2: 173-188.

Tanchoco, J., Egbelu, P., and Taghaboni, F., 1987. "Determination of the Total Number of Vehicles in an AGV-Based System," ***Material Flow***, 4.1-2: 33-51.

Usher, J., Evans, G., and Wilhelm, M., 1988. "AGV Flow Path Design and Load Transfer Point Location," ***Proceedings of IIE Conference***, 174-179.

Wilhelm, M., and Evans, G., 1987. "The State-of the-Art in AGV Systems Analysis and Planning," Proceedings of the AGVS Seminar on October 1987, held in Pittsburgh, PA. pp. 51-70.

Whitt, W., 1983. "The Queueing Network Analyzer," The Bell System Technical Journal, 62.9: 2779-2812.

Zeng, Lai-Guang, Wang, Hsu-Pin, and Jin, Song, 1991. "Conflict Detection of Automated Guided Vehicles: A Petri Net Approach," ***International Journal Production Research***, 29.5: 865-879.

APPENDICES

APPENDIX A: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE NEAREST RULE

State	Possible Transitions (Nearest Rule)			
(5,0,0,0)	(4,1,0,0) [p12,p32,p42]	(4,0,1,0) [p13,p23,p43]	(4,0,0,1) [p14,p24,p34]	(5,0,0,0) [p21,p31,p41]
(0,5,0,0)	(1,4,0,0) [p21,p31,p41]	(0,4,1,0) [p23,p13,p43]	(0,4,0,1) [p24,p14,p34]	(0,5,0,0) [p12,p32,p42]
(0,0,5,0)	(1,0,4,0) [p31,p21,p41]	(0,1,4,0) [p32,p12,p42]	(0,0,4,1) [p34,p14,p24]	(0,0,5,0) [p13,p23,p43]
(0,0,0,5)	(1,0,0,4) [p41,p21,p31]	(0,1,0,4) [p42,p12,p32]	(0,0,1,4) [p43,p13,p23]	(0,0,0,5) [p14,p24,p34]
(4,1,0,0)	(4,0,1,0) [p23]	(4,0,0,1) [p24]	(4,1,0,0) [p31,p41]	(3,1,0,1) [p14,p34]
(4,0,1,0)	(4,1,0,0) [p32]	(4,0,1,0) [p21,p41]	(4,0,0,1) [p34]	(3,1,1,0) [p12,p42]
(4,0,0,1)	(4,1,0,0) [p42]	(4,0,1,0) [p43]	(4,0,0,1) [p21,p31]	(3,1,0,1) [p12,p32]
(1,4,0,0)	(0,4,1,0) [p13,p43]	(0,4,0,1) [p14,p34]	(2,3,0,0) [p21]	(1,4,0,0) [p31,p41]
(0,4,1,0)	(0,3,2,0) [p13,p23,p43]	(0,4,1,0) [p12,p42]	(1,3,1,0) [p21,p41]	(0,3,1,1) [p14,p24]
(0,4,0,1)	(0,4,0,1) [p12,p34]	(0,3,0,2) [p14,p24]	(1,3,0,1) [p21]	(0,3,1,1) [p13,p23]
(1,0,4,0)	(0,0,5,0) [p13,p23,p43]	(0,0,4,1) [p14,p24]	(1,0,4,0) [p21,p41]	(0,1,4,0) [p12,p42]
(0,1,4,0)	(0,0,5,0) [p13,p23,p43]	(0,0,4,1) [p14,p24]	(1,0,4,0) [p21,p41]	(0,1,4,0) [p12,p42]
(0,0,4,1)	(0,0,4,1) [p13,p24]	(0,0,3,2) [p14,p34]	(1,0,4,0) [p21,p41]	(0,0,5,0) [p23,p43]
(1,0,0,4)	(0,0,1,4) [p13,p23]	(0,0,0,5) [p14,p24,p34]	(1,0,0,4) [p21,p31]	(2,0,0,3) [p41]
(0,1,0,4)	(0,0,1,4) [p13,p23]	(0,0,0,5) [p14,p24]	(1,0,0,4) [p21]	(1,1,0,3) [p31,p41]
(0,0,1,4)	(0,0,1,4) [p13,p24]	(0,0,0,5) [p14,p34]	(1,0,1,3) [p21,p41]	(0,0,2,3) [p23,p43]
(3,2,0,0)	(2,2,1,0) [p13,p43]	(2,2,0,1) [p14,p34]	(2,3,0,0) [p12,p32,p42]	(3,1,1,0) [p23]
(3,0,2,0)	(2,0,3,0) [p13,p23,p43]	(2,0,2,1) [p14,p24]	(3,0,2,0) [p21,p41]	(2,1,2,0) [p12,p42]
(3,0,0,2)	(2,0,1,2) [p13,p23]	(2,0,0,3) [p14,p24,p34]	(3,0,0,2) [p21,p31]	(4,0,0,1) [p41]
(2,3,0,0)	(1,3,1,0) [p13,p43]	(1,3,0,1) [p14,p34]	(1,4,0,0) [p12,p32,p42]	(2,2,1,0) [p23]
(0,3,2,0)	(0,2,3,0) [p13,p23,p43]	(02,2,1) [p14,p24]	(1,2,2,0) [p21,p41]	(0,3,2,0) [p12,p42]
(0,3,0,2)	(0,2,1,2) [p13,p23]	(0,2,0,3) [p14,p24]	(0,3,0,2) [p12,p34]	(1,3,0,1) [p31,p41]
(2,0,3,0)	(1,0,4,0) [p13,p23,p43]	(1,0,3,1) [p14,p24]	(2,0,3,0) [p21,p41]	(1,1,3,0) [p12,p42]
(0,2,3,0)	(0,1,4,0) [p13,p23,p43]	(0,1,3,1) [p14,p24]	(1,1,3,0) [p21,p41]	(0,2,3,0) [p12,p42]
(0,0,3,2)	(0,0,3,2) [p13,p24]	(0,0,2,3) [p14,p34]	(1,0,3,1) [p21,p41]	(0,0,4,1) [p23,p43]
(2,0,0,3)	(1,0,1,3) [p13,p23]	(1,0,0,4) [p14,p24,p34]	(2,0,0,3) [p21,p31]	(3,0,0,2) [p41]
(0,2,0,3)	(0,1,1,3) [p13,p23]	(0,1,0,4) [p14,p24]	(0,2,0,3) [p12,p34]	(1,2,0,2) [p31,p41]
(0,0,2,3)	(0,0,2,3) [p13,p24]	(0,0,1,4) [p14,p34]	(1,0,2,2) [p21,p41]	(0,0,3,2) [p23,p43]
(3,1,1,0)	(2,1,2,0) [p13,p43]	(2,2,1,0) [p12,p42]	(4,0,1,0) [p21]	(3,0,2,0) [p23]
	(2,1,1,1) [p14]			
(3,1,0,1)	(2,2,0,1) [p12,p32]	(2,1,1,1) [p13]	(2,1,0,2) [p14,p34]	(4,0,0,1) [p21]
	(3,1,1,0) [p43]			
(3,0,1,1)	(2,1,1,1) [p12]	(2,0,2,1) [p13,p23]	(2,0,1,2) [p14,p24]	(3,0,1,1) [p21]
	(4,0,1,0) [p41]			

APPENDIX A: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE NEAREST RULE (CONTINUED)

State	Possible Transitions (Nearest Rule)			
(1,3,1,0)	(0,4,1,0) [p12,p42]	(0,3,2,0) [p13,p43]	(0,3,1,1) [p14]	(2,2,1,0) [p21]
	(1,3,1,0) [p41]			
(1,3,0,1)	(0,4,0,1) [p12,p32]	(0,3,1,1) [p13]	(0,3,0,2) [p14,p34]	(2,2,0,1) [p21]
	(1,3,1,0) [p43]			
(0,3,1,1)	(0,3,1,1) [p12]	(0,2,2,1) [p13,p23]	(0,2,1,2) [p14,p24]	(1,2,1,1) [p21]
	(0,3,2,0) [p43]			
(1,1,3,0)	(0,2,3,0) [p12,p42]	(0,1,4,0) [p13,p43]	(0,1,3,1) [p14]	(2,0,3,0) [p21]
	(1,1,3,0) [p41]			
(1,0,3,1)	(0,1,3,1) [p12]	(0,0,4,1) [p13,p23]	(0,0,3,2) [p14,p24]	(1,0,3,1) [p21]
	(1,0,4,0) [p43]			
(0,1,3,1)	(0,1,3,1) [p12]	(0,0,4,1) [p13,p23]	(0,0,3,2) [p14,p24]	(1,0,3,1) [p21]
	(0,1,4,0) [p43]			
(1,1,0,3)	(0,2,0,3) [p12,p32]	(0,1,1,3) [p13]	(0,1,0,4) [p14,p34]	(2,0,0,3) [p21]
	(1,1,1,2) [p43]			
(1,0,1,3)	(0,1,1,3) [p12]	(0,0,2,3) [p13,p23]	(0,0,1,4) [p14,p24]	(1,0,1,3) [p21]
	(1,0,2,2) [p43]			
(0,1,1,3)	(0,1,1,3) [p12]	(0,0,2,3) [p13,p23]	(0,0,1,4) [p14,p24]	(1,0,1,3) [p21]
	(0,1,2,2) [p43]			
(2,2,1,0)	(1,3,1,0) [p12,p42]	(1,2,2,0) [p13,p43]	(1,2,1,1) [p14]	(3,1,1,0) [p21]
	(2,2,1,0) [p41]			
(2,2,0,1)	(1,3,0,1) [p12,p32]	(1,2,1,1) [p13]	(1,2,0,2) [p14,p34]	(3,1,0,1) [p21]
	(2,2,1,0) [p43]			
(2,1,2,0)	(1,2,2,0) [p12,p42]	(1,1,3,0) [p13,p43]	(1,1,2,1) [p14]	(3,0,2,0) [p21]
	(2,1,2,0) [p41]			
(2,0,2,1)	(1,1,2,1) [p12]	(1,0,3,1) [p13,p23]	(1,0,2,2) [p14,p24]	(2,0,2,1) [p21]
	(2,0,3,0) [p43]			
(2,1,0,2)	(1,2,0,2) [p12,p32]	(1,1,1,2) [p13]	(1,1,0,3) [p14,p34]	(3,0,0,2) [p21]
	(2,1,1,1) [p43]			
(2,0,1,2)	(1,1,1,2) [p12]	(1,0,2,2) [p13,p23]	(1,0,1,3) [p14,p24]	(2,0,1,2) [p21]
	(2,0,2,1) [p43]			
(0,2,2,1)	(0,2,2,1) [p12]	(0,1,3,1) [p13,p23]	(0,1,2,2) [p14,p24]	(1,1,2,1) [p21]
	(0,2,3,0) [p43]			
(1,2,2,0)	(0,3,2,0) [p12,p42]	(0,2,3,0) [p13,p43]	(0,2,2,1) [p14]	(2,1,2,0) [p21]
	(1,2,2,0) [p41]			
(1,2,0,2)	(0,3,0,2) [p12,p32]	(0,2,1,2) [p13]	(0,2,0,3) [p14,p34]	(2,1,0,2) [p21]
	(1,2,1,1) [p43]			
(0,2,1,2)	(0,2,1,2) [p12]	(0,1,2,2) [p13,p23]	(0,1,1,3) [p14,p24]	(1,1,1,2) [p21]
	(0,2,2,1) [p43]			
(1,0,2,2)	(0,1,2,2) [p12]	(0,0,3,2) [p13,p23]	(0,0,2,3) [p14,p24]	(1,0,2,2) [p21]
	(1,1,2,1) [p42]			
(0,1,2,2)	(0,1,2,2) [p12]	(0,0,3,2) [p13,p23]	(0,0,2,3) [p14,p24]	(1,0,2,2) [p21]
	(0,1,3,1) [p43]			
(2,1,1,1)	(1,2,1,1) [p12]	(1,1,2,1) [p13]	(1,1,1,2) [p14]	(3,0,1,1) [p21]
	(3,1,1,0) [p41]	(2,2,1,0) [p42]	(2,1,2,0) [p43]	
(1,2,1,1)	(0,3,1,1) [p12]	(0,2,2,1) [p13]	(0,2,1,2) [p14]	(2,1,1,1) [p21]
	(2,2,1,0) [p41]	(1,3,1,0) [p42]	(1,2,2,0) [p43]	
(1,1,2,1)	(0,2,2,1) [p12]	(0,1,3,1) [p13]	(0,1,2,2) [p14]	(2,0,2,1) [p21]
	(2,1,2,0) [p41]	(1,2,2,0) [p42]	(1,1,3,0) [p43]	
(1,1,1,2)	(0,2,1,2) [p12]	(0,1,2,2) [p13]	(0,1,1,3) [p14]	(2,0,1,2) [p21]
	(2,1,1,1) [p41]	(1,2,1,1) [p42]	(1,1,2,1) [p43]	

APPENDIX A: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE NEAREST RULE (CONTINUED)

State	Possible Transitions (Nearest Rule)				
(4,1,0,0)	(3,1,1,0) [p13,p43]	(3,2,0,0) [p12,p32,p42]		(5,0,0,0) [p21]	
(4,0,1,0)	(3,0,1,1) [p14,p24]	(3,0,2,0) [p13,p23,p43]		(5,0,0,0) [p31]	
(4,0,0,1)	(3,0,1,1) [p13,p23]	(3,0,0,2) [p14,p24,p34]		(5,0,0,0) [p41]	
(1,4,0,0)	(1,3,0,1) [p24]	(0,5,0,0) [p12,p32,p42]		(1,3,1,0) [p23]	
(0,4,1,0)	(1,4,0,0) [p31]	(0,5,0,0) [p32]	(0,4,0,1) [p34]		
(0,4,0,1)	(1,4,0,0) [p41,p31]	(0,5,0,0) [p42,p32]		(0,4,1,0) [p43]	
(1,0,4,0)	(2,0,3,0) [p31]	(1,1,3,0) [p32]	(1,0,3,1) [p34]		
(0,1,4,0)	(1,1,3,0) [p31]	(0,2,3,0) [p32]	(0,1,3,1) [p34]		
(0,0,4,1)	(0,1,3,1) [p12,p32]	(1,0,3,1) [p31]	(0,1,4,0) [p42]		
(1,0,0,4)	(1,1,0,3) [p42]	(0,1,0,4) [p12,p32]		(1,0,1,3) [p43]	
(0,1,0,4)	(0,2,0,3) [p32,p42]	(0,1,0,4) [p12,p34]		(0,1,1,3) [p43]	
(0,0,1,4)	(1,0,0,4) [p31]	(0,1,0,4) [p12,p32]		(0,1,1,3) [p42]	
(3,2,0,0)	(3,1,0,1) [p24]	(3,2,0,0) [p31,p41]		(4,1,0,0) [p21]	
(3,0,2,0)	(3,1,1,0) [p32]	(3,0,1,1) [p34]	(4,0,1,0) [p31]		
(3,0,0,2)	(3,1,0,1) [p42]	(2,1,0,2) [p12,p32]		(3,0,1,1) [p43]	
(2,3,0,0)	(2,2,0,1) [p24]	(2,3,0,0) [p31,p41]		(3,2,0,0) [p21]	
(0,3,2,0)	(0,4,1,0) [p32]	(0,3,1,1) [p34]	(1,3,1,0) [p31]		
(0,3,0,2)	(0,4,0,1) [p32,p42]	(0,3,1,1) [p43]	(1,2,0,2) [p21]		
(2,0,3,0)	(2,1,2,0) [p32]	(2,0,2,1) [p34]	(3,0,2,0) [p31]		
(0,2,3,0)	(0,3,2,0) [p32]	(0,2,2,1) [p34]	(1,2,2,0) [p31]		
(0,0,3,2)	(0,1,2,2) [p12,p32]	(0,1,3,1) [p42]	(1,0,2,2) [p31]		
(2,0,0,3)	(2,1,0,2) [p42]	(1,1,0,3) [p12,p32]		(2,0,1,2) [p43]	
(0,2,0,3)	(0,3,0,2) [p32,p42]	(1,1,0,3) [p21]	(0,2,1,2) [p43]		
(0,0,2,3)	(0,1,1,3) [p12,p32]	(0,1,2,2) [p42]	(1,0,1,3) [p31]		
(3,1,1,0)	(3,0,1,1) [p24]	(4,1,0,0) [p31]	(3,2,0,0) [p32]	(3,1,0,1) [p34]	(3,1,1,0) [p41]
(3,1,0,1)	(3,0,1,1) [p23]	(3,0,0,2) [p24]	(3,1,0,1) [p31]	(4,1,0,0) [p41]	(3,2,0,0) [p42]
(3,0,1,1)	(3,0,2,0) [p43]	(3,1,0,1) [p32]	(4,0,0,1) [p31]	(3,1,1,0) [p42]	(3,0,0,2) [p34]
(1,3,1,0)	(1,2,2,0) [p23]	(1,2,1,1) [p24]	(2,3,0,0) [p31]	(1,4,0,0) [p32]	(1,3,0,1) [p34]
(1,3,0,1)	(1,2,1,1) [p23]	(1,2,0,2) [p24]	(1,3,0,1) [p31]	(2,3,0,0) [p41]	(1,4,0,0) [p42]
(0,3,1,1)	(1,3,0,1) [p31]	(0,4,0,1) [p32]	(0,3,0,2) [p34]	(1,3,1,0) [p41]	(0,4,1,0) [p42]
(1,1,3,0)	(1,0,4,0) [p23]	(1,0,3,1) [p24]	(2,1,2,0) [p31]	(1,2,2,0) [p32]	(1,1,2,1) [p34]
(1,0,3,1)	(2,0,2,1) [p31]	(1,1,2,1) [p32]	(1,0,2,2) [p34]	(2,0,3,0) [p41]	(1,1,3,0) [p42]
(0,1,3,1)	(1,1,2,1) [p31]	(0,2,2,1) [p32]	(0,1,2,2) [p34]	(1,1,3,0) [p41]	(0,2,3,0) [p42]
(1,1,0,3)	(1,0,1,3) [p23]	(1,0,0,4) [p24]	(1,1,0,3) [p31]	(2,1,0,2) [p41]	(1,2,0,2) [p42]
(1,0,1,3)	(2,0,0,3) [p31]	(1,1,0,3) [p32]	(1,0,0,4) [p34]	(2,0,1,2) [p41]	(1,1,1,2) [p42]
(0,1,1,3)	(1,1,0,3) [p31]	(0,2,0,3) [p32]	(0,1,0,4) [p34]	(1,1,1,2) [p41]	(0,2,1,2) [p42]
(2,2,1,0)	(2,1,2,0) [p23]	(2,1,1,1) [p24]	(3,2,0,0) [p31]	(2,3,0,0) [p32]	(2,2,0,1) [p34]
(2,2,0,1)	(2,1,1,1) [p23]	(2,1,0,2) [p24]	(2,2,0,1) [p31]	(3,2,0,0) [p41]	(2,3,0,0) [p42]
(2,1,2,0)	(2,0,3,0) [p23]	(2,0,2,1) [p24]	(3,1,1,0) [p31]	(2,2,1,0) [p32]	(2,1,1,1) [p34]
(2,0,2,1)	(3,0,1,1) [p31]	(2,1,1,1) [p32]	(2,0,1,2) [p34]	(3,0,2,0) [p41]	(2,1,2,0) [p42]
(2,1,0,2)	(2,0,1,2) [p23]	(2,0,0,3) [p24]	(2,1,0,2) [p31]	(3,1,0,1) [p41]	(2,2,0,1) [p42]
(2,0,1,2)	(3,0,0,2) [p31]	(2,1,0,2) [p32]	(2,0,0,3) [p34]	(3,0,1,1) [p41]	(2,1,1,1) [p42]
(0,2,2,1)	(1,2,1,1) [p31]	(0,3,1,1) [p32]	(0,2,1,2) [p34]	(1,2,2,0) [p41]	(0,3,2,0) [p42]
(1,2,2,0)	(1,1,3,0) [p23]	(1,1,2,1) [p24]	(2,2,1,0) [p31]	(1,3,1,0) [p32]	(1,2,1,1) [p34]
(1,2,0,2)	(1,1,1,2) [p23]	(1,1,0,3) [p24]	(1,2,0,2) [p31]	(2,2,0,1) [p41]	(1,3,0,1) [p42]
(0,2,1,2)	(1,2,0,2) [p31]	(0,3,0,2) [p32]	(0,2,0,3) [p34]	(1,2,1,1) [p41]	(0,3,1,1) [p42]
(1,0,2,2)	(1,0,3,1) [p43]	(2,0,1,2) [p31]	(1,1,1,2) [p32]	(1,0,1,3) [p34]	(2,0,2,1) [p41]
(0,1,2,2)	(1,1,1,2) [p31]	(0,2,1,2) [p32]	(0,1,1,3) [p34]	(1,1,2,1) [p41]	(0,2,2,1) [p42]
(2,1,1,1)	(2,0,2,1) [p23]	(2,0,1,2) [p24]	(3,1,0,1) [p31]	(2,2,0,1) [p32]	(2,1,0,2) [p34]
(1,2,1,1)	(1,1,2,1) [p23]	(1,1,1,2) [p24]	(2,2,0,1) [p31]	(1,3,0,1) [p32]	(1,2,0,2) [p34]
(1,1,2,1)	(1,0,3,1) [p23]	(1,0,2,2) [p24]	(2,1,1,1) [p31]	(1,2,1,1) [p32]	(1,1,1,2) [p34]
(1,1,1,2)	(1,0,2,2) [p23]	(1,0,1,3) [p24]	(2,1,0,2) [p31]	(1,2,0,2) [p32]	(1,1,0,3) [p34]

