TUAN LE ANH

Intelligent Control of Vehicle-Based Internal Transport Systems

Intelligent Control of Vehicle-Based Internal Transport Systems

Intelligent Control of Vehicle-Based Internal Transport Systems

Intelligente besturing van
interne voertuigtransportsystemen

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
Rector Magnificus
Prof. dr. S.W.J. Lamberts
and according to the decision of the Doctorate Board

The public defence shall be held on
Thursday 21 April 2005 at 13.30 hrs

by

Tuan Le Anh

Born at Hanoi, Vietnam

Doctoral Committee

Promotors:	Prof.dr. M.B.M. de Koster
	Prof.dr.ir. R. Dekker
Other members:	Prof.dr. S.L. van de Velde
	Prof.dr.ir. J. Ashayeri
	Prof.dr. Y. Pochet

Erasmus Research Institute of Management (ERIM)
RSM Erasmus University / Erasmus School of Economics
Erasmus University Rotterdam

Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM Ph.D. Series Research in Management 51
TRAIL Thesis series T2005/2, The Netherlands TRAIL Research School

ISBN 90-5892-079-8

Acknowledgements

Doing a Ph.D project in management not only means a lot of hard work but also means plenty of enjoyment. Now, when I am so close to the finishing line, it is the right moment to look back a little bit. It started after I finished my electrical engineering study at the Hanoi University of Technology. At that time, I wanted to have some knowledge on management. Therefore, I took the "Industrial engineering" program of the Asian Institute of Technology and then the "Industrial management" program at the Catholic University of Leuven where I got my master degree. The Erasmus University Rotterdam (EUR) and the Erasmus Research Institute of Management (ERIM) came into my mind just by chance when I was looking for research projects in operations management. For some reasons, at last the research project of "Control of material handling system", which I have been doing during my four years at EUR under the supervision of René De Koster and Rommert Dekker, became my serious interest.

René has been both my first promotor and my daily supervisor. I would like to thank him for welcoming me here in the first place and for his great support and patience during my time at RSM Erasmus University (formerly Rotterdam School of Management/ Erasmus University Rotterdam). His knowledge and experience prove to be invaluable for my research. I would like to thank Rommert for his useful discussions which certainly helped to clarify my research problem. He also pointed out some weaknesses of my study and gave me many useful advices to improve it.

I highly appreciate all committee members for their time and suggestions. Particularly, I would like to thank Prof. Steef van de Velde for his advices on the complexity analysis, Prof. Yves Pochet for his recommendations on the simulation study and Prof. Jalal Ashayeri for his suggestions on literature and the material handling system classification. I also thank other committee members Prof. Albert Wagelmans and Prof. Aart van Harten for taking part in my doctoral committee.

During my four years at RSM Erasmus University, I have had a very pleasant working environment. I would like to thank all of my colleagues at the Department of Management of Technology and Innovation for that. Particularly, I thank Julien and Janneke for their helps and amusing talks, Julien for the amazing "samenvatting", Iris for reading my thesis draft, Kees Jan for his useful advices, Thọ for sharing "snooker time" and for traveling with me to many conferences. I thank Thọ, An, Julien and Eric for working on "interesting" LNMB exercises with me. I also appreciate Robert, Dennis and Ramon for their helps. Robert gave me advices and assistance on my simulation study. Dennis provided me some useful explanations on column generation. Ramon shared with me some CPLEX experiences. I would like to thank Carmen and Tineke for their helps during my studies at RSM Erasmus University. I also acknowledge ERIM for funding my Ph.D research.

Four years have been quite a long time for me with many ups and downs. During that period, many people have been helpful to me. I would like to express my gratitude to my parents and my sister for their supports. I thank my friends (Tùng, Thái and Khôi) for our good time in Germany. I acknowledge all of my friends (just name a few: Thọ, An, Thy, Nga, My Lan, Nam, Kiên, Hùng, Thuỷ …) for sharing "funny" time during my four years. I cannot name all of you here; however, it does not mean that I forget anyone.

For all of you: "*cám ơn*"! Thank you!

Tuan Le Anh - Lê Anh Tuấn
Rotterdam, February 2005

Table of contents

List of figures

List of tables

Chapter 1

Introduction

This thesis focuses on improving the material handling activities in supply chains, particularly within facilities. According to Handfield and Nichols (1999), the costs of the material flow can approach 75% of the total cost of a supply chain. This suggests that improving material handling activities is a subject prone for improvement. The internal flow within facilities such as warehouses, production plants is the main focus of this thesis.