APPENDIX B: LINEAR EQUATIONS FOR FOUR STATIONS AND FIVE VEHICLES USING THE NEAREST RULE

X40; (Objective Function)
(c1 to c56 are constraints)
c1: X1 + X2 + X3+ X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13
+ X14 + X15 + X16 + X17 + X18 + X19 + X20 + X21 + X22 + X23 + X24
+ X25 + X26 + X27 + X28 + X29 + X30 + X31 + X32 + X33 + X34 + X35
+ X36 + X37 + X38 + X39 + X40 + X41 + X42 + X43 + X44 + X45 + X46
+ X47 + X48 + X49 + X50 + X51 + X52 + X53 + X54 + X55 + X56 = 1;
c2: 16 X1 - 2 X5 - 2 X6 - 3 X7 = 0;
c3: 16 X2 - 7 X8 - 2 X9 - 3 X10 = 0;
c4: 19 X3 - 4 X11 - 4 X12 - 2 X13 = 0;
c5: 18 X4 - 5 X14 - 4 X15 - 3 X16 = 0;
c6: 7 X1 -18 X5 + 2 X6 + 1 X7 + 2 X17 + 2 X29 + 3 X30 = 0;
c7: 4 X1 + 1 X5 -18 X6 + 1 X7 + 2 X18 + 2 X29 + 3 X31 = 0;
c8: 5 X1 + 2 X5 + 1 X6 -19 X7 + 3 X19 + 2 X30 + 2 X31 = 0;
c9: 7 X2 -18 X8 + 2 X9 + 5 X10 + 7 X20 + 2 X32 + 1 X33 = 0;
c10: 4 X2 + 3 X8 -18 X9 + 1 X10 + 2 X21 + 5 X32 + 1 X34 = 0;
c11: 5 X2 + 3 X8 + 1 X9 -18 X10 + 3 X22 + 6 X33 + 2 X34 = 0;
c12: 7 X3 -18 X11 + 5 X12 + 5 X13 + 4 X23 + 1 X35 + 1 X36 = 0;
c13: 7 X3 + 5 X11 -18 X12 + 1 X13 + 4 X24 + 3 X35 + 1 X37 = 0;
c14: 5 X3 + 4 X11 + 4 X12 -19 X13 + 2 X25 + 3 X36 + 3 X37 = 0;
c15: 7 X4 -19 X14 + 2 X15 + 2 X16 + 5 X26 + 2 X38 + 1 X39 = 0;
c16: 7 X4 + 6 X14 -18 X15 + 6 X16 + 4 X27 + 3 X38 + 1 X40 = 0;
c17: 4 X4 + 3 X14 + 3 X15 -19 X16 + 3 X28 + 4 X39 + 4 X40 = 0;
c18: 7 X5 -18 X17 + 2 X20 + 2 X29 + 1 X30 + 2 X41 + 3 X42 = 0;
c19: 4 X6 -18 X18 + 2 X23 + 1 X29 + 1 X31 + 2 X43 + 3 X44 = 0;
c20: 5 X7 -19 X19 + 3 X26 + 2 X30 + 1 X31 + 2 X45 + 2 X46 = 0;
c21: 2 X8 + 7 X17 -18 X20 + 2 X32 + 3 X33 + 2 X41 + 1 X42 = 0;
c22: 4 X9 -18 X21 + 2 X24 + 3 X32 + 1 X34 + 1 X47 + 5 X48 = 0;
c23: 4 X10 -18 X22 + 3 X27 + 3 X33 + 1 X34 + 6 X49 + 2 X50 = 0;
c24: 2 X11 + 4 X18 -18 X23 + 2 X35 + 3 X36 + 1 X43 + 1 X44 = 0;
c25: 2 X12 + 4 X21 -18 X24 + 5 X35 + 1 X37 + 1 X47 + 3 X48 = 0;
c26: 3 X13 -19 X25 + 2 X28 + 4 X36 + 4 X37 + 3 X51 + 3 X52 = 0;
c27: 3 X14 + 5 X19 -19 X26 + 2 X38 + 2 X39 + 2 X45 + 1 X46 = 0;
c28: 3 X15 + 4 X22 -18 X27 + 6 X38 + 2 X40 + 3 X49 + 1 X50 = 0;
c29: 2 X16 + 3 X25 -19 X28 + 3 X39 + 3 X40 + 4 X51 + 4 X52 = 0;
c30: 3 X5 + 5 X6 + 1 X17 + 2 X18 - 20 X29+ 1 X30 + 1 X31 + 2 X41 + 2 X43 + 3 X53 = 0;
c31: 3 X5 + 6 X7 + 2 X17 + 1 X19 + 1 X29 -21 X30 + 2 X31 + 2 X42 + 3 X45 + 2 X53 = 0;
c32: 4 X6 + 3 X7 + 1 X18 + 1 X19 + 2 X29 + 1 X30 -21 X31 + 2 X44 + 3 X46 + 2 X53 = 0;
c33: 1 X8 + 5 X9 + 3 X20 + 2 X21 - 20 X32+ 1 X33 + 3 X34 + 5 X41 + 2 X48 + 1 X54 = 0;
c34: 2 X8 + 2 X10 + 3 X20 + 5 X22 + 1 X32 -21 X33 + 2 X34 + 6 X42 +1 X49 + 2 X54 = 0;
c35: 4 X9 + 3 X10 + 1 X21 + 1 X22 + 2 X32 + 2 X33 -19 X34 + 2 X47 +1 X50 + 4 X54 = 0;

APPENDIX B: LINEAR EQUATIONS FOR FOUR STATIONS AND FIVE VEHICLES USING THE NEAREST RULE (CONTINUED)

c36: 2 X11 + 2 X12 + 5 X23 + 5 X24 -20 X35 + 1 X36 + 3 X37 + 3 X43 + 1 X48 + 1 X55 = 0;
c37: 1 X11 + 2 X13 + 4 X23 + 5 X25 + 2 X35 -21 X36 + 2 X37 + 3 X44 + 1 X51 + 1 X55 = 0;
c38: 1 X12 + 6 X13 + 4 X24 + 1 X25 + 2 X35 + 4 X36 -19 X37 + 3 X47 + 1 X52 + 2 X55 = 0;
c39: 1 X14 + 5 X15 + 6 X26 + 2 X27 -21 X38 + 2 X39 + 2 X40 + 3 X45 + 2 X49 + 1 X56 = 0;
c40: 1 X14 + 5 X16 + 3 X26 + 2 X28 + 1 X38 -21 X39 + 2 X40 + 4 X46 + 1 X51 + 2 X56 = 0;
c41: 1 X15 + 1 X16 + 3 X27 + 6 X28 + 2 X38 + 4 X39 -19 X40 + 4 X50 + 1 X52 + 2 X56 = 0;
c42: 3 X17 + 1 X20 + 5 X29 + 2 X32 -20 X41 + 1 X42 + 2 X43 + 2 X48 + 1 X53 + 3 X54 = 0;
c43: 3 X17 + 2 X20 + 6 X30 + 2 X33 + 1 X41 -21 X42 + 1 X45 + 3 X49 + 2 X53 + 2 X54 = 0;
c44: 5 X18 + 2 X23 + 3 X29 + 2 X35 + 1 X41 -20 X43 + 1 X44 + 2 X48 + 1 X53 + 3 X55 = 0;
c45: 4 X18 + 1 X23 + 3 X31 + 2 X36 + 2 X43 -21 X44 + 1 X46 + 3 X51 + 1 X53 + 2 X55 = 0;
c46: 6 X19 + 1 X26 + 3 X30 + 3 X38 + 2 X42 -21 X45 + 2 X46 + 2 X49 + 1 X53 + 2 X56 = 0;
c47: 3 X19 + 1 X26 + 4 X31 + 3 X39 + 1 X44 + 1 X45 -21 X46 + 2 X51 + 2 X53 + 2 X56 = 0;
c48: 4 X21 + 1 X24 + 3 X34 + 2 X37 -19 X47 + 2 X48 + 1 X50 + 1 X52 + 2 X54 + 4 X55 = 0;
c49: 5 X21 + 2 X24 + 1 X32 + 2 X35 + 3 X41 + 5 X43 + 3 X47 -20 X48 + 1 X54 + 1 X55 = 0;
c50: 2 X22 + 5 X27 + 2 X33 + 1 X38 + 3 X42 + 6 X45 -21 X49 + 2 X50 + 1 X54 + 2 X56 = 0;
c51: 3 X22 + 1 X27 + 4 X34 + 1 X40 + 1 X47 + 2 X49 -19 X50 + 2 X52 + 2 X54 + 4 X56 = 0;
c52: 2 X25 + 5 X28 + 1 X36 + 1 X39 + 4 X44 + 3 X46 -21 X51 + 2 X52 + 2 X55 + 1 X56 = 0;
c53: 6 X25 + 1 X28 + 1 X37 + 1 X40 + 4 X47 + 3 X50 + 4 X51 -19 X52 + 2 X55 + 2 X56 = 0;
c54: 2 X29 + 2 X30 + 4 X31 + 2 X41 + 1 X42 + 1 X43 + 2 X44 + 1 X45 + 1 X46 -23 X53 + 2 X54 + 2 X55 + 3 X56 = 0;
c55: 2 X32 + 1 X32 + 2 X34 + 2 X41 + 2 X42 + 2 X47 + 1 X48 + 1 X49 + 3 X50 + 4 X53 -23 X54 + 2 X55 + 1 X56 = 0;
c56: 1 X35 + 2 X36 + 2 X37 + 2 X43 + 4 X44 + 2 X47 + 2 X48 + 1 X51 + 3 X52 + 2 X53 + 1 X54 -23 55 + 1 X56 = 0;

APPENDIX C: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE NEAREST RULE

State	Possible Transitions (Nearest Rule)			
(4,0,0,0)	(3,1,0,0) [p12,p32,p42]	(3,0,1,0) [p13,p23,p43]	(3,0,0,1) [p14,p24,p34]	(4,0,0,0) [p21,p31,p41]
(0,4,0,0)	(0,4,0,0) [p12,p32,p42]	(0,3,1,0) [p13,p23,p43]	(0,3,0,1) [p14,p24,p34]	(1,3,0,0) [p21,p31,p41]
(0,0,4,0)	(0,1,3,0) [p12,p32,p42]	(0,0,4,0) [p13,p23,p43]	(0,0,3,1) [p14,p24,p34]	(1,0,3,0) [p21,p31,p41]
(0,0,0,4)	(0,1,0,3) [p12,p32,p42]	(0,0,1,3) [p13,p23,p43]	(0,0,0,4) [p14,p24,p34]	(1,0,0,3) [p21,p31,p41]
(3,1,0,0)	(2,2,0,0) [p12,p32,p42]	(2,1,1,0) [p13,p43]	(2,1,0,1) [p14,p34]	(4,0,0,0) [p21]
(3,0,1,0)	(2,1,1,0) [p12,p42]	(2,0,2,0) [p13,p23,p43]	(2,0,1,1) [p14,p24]	(3,0,1,0) [p21,p41]
(3,0,0,1)	(2,1,0,1) [p12,p32]	(2,0,1,1) [p13,p23]	(2,0,0,2) [p14,p24,p34]	(3,0,0,1) [p21,p31]
(1,3,0,0)	(0,4,0,0) [p12,p32,p42]	(0,3,1,0) [p13,p43]	(0,3,0,1) [p14,p34]	(2,2,0,0) [p21]
(0,3,1,0)	(0,3,1,0) [p12,p42]	(0,2,2,0) [p13,p23,p43]	(0,2,1,1) [p14,p24]	(1,2,1,0) [p21,p41]
(0,3,0,1)	(0,3,0,1) [p12,p34]	(0,2,1,1) [p13,p23]	(0,2,0,2) [p14,p24]	(1,2,0,1) [p21]
(1,0,3,0)	(0,1,3,0) [p12,p42]	(0,0,4,0) [p13,p23,p43]	(0,0,3,1) [p14,p24]	(1,0,2,1) [p34]
(0,1,3,0)	(0,1,3,0) [p12,p42]	(0,0,4,0) [p13,p23,p43]	(0,0,3,1) [p14,p24]	(1,0,3,0) [p21,p41]
(0,0,3,1)	(0,1,2,1) [p12,p32]	(0,0,3,1) [p13,p24]	(0,0,2,2) [p14,p34]	(1,0,3,0) [p21,p41]
(1,0,0,3)	(0,1,0,3) [p12,p32]	(0,0,1,3) [p13,p23]	(0,0,0,4) [p14,p24,p34]	(1,0,0,3) [p21,p31]
(0,1,0,3)	(0,1,0,3) [p12,p34]	(0,0,1,3) [p13,p23]	(0,0,0,4) [p14,p24]	(1,0,0,3) [p21]
(0,0,1,3)	(0,1,0,3) [p12,p32]	(0,0,1,3) [p13,p24]	(0,0,0,4) [p14,p34]	(1,0,1,2) [p21,p41]
(2,2,0,0)	(1,3,0,0) [p12,p32,p42]	(1,2,1,0) [p13,p43]	(1,2,0,1) [p14,p34]	(3,1,0,0) [p21]
(2,0,2,0)	(1,1,2,0) [p12,p42]	(1,0,3,0) [p13,p23,p43]	(1,0,2,1) [p14,p24]	(2,0,2,0) [p21,p41]
(2,0,0,2)	(1,1,0,2) [p12,p32]	(1,0,1,2) [p13,p23]	(1,0,0,3) [p14,p24,p34]	(2,0,0,2) [p21,p31]
(0,2,2,0)	(0,2,2,0) [p12,p42]	(0,1,3,0) [p13,p23,p43]	(0,1,2,1) [p14,p24]	(1,1,2,0) [p21,p41]
(0,2,0,2)	(0,2,0,2) [p12,p34]	(0,1,1,2) [p13,p23]	(0,1,0,3) [p14,p24]	(1,1,0,2) [p21]
(0,0,2,2)	(0,1,1,2) [p12,p32]	(0,0,2,2) [p13,p24]	(0,0,1,3) [p14,p34]	(1,0,2,1) [p21,p41]
(2,1,1,0)	(1,2,1,0) [p12,p42]	(1,1,2,0) [p13,p43]	(1,1,1,1) [p14]	(3,0,1,0) [p21]
	(2,1,1,0) [p41]			
(2,1,0,1)	(1,2,0,1) [p12,p32]	(1,1,1,1) [p13]	(1,1,0,2) [p14,p34]	(3,0,0,1) [p21]
	(2,1,1,0) [p43]			
(2,0,1,1)	(1,1,1,1) [p12]	(1,0,2,1) [p13,p23]	(1,0,1,2) [p14,p24]	(2,0,1,1) [p21]
	(2,0,2,0) [p43]			
(1,2,1,0)	(0,3,1,0) [p12,p42]	(0,2,2,0) [p13,p43]	(0,2,1,1) [p14]	(2,1,10) [p21]
	(1,2,1,0) [p41]			
(1,2,0,1)	(0,3,0,1) [p12,p32]	(0,2,1,1) [p13]	(0,2,0,2) [p14,p34]	(2,1,0,1) [p21]
	(1,2,1,0) [p43]			
(0,2,1,1)	(0,2,1,1) [p12]	(0,1,2,1) [p13,p23]	(0,1,1,2) [p14,p24]	(1,1,1,1) [p21]
	(0,2,2,0) [p43]			
(1,1,2,0)	(0,2,2,0) [p12,p42]	(0,1,3,0) [p13,p43]	(0,1,2,1) [p14]	(2,0,2,0) [p21]
	(1,1,2,0) [p41]			
(1,0,2,1)	(0,1,2,1) [p12]	(0,0,3,1) [p13,p23]	(0,0,2,2) [p14,p24]	(1,0,2,1) [p21]
	(1,0,3,0) [p43]			
(0,1,2,1)	(0,1,2,1) [p12,p21]	(1,0,2,1) [p21]	(0,0,2,2) [p14,p24]	(0,1,3,0) [p43]
	(0,2,2,0) [p42]			
(1,1,0,2)	(0,2,0,2) [p12,p32]	(0,1,1,2) [p13]	(0,1,0,3) [p14,p34]	(2,0,0,2) [p21]
	(1,1,1,1) [p43]			
(1,0,1,2)	(0,1,1,2) [p12]	(0,0,2,2) [p13,p23]	(0,0,1,3) [p14,p24]	(1,0,1,2) [p21]
	(1,0,2,1) [p43]			
(0,1,1,2)	(0,1,1,2) [p12]	(0,0,2,2) [p13,p23]	(0,0,1,3) [p14,p24]	(1,0,1,2) [p21]
	(0,1,2,1) [p43]			
(1,1,1,1)	(0,2,1,1) [p12]	(0,1,2,1) [p13]	(0,1,1,2) [p14]	(2,0,1,1) [p21]
	(2,1,1,0) [p41]	(1,2,1,0) [p42]	(1,1,2,0) [p43]	

APPENDIX C: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FOUR VEHICLES USING THE NEAREST RULE (CONTINUED)

State	Possible Transitions (Nearest Rule)				
(3,1,0,0)	(3,0,1,0) [p23]	(3,0,0,1) [p24]	(3,1,0,0) [p31,p41]		
(3,0,1,0)	(4,0,0,0) [p31]	(3,1,0,0) [p32]	(3,0,0,1) [p34]		
(3,0,0,1)	(4,0,0,0) [p41]	(3,1,0,0) [p42]	(3,0,1,0) [p43]		
(1,3,0,0)	(1,2,1,0) [p23]	(1,2,0,1) [p24]	(1,3,0,0) [p31,p41]		
(0,3,1,0)	(1,3,0,0) [p31]	(0,4,0,0) [p32]	(0,3,0,1) [p34]		
(0,3,0,1)	(1,3,0,0) [p31,p41]	(0,4,0,0) [p32,p42]	(0,3,1,0) [p43]		
(1,0,3,0)	(2,0,2,0) [p31]	(1,1,2,0) [p32]	(1,0,3,0) [p21,p41]		
(0,1,3,0)	(1,1,2,0) [p31]	(0,2,2,0) [p32]	(0,1,2,1) [p34]		
(0,0,3,1)	(0,0,4,0) [p23,p43]	(1,0,2,1) [p31]	(0,1,3,0) [p42]		
(1,0,0,3)	(1,1,0,2) [p42]	(2,0,0,2) [p41]	(1,0,1,2) [p43]		
(0,1,0,3)	(0,2,0,2) [p32,p42]	(1,1,0,2) [p31,p41]	(0,1,1,2) [p43]		
(0,0,1,3)	(0,0,2,2) [p23,p43]	(1,0,0,3) [p31]	(0,1,1,2) [p42]		
(2,2,0,0)	(2,1,1,0) [p23]	(2,1,0,1) [p24]	(2,2,0,0) [p31,p41]		
(2,0,2,0)	(3,0,1,0) [p31]	(2,1,1,0) [p32]	(2,0,1,1) [p34]		
(2,0,0,2)	(3,0,0,1) [p41]	(2,1,0,1) [p42]	(2,0,1,1) [p43]		
(0,2,2,0)	(1,2,1,0) [p31]	(0,3,1,0) [p32]	(0,2,1,1) [p34]		
(0,2,0,2)	(1,2,0,1) [p31,p41]	(0,3,0,1) [p32,p42]	(0,2,1,1) [p43]		
(0,0,2,2)	(0,0,3,1) [p23,p43]	(0,1,2,1) [p42]	(1,0,1,2) [p31]		
(2,1,1,0)	(2,0,2,0) [p23]	(2,0,1,1) [p24]	(3,1,0,0) [p31]	(2,2,0,0) [p32]	(2,1,0,1) [p34]
(2,1,0,1)	(2,0,1,1) [p23]	(2,0,0,2) [p24]	(2,1,0,1) [p31]	(3,1,0,0) [p41]	(2,2,0,0) [p42]
(2,0,1,1)	(3,0,0,1) [p31]	(2,1,0,1) [p32]	(2,0,0,2) [p34]	(3,0,1,0) [p41]	(2,1,1,0) [p42]
(1,2,1,0)	(1,1,2,0) [p23]	(1,1,1,1) [p24]	(2,2,0,0) [p31]	(1,3,0,0) [p32]	(1,2,0,1) [p34]
(1,2,0,1)	(1,1,1,1) [p23]	(1,1,0,2) [p24]	(1,2,0,1) [p31]	(2,2,0,0) [p41]	(1,3,0,0) [p42]
(0,2,1,1)	(1,2,0,1) [p31]	(0,3,0,1) [p32]	(0,2,0,2) [p34]	(1,2,1,0) [p41]	(0,3,1,0) [p42]
(1,1,2,0)	(1,0,3,0) [p23]	(1,0,2,1) [p24]	(2,1,1,0) [p31]	(1,2,10) [p32]	(1,1,1,1) [p34]
(1,0,2,1)	(2,0,1,1) [p31]	(1,1,1,1) [p32]	(1,0,1,2) [p34]	(2,0,2,0) [p41]	(1,1,2,0) [p42]
(0,1,2,1)	(0,0,3,1) [p23,p13]	(1,1,1,1) [p31]	(0,2,1,1) [p32]	(0,1,2,2) [p34]	(1,1,2,0) [p41]
(1,1,0,2)	(1,0,1,2) [p23]	(1,0,0,3) [p24]	(1,1,0,2) [p31]	(2,1,0,1) [p41]	(1,2,0,1) [p42]
(1,0,1,2)	(2,0,0,2) [p31]	(1,1,0,2) [p32]	(1,0,0,3) [p34]	(2,0,1,1) [p41]	(1,1,1,1) [p42]
(0,1,1,2)	(1,1,0,2) [p31]	(0,2,0,2) [p32]	(0,1,0,3) [p34]	(1,1,1,1) [p41]	(0,2,1,1) [p42]
(1,1,1,1)	(1,0,2,1) [p23]	(1,0,1,2) [p24]	(2,1,0,1) [p31]	(1,2,0,1) [p32]	(1,1,0,2) [p34]

APPENDIX D: LINEAR EQUATIONS FOR FOUR STATIONS AND FOUR VEHICLES USING THE NEAREST RULE

X4; (Objective Function)
(c1 to c35 are constraints)
c1: 2 X23 + 2 X24 + 4 X25 + 2 X26 + 1 X27 + 2 X28 + 1 X29
+ 2 X30 + 2 X31 + 1 X32 + 1 X33 + 3 X34 -23 X35 = 0;
c2: 16 X1 - 2 X5 - 2 X6 - 3 X7 = 0;
c3: 16 X2 - 7 X8 - 2 X9 - 3 X10 = 0;
c4: 19 X3 - 4 X11 - 4 X12 - 2 X13 = 0;
c5: 18 X4 - 5 X14 - 4 X15 - 3 X16 = 0;
c6: 7 X1 -18 X5 + 2 X6 + 1 X7 + 2 X17 + 2 X23 + 3 X24 = 0;
c7: 4 X1 + 1 X5 -18 X6 + 1 X7 + 2 X18 + 2 X23 + 3 X25 = 0;
c8: 5 X1 + 2 X5 + 1 X6 -19 X7 + 3 X19 + 2 X24 + 2 X25 = 0;
c9: 7 X2 -18 X8 + 2 X9 + 5 X10 + 7 X17 + 2 X26 + 1 X27 = 0;
c10: 4 X2 + 3 X8 -18 X9 + 1 X10 + 2 X20 + 5 X26 + 1 X28 = 0;
c11: 5 X2 + 3 X8 + 1 X9 -18 X10 + 3 X21 + 6 X27 + 2 X28 = 0;
c12: 7 X3 -18 X11 + 5 X12 + 5 X13 + 4 X18 + 1 X29 + 1 X30 = 0;
c13: 7 X3 + 5 X11 -18 X12 + 1 X13 + 4 X20 + 3 X29 + 1 X31 = 0;
c14: 5 X3 + 4 X11 + 4 X12 -19 X13 + 2 X22 + 3 X30 + 3 X31 = 0;
c15: 7 X4 -19 X14 + 2 X15 + 2 X16 + 5 X19 + 2 X32 + 1 X33 = 0;
c16: 7 X4 + 6 X14 -18 X15 + 6 X16 + 4 X21 + 3 X32 + 1 X34 = 0;
c17: 4 X4 + 3 X14 + 3 X15 -19 X16 + 3 X22 + 4 X33 + 4 X34 = 0;
c18: 7 X5 + 2 X8 -18 X17 + 2 X23 + 1 X24 + 2 X26 + 3 X27 = 0;
c19: 4 X6 + 2 X11 -18 X18 + 1 X23 + 1 X25 + 2 X29 + 3 X30 = 0;
c20: 5 X7 + 3 X14 -19 X19 + 2 X24 + 1 X25 + 2 X32 + 2 X33 = 0;
c21: 4 X9 + 2 X12 -18 X20 + 3 X26 + 1 X28 + 5 X29 + 1 X31 = 0;
c22: 4 X10 + 3 X15 -18 X21 + 3 X27 + 1 X28 + 6 X32 + 2 X34 = 0;
c23: 3 X13 + 2 X16 -19 X22 + 4 X30 + 4 X31 + 3 X33 + 3 X34 = 0;
c24: 3 X5 + 5 X6 + 1 X17 + 2 X18 -20 X23 + 1 X24 + 1 X25 + 2 X26 + 2 X29 + 3 X35 = 0;
c25: 3 X5 + 6 X7 + 2 X17 + 1 X19 + 1 X23 -21 X24 + 2 X25 + 2 X27 + 3 X32 + 2 X35 = 0;
c26: 4 X6 + 3 X7 + 1 X18 + 1 X19 + 2 X23 + 1 X24 -21 X25 + 2 X30 + 3 X33 + 2 X35 = 0;
c27: 1 X8 + 5 X9 + 3 X17 + 2 X20 + 5 X23 -20 X26 + 1 X27 + 3 X28 + 2 X29 + 1 X35 = 0;
c28: 2 X8 + 2 X10 + 3 X17 + 5 X21 + 6 X24 + 1 X26 -21 X27 + 2 X28+ 1 X32 + 2 X35 = 0;
c29: 4 X9 + 3 X10 + 1 X20 + 1 X21 + 2 X26 + 2 X27 -19 X28 + 2 X31+ 1 X34 + 4 X35 = 0;
c30: 2 X11 + 2 X12 + 5 X18 + 5 X20 + 3 X23 + 1 X26 -20 X29 + 1 X30 + 3 X31 + 1 X35 = 0;
c31: 1 X11 + 2 X13 + 4 X18 + 5 X22 + 3 X25 + 2 X29 -21 X21 + 2 X31+ 1 X33 + 1 X35 = 0;
c32: 1 X12 + 6 X13 + 4 X20 + 1 X22 + 3 X28 + 2 X29 + 4 X30 -19 X31 + 1 X34 + 2 X35 = 0;
c33: 1 X14 + 5 X15 + 6 X19 + 2 X21 + 3 X24 + 2 X27 -21 X32 + 2 X33 + 2 X34 + 1 X35 = 0;
c34: 1 X14 + 5 X16 + 3 X19 + 2 X22 + 4 X25 + 1 X30 + 1 X32 -21 X33 + 2 X34 + 2 X35 = 0;
c35: X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16+X17+X18
+X19+X20+X21+X22+X23+X24+X25+X26+X27+X28+X29+X30+X31+X32+X33+X34
+X35=1;

APPENDIX E: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND THREE VEHICLES USING THE NEAREST RULE

	Possible Transitions (Nearest Rule)			
(3,0,0,0)	(2,1,0,0) [p12,p32,p42]	(2,0,1,0) [p13,p23,p43]	(2,0,0,1) [p14,p24,p34]	(3,0,0,0) [p21,p31,p41]
(0,3,0,0)	(0,3,0,0) [p12,p32,p42]	(0,2,1,0) [p13,p23,p43]	(0,2,0,1) [p14,p24,p34]	(1,2,0,0) [p21,p31,p41]
(0,0,3,0)	(0,1,2,0) [p12,p32,p42]	(0,0,3,0) [p13,p23,p43]	(0,0,2,1) [p14,p24,p34]	(1,0,2,0) [p21,p31,p41]
(0,0,0,3)	(0,1,0,2) [p12,p32,p42]	(0,0,1,2) [p13,p23,p43]	(0,0,0,3) [p14,p24,p34]	(1,0,0,2) [p21,p31,p41]
(2,1,0,0)	(1,2,0,0) [p12,p32,p42]	(1,1,1,0) [p13,p43]	(1,1,0,1) [p14,p34]	(2,0,0,0) [p21]
	(2,0,1,0) [p23]	(2,0,0,1) [p24]	(3,1,0,0) [p31,p41]	
(2,0,1,0)	(1,1,1,0) [p12,p42]	(1,0,2,0) [p13,p23,p43]	(1,0,1,1) [p14,p24]	(2,0,1,0) [p21,p41]
	(3,0,0,0) [p31]	(2,1,0,0) [p32]	(3,0,0,1) [p34]	
(2,0,0,1)	(1,1,0,1) [p12,p32]	(1,0,1,1) [p13,p23]	(1,0,0,2) [p14,p24,p34]	(2,0,0,1) [p21,p31]
	(3,0,0,0) [p41]	(2,1,0,0) [p42]	(3,0,1,0) [p43]	
(1,2,0,0)	(0,3,0,0) [p12,p32,p42]	(0,2,1,0) [p13,p43]	(0,2,0,1) [p14,p34]	(2,1,0,0) [p21]
	(1,1,1,0) [p23]	(1,1,0,1) [p24]	(1,3,0,0) [p31,p41]	
(0,2,1,0)	(0,2,1,0) [p12,p42]	(0,1,2,0) [p13,p23,p43]	(0,1,1,1) [p14,p24]	(1,1,1,0) [p21,p41]
	(1,2,0,0) [p31]	(0,3,0,0) [p32]	(0,3,0,1) [p34]	
(0,2,0,1)	(0,2,0,1) [p12,p34]	(0,1,1,1) [p13,p23]	(0,1,0,2) [p14,p24]	(1,1,0,1) [p21]
	(1,2,0,0) [p31,p41]	(0,3,0,0) [p32,p42]	(0,3,1,0) [p43]	
(1,0,2,0)	(0,1,2,0) [p12,p42]	(0,0,3,0) [p13,p23,p43]	(0,0,2,1) [p14,p24]	(1,0,1,1) [p34]
	(2,0,1,0) [p31]	(1,1,1,0) [p32]	(1,0,2,0) [p21,p41]	
(0,1,2,0)	(0,1,2,0) [p12,p42]	(0,0,3,0) [p13,p23,p43]	(0,0,2,1) [p14,p24]	(1,0,2,0) [p21,p41]
	(1,1,1,0) [p31]	(0,2,1,0) [p32]	(0,1,1,1) [p34]	
(0,0,2,1)	(0,1,1,1) [p12,p32]	(0,0,2,1) [p13,p24]	(0,0,1,2) [p14,p34]	(1,0,2,0) [p21,p41]
	(0,0,3,0) [p23,p43]	(1,0,1,1) [p31]	(0,1,2,0) [p42]	
(1,0,0,2)	(0,1,0,2) [p12,p32]	(0,0,1,2) [p13,p23]	(0,0,0,3) [p14,p24,p34]	(1,0,0,2) [p21,p31]
	(1,1,0,1) [p42]	(2,0,0,1) [p41]	(1,0,1,1) [p43]	
(0,1,0,2)	(0,1,0,2) [p12,p34]	(0,0,1,2) [p13,p23]	(0,0,0,3) [p14,p24]	(1,0,0,2) [p21]
	(0,2,0,1) [p32,p42]	(1,1,0,1) [p31,p41]	(0,1,1,1) [p43]	
(0,0,1,2)	(0,1,0,2) [p12,p32]	(0,0,1,2) [p13,p24]	(0,0,0,3) [p14,p34]	(1,0,1,1) [p21,p41]
	(0,0,2,1) [p23,p43]	(1,0,0,2) [p31]	(0,1,1,1) [p42]	
(1,1,1,0)	(0,2,1,0) [p12,p42]	(0,1,2,0) [p13,p43]	(0,1,1,1) [p14]	(2,0,1,0) [p21]
	(1,1,1,0) [p41]	(1,0,2,0) [p23]	(1,0,1,1) [p24]	(2,1,0,0) [p31]
	(1,2,0,0) [p32]	(1,1,0,1) [p34]		
(1,1,0,1)	(0,2,0,1) [p12,p32]	(0,1,1,1) [p13]	(0,1,0,2) [p14,p34]	(2,0,0,1) [p21]
	(1,1,1,0) [p43]	(1,0,1,1) [p23]	(1,0,0,2) [p24]	(1,1,0,1) [p31]
	(2,1,0,0) [p41]	(1,2,0,0) [p42]		
(1,0,1,1)	(0,1,1,1) [p12]	(0,0,2,1) [p13,p23]	(0,0,1,2) [p14,p24]	(1,0,1,1) [p21]
	(1,0,2,0) [p43]	(2,0,0,1) [p31]	(1,1,0,1) [p32]	(1,0,0,2) [p34]
	(2,0,1,0) [p41]	(1,1,1,0) [p42]		
(0,1,1,1)	(0,1,1,1) [p12]	(0,0,2,1) [p13,p23]	(0,0,1,2) [p14,p24]	(1,0,1,1) [p21]
	(1,1,0,1) [p31]	(0,2,0,1) [p32]	(0,1,0,2) [p34]	(0,,1,2,0) [p43]
	(1,1,1,0) [p41]	(0,2,1,0) [p42]		