1.1 Material handling

Material Handling (MH) activities can be seen everywhere: raw materials and (intermediate) products circulate in production plants, products are transported within warehouses, distribution centers or between them, luggage is transferred between airport internal locations, etc. These are only few representative examples explaining why material handling plays an important role in real-life. In practice, material handling contributes a big percentage into a product value. This value is estimated to represent between 15% and 70% of the total cost of a manufactured product (Tompkins et al., 2003). Tompkins et al. (2003) also indicate that 20% - 50% of total operating expenses in manufacturing can be attributed to material handling expenses. In general, material handling can be seen as a means to reduce the total manufacturing (or service) cost through more efficient flow control, lower inventories, higher operation efficiency and improved safety. It should be considered as a tool to gain competitive advantage in business.

Material Handling Industry of America (MHIA, 2004a) defines Material Handling as:

> "[…] the movement, storage, control and protection of materials, goods and products throughout the process of manufacturing, distribution, consumption and disposal. The focus is on the methods, mechanical equipment, systems and related controls used to achieve these functions."

This definition is very broad and is close to the definition of logistics. However, logistics focuses more on organization, integration, and utilization. Material handling, on the other hand, stretches more to equipment and physical movements of materials. Material handling is not necessarily only related to manufacturing or distribution of goods. It is also important for service facilities (e.g. airport terminals).

In this thesis, we study material handling systems which are used popularly in facilities such as warehouses, manufacturing plants, airport and transshipment terminals. Thus, we limit the definition of material handling as follows:

> "Material handling is the movement, storage, control and protection of materials, goods and products within a facility. These activities are performed by means of material handling equipment under supervision of a material handling control system."

Material handling systems (MHS) are an integral part of all production and movement systems. A typical material handling system consists of material handling infrastructure, equipment, personnel, a planning and control system, a communication system and products on product carriers (can be seen as single loads). Increasing the efficiency, flexibility and safety of material handling activities and reducing material handling cost are the main objectives of most material handling systems.

In real-life, manual handling of materials is still the current practice in many places. In manual material handling systems, people play a major role in transporting materials. In mechanized material handling systems, machines are responsible for most handling tasks. Mechanized material handling systems using mechanical equipment, such as conveyors or forklift trucks, have significantly higher throughput in comparison with manual systems. The mechanized material handling system also has a higher safety level, particularly, when handling heavy or dangerous materials is required.

An *Automated Material Handling* (AMH) system is a type of mechanized material handling system in which material handling equipment can be controlled automatically. In automated production environments, an AMH can be an integral part of a *Flexible Manufacturing System* (FMS). A FMS refers to a set of computer numerically controlled

(CNC) machines, storage systems for tools and supporting workstations that are connected by a material handing system and is controlled by a central computer (Askin and Standridge, 1993). The most popular type of AMH systems in FMS environments are *Automated Guided Vehicle Systems* (AGVSs). Some modern warehouses also use AMH systems for handling material. Automated storage and retrieval system (AS/RS) is an example of this. An AMH system requires a higher investment than a traditional one. Automated equipment and their control system are also more expensive than manual ones. The cost, which can significantly be reduced, is the personnel cost. In general, AMH systems have much higher throughput than mechanized systems. These benefits, in many cases, justify the high investment in new automated systems.