APPENDIX F: STATIONARY PROBABILITIES OF THE MARKOV CHAIN FOR FOUR STATIONS AND THREE VEHICLES USING THE NEAREST DISPATCHING RULE

State	S1	S2	S3	S4	S5	S6	S7
Probability	0.0125	0.0484	0.0281	0.0424	0.0331	0.0263	0.0274

State	S8	S9	S10	S11	S12	S13	S14
Probability	0.0663	0.0508	0.0699	0.0528	0.0531	0.0557	0.0475

State	S15	S16	S17	S18	S19	S20
Probability	0.0835	0.0640	0.0546	0.0619	0.0493	0.0725

APPENDIX G: LINEAR EQUATIONS FOR FOUR STATIONS AND THREE VEHICLES USING THE NEAREST RULE

X4 (Objective Function);
(c1 to c20 are constraints)
c1: X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10
+ X11 + X12 + X13 + X14 + X15 + X16 +X17 + X18 + X19 + X20 = 1;
c2: 16 X1 - 2 X5 - 2 X6 - 3 X7 = 0;
c3: 16 X2 - 7 X8 - 2 X9 - 3 X10 = 0;
c4: 19 X3 - 4 X11 - 4 X12 - 2 X13 = 0;
c5: 18 X4 - 5 X14 - 4 X15 - 3 X16 = 0;
c6: 7 X1 -18 X5 + 2 X6 + 1 X7 + 2 X8 + 2 X17 + 3 X18 = 0;
c7: 4 X1 + 1 X5 -18 X6 + 1 X7 + 2 X11 + 2 X17 + 3 X19 = 0;
c8: 5 X1 + 2 X5 + 1 X6 -19 X7 + 3 X14 + 2 X18 + 2 X19 = 0;
c9: 7 X2 + 7 X5 -18 X8 + 2 X9 + 5 X10 + 2 X17 + 1 X18 = 0;
c10: 4 X2 + 3 X8 -18 X9 + 1 X10 + 2 X12 + 5 X17 + 1 X20 = 0;
c11: 5 X2 + 3 X8 + 1 X9 -18 X10 + 3 X15 + 6 X18 + 2 X20 = 0;
c12: 7 X3 + 4 X6 -18 X11 + 5 X12 + 5 X13 + 1 X17 + 1 X19 = 0;
c13: 7 X3 + 4 X9 + 5 X11 -18 X12 + 1 X13 + 3 X17 + 1 X20 = 0;
c14: 5 X3 + 4 X11 + 4 X12 -19 X13 + 2 X16 + 3 X19 + 3 X20 = 0;
c15: 7 X4 + 5 X7 -19 X14 + 2 X15 + 2 X16 + 2 X18 + 1 X19 = 0;
c16: 7 X4 + 4 X10 + 6 X14 -18 X15 + 6 X16 + 3 X18 + 1 X20 = 0;
c17: 4 X4 + 3 X13 + 3 X14 + 3 X15 -19 X16 + 4 X19 + 4 X20 = 0;
c18: 3 X5 + 5 X6 + 1 X8 + 5 X9 + 2 X11 + 2 X12 -20 X17 + 1 X18 + 1 X19 +3 X20= 0;
c19: 3 X5 + 6 X7 + 2 X8 + 2 X10 + 1 X14 + 5 X15 + 1 X17 -21 X18 + 2 X19 + 2 X20 = 0;
c20: 4 X6 + 3 X7 + 1 X11 + 2 X13 + 1 X14 + 5 X16 + 2 X17 + 1 X18 - 21 X19 + 2 X20 = 0;

APPENDIX H: PROBABILITY DISTRIBUTION OF THE LOAD REQUESTS FOR FOUR STATIONS AND THREE VEHICLES USING THE NEAREST DISPATCHING RULE

n (# of Load Requests)	**0**	**1**	**2**	**3**	**4**	**5**
P(n)	0.1027	0.2109	0.2165	0.1482	0.1015	0.0695
n (# of Load Requests)	**6**	**7**	**8**	**9**	**10**	**11**
P(n)	0.0475	0.0326	0.0223	0.0153	0.0104	0.0072
n (# of Load Requests)	**12**	**13**	**14**	**15**	**16**	**17**
P(n)	0.0049	0.0034	0.0023	0.0016	0.0011	0.0007

APPENDIX I: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE FARTHEST RULE

	Possible Transitions (Farthest Rule)			
(5,0,0,0)	(4,1,0,0) [p12,p32,p42]	(4,0,1,0) [p13,p23,p43]	(4,0,0,1) [p14,p24,p34]	(5,0,0,0) [p21,p31,p41]
(0,5,0,0)	(1,4,0,0) [p21,p31,p41]	(0,4,1,0) [p23,p13,p43]	(0,4,0,1) [p24,p14,p34]	(0,5,0,0) [p12,p32,p42]
(0,0,5,0)	(1,0,4,0) [p31,p21,p41]	(0,1,4,0) [p32,p12,p42]	(0,0,4,1) [p34,p14,p24]	(0,0,5,0) [p13,p23,p43]
(0,0,0,5)	(1,0,0,4) [p41,p21,p31]	(0,1,0,4) [p42,p12,p32]	(0,0,1,4) [p43,p13,p23]	(0,0,0,5) [p14,p24,p34]
(4,1,0,0)	(3,2,0,0) [p12,p32]	(3,1,1,0) [p13]	(3,1,0,1) [p14]	(5,0,0,0) [p21,p31,p41]
(4,0,1,0)	(3,1,1,0) [p12]	(3,0,2,0) [p13]	(3,0,1,1) [p14]	(5,0,0,0) [p21,p31,p41]
(4,0,0,1)	(3,1,0,1) [p12]	(3,0,1,1) [p13]	(3,0,0,2) [p14]	(5,0,0,0) [p21,p31,p41]
(1,4,0,0)	(0,5,0,0) [p12]	(0,4,1,0) [p13]	(0,4,0,1) [p14]	(2,3,0,0) [p21,p31,p41]
(0,4,1,0)	(0,5,0,0) [p12,p32,p42]	(0,4,1,0) [p13,p43]	(0,4,0,1) [p14,p34]	(1,3,1,0) [p21]
(0,4,0,1)	(0,5,0,0) [p12,p42]	(0,4,1,0) [p13,p43]	(0,4,0,1) [p14,p32]	(1,3,0,1) [p21,p31]
(1,0,4,0)	(0,1,4,0) [p12]	(0,0,5,0) [p13]	(0,0,4,1) [p14]	(2,0,3,0) [p21,p31,p41]
(0,1,4,0)	(0,2,3,0) [p12,p32,p42]	(0,1,4,0) [p13,p43]	(0,1,3,1) [p14,p34]	(1,0,4,0) [p21]
(0,0,4,1)	(0,1,4,0) [p12,p42]	(0,0,5,0) [p13,p43]	(0,0,4,1) [p14,p23]	(1,0,3,1) [p21,p31]
(1,0,0,4)	(0,1,0,4) [p12]	(0,0,1,4) [p13]	(0,0,0,5) [p14]	(2,0,0,3) [p21,p31,p41]
(0,1,0,4)	(0,2,0,3) [p12,p42]	(0,1,1,3) [p13,p43]	(0,1,0,4) [p14,p32]	(1,0,0,4) [p21,p31]
(0,0,1,4)	(0,1,1,3) [p12,p42]	(0,0,2,3) [p13,p43]	(0,0,1,4) [p14,p23]	(1,0,0,4) [p21,p31]
(3,2,0,0)	(2,3,0,0) [p12]	(2,2,1,0) [p13]	(2,2,0,1) [p14]	(4,1,0,0) [p21,p31,p41]
(3,0,2,0)	(2,1,2,0) [p12]	(2,0,3,0) [p13]	(2,0,2,1) [p14]	(4,0,1,0) [p21,p31,p41]
(3,0,0,2)	(2,1,0,2) [p12]	(2,0,1,2) [p13]	(2,0,0,3) [p14]	(4,0,0,1) [p21,p31,p41]
(2,3,0,0)	(1,4,0,0) [p12]	(1,3,1,0) [p13]	(1,3,0,1) [p14]	(3,2,0,0) [p21,p31,p41]
(0,3,2,0)	(0,4,1,0) [p12,p32,p42]	(0,3,2,0) [p13,p43]	(0,3,1,1) [p14,p34]	(1,2,2,0) [p21]
(0,3,0,2)	(0,4,0,1) [p12,p42]	(0,3,1,1) [p13,p43]	(0,3,0,2) [p14,p32]	(1,2,0,2) [p21,p31]
(2,0,3,0)	(1,1,3,0) [p12]	(1,0,4,0) [p13]	(1,0,3,1) [p14]	(3,0,2,0) [p21,p31,p41]
(0,2,3,0)	(0,3,2,0) [p12,p32,p42]	(0,2,3,0) [p13,p43]	(0,2,2,1) [p14,p34]	(1,1,3,0) [p21]
(0,0,3,2)	(0,1,3,1) [p12,p42]	(0,0,4,1) [p13,p43]	(0,0,3,2) [p14,p23]	(1,0,2,2) [p21,p31]
(2,0,0,3)	(1,1,0,3) [p12]	(1,0,1,3) [p13]	(1,0,0,4) [p14]	(3,0,0,2) [p21,p31,p41]
(0,2,0,3)	(0,3,0,2) [p12,p42]	(0,2,1,2) [p13,p43]	(0,2,0,3) [p14,p32]	(1,1,0,3) [p21,p31]
(0,0,2,3)	(0,1,2,2) [p12,p42]	(0,0,3,2) [p13,p43]	(0,0,2,3) [p14,p23]	(0,1,1,3) [p32]
(3,1,1,0)	(2,2,1,0) [p12]	(2,1,2,0) [p13]	(2,1,1,1) [p14]	(4,0,1,0) [p21]
	(3,1,0,1) [p34]	(3,1,1,0) [p43]		
(3,1,0,1)	(2,2,0,1) [p12]	(2,1,1,1) [p13]	(2,1,0,2) [p14]	(4,0,0,1) [p21,p31]
	(3,2,0,0) [p42]	(3,1,1,0) [p43]		
(3,0,1,1)	(2,1,1,1) [p12]	(2,0,2,1) [p13]	(2,0,1,2) [p14]	(4,0,0,1) [p21,p31]
	(3,1,1,0) [p42]	(3,0,2,0) [p43]		
(1,3,1,0)	(0,4,1,0) [p12]	(0,3,2,0) [p13]	(0,3,1,1) [p14]	(2,2,1,0) [p21]
	(1,3,0,1) [p34]	(1,3,1,0) [p43]		
(1,3,0,1)	(0,4,0,1) [p12]	(0,3,1,1) [p13]	(0,3,0,2) [p14]	(2,2,0,1) [p21,p31]
	(1,4,0,0) [p42]	(1,3,1,0) [p43]		
(0,3,1,1)	(0,4,1,0) [p12,p42]	(0,3,2,0) [p13,p43]	(0,3,1,1) [p14]	(1,2,1,1) [p21]
	(0,3,0,2) [p34]	(1,3,1,0) [p41]		
(1,1,3,0)	(0,2,3,0) [p12]	(0,1,4,0) [p13]	(0,1,3,1) [p14]	(2,0,3,0) [p21]
	(1,1,2,1) [p34]	(1,1,3,0) [p43]		
(1,0,3,1)	(0,1,3,1) [p12]	(0,0,4,1) [p13]	(0,0,3,2) [p14]	(2,0,2,1) [p21,p31]
	(1,1,3,0) [p42]	(1,0,4,0) [p43]		
(0,1,3,1)	(0,2,3,0) [p12,p42]	(0,1,4,0) [p13,p43]	(0,1,3,1) [p14]	(1,0,3,1) [p21]
	(0,1,2,2) [p34]	(1,1,3,0) [p41]		
(1,1,0,3)	(0,2,0,3) [p12]	(0,1,1,3) [p13]	(0,1,0,4) [p14]	(2,0,0,3) [p21,p31]
	(1,2,0,2) [p42]	(1,1,1,2) [p43]		
(1,0,1,3)	(0,1,1,3) [p12]	(0,0,2,3) [p13]	(0,0,1,4) [p14]	(2,0,0,3) [p21,p31]
	(1,1,1,2) [p42]	(1,0,2,2) [p43]		
(0,1,1,3)	(0,2,1,2) [p12,p42]	(0,1,2,2) [p13,p43]	(0,1,1,3) [p14]	(1,0,1,3) [p21]
	(0,1,0,4) [p34]	(1,1,1,2) [p41]		

APPENDIX I: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE FARTHEST RULE (CONTINUED)

State	Possible Transitions (Farthest Rule)			
(2,2,1,0)	(1,3,1,0) [p12]	(1,2,2,0) [p13]	(1,2,1,1) [p14]	(3,1,1,0) [p21]
	(2,2,0,1) [p34]	(2,2,1,0) [p43]		
(2,2,0,1)	(1,3,0,1) [p12]	(1,2,1,1) [p13]	(1,2,0,2) [p14]	(3,1,0,1) [p21,p31]
	(2,3,0,0) [p42]	(2,2,1,0) [p43]		
(2,1,2,0)	(1,2,2,0) [p12]	(1,1,3,0) [p13]	(1,1,2,1) [p14]	(3,0,2,0) [p21]
	(2,1,1,1) [p34]	(2,1,2,0) [p43]		
(2,0,2,1)	(1,1,2,1) [p12]	(1,0,3,1) [p13]	(1,0,2,2) [p14]	(3,0,1,1) [p21,p31]
	(2,1,2,0) [p42]	(2,0,3,0) [p43]		
(2,1,0,2)	(1,2,0,2) [p12]	(1,1,1,2) [p13]	(1,1,0,3) [p14]	(3,0,0,2) [p21,p31]
	(2,2,0,1) [p42]	(2,1,1,1) [p43]		
(2,0,1,2)	(1,1,1,2) [p12]	(1,0,2,2) [p13]	(1,0,1,3) [p14]	(3,0,0,2) [p21,p31]
	(2,1,1,1) [p42]	(2,0,2,1) [p43]		
(0,2,2,1)	(0,3,2,0) [p12,p42]	(0,2,3,0) [p13,p43]	(0,2,2,1) [p14]	(1,1,2,1) [p21]
	(0,2,1,2) [p34]	(1,2,2,0) [p41]		
(1,2,2,0)	(0,3,2,0) [p12]	(0,2,3,0) [p13]	(0,2,2,1) [p14]	(2,1,2,0) [p21]
	(1,2,1,1) [p34]	(1,2,2,0) [p43]		
(1,2,0,2)	(0,3,0,2) [p12]	(0,2,1,2) [p13]	(0,2,0,3) [p14]	(2,1,0,2) [p21,p31]
	(1,3,0,1) [p42]	(1,2,1,1) [p43]		
(0,2,1,2)	(0,3,1,1) [p12,p42]	(0,2,2,1) [p13,p43]	(0,2,1,2) [p14]	(1,1,1,2) [p21]
	(0,2,0,3) [p34]	(1,2,1,1) [p41]		
(1,0,2,2)	(0,1,2,2) [p12]	(0,0,3,2) [p13]	(0,0,2,3) [p14]	(2,0,1,2) [p21,p31]
	(1,1,2,1) [p42]	(1,0,3,1) [p43]		
(0,1,2,2)	(0,2,2,1) [p12,p42]	(0,1,3,1) [p13,p43]	(0,1,2,2) [p14]	(1,0,2,2) [p21]
	(0,1,1,3) [p34]	(1,1,2,1) [p41]		
(2,1,1,1)	(1,2,1,1) [p12]	(1,1,2,1) [p13]	(1,1,1,2) [p14]	(3,0,1,1) [p21]
	(3,1,1,0) [p41]	(2,2,1,0) [p42]	(2,1,2,0) [p43]	(2,1,0,2) [p34]
(1,2,1,1)	(0,3,1,1) [p12]	(0,2,2,1) [p13]	(0,2,1,2) [p14]	(2,1,1,1) [p21]
	(2,2,1,0) [p41]	(1,3,1,0) [p42]	(1,2,2,0) [p43]	(1,2,0,2) [p34]
(1,1,2,1)	(0,2,2,1) [p12]	(0,1,3,1) [p13]	(0,1,2,2) [p14]	(2,0,2,1) [p21]
	(2,1,2,0) [p41]	(1,2,2,0) [p42]	(1,1,3,0) [p43]	(1,1,1,2) [p34]
(1,1,1,2)	(0,2,1,2) [p12]	(0,1,2,2) [p13]	(0,1,1,3) [p14]	(2,0,1,2) [p21]
	(2,1,1,1) [p41]	(1,2,1,1) [p42]	(1,1,2,1) [p43]	(1,1,0,3) [p34]

APPENDIX I: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FIVE VEHICLES USING THE FARTHEST RULE (CONTINUED)

State	Possible Transitions (Farthest Rule)			
(4,1,0,0)	(4,0,1,0) [p23,p43]	(4,0,0,1) [p24,p34]	(4,1,0,0) [p42]	
(4,0,1,0)	(4,0,1,0) [p23,p43]	(4,0,0,1) [p24,p34]	(4,1,0,0) (p32,p42]	
(4,0,0,1)	(4,0,1,0) [p23,p43]	(4,0,0,1) [p24,p34]	(4,1,0,0) [p32,p42]	
(1,4,0,0)	(1,3,1,0) [p23,p43]	(1,3,0,1) [p24,p34]	(1,4,0,0) [p32,p42]	
(0,4,1,0)	(0,3,2,0) [p23]	(0,3,1,1) [p24]	(1,4,0,0) [p31,p41]	
(0,4,0,1)	(0,3,1,1) [p23]	(0,3,0,2) [p24,p34]	(1,4,0,0) [p41]	
(1,0,4,0)	(1,0,4,0) [p23,p43]	(1,0,3,1) [p24,p34]	(1,1,3,0) [p32,p42]	
(0,1,4,0)	(0,0,5,0) [p23]	(0,0,4,1) [p24]	(1,1,3,0) [p31,p41]	
(0,0,4,1)	(0,0,3,2) [p24,p34]	(0,1,3,1) [p32]	(1,0,4,0) [p41]	
(1,0,0,4)	(1,0,1,3) [p23,p43]	(1,0,0,4) [p24,p34]	(1,1,0,3) [p32,p42]	
(0,1,0,4)	(0,0,1,4) [p23]	(0,0,0,5) [p24,p34]	(1,1,0,3) [p41]	
(0,0,1,4)	(0,0,0,5) [p24,p34]	(0,1,0,4) [p32]	(1,0,1,3) [p41]	
(3,2,0,0)	(3,1,1,0) [p23,p43]	(3,1,0,1) [p24,p34]	(3,2,0,0) [p32,p42]	
(3,0,2,0)	(3,0,2,0) [p23,p43]	(3,0,1,1) [p24,p34]	(3,1,10) [p32,p42]	
(3,0,0,2)	(3,0,1,1) [p23,p43]	(3,0,0,2) [p24,p34]	(3,1,0,1) [p32,p42]	
(2,3,0,0)	(2,2,1,0) [p23,p43]	(2,2,0,1) [p24,p34]	(2,3,0,0) [p32,p42]	
(0,3,2,0)	(0,2,3,0) [p23]	(0,2,2,1) [p24]	(1,3,1,0) [p31,p41]	
(0,3,0,2)	(0,2,1,2) [p23]	(0,2,0,3) [p24,p34]	(1,3,0,1) [p41]	
(2,0,3,0)	(2,0,3,0) [p23,p43]	(2,0,2,1) [p24,p34]	(2,1,2,0) [p32,p42]	
(0,2,3,0)	(0,1,4,0) [p23]	(0,1,3,1) [p24]	(1,2,2,0) [p31,p41]	
(0,0,3,2)	(0,0,2,3) [p24,p34]	(0,1,2,2) [p32]	(1,0,3,1) [p41]	
(2,0,0,3)	(2,0,1,2) [p23,p43]	(2,0,0,3) [p24,p34]	(2,1,0,2) [p32,p42]	
(0,2,0,3)	(0,1,1,3) [p23]	(0,1,0,4) [p24,p34]	(1,2,0,2) [p41]	
(0,0,2,3)	(0,0,1,4) [p24,p34]	(1,0,1,3) [p31,p21]	(1,0,2,2) [p41]	
(3,1,1,0)	(3,0,2,0) [p23]	(3,0,1,1) [p24]	(4,1,0,0) [p31,p41]	(3,2,0,0) [p32,p42]
(3,1,0,1)	(3,0,1,1) [p23]	(3,0,0,2) [p24,p34]	(3,1,0,1) [p32]	(4,1,0,0) [p41]
(3,0,1,1)	(3,0,1,1) [p23]	(3,0,0,2) [p24,p34]	(3,1,0,1) [p32]	(4,0,1,0) [p41]
(1,3,1,0)	(1,2,2,0) [p23]	(1,2,1,1) [p24]	(2,3,0,0) [p31,p41]	(1,4,0,0) [p32,p42]
(1,3,0,1)	(1,2,1,1) [p23]	(1,2,0,2) [p24,p34]	(1,3,0,1) [p32]	(2,3,0,0) [p41]
(0,3,1,1)	(0,2,2,1) [p23]	(0,2,1,2) [p24]	(1,3,0,1) [p31]	(0,4,0,1) [p32]
(1,1,3,0)	(1,0,4,0) [p23]	(1,0,3,1) [p24]	(2,1,2,0) [p31,p41]	(1,2,2,0) [p32,p42]
(1,0,3,1)	(1,0,3,1) [p23]	(1,0,2,2) [p24,p34]	(1,1,2,1) [p32]	(2,0,3,0) [p41]
(0,1,3,1)	(0,0,4,1) [p23]	(0,0,3,2) [p24]	(1,1,2,1) [p31]	(0,2,2,1) [p32]
(1,1,0,3)	(1,0,1,3) [p23]	(1,0,0,4) [p24,p34]	(1,1,0,3) [p32]	(2,1,0,2) [p41]
(1,0,1,3)	(1,0,1,3) [p23]	(1,0,0,4) [p24,p34]	(1,1,0,3) [p32]	(2,0,1,2) [p41]
(0,1,1,3)	(0,0,2,3) [p23]	(0,0,1,4) [p24]	(1,1,0,3) [p31]	(0,2,0,3) [p32]
(2,2,1,0)	(2,1,2,0) [p23]	(2,1,1,1) [p24]	(3,2,0,0) [p31,p41]	(2,3,0,0) [p32,p42]
(2,2,0,1)	(2,1,1,1) [p23]	(2,1,0,2) [p24,p34]	(2,2,0,1) [p32]	(3,2,0,0) [p41]
(2,1,2,0)	(2,0,3,0) [p23]	(2,0,2,1) [p24]	(3,1,1,0) [p31,p41]	(2,2,1,0) [p32,p42]
(2,0,2,1)	(2,0,2,1) [p23]	(2,0,1,2) [p24,p34]	(2,1,1,1) [p32]	(3,0,2,0) [p41]
(2,1,0,2)	(2,0,1,2) [p23]	(2,0,0,3) [p24,p34]	(2,1,0,2) [p32]	(3,1,0,1) [p41]
(2,0,1,2)	(2,0,1,2) [p23]	(2,0,0,3) [p24,p34]	(2,1,0,2) [p32]	(3,0,1,1) [p41]
(0,2,2,1)	(0,1,3,1) [p23]	(0,1,2,2) [p24]	(1,2,1,1) [p31]	(0,3,1,1) [p32]
(1,2,2,0)	(1,1,3,0) [p23]	(1,1,2,1) [p24]	(2,2,1,0) [p31,p41]	(1,3,1,0) [p32,p42]
(1,2,0,2)	(1,1,1,2) [p23]	(1,1,0,3) [p24,p34]	(1,2,0,2) [p32]	(2,2,0,1) [p41]
(0,2,1,2)	(0,1,2,2) [p23]	(0,1,1,3) [p24]	(1,2,0,2) [p31]	(0,3,0,2) [p32]
(1,0,2,2)	(1,0,2,2) [p23]	(1,0,1,3) [p24,p34]	(1,1,1,2) [p32]	(2,0,2,1) [p41]
(0,1,2,2)	(0,0,3,2) [p23]	(0,0,2,3) [p24]	(1,1,1,2) [p31]	(0,2,1,2) [p32]
(2,1,1,1)	(2,0,2,1) [p23]	(2,0,1,2) [p24]	(3,1,0,1) [p31]	(2,2,0,1) [p32]
(1,2,1,1)	(1,1,2,1) [p23]	(1,1,1,2) [p24]	(2,2,0,1) [p31]	(1,3,0,1) [p32]
(1,1,2,1)	(1,0,3,1) [p23]	(1,0,2,2) [p24]	(2,1,1,1) [p31]	(1,2,1,1) [p32]
(1,1,1,2)	(1,0,2,2) [p23]	(1,0,1,3) [p24]	(2,1,0,2) [p31]	(1,2,0,2) [p32]

APPENDIX J: LINEAR EQUATIONS FOR FOUR STATIONS AND FIVE VEHICLES USING THE FARTHEST RULE

X50; (Objective Function)

(c1 to c56 are constraints)

c1: X1 + X2 + X3+ X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13
+ X14 + X15 + X16 + X17 + X18 + X19 + X20 + X21 + X22 + X23 + X24
+ X25 + X26 + X27 + X28 + X29 + X30 + X31 + X32 + X33 + X34 + X35
+ X36 + X37 + X38 + X39 + X40 + X41 + X42 + X43 + X44 + X45 + X46
+ X47 + X48 + X49 + X50 + X51 + X52 + X53 + X54 + X55 + X56 = 1;

c2: 16 X1 - 7 X5 - 7 X6 - 7 X7 = 0;

c3: 16 X2 - 4 X8 - 7 X9 - 5 X10 = 0;

c4: 19 X3 - 2 X11 - 1 X12 - 3 X13 = 0;

c5: 18 X4 - 2 X14 - 3 X15 - 3 X16 = 0;

c6: 7 X1 -22 X5 + 3 X6 + 3 X7 + 7 X17 + 5 X29 + 3 X30 = 0;

c7: 4 X1 + 2 X5 -21 X6 + 2 X7 + 7 X18 + 2 X29 + 3 X31 = 0;

c8: 5 X1 + 3 X5 + 3 X6 -20 X7 + 7 X19 + 4 X30 + 4 X31 = 0;

c9: 7 X2 -20 X8 + 5 X9 + 3 X10 + 4 X20 + 3 X32 + 1 X33 = 0;

c10: 4 X2 + 2 X8 -20 X9 + 3 X10 + 7 X21 + 4 X32 + 5 X34 = 0;

c11: 5 X2 + 2 X8 + 3 X9 -19 X10 + 5 X22 + 4 X33 + 2 X34 = 0;

c12: 7 X3 -21 X11 + 2 X12 + 3 X13 + 2 X23 + 1 X35 + 1 X36 = 0;

c13: 7 X3 + 4 X11 -20 X12 + 5 X13 + 1 X24 + 2 X35 + 3 X37 = 0;

c14: 5 X3 + 2 X11 + 2 X12 -20 X13 + 3 X25 + 2 X36 + 1 X37 = 0;

c15: 7 X4 -20 X14 + 4 X15 + 4 X16 + 2 X26 + 3 X38 + 3 X39 = 0;

c16: 7 X4 + 4 X14 -19 X15 + 2 X16 + 3 X27 + 2 X38 + 1 X40 = 0;

c17: 4 X4 + 2 X14 + 1 X15 -20 X16 + 3 X28 + 2 X39 + 2 X40 = 0;

c18: 6 X5 -20 X17 + 7 X20 + 3 X29 + 1 X30 + 5 X41 + 3 X42 = 0;

c19: 2 X6 -21 X18 + 7 X23 + 1 X29 + 1 X31 + 2 X43 + 3 X44 = 0;

c20: 2 X7 -20 X19 + 7 X26 + 3 X30 + 3 X31 + 4 X45 + 4 X46 = 0;

c21: 7 X8 + 4 X17 -20 X20 + 5 X32 + 3 X33 + 3 X41 + 1 X42 = 0;

c22: 1 X9 -20 X21 + 7 X24 + 2 X32 + 3 X34 + 5 X47 + 4 X48 = 0;

c23: 3 X10 -19 X22 + 5 X27 + 2 X33 + 1 X34 + 4 X49 + 2 X50 = 0;

c24: 7 X11 + 2 X18 -21 X23 + 2 X35 + 3 X36 + 1 X43 + 1 X44 = 0;

c25: 7 X12 + 1 X21 -20 X24 + 4 X35 + 5 X37 + 3 X47 + 2 X48 = 0;

c26: 3 X13 -20 X25 + 3 X28 + 2 X36 + 2 X37 + 2 X51 + 1 X52 = 0;

c27: 7 X14 + 2 X19 -20 X26 + 4 X38 + 4 X39 + 3 X45 + 3 X46 = 0;

c28: 5 X15 + 3 X22 -19 X27 + 4 X38 + 2 X40 + 2 X49 + 1 X50 = 0;

c29: 3 X16 + 3 X25 -20 X28 + 2 X39 + 1 X40 + 2 X51 + 2 X52 = 0;

c30: 2 X5 + 4 X6 + 2 X17 + 3 X18 - 22 X29+ 1 X30 + 1 X31 + 2 X41 + 5 X43 + 3 X53 = 0;

c31: 2 X5 + 4 X7 + 3 X17 + 3 X19 + 1 X29 -21 X30 + 2 X31 + 4 X42 + 3 X45 + 2 X53 = 0;

c32: 2 X6 + 2 X7 + 3 X18 + 2 X19 + 2 X29 + 1 X30 -22 X31 + 4 X44 + 3 X46 + 2 X53 = 0;

c33: 2 X8 + 2 X9 + 2 X20 + 5 X21 - 22 X32+ 1 X33 + 3 X34 + 4 X41 + 3 X48 + 1 X54 = 0;

c34: 3 X8 + 4 X10 + 2 X20 + 3 X22 + 1 X32 -21 X33 + 2 X34 + 4 X42 + 1 X49 + 2 X54 = 0;

c35: 2 X9 + 1 X10 + 3 X21 + 3 X22 + 2 X32 + 2 X33 -21 X34 + 2 X47 + 5 X50 + 4 X54 = 0;

APPENDIX J: LINEAR EQUATIONS FOR FOUR STATIONS AND FIVE VEHICLES USING THE FARTHEST RULE (CONTINUED)

c36: 3 X11 + 5 X12 + 4 X23 + 2 X24 -22 X35 + 1 X36 + 3 X37 + 2 X43 + 1 X48 + 1 X55 = 0;
c37: 3 X11 + 4 X13 + 2 X23 + 3 X25 + 2 X35 -22 X36 + 2 X37 + 2 X44 + 1 X51 + 1 X55 = 0;
c38: 3 X12 + 2 X13 + 2 X24 + 5 X25 + 2 X35 + 4 X36 -21 X37 + 1 X47 + 3 X52 + 2 X55 = 0;
c39: 3 X14 + 3 X15 + 4 X26 + 4 X27 -21 X38 + 2 X39 + 2 X40 + 2 X45 + 3 X49 + 1 X56 = 0;
c40: 2 X14 + 3 X16 + 2 X26 + 4 X28 + 1 X38 -22 X39 + 2 X40 + 2 X46 + 3 X51 + 2 X56 = 0;
c41: 3 X15 + 5 X16 + 1 X27 + 2 X28 + 2 X38 + 4 X39 -21 X40 + 2 X50 + 1 X52 + 2 X56 = 0;
c42: 2 X17 + 2 X20 + 4 X29 + 2 X32 -22 X41 + 1 X42 + 3 X43 + 5 X48 + 1 X53 + 3 X54 = 0;
c43: 2 X17 + 3 X20 + 4 X30 + 4 X33 + 1 X41 -21 X42 + 1 X45 + 3 X49 + 2 X53 + 2 X54 = 0;
c44: 4 X18 + 3 X23 + 2 X29 + 5 X35 + 1 X41 -22 X43 + 1 X44 + 2 X48 + 1 X53 + 3 X55 = 0;
c45: 2 X18 + 3 X23 + 2 X31 + 4 X36 + 2 X43 -22 X44 + 1 X46 + 3 X51 + 1 X53 + 2 X55 = 0;
c46: 4 X19 + 3 X26 + 2 X30 + 3 X38 + 3 X42 -21 X45 + 2 X46 + 4 X49 + 1 X53 + 2 X56 = 0;
c47: 2 X19 + 2 X26 + 2 X31 + 3 X39 + 3 X44 + 1 X45 -22 X46 + 4 X51 + 2 X53 + 2 X56 = 0;
c48: 2 X21 + 3 X24 + 1 X34 + 2 X37 -21 X47 + 2 X48 + 3 X50 + 5 X52 + 2 X54 + 4 X55 = 0;
c49: 2 X21 + 5 X24 + 1 X32 + 3 X35 + 2 X41 + 4 X43 + 3 X47 -22 X48 + 1 X54 + 1 X55 = 0;
c50: 4 X22 + 3 X27 + 3 X33 + 1 X38 + 2 X42 + 4 X45 -21 X49 + 2 X50 + 1 X54 + 2 X56 = 0;
c51: 1 X22 + 3 X27 + 2 X34 + 5 X40 + 1 X47 + 2 X49 -21 X50 + 2 X52 + 2 X54 + 4 X56 = 0;
c52: 4 X25 + 3 X28 + 3 X36 + 1 X39 + 2 X44 + 2 X46 -22 X51 + 2 X52 + 2 X55 + 1 X56 = 0;
c53: 2 X25 + 5 X28 + 1 X37 + 3 X40 + 2 X47 + 1 X50 + 4 X51 -21 X52 + 2 X55 + 2 X56 = 0;
c54: 2 X29 + 2 X30 + 4 X31 + 2 X41 + 1 X42 + 1 X43 + 2 X44 + 1 X45 + 1 X46 -23 X53 + 2 X54 + 2 X55 + 3 X56 = 0;
c55: 2 X32 + 1 X32 + 2 X34 + 2 X41 + 2 X42 + 2 X47 + 1 X48 + 1 X49 + 3 X50 + 4 X53 -23 X54 + 2 X55 + 1 X56 = 0;
c56: 1 X35 + 2 X36 + 2 X37 + 2 X43 + 4 X44 + 2 X47 + 2 X48 + 1 X51 + 3 X52 + 2 X53 + 1 X54 -23 55 + 1 X56 = 0;