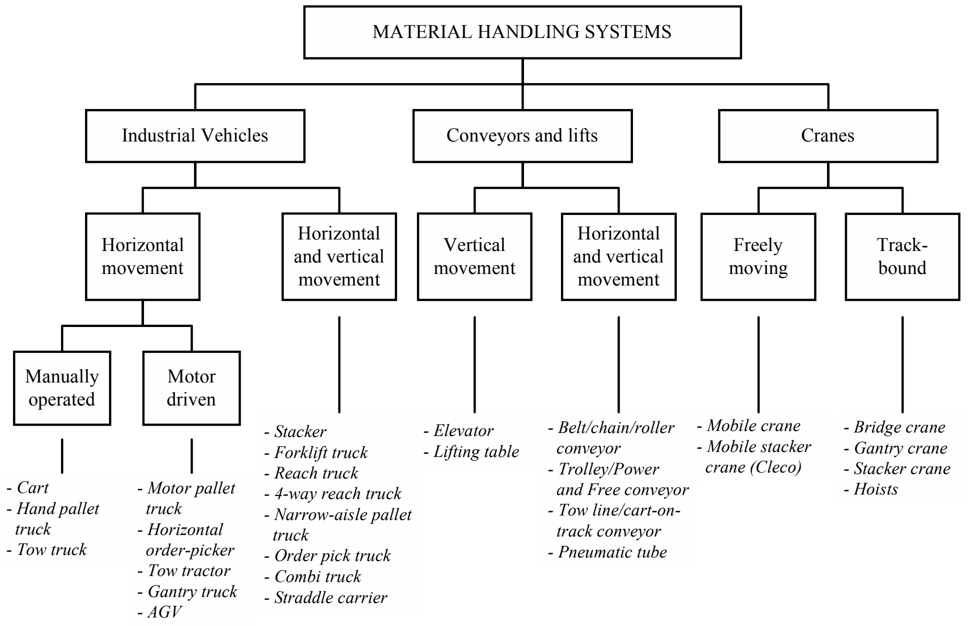


Figure 1.1 A classification of Material Handling Systems

Material handling systems can be classified according to criteria such as the type of material handling equipment, the degree of automation, the guide-path system, etc. Figure 1.1 provides a classification of material handling systems based on the type transportation equipment used (adapted from De Koster, 1995).

Figure 1.2 A forklift truck in a warehouse (left) and AGVs (right) (courtesy of the Kingdom Group and Siemens Dematic)

The first major task of a material handling designer is to select a particular type of material handling system. This task starts by identifying proper material handling equipment based on the facility infrastructure and requirements. In practice, conveyors are frequently used for moving materials of relatively uniform size with moderate to high transportation frequency between a specified set of locations over a fixed path (Askin and Standridge, 1993). Cranes are overhead lifting devices used for intermittent moving of materials varying in size and weight within a fixed space. Industrial vehicles are used for intermittent transport of materials over varying paths. Figure 1.2 presents two typical industrial vehicles which are used popularly in practice.

In practice, the most popular types of material handling equipment are conveying equipment and industrial vehicles. According to the *Material Handling Industry of America*, in the US domestic market, the conveying equipment has the largest shipped value among material handling equipment, follows by industrial vehicles and cranes (MHIA, 2004b).

Van der Meer (2000) identified seven tasks of most material handling systems within a facility: (1) receiving materials; (2) transportation from receiving to storage areas; (3) storage of materials; (4) picking materials; (5) transport (internal) materials between different areas within the facility; (6) adding value to materials or products through customization; (7) shipment of materials. In addition, protection of materials is also important (8). Depending on the particular type of material handling system some tasks might be more important than others. In this research, we focus on a major task (including (2) and (5)) in material handling systems - moving materials internally (by means of an internal transport system). The internal transport system has a crucial role in facilities including warehouses, manufacturing plants, airport and transshipment terminals. The internal transport systems studied in this thesis use industrial vehicles as the means of transport. Among industrial vehicles, we focus on those vehicles which can be guided remotely by a system controller through a communication means such as *Radio-Frequency* (RF), infrared, induction wire and laser. Since these internal transport systems use vehicles as the means of transport, we call them *Vehicle-Based Internal Transport* systems or VBIT systems (or VBITSs).

Next section gives some examples of VBIT systems in some typical facilities in real-life.

1.2 VBIT systems in some typical facilities

1.2.1 Warehouse

A warehouse is a facility which holds inventories such as raw materials, intermediate and finished products. A distribution center is a type of warehouse which is used to store finished products at a distributor before delivering them to customers (wholesalers, retailers, stores, consumers).

In a production warehouse, materials and intermediate products are stored to support manufacturing operations. Finished products can also be held temporarily before being moved to distributors. Production warehouses can be classified into three categories: raw material, in-process inventory and finished good warehouses. A raw material warehouse receives goods from outside sources and ships to internal users. An in-process inventory warehouse receives intermediate products from internal sources and ships to inside users. A finished good warehouse stores end products and ships them to customers. In many cases, these functions are combined in a single facility.