APPENDIX K: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FOUR VEHICLES USING THE FARTHEST RULE

State	Possible Transitions (Farthest Rule)			
(4,0,0,0)	(3,1,0,0) [p12,p32,p42]	(3,0,1,0) [p13,p23,p43]	(3,0,0,1) [p14,p24,p34]	(4,0,0,0) [p21,p31,p41]
(0,4,0,0)	(0,4,0,0) [p12,p32,p42]	(0,3,1,0) [p13,p23,p43]	(0,3,0,1) [p14,p24,p34]	(1,3,0,0) [p21,p31,p41]
(0,0,4,0)	(0,1,3,0) [p12,p32,p42]	(0,0,4,0) [p13,p23,p43]	(0,0,3,1) [p14,p24,p34]	(1,0,3,0) [p21,p31,p41]
(0,0,0,4)	(0,1,0,3) [p12,p32,p42]	(0,0,1,3) [p13,p23,p43]	(0,0,0,4) [p14,p24,p34]	(1,0,0,3) [p21,p31,p41]
(3,1,0,0)	(2,2,0,0) [p12]	(2,1,1,0) [p13]	(2,1,0,1) [p14]	(4,0,0,0) [p21,p31,p41]
(3,0,1,0)	(2,1,1,0) [p12]	(2,0,2,0) [p13]	(2,0,1,1) [p14]	(4,0,0,0) [p21,p31,p41]
(3,0,0,1)	(2,1,0,1) [p12]	(2,0,1,1) [p13]	(2,0,0,2) [p14]	(4,0,0,0) [p21,p31,p41]
(1,3,0,0)	(0,4,0,0) [p12]	(0,3,1,0) [p13]	(0,3,0,1) [p14]	(2,2,0,0) [p21,p31,p41]
(0,3,1,0)	(0,4,0,0) [p12,p32,p42]	(0,3,1,0) [p13,p43]	(0,3,0,1) [p14,p34]	(1,2,1,0) [p21]
(0,3,0,1)	(0,4,0,0) [p12,p42]	(0,3,1,0) [p13,p43]	(0,3,0,1) [p14,p32]	(1,2,0,1) [p21,p31]
(1,0,3,0)	(0,1,3,0) [p12]	(0,0,4,0) [p13]	(0,0,3,1) [p14]	(2,0,2,0) [p21,p31,p41]
(0,1,3,0)	(0,2,2,0) [p12,p32,p42]	(0,1,3,0) [p13,p43]	(0,1,2,1) [p14,p34]	(1,0,3,0) [p21]
(0,0,3,1)	(0,1,3,0) [p12,p42]	(0,0,4,0) [p13,p43]	(0,0,3,1) [p14,p23]	(1,0,2,1) [p21,p31]
(1,0,0,3)	(0,1,0,3) [p12]	(0,0,1,3) [p13]	(0,0,0,4) [p14]	(2,0,0,2) [p21,p31,p41]
(0,1,0,3)	(0,2,0,2) [p12,p42]	(0,1,1,2) [p13,p43]	(0,1,0,3) [p14,p32]	(1,0,0,3) [p21,p31]
(0,0,1,3)	(0,1,1,2) [p12,p42]	(0,0,2,2) [p13]	(0,0,1,3) [p14,p23,p43]	(1,0,0,3) [p21,p31]
(2,2,0,0)	(1,3,0,0) [p12]	(1,2,1,0) [p13]	(1,2,0,1) [p14]	(3,1,0,0) [p21,p31,p41]
(2,0,2,0)	(1,1,2,0) [p12]	(1,0,3,0) [p13]	(1,0,2,1) [p14]	(3,0,1,0) [p21,p31,p41]
(2,0,0,2)	(1,1,0,2) [p12]	(1,0,1,2) [p13]	(1,0,0,3) [p14]	(3,0,0,1) [p21,p31,p41]
(0,2,2,0)	(0,3,1,0) [p12,p32,p42]	(0,2,2,0) [p13,p43]	(0,2,1,1) [p14,p34]	(1,1,2,0) [p21]
(0,2,0,2)	(0,3,0,1) [p12,p42]	(0,2,1,1) [p13,p43]	(0,2,0,2) [p14,p32]	(1,1,0,2) [p21,p31]
(0,0,2,2)	(0,1,2,1) [p12,p42]	(0,0,3,1) [p13,p43]	(0,0,2,2) [p14,p23]	(1,0,1,2) [p21,p31]
(2,1,1,0)	(1,2,1,0) [p12]	(1,1,2,0) [p13]	(1,1,1,1) [p14]	(3,0,1,0) [p21]
	(2,1,1,0) [p43]	(2,1,0,1) [p34]		
(2,1,0,1)	(1,2,0,1) [p12]	(1,1,1,1) [p13]	(1,1,0,2) [p14]	(3,0,0,1) [p21,p31]
	(2,2,0,0) [p42]	(2,1,1,0) [p43]		
(2,0,1,1)	(1,1,1,1) [p12]	(1,0,2,1) [p13]	(1,0,1,2) [p14]	(3,0,0,1) [p21,p31]
	(2,1,1,0) [p42]	(2,0,2,0) [p43]		
(1,2,1,0)	(0,3,1,0) [p12]	(0,2,2,0) [p13]	(0,2,1,1) [p14]	(2,1,1,0) [p21]
	(1,2,0,1) [p34]	(1,2,1,0) [p43]		
(1,2,0,1)	(0,3,0,1) [p12]	(0,2,1,1) [p13]	(0,2,0,2) [p14]	(2,1,0,1) [p21,p31]
	(2,2,0,0) [p41]	(1,3,0,0) [p42]		
(0,2,1,1)	(0,3,1,0) [p12,p42]	(0,2,2,0) [p13,p43]	(0,2,1,1) [p14]	(1,1,1,1) [p21]
	(0,2,0,2) [p34]	(1,2,1,0) [p41]		
(1,1,2,0)	(0,2,2,0) [p12]	(0,1,3,0) [p13]	(0,1,2,1) [p14]	(2,0,2,0) [p21]
	(1,1,1,1) [p34]	(1,1,2,0) [p43]		
(1,0,2,1)	(0,1,2,1) [p12]	(0,0,3,1) [p13]	(0,0,2,2) [p14]	(2,0,1,1) [p21,p31]
	(1,1,2,0) [p42]	(1,0,3,0) [p43]		
(0,1,2,1)	(0,2,2,0) [p12,p42]	(0,1,3,0) [p13,p43]	(0,1,2,1) [p14]	(1,0,2,1) [p21]
	(0,1,1,2) [p34]	(1,1,2,0) [p41]		
(1,1,0,2)	(0,2,0,2) [p12]	(0,1,1,2) [p13]	(0,1,0,3) [p14]	(2,0,0,2) [p21,p31]
	(1,2,0,1) [p42]	(1,1,1,1) [p43]		
(1,0,1,2)	(0,1,1,2) [p12]	(0,0,2,2) [p13]	(0,0,1,3) [p14]	(2,0,0,2) [p21,p31]
	(1,1,1,1) [p42]	(1,0,2,1) [p43]		
(0,1,1,2)	(0,2,1,1) [p12,p42]	(0,1,2,1) [p13,p43]	(0,1,1,2) [p14]	(1,0,1,2) [p21]
	(0,1,0,3) [p34]	(1,1,1,1) [p41]		
(1,1,1,1)	(0,2,1,1) [p12]	(0,1,2,1) [p13]	(0,1,1,2) [p14]	(2,0,1,1) [p21]
	(2,1,1,0) [p41]	(1,2,1,0) [p42]	(1,1,2,0) [p43]	(1,1,0,2) [p34]

APPENDIX K: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND FOUR VEHICLES USING THE FARTHEST RULE (CONTINUED)

(3,1,0,0)	(3,0,1,0) [p23,p43]	(3,0,0,1) [p24,p34]	(3,1,0,0) [p32,p42]	
(3,0,1,0)	(3,0,1,0) [p23,p43]	(3,0,0,1) [p24,p34]	(3,1,0,0) [p32,p42]	
(3,0,0,1)	(3,0,1,0) [p23,p43]	(3,0,0,1) [p24,p34]	(3,1,0,0) [p32,p42]	
(1,3,0,0)	(1,2,1,0) [p23,p43]	(1,2,0,1) [p24,p34]	(1,3,0,0) [p32,p42]	
(0,3,1,0)	(0,2,2,0) [p23]	(0,2,1,1) [p24]	(1,3,0,0) [p31,p41]	
(0,3,0,1)	(0,2,1,1) [p23]	(0,2,0,2) [p24,p34]	(1,3,0,0) [p41]	
(1,0,3,0)	(1,0,3,0) [p23,p43]	(1,0,2,1) [p24,p34]	(1,1,2,0) [p32,p42]	
(0,1,3,0)	(0,0,4,0) [p23]	(0,0,3,1) [p24]	(1,1,2,0) [p31,p41]	
(0,0,3,1)	(0,0,2,2) [p24,p34]	(0,1,2,1) [p32]	(1,0,3,0) [p41]	
(1,0,0,3)	(1,0,1,2) [p23,p43]	(1,0,0,3) [p24,p34]	(1,1,0,2) [p32,p42]	
(0,1,0,3)	(0,0,1,3) [p23]	(0,0,0,4) [p24,p34]	(1,1,0,2) [p41]	
(0,0,1,3)	(0,0,0,4) [p24,p34]	(0,1,0,3) [p32]	(1,0,1,2) [p41]	
(2,2,0,0)	(2,1,1,0) [p23,p43]	(2,1,0,1) [p24,p34]	(2,2,0,0) [p32,p42]	
(2,0,2,0)	(2,0,2,0) [p23,p43]	(2,0,1,1) [p24,p34]	(2,1,1,0) [p32,p42]	
(2,0,0,2)	(2,0,1,1) [p23,p43]	(2,0,0,2) [p24,p34]	(2,1,0,1) [p32,p42]	
(0,2,2,0)	(0,1,3,0) [p23]	(0,1,2,1) [p24]	(1,2,1,0) [p31,p41]	
(0,2,0,2)	(0,1,1,2) [p23]	(0,1,0,3) [p24,p34]	(1,2,0,1) [p41]	
(0,0,2,2)	(0,0,1,3) [p24,p34]	(1,0,2,1) [p41]	(0,1,1,2) [p32]	
(2,1,1,0)	(2,0,2,0) [p23]	(2,0,1,1) [p24]	(3,1,0,0) [p31,p41]	(2,2,0,0) [p32,p42]
(2,1,0,1)	(2,0,1,1) [p23]	(2,0,0,2) [p24,p34]	(2,1,0,1) [p32]	(3,1,0,0) [p41]
(2,0,1,1)	(2,0,1,1) [p23]	(2,0,0,2) [p24,p34]	(2,1,0,1) [p32]	(3,0,1,0) [p41]
(1,2,1,0)	(1,1,2,0) [p23]	(1,1,1,1) [p24]	(2,2,0,0) [p31,p41]	(1,3,0,0) [p32,p42]
(1,2,0,1)	(1,1,1,1) [p23]	(1,1,0,2) [p24,p34]	(1,2,0,1) [p32]	(1,2,1,0) [p43]
(0,2,1,1)	(0,1,2,1) [p23]	(0,1,1,2) [p24]	(1,2,0,1) [p31]	(0,3,0,1) [p32]
(1,1,2,0)	(1,0,3,0) [p23]	(1,0,2,1) [p24]	(2,1,1,0) [p31,p41]	(1,2,1,0) [p32,p42]
(1,0,2,1)	(1,0,2,1) [p23]	(1,0,1,2) [p24,p34]	(1,1,1,1) [p32]	(2,0,2,0) [p41]
(0,1,2,1)	(0,0,3,1) [p23]	(0,0,2,2) [p24]	(1,1,1,1) [p31]	(0,2,1,1) [p32]
(1,1,0,2)	(1,0,1,2) [p23]	(1,0,0,3) [p24,p34]	(1,1,0,2) [p32]	(2,1,0,1) [p41]
(1,0,1,2)	(1,0,1,2) [p23]	(1,0,0,3) [p24,p34]	(1,1,0,2) [p32]	(2,0,1,1) [p41]
(0,1,1,2)	(0,0,2,2) [p23]	(0,0,1,3) [p24]	(1,1,0,2) [p31]	(0,2,0,2) [p32]
(1,1,1,1)	(1,0,2,1) [p23]	(1,0,1,2) [p24]	(2,1,0,1) [p31]	(1,2,0,1) [p32]

APPENDIX L: LINEAR EQUATIONS FOR FOUR STATIONS AND FOUR VEHICLES USING THE FARTHEST RULE

X4; (Objective Function)
(c1 to c36 are constraints)
c1: 16 X1 - 7 X5 - 7 X6 - 7 X7 = 0;
c2: 16 X2 - 4 X8 - 7 X9 - 5 X10 = 0;
c3: 19 X3 - 2 X11 - 1 X12 - 3 X13 = 0;
c4: 18 X4 - 2 X14 - 3 X15 - 3 X16 = 0;
c5: 7 X1 -20 X5 + 3 X6 + 3 X7 + 7 X17 + 5 X23 + 3 X24 = 0;
c6: 4 X1 + 2 X5 -21 X6 + 2 X7 + 7 X18 + 2 X23 + 3 X25 = 0;
c7: 5 X1 + 3 X5 + 3 X6 -20 X7 + 7 X19 + 4 X24 + 4 X25 = 0;
c8: 7 X2 -20 X8 + 5 X9 + 3 X10 + 4 X17 + 3 X26 + 1 X27 = 0;
c9: 4 X2 + 2 X8 -20 X9 + 3 X10 + 7 X20 + 4 X26 + 5 X28 = 0;
c10: 5 X2 + 2 X8 + 3 X9 -19 X10 + 5 X21 + 4 X27 + 2 X28 = 0;
c11: 7 X3 -21 X11 + 2 X12 + 3 X13 + 2 X18 + 1 X29 + 1 X30 = 0;
c12: 7 X3 + 4 X11 -20 X12 + 5 X13 + 1 X20 + 2 X29 + 3 X31 = 0;
c13: 5 X3 + 2 X11 + 2 X12 -20 X13 + 3 X22 + 2 X30 + 1 X31 = 0;
c14: 7 X4 -20 X14 + 4 X15 + 4 X16 + 2 X19 + 3 X32 + 3 X33 = 0;
c15: 7 X4 + 4 X14 -19 X15 + 2 X16 + 3 X21 + 2 X32 + 1 X34 = 0;
c16: 4 X4 + 2 X14 + 1 X15 -19 X16 + 3 X22 + 2 X33 + 2 X34 = 0;
c17: 4 X5 + 7 X8 -20 X17 + 3 X23 + 1 X24 + 5 X26 + 3 X27 = 0;
c18: 2 X6 + 7 X11 -21 X18 + 1 X23 + 1 X25 + 2 X29 + 3 X30 = 0;
c19: 2 X7 + 7 X14 -20 X19 + 3 X24 + 3 X25 + 4 X32 + 4 X33 = 0;
c20: 1 X9 + 7 X12 -20 X20 + 2 X26 + 3 X28 + 4 X29 + 5 X31 = 0;
c21: 3 X10 + 5 X15 -19 X21 + 2 X27 + 1 X28 + 4 X32 + 2 X34 = 0;
c22: 3 X13 + 2 X16 -20 X22 + 2 X30 + 2 X31 + 2 X33 + 1 X34 = 0;
c23: 2 X5 + 4 X6 + 2 X17 + 3 X18 -22 X23 + 1 X24 + 1 X25 + 2 X26 + 5 X29 + 3 X35 = 0;
c24: 2 X5 + 4 X7 + 3 X17 + 3 X19 + 1 X23 -21 X24 + 2 X25 + 4 X27 + 3 X32 + 2 X35 = 0;
c25: 2 X6 + 2 X7 + 3 X18 + 2 X19 + 2 X23 + 1 X24 -22 X25 + 4 X30 + 3 X33 + 2 X35 = 0;
c26: 2 X8 + 2 X9 + 2 X17 + 5 X20 + 4 X23 -22 X26 + 1 X27 + 3 X28 + 3 X29 + 1 X35 = 0;
c27: 3 X8 + 4 X10 + 2 X17 + 3 X21 + 4 X24 + 1 X26 -21 X27 + 2 X28 + 1 X32 + 2 X35 = 0;
c28: 2 X9 + 1 X10 + 3 X20 + 3 X21 + 2 X26 + 2 X27 -21 X28 + 2 X31 + 5 X34 + 4 X35 = 0;
c29: 3 X11 + 5 X12 + 4 X18 + 2 X20 + 2 X23 + 1 X26 -22 X29 + 1 X30+ 3 X31 + 1 X35 = 0;
c30: 3 X11 + 4 X13 + 2 X18 + 3 X22 + 2 X25 + 2 X29 -22 X30 + 2 X31+ 1 X33 + 1 X35 = 0;
c31: 3 X12 + 2 X13 + 2 X20 + 5 X22 + 1 X28 + 2 X29 + 4 X30 -21 X31 + 3 X34 + 2 X35 = 0;
C32: 3 X14 + 3 X15 + 4 X19 + 4 X21 + 2 X24 + 3 X27 -21 X32 + 2 X33 + 2 X34 + 1 X35 = 0;
c33: 2 X14 + 3 X16 + 2 X19 + 4 X22 + 2 X25 + 3 X30 + 1 X32 -22 X33 + 2 X34 + 2 X35 = 0;
c34: 2 X23 + 2 X24 + 4 X25 + 2 X26 + 1 X27 + 2 X28 + 1 X29 + 2 X30 + 2 X31 + 1 X32 +1 X33 + 3 X34 -23 X35 = 0;
c35: X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16+X17+X18 +X19+X20+X21+X22+X23+X24+X25+X26+X27+X28+X29+X30+X31+X32+X33+X34 +X35=1;

APPENDIX M: STATE SPACE AND TRANSITION PROBABILITIES FOR FOUR STATIONS AND THREE VEHICLES USING THE FARTHEST RULE

	Possible Transitions (Nearest Rule)			
(3,0,0,0)	(2,1,0,0) [p12,p32,p42]	(2,0,1,0) [p13,p23,p43]	(2,0,0,1) [p14,p24,p34]	(3,0,0,0) [p21,p31,p41]
(0,3,0,0)	(0,3,0,0) [p12,p32,p42]	(0,2,1,0) [p13,p23,p43]	(0,2,0,1) [p14,p24,p34]	(1,2,0,0) [p21,p31,p41]
(0,0,3,0)	(0,1,2,0) [p12,p32,p42]	(0,0,3,0) [p13,p23,p43]	(0,0,2,1) [p14,p24,p34]	(1,0,2,0) [p21,p31,p41]
(0,0,0,3)	(0,1,0,2) [p12,p32,p42]	(0,0,1,2) [p13,p23,p43]	(0,0,0,3) [p14,p24,p34]	(1,0,0,2) [p21,p31,p41]
(2,1,0,0)	(1,2,0,0) [p12]	(1,1,1,0) [p13]	(1,1,0,1) [p14]	(3,0,0,0) [p21,p31,p41]
	(2,0,1,0) [p23,p43]	(2,0,0,1) [p24,p34]	(2,1,0,0) [p32,p42]	
(2,0,1,0)	(1,1,1,0) [p12]	(1,0,2,0) [p13]	(1,0,1,1) [p14]	(3,0,0,0) [p21,p31,p41]
	(2,0,1,0) [p23,p43]	(2,0,0,1) [p24,p34]	(2,1,0,0) [p32,p42]	
(2,0,0,1)	(1,1,0,1) [p12]	(1,0,1,1) [p13]	(1,0,0,2) [p14]	(3,0,0,0) [p21,p31,p41]
	(2,0,1,0) [p23,p43]	(2,0,0,1) [p24,p34]	(2,1,0,0) [p32,p42]	
(1,2,0,0)	(0,3,0,0) [p12]	(0,2,1,0) [p13]	(0,2,0,1) [p14]	(2,1,0,0) [p21,p31,p41]
	(1,1,1,0) [p23,p43]	(1,1,0,1) [p24,p34]	(1,2,0,0) [p32,p42]	
(0,2,1,0)	(0,3,0,0) [p12,p32,p42]	(0,2,1,0) [p13,p43]	(0,2,0,1) [p14,p34]	(1,1,1,0) [p21]
	(0,1,2,0) [p23]	(0,1,1,1) [p24]	(1,2,0,0) [p31,p41]	
(0,2,0,1)	(0,3,0,0) [p12,p42]	(0,2,1,0) [p13,p43]	(0,2,0,1) [p14,p32]	(1,1,0,1) [p21,p31]
	(0,1,1,1) [p23]	(0,1,0,2) [p24,p34]	(1,2,0,0) [p41]	
(1,0,2,0)	(0,1,2,0) [p12]	(0,0,3,0) [p13]	(0,0,2,1) [p14]	(2,0,1,0) [p21,p31,p41]
	(1,0,2,0) [p23,p43]	(1,0,1,1) [p24,p34]	(1,1,1,0) [p32,p42]	
(0,1,2,0)	(0,2,1,0) [p12,p32,p42]	(0,1,2,0) [p13,p43]	(0,1,1,1) [p14,p34]	(1,0,2,0) [p21]
	(0,0,3,0) [p23]	(0,0,2,1) [p24]	(1,1,1,0) [p31,p41]	
(0,0,2,1)	(0,1,2,0) [p12,p42]	(0,0,3,0) [p13,p43]	(0,0,2,1) [p14,p23]	(1,0,1,1) [p21,p31]
	(0,0,1,2) [p24,p34]	(0,1,1,1) [p32]	(1,0,2,0) [p41]	
(1,0,0,2)	(0,1,0,2) [p12]	(0,0,1,2) [p13]	(0,0,0,3) [p14]	(2,0,0,1) [p21,p31,p41]
	(1,0,1,1) [p23,p43]	(1,0,0,2) [p24,p34]	(1,1,0,1) [p32,p42]	
(0,1,0,2)	(0,2,0,1) [p12,p42]	(0,1,1,1) [p13,p43]	(0,1,0,2) [p14,p32]	(1,0,0,2) [p21,p31]
	(0,0,1,2) [p23]	(0,0,0,3) [p24,p34]	(1,1,0,1) [p41]	
(0,0,1,2)	(0,1,1,1) [p12,p42]	(0,0,2,1) [p13]	(0,0,1,2) [p14,p23,p43]	(1,0,0,2) [p21,p31]
	(0,0,0,3) [p24,p34]	(0,1,0,2) [p32]	(1,0,1,1) [p41]	
(1,1,1,0)	(0,2,1,0) [p12,p42]	(0,1,2,0) [p13,p43]	(0,1,1,1) [p14]	(2,0,1,0) [p21]
	(1,1,1,0) [p41]	(1,1,0,1) [p34]	(2,1,0,0) [p31]	(1,2,0,0) [p32]
	(1,0,2,0) [p23]	(1,0,1,1) [p24]		
(1,1,0,1)	(0,2,0,1) [p12,p32]	(0,1,1,1) [p13]	(0,1,0,2) [p14,p34]	(2,0,0,1) [p21]
	(1,1,1,0) [p43]	(1,2,0,0) [p42]	(1,1,0,1) [p31]	(2,1,0,0) [p41]
	(1,0,1,1) [p23]	(1,0,0,2) [p24]		
(1,0,1,1)	(0,1,1,1) [p12]	(0,0,2,1) [p13,p23]	(0,0,1,2) [p14,p24]	(1,0,1,1) [p21]
	(1,0,2,0) [p43]	(1,1,1,0) [p42]	(1,0,0,2) [p34]	(2,0,1,0) [p41]
	(2,0,0,1) [p31]	(1,1,0,1) [p32]		
(0,1,1,1)	(0,1,1,1) [p12]	(0,0,2,1) [p13,p23]	(0,0,1,2) [p14,p24]	(1,0,1,1) [p21]
	(1,1,0,1) [p31]	(0,2,0,1) [p32]	(0,1,0,2) [p34]	(1,1,1,0) [p41]
	(0,,1,2,0) [p43]	(0,2,1,0) [p42]		

APPENDIX N: STATIONARY PROBABILITIES OF THE MARKOV CHAIN FOR FOUR STATIONS AND THREE VEHICLES USING THE FARTHEST DISPATCHING RULE

State	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
Probability	0.1126	0.0660	0.0046	0.0136	0.1129	0.0554	0.0890

State	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**	**S14**
Probability	0.0811	0.0588	0.0639	0.0147	0.0217	0.0123	0.0406

State	**S15**	**S16**	**S17**	**S18**	**S19**	**S20**
Probability	0.0354	0.0189	0.0504	0.0720	0.0352	0.0408

APPENDIX O: LINEAR EQUATIONS FOR FOUR STATIONS AND THREE VEHICLES USING THE FARTHEST RULE

X4 (Objective Function);
(c1 to c20 are constraints)
c1: X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10
 + X11 + X12 + X13 + X14 + X15 + X16 +X17 + X18 + X19 + X20 = 1;
c2: 16 X1 - 7 X5 - 7 X6 - 7 X7 = 0;
c3: 16 X2 - 4 X8 - 7 X9 - 5 X10 = 0;
c4: 19 X3 - 2 X11 - 1 X12 - 3 X13 = 0;
c5: 18 X4 - 2 X14 - 3 X15 - 3 X16 = 0;
c6: 7 X1 -20 X5 + 3 X6 + 3 X7 + 7 X8 + 5 X17 + 3 X18 = 0;
c7: 4 X1 + 2 X5 -21 X6 + 2 X7 + 7 X11 + 2 X17 + 3 X19 = 0;
c8: 5 X1 + 3 X5 + 3 X6 -20 X7 + 7 X14 + 4 X18 + 4 X19 = 0;
c9: 7 X2 + 4 X5 -20 X8 + 5 X9 + 3 X10 + 3 X17 + 1 X18 = 0;
c10: 4 X2 + 2 X8 -20 X9 + 3 X10 + 7 X12 + 4 X17 + 5 X20 = 0;
c11: 5 X2 + 2 X8 + 3 X9 -19 X10 + 5 X15 + 4 X18 + 2 X20 = 0;
c12: 7 X3 + 2 X6 -21 X11 + 2 X12 + 3 X13 + 1 X17 + 1 X19 = 0;
c13: 7 X3 + 1 X9 + 4 X11 -20 X12 + 5 X13 + 2 X17 + 3 X20 = 0;
c14: 5 X3 + 2 X11 + 2 X12 -20 X13 + 2 X16 + 2 X19 + 1 X20 = 0;
c15: 7 X4 + 2 X7 -20 X14 + 4 X15 + 4 X16 + 3 X18 + 3 X19 = 0;
c16: 7 X4 + 3 X10 + 4 X14 -19 X15 + 2 X16 + 2 X18 + 1 X20 = 0;
c17: 4 X4 + 3 X13 + 2 X14 + 1 X15 -19 X16 + 2 X19 + 2 X20 = 0;
c18: 2 X5 + 4 X6 + 2 X8 + 2 X9 + 3 X11 + 5 X12 -22 X17 + 1 X18 + 1 X19 +3 X20= 0;
c19: 2 X5 + 4 X7 + 3 X8 + 4 X10 + 3 X14 + 3 X15 + 1 X17 -21 X18 + 2 X19 + 2 X20 = 0;
c20: 2 X6 + 2 X7 + 3 X11 + 4 X13 + 2 X14 + 3 X16 + 2 X17+ 1 X18 - 22 X19 + 2 X20 = 0;

APPENDIX P: PROBABILITY DISTRIBUTION OF THE LOAD REQUESTS FOR FOUR STATIONS AND THREE VEHICLES USING THE FARTHEST DISPATCHING RULE

n (# of Load Requests)	**0**	**1**	**2**	**3**	**4**	**5**
P(n)	0.0851	0.1849	0.2010	0.1457	0.1056	0.0765
n (# of Load Requests)	**6**	**7**	**8**	**9**	**10**	**11**
P(n)	0.0554	0.0402	0.0291	0.0211	0.0153	0.0111
n (# of Load Requests)	**12**	**13**	**14**	**15**	**16**	**17**
P(n)	0.0080	0.0058	0.0042	0.0031	0.0022	0.0016

APPENDIX Q: SIMULATION PROGRAM FOR NW/NV DISPATCHING RULE COMBINATION

```
/*Nearest WC,Nearest V*/
float rand(int stream);
void randst(long zset,int stream);
long randgt(int stream);
#include<stdio.h>
#include<math.h>
#include"c:\shen\rand.cpp"
#define nagv 4
#define lut 12/60
#define simtime 6600
#define ct 600

struct agv
        {
        int status;
        int station;
        };
struct ev
        {
        int                     ec;
        double          et;
        int                     station;
        struct ev       *next;
        };

struct job
        {
        double  at;
        int             station;
        int             desti;
        struct  job *next;
        };

struct job      *q[12];
struct agv      a[nagv];
struct ev       *p1,*p2;
double time;
int mean[12]={15,30,30,30,30,60,30,30,60,20,60,60};
float t[4][4];
```

```
float expon(float u,int lambda)
        {
        return(-lambda*log(u));
        }


void eli(struct ev *p3)
        {
        struct ev *p4;
        if(p1==NULL&&p2==NULL)
                {
                p1=p3;
                p2=p3;
                p3->next=NULL;
                }
        else if(p3->et<p1->et) /*Insert at head of the list*/
                {
                p3->next=p1;
                p1=p3;
                }
        else if(p3->et>p2->et) /*Insert at the end of the list*/
                {
                p2->next=p3;
                p2=p3;
                p3->next=NULL;
                }
        else /*Insert at the middle of the list*/
                {
                p4=p1;
                while(p3->et>p4->et)
                        {
                        p3->next=p4;
                        p4=p4->next;
                        }
                p3->next->next=p3;
                p3->next=p4;
                }
        }
```

```
void initialize()
        {
        float u,arrivaltime;
        int i;
        struct ev *p3;

        p1=NULL;
        p2=NULL;
        time=0;
        for(i=0;i<12;++i)
                {
                u=rand(i);
                p3=new struct ev;
                p3->et=expon(u,mean[i]);
                p3->ec=i;
                p3->station=(int)(i/3);
                eli(p3);
                }
        for(i=0;i<4;++i)
                {
                a[i].status=1;
                a[i].station=i;
                }
        if(nagv>4)
                {
                for(i=4;i<nagv;++i)
                        {
                        a[i].status=1;
                        a[i].station=0;
                        }
                }
        t[0][1]=t[1][0]=140/60;
        t[0][2]=t[2][0]=160/60;
        t[0][3]=150/60;t[3][0]=340/60;
        t[1][2]=255/60;t[2][1]=255/60;
        t[1][3]=240/60;t[3][1]=240/60;
        t[2][3]=t[3][2]=250/60;
        for(i=0;i<4;++i) t[i][i]=0;
        for(i=0;i<12;++i) q[i]=NULL;
        }
```

```
unsigned long int arrivals=0,wait=0;
unsigned long int nowait=0;
unsigned long int serviced=0;
double tt=0.0;
unsigned long int same=0;

void arrival(struct ev *p)
        {
        int                 jobstation,jobcode,jobdesti,i,j,fvi,vt;
        int                 mean[]={15,30,30,30,30,60,30,30,60,20,60,60};
        float               u;
        double        fvt,traveltime;
        struct ev   *pnew,*pagv;
        struct job       *pjob;

        /*Setting the parameters and freeing heap*/
        time=p->et;jobstation=p->station;jobcode=p->ec;
        if(jobcode==3||jobcode==6||jobcode==9) jobdesti=0;
        if(jobcode==0||jobcode==7||jobcode==10) jobdesti=1;
        if(jobcode==1||jobcode==4||jobcode==11) jobdesti=2;
        if(jobcode==2||jobcode==5||jobcode==8) jobdesti=3;
        p1=p1->next;
        delete(p);
        /*Scheduling the next arrival*/
        pnew=new struct ev;
        pnew->ec=jobcode;pnew->station=jobstation;u=rand(jobcode);pnew-
>et=time+expon(u,mean[jobcode]);
        eli(pnew);
        /*Searching for the nearest available vehicle*/
        fvt=100;fvi=-1;
        for(i=0;i<nagv;++i)
                {
                if(a[i].status==1)
                        {
                        vt=a[i].station;
                        if(fvt>t[vt][jobstation])
                                {
                                fvt=t[vt][jobstation];
                                fvi=i;
                                }
                        }
                }
```

```
/*Checking if vehicle available at the station*/
        for(i=0;i<nagv;++i)
                {
                if(a[i].status==1&&a[i].station==jobstation)
                        {
                        fvt=0;fvi=i;
                        }
                }

        if(fvi==-1)/*Put the load in queue*/
                {
                pjob=new struct job;
                pjob->at=time;pjob->station=jobstation;pjob-
>next=q[jobcode];q[jobcode]=pjob;
                pjob->desti=jobdesti;
                }
        else/*Begin Service*/
                {
                a[fvi].status=0;a[fvi].station=jobdesti;
                tt=tt+fvt;
                if(fvt==0&&time>ct) ++same;
                traveltime=time+fvt+2*lut+t[jobstation][jobdesti];
                if(traveltime<simtime&&traveltime>ct) ++nowait;
                pagv=new struct ev;
                pagv->ec=12+fvi;pagv->et=traveltime;pagv->station=jobdesti;
                eli(pagv);
                }
        }

double waittime=0;

void end(struct ev *p)
        {
        int i,agvat,agvnos,fwc,stat,fws,fwd;
        double fwt,traveltime;
        struct ev *pnew;
        struct job *pjob;

        /*Setting the parameters*/
        time=p->et;agvat=p->station;agvnos=p->ec-12;p1=p1->next;delete(p);
        /*Search for the nearest job*/
```

```
fwc=-1;fwt=1000;
for(i=0;i<12;++i)
        {
        if(q[i]!=NULL)
                {
                stat=q[i]->station;
                if(t[agvat][stat]<fwt)
                        {
                        fwt=t[agvat][stat];
                        fwc=i;
                        fws=q[i]->station;
                        }
                }
        }
/*Search for a job in the station AGV is at*/
for(i=0;i<12;++i)
        {
        if(q[i]!=NULL)
                {
                stat=q[i]->station;
                if(stat==agvat)
                        {
                        fwt=t[agvat][stat];
                        fwc=i;
                        fws=q[i]->station;
                        }
                }
        }
if(fwc==-1)/*Set AGV free*/
        {
        a[agvnos].status=1;
        }
else/*Begin Service*/
        {
        fwd=q[fwc]->desti;
        a[agvnos].station=fwd;
        traveltime=time+t[agvat][fws]+2*lut+t[fws][fwd];
        if(traveltime<simtime&&traveltime>ct) waittime=waittime+time-(q[fwc]->at);
        if(traveltime<simtime&&traveltime>ct) ++wait;
        pnew=new struct ev;
        pnew->ec=12+agvnos;
```

```
pnew->et=traveltime;
        pnew->station=fwd;
        eli(pnew);
        pjob=q[fwc];
        q[fwc]=q[fwc]->next;
        delete pjob;
        }
}


void main()
        {
        double  waitfrac,avgwait;

        initialize();
        printf("\n\n");
        while(p1->et<simtime)
                {
                if(p1->ec<12)
                        {
                        ++arrivals;
                        arrival(p1);
                        }
                else
                        {
                        ++serviced;
                        end(p1);
                        }
                }
        printf("\nWait is %ld,nowait is %ld arrivals is %ld and serviced is %ld", wait, nowait,
        arrivals, serviced);
        waitfrac=(double) wait/(serviced);
        avgwait=waittime/(wait);
        printf("\n\nFraction Waiting  %f ",waitfrac);
        printf("\nAverage wait %f",avgwait*60);
        printf("\nThe empty vehicle travel time is %f",tt/nowait*60);
        getchar();
        }
```

APPENDIX R: SIMULATION DATA FOR NV/NW DISPATCHING RULE COMBINATION ($\alpha = 0.14$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	17.28	19.44	20.84	22.03	24.19	26.24	24.54	29.53	27.28	28.86
2	20.09	21.01	19.83	23.18	24.66	27.73	23.74	28.39	27.57	28.62
3	19.49	22.58	21.88	24.18	22.75	27.29	24.44	27.91	27.88	29.86
4	21.10	22.20	19.51	23.33	24.77	26.28	24.69	29.21	28.12	29.54
5	16.76	23.30	20.47	22.92	25.08	26.26	25.57	26.94	27.30	30.59
6	17.37	19.29	21.89	23.98	23.68	24.59	26.76	27.23	26.71	29.42
7	17.23	24.12	25.26	25.17	25.14	25.52	24.96	28.13	26.44	31.73
8	16.76	18.21	22.78	26.63	24.32	26.97	26.61	29.31	23.98	32.29
9	18.02	19.23	23.56	25.53	25.24	26.79	28.36	24.75	26.24	30.53
10	16.46	22.02	22.78	25.89	24.18	26.69	27.06	25.32	27.88	31.68
11	17.94	18.40	20.69	22.52	24.2	25.18	28.17	26.31	25.77	29.99
12	17.44	18.93	20.74	23.98	22.45	25.24	26.54	26.84	25.30	30.89
13	17.65	20.66	23.56	25.27	22.42	27.78	25.16	27.10	24.25	28.86
14	18.48	20.73	23.14	25.09	22.9	27.59	26.73	25.14	26.97	29.98
15	17.20	16.93	23.37	25.52	21.52	27.32	26.64	26.01	27.18	31.94
16	15.85	18.12	23.00	24.67	23.99	26.72	24.26	27.92	24.47	31.86
17	16.21	17.13	21.43	24.59	23.35	25.99	22.66	27.85	26.40	32.82
18	19.02	19.87	21.01	23.88	21.31	25.88	27.07	26.23	26.77	30.39
19	16.21	18.43	19.88	25.95	21.47	26.49	26.06	27.01	26.74	30.57
20	15.87	20.16	21.00	25.76	24.55	26.97	25.90	28.40	28.14	31.38

APPENDIX S: SIMULATION DATA FOR NV/NW DISPATCHING RULE COMBINATION ($\alpha = 0.12$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	19.87	19.61	20.84	22.73	24.30	26.24	24.54	29.53	27.28	28.98
2	20.23	21.18	20.48	25.16	24.77	27.73	23.74	28.39	27.57	28.74
3	19.63	22.75	21.88	24.18	23.89	27.29	24.44	27.91	27.88	29.98
4	21.24	22.37	19.51	23.33	24.88	26.28	24.69	29.21	28.12	29.66
5	18.66	23.47	21.72	24.92	25.19	26.26	25.57	26.94	27.30	30.71
6	17.57	19.46	21.59	23.98	23.79	24.59	26.76	27.23	26.71	29.54
7	17.74	24.29	24.61	25.17	25.25	25.52	25.63	28.13	26.44	31.85
8	16.90	18.38	22.72	26.43	24.43	28.97	26.61	29.31	25.73	32.41
9	18.15	19.40	23.56	25.53	25.35	26.79	28.36	24.75	26.24	30.65
10	16.60	22.19	22.78	25.89	24.29	26.69	27.06	25.32	27.88	31.80
11	18.81	18.57	20.69	23.52	26.29	25.18	28.17	26.31	25.77	30.11
12	17.58	19.10	20.74	23.98	22.56	25.24	26.54	26.84	25.30	31.01
13	17.59	20.83	23.56	25.27	25.85	27.78	25.16	27.10	27.64	28.98
14	18.62	20.90	23.14	25.09	23.91	28.59	26.73	25.14	26.97	30.10
15	17.34	17.10	23.37	25.52	21.63	27.32	26.64	26.01	27.18	32.06
16	15.99	18.29	23.00	24.67	26.10	26.72	24.26	27.92	24.47	31.98
17	16.35	17.30	21.43	24.59	23.46	25.99	22.66	27.85	26.40	32.94
18	19.16	20.04	21.01	23.88	24.75	25.88	27.07	26.23	26.77	30.51
19	16.42	18.60	19.88	25.95	21.58	26.49	26.06	27.01	26.74	30.69
20	16.01	20.33	21.00	25.76	24.66	26.97	25.90	28.40	28.14	31.50

APPENDIX T: SIMULATION DATA FOR NV/NW DISPATCHING RULE COMBINATION ($\alpha = 0.10$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	17.28	21.37	20.84	22.24	24.19	26.91	3.85	5.63	5.28	5.86
2	20.09	22.37	19.83	23.39	24.66	27.73	3.05	4.49	5.57	5.62
3	19.49	22.58	21.88	24.39	22.75	27.29	3.75	6.21	5.88	6.86
4	21.10	22.25	19.51	23.54	24.77	26.28	4.00	5.31	6.12	6.54
5	16.76	23.30	20.47	23.13	25.68	26.73	4.88	5.04	5.30	5.59
6	19.37	19.69	21.89	24.19	23.68	24.59	6.07	4.33	5.71	6.42
7	17.23	24.12	25.26	25.38	25.14	25.52	4.27	5.23	6.44	5.73
8	19.79	17.95	22.78	26.84	24.32	26.97	5.92	5.41	5.98	6.29
9	18.02	21.23	23.56	25.74	25.73	26.79	7.67	5.84	4.24	5.53
10	16.46	22.02	22.78	26.10	24.18	26.69	6.37	6.42	5.88	5.68
11	17.94	20.51	20.69	22.73	24.20	26.19	7.48	5.21	4.77	5.99
12	19.54	18.93	23.74	28.19	22.45	25.24	5.85	5.94	5.13	7.89
13	17.65	21.19	23.56	25.48	23.16	27.78	4.47	4.28	6.25	5.86
14	18.48	22.46	23.14	26.30	22.99	27.59	6.04	5.24	5.97	5.98
15	17.20	17.48	23.37	25.73	25.52	27.32	5.95	6.51	5.18	5.94
16	18.85	18.12	23.00	24.88	23.99	26.72	3.57	6.02	6.47	6.86
17	18.21	19.73	21.43	24.80	23.35	25.99	1.97	4.95	5.46	6.82
18	19.02	19.87	21.01	24.09	21.31	26.43	6.38	5.33	5.77	5.39
19	16.21	18.43	19.88	26.16	23.47	26.49	5.37	5.11	6.74	5.57
20	20.78	20.16	21.00	25.97	24.59	26.97	5.21	4.50	6.14	6.38

APPENDIX U: SIMULATION DATA FOR NV/NW DISPATCHING RULE COMBINATION ($\alpha = 0.08$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	18.28	19.44	21.84	22.03	4.95	5.01	5.75	6.41	6.94	6.79
2	21.19	21.01	19.83	23.18	5.42	7.50	6.20	8.32	7.66	8.21
3	19.49	22.58	21.88	24.18	3.51	6.06	4.32	6.24	5.45	7.75
4	21.15	22.20	19.51	23.33	5.53	5.05	6.33	9.29	7.38	6.81
5	16.76	23.30	20.47	22.92	5.84	5.03	6.65	7.30	7.80	6.71
6	17.74	19.29	22.58	24.03	4.44	4.45	5.26	6.25	6.45	5.05
7	17.23	24.12	25.26	25.17	5.90	4.29	6.71	8.12	8.04	6.08
8	16.76	18.21	22.78	26.63	5.08	5.74	5.81	5.21	6.72	5.97
9	18.02	19.23	23.56	25.71	6.00	5.56	6.81	6.23	7.94	7.24
10	17.39	22.02	22.78	25.89	4.94	5.46	5.75	7.72	6.88	7.19
11	17.94	18.4	20.69	22.52	5.05	4.58	5.74	5.40	6.97	5.66
12	17.44	18.93	20.74	23.98	3.21	4.01	4.02	5.13	5.15	5.74
13	17.73	20.66	23.56	25.27	3.18	6.55	3.92	7.64	5.10	7.83
14	18.48	20.73	23.14	25.47	3.91	6.36	4.47	7.73	5.60	8.09
15	17.91	17.38	22.97	25.52	2.28	5.09	3.06	3.93	4.19	8.24
16	17.85	18.12	23.89	24.67	4.75	5.49	5.56	5.42	6.69	7.22
17	16.21	17.13	21.43	24.59	4.11	4.76	4.92	4.13	6.00	5.93
18	19.02	20.18	21.01	23.88	4.93	4.65	2.83	6.17	4.01	6.38
19	16.21	18.43	19.88	25.95	2.23	5.26	3.04	5.43	4.15	6.95
20	16.86	22.16	21.54	25.76	5.31	6.34	6.15	5.86	7.25	7.41

APPENDIX V: SIMULATION DATA FOR FV/FW DISPATCHING RULE COMBINATION ($\alpha = 0.14$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	23.02	24.54	24.83	25.87	29.72	32.42	9.11	8.71	10.54	11.25
2	25.79	26.53	23.79	26.91	30.19	33.74	9.14	9.54	10.53	11.83
3	22.24	27.70	25.91	27.63	28.28	33.27	9.17	10.42	11.12	11.81
4	21.86	27.33	23.84	27.13	30.30	34.19	9.13	8.33	11.31	12.63
5	22.51	28.49	24.55	26.76	30.61	32.27	10.04	8.17	10.45	11.28
6	23.08	24.39	25.70	27.78	29.21	30.59	8.01	9.21	10.59	10.37
7	23.02	29.24	29.09	28.74	30.67	31.57	8.79	9.79	9.83	10.57
8	22.51	23.39	26.81	30.43	29.85	34.97	9.73	11.88	7.27	8.24
9	23.28	24.35	27.64	29.53	30.77	32.74	8.82	10.42	9.48	10.23
10	22.16	25.74	26.82	29.61	29.71	32.62	9.68	11.88	11.52	12.71
11	23.69	23.53	24.72	26.36	29.73	31.78	9.56	8.76	9.29	10.29
12	23.21	23.95	25.09	27.78	27.98	31.24	9.01	10.21	8.54	9.92
13	20.14	21.76	27.59	29.77	27.95	33.77	9.51	11.31	7.54	8.76
14	24.18	25.90	27.10	28.81	28.43	34.59	10.71	8.52	10.21	10.85
15	22.96	21.55	28.08	29.32	27.05	33.36	9.62	10.78	10.46	11.27
16	22.07	23.24	27.03	28.45	29.52	32.72	9.58	9.78	7.71	8.47
17	21.96	22.25	25.56	28.39	28.88	31.85	9.09	10.69	9.66	10.43
18	21.77	24.93	25.05	27.69	26.84	35.88	10.96	9.21	10.01	10.55
19	22.01	23.55	23.91	29.35	27.00	32.48	9.63	10.83	9.84	10.54
20	21.72	25.24	25.01	29.51	30.08	32.73	10.81	11.04	11.32	12.77

APPENDIX W: SIMULATION DATA FOR FV/FW DISPATCHING RULE COMBINATION ($\alpha = 0.12$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	25.87	25.37	24.97	26.03	8.31	6.94	8.30	12.53	10.61	11.25
2	20.09	21.01	23.96	27.18	8.78	8.43	7.50	11.39	10.53	11.83
3	23.49	22.58	26.01	28.18	6.87	7.99	8.20	10.91	11.12	11.81
4	21.10	24.20	23.64	27.33	8.89	8.98	8.45	12.21	11.32	12.63
5	22.38	23.30	24.60	26.92	9.20	6.96	9.33	9.94	10.45	11.28
6	20.37	20.83	26.02	27.98	7.80	8.29	10.52	10.23	10.53	10.37
7	23.53	24.12	29.39	29.17	9.26	8.35	8.72	11.13	9.77	10.57
8	21.76	24.58	26.91	30.63	8.44	7.67	10.37	12.31	7.27	8.24
9	20.02	23.47	27.69	29.53	9.36	9.49	11.12	7.75	9.48	10.23
10	25.46	22.02	26.91	29.89	8.30	7.39	10.78	8.32	11.52	12.71
11	21.16	23.69	24.82	26.52	8.32	7.88	11.89	9.31	9.29	10.29
12	22.44	24.39	24.87	27.98	6.57	7.94	10.30	9.84	8.54	9.92
13	21.65	20.66	27.69	29.27	6.54	8.48	8.95	10.10	8.56	8.76
14	22.58	24.73	27.27	29.49	7.02	9.29	9.49	8.14	10.24	10.85
15	27.34	25.93	27.50	29.52	7.64	8.82	10.40	9.01	10.46	11.27
16	21.58	21.08	27.13	28.67	8.11	7.42	8.22	10.92	7.63	8.47
17	22.19	23.31	25.56	28.59	7.47	8.69	6.42	10.85	9.28	10.43
18	21.02	21.87	25.14	27.88	6.43	8.58	10.83	9.23	10.23	10.55
19	23.13	24.34	24.01	29.95	5.59	8.19	9.82	10.01	9.84	10.54
20	24.59	24.16	25.13	29.76	8.67	7.67	9.16	11.40	11.36	12.77

APPENDIX X: SIMULATION DATA FOR FV/FW DISPATCHING RULE COMBINATION ($\alpha = 0.10$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	23.02	24.54	6.72	7.87	8.72	8.28	8.30	12.48	10.61	11.25
2	25.79	26.53	5.68	6.91	9.19	9.74	7.52	11.84	10.53	11.83
3	25.24	27.70	7.80	5.63	7.28	9.27	8.20	10.92	11.12	11.81
4	26.86	27.33	5.73	6.13	9.30	9.78	8.75	12.31	11.32	12.63
5	22.51	28.49	6.44	7.76	9.61	8.27	9.33	9.94	10.45	11.28
6	23.08	24.39	7.59	5.78	8.21	6.59	10.52	10.23	10.53	10.37
7	23.02	29.24	6.98	6.94	9.67	7.51	8.72	11.13	9.77	10.57
8	22.51	23.39	7.17	8.48	8.85	10.97	10.37	12.31	7.27	8.24
9	23.80	24.35	6.53	7.53	9.77	8.74	11.12	7.68	9.48	10.23
10	22.16	27.74	7.71	7.67	8.71	8.62	10.82	8.32	11.52	12.71
11	23.69	23.53	6.61	6.36	8.73	7.78	11.93	9.34	9.29	10.29
12	23.21	23.95	5.98	6.31	6.98	7.24	10.30	9.84	8.54	9.92
13	23.40	25.76	6.48	7.77	6.95	9.77	8.79	10.10	8.56	8.76
14	24.18	25.90	6.99	6.82	7.43	10.59	9.44	9.12	10.24	10.85
15	22.96	21.55	5.97	7.32	6.05	9.36	10.46	9.01	10.46	11.27
16	22.07	23.24	6.92	6.53	8.52	8.64	8.20	10.84	7.63	8.47
17	21.96	22.25	7.45	6.39	7.88	7.85	6.13	10.85	9.28	10.43
18	24.77	24.93	6.94	5.68	5.84	11.86	10.39	9.23	10.23	10.55
19	22.01	23.55	5.80	7.35	6.00	8.64	9.32	10.01	9.84	10.54
20	21.72	25.24	6.90	7.51	9.08	7.98	9.16	11.44	11.36	12.77

APPENDIX Y: SIMULATION DATA FOR FV/FW DISPATCHING RULE COMBINATION ($\alpha = 0.08$)

Material Flow	23		24		25		26		27	
Rule Combination	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW	NV/NW	NV/FW
Replication \ 1	5.78	6.34	6.84	7.03	8.19	8.24	9.22	10.32	10.28	10.65
2	8.59	7.91	5.83	8.18	8.66	9.73	8.42	9.18	10.57	11.41
3	7.99	9.48	6.75	7.38	6.75	9.29	9.12	8.70	10.88	12.65
4	9.60	9.10	5.51	6.21	8.77	8.28	9.37	10.00	11.12	10.47
5	5.26	9.25	6.47	7.92	9.08	8.26	9.38	7.73	10.30	13.38
6	5.87	6.19	7.89	8.98	7.68	6.59	9.31	11.39	9.71	12.21
7	5.73	9.34	6.82	6.83	9.14	7.52	9.64	8.92	9.44	11.83
8	5.26	5.11	6.59	7.21	8.32	8.97	11.29	10.10	11.87	12.39
9	6.52	6.13	7.61	6.79	9.24	8.79	8.39	12.54	9.24	13.32
10	4.96	8.92	8.78	6.79	8.18	8.69	7.58	11.83	10.88	11.67
11	6.44	5.30	6.69	7.52	8.20	7.18	9.71	11.28	10.77	12.78
12	5.94	5.83	6.74	7.98	6.95	7.24	11.22	9.47	8.30	10.27
13	6.15	7.56	7.36	6.48	6.42	9.78	9.84	9.51	10.54	11.65
14	6.98	7.63	8.14	8.25	6.90	9.59	7.52	11.93	9.97	12.77
15	5.70	3.83	7.36	6.37	7.19	9.32	8.73	10.85	10.18	10.95
16	4.35	5.02	8.74	6.72	7.99	8.72	8.94	8.71	9.84	9.57
17	4.71	4.03	7.43	9.59	8.52	7.99	7.34	8.64	9.40	10.91
18	7.52	6.77	7.01	7.27	7.45	7.88	11.75	9.72	9.77	11.18
19	4.71	5.33	5.88	7.84	6.83	8.49	8.19	9.82	10.74	9.66
20	4.37	7.06	6.45	8.49	8.55	8.97	10.58	10.74	11.14	10.32

VITA

Yu-Cheng Shen was born in Taiwan, R.O.C., on May 31, 1964. He received the Bachelor of Science degree in Industrial Engineering from Chung Yuan University, Taiwan, R.O.C. in 1987. From 1987 to 1989, he served in the Artillery Division of the Chinese Army. Immediately after the military service, he went to Rensselaer Polytechnic Institute (RPI) in Troy, NY, to pursue his graduate studies. In December 1991, he received his Master's degree in Industrial and Management Engineering. Since 1993, he had been a graduate student in the Industrial and Systems Engineering Department at Virginia Polytechnic Institute and State University, where he recently received his Ph.D. in Industrial Engineering. His research interests include production planning and inventory control, material handling systems, and facility layout.