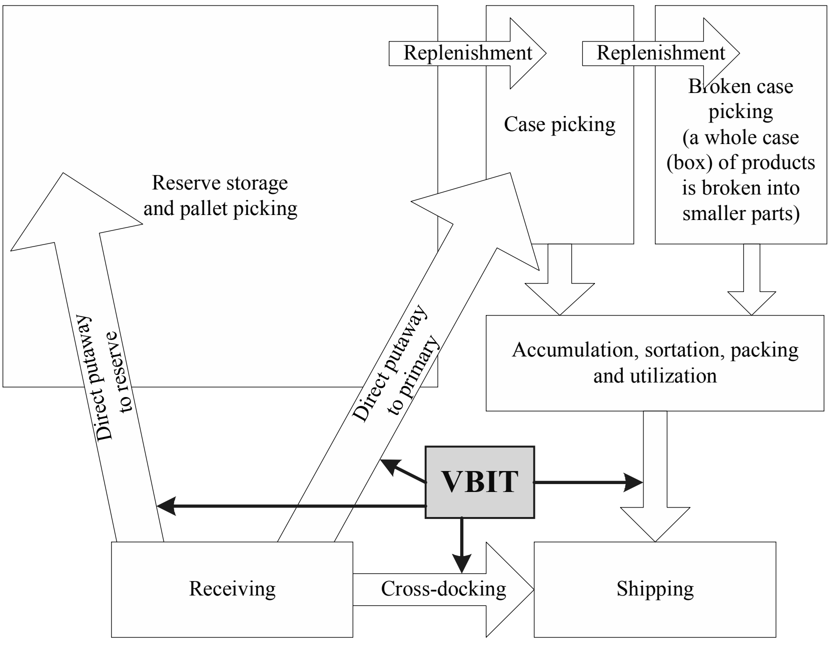


Figure 1.3 Typical warehouse functions, flows and the position of the VBIT system (adapted from Tompkins et al., 2003)

Figure 1.3 shows main activities in a warehouse (Tompkins et al., 2003). The VBIT system's role is to move pallet loads as quick as possible between internal locations such as between receiving and storage areas.

1.2.2 Manufacturing system

As indicated before, VBIT systems play a role in production warehouses. Besides that role, VBIT systems also have another important role in manufacturing processes, particularly in flexible manufacturing systems. A FMS, in general, consists of machines, a part movement system (or AGV system), supporting workstations (e.g. load/ unload station) and a system controller (Figure 1.4). The system controller controls and monitors all operations in the FMS. In a FMS, the AGV (or VBIT) system is responsible for moving parts and tools between machines and a central storage area. In a FMS system, a vehicle generally transports one pallet load which contains one or more fixtures or parts. AGVs in manufacturing environments usually follow wire paths embedded in the floor. Such guide paths are normally one way tracks. The main objective is to serve machines as quickly as possible and avoid deadlocking.

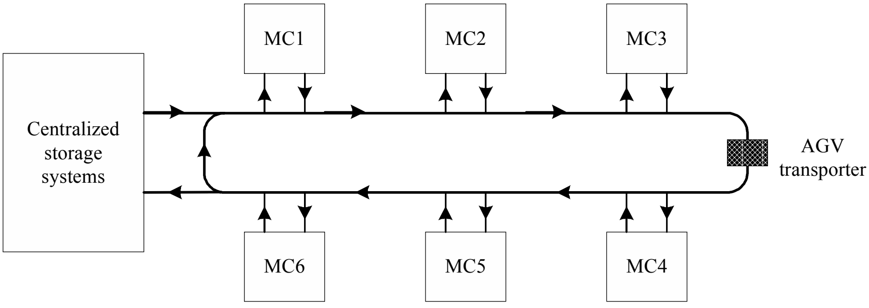


Figure 1.4 An FMS system layout (Tompkins et al., 2003)

1.2.3 Container terminal

A container terminal plays a role as an interfacing node between container vessels and other transportation means. Figure 1.5 shows basic operations at an automated container terminal.

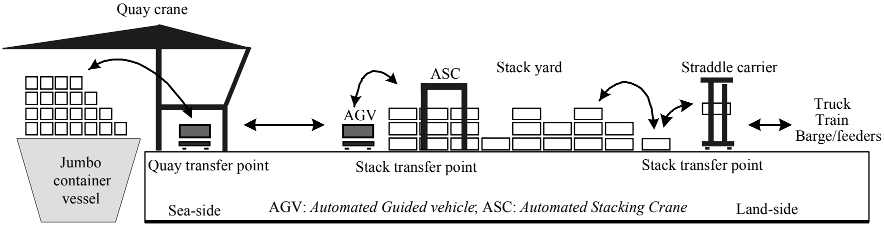


Figure 1.5 Overview of container moves at a typical transshipment terminal (Van der Meer, 2000)

At a container terminal, ships need to be loaded and unloaded. When a ship arrives at the port, the import containers are removed from the ship by quay cranes (QCs). QCs then transfer containers to vehicles (AGV in Figure 1.5) which move containers to the stack. The stack consists of a number of lanes, where containers can be stored for a certain period. The stacking crane (SC - ASC in Figure 1.5) is responsible for moving and stacking containers. After a certain period, containers are retrieved by SCs and transported by vehicles to other transportation modes like barges, trucks, trains. A vehicle can normally carry only one container at a time. More than one container can be transported when multi-trailer vehicles are used. The objective, again, is to move containers as quickly as possible satisfying time-window constrains. More about operations at container terminals can be found in Meersmans and Dekker (2001) and Vis and De Koster (2003).

1.2.4 Airport terminal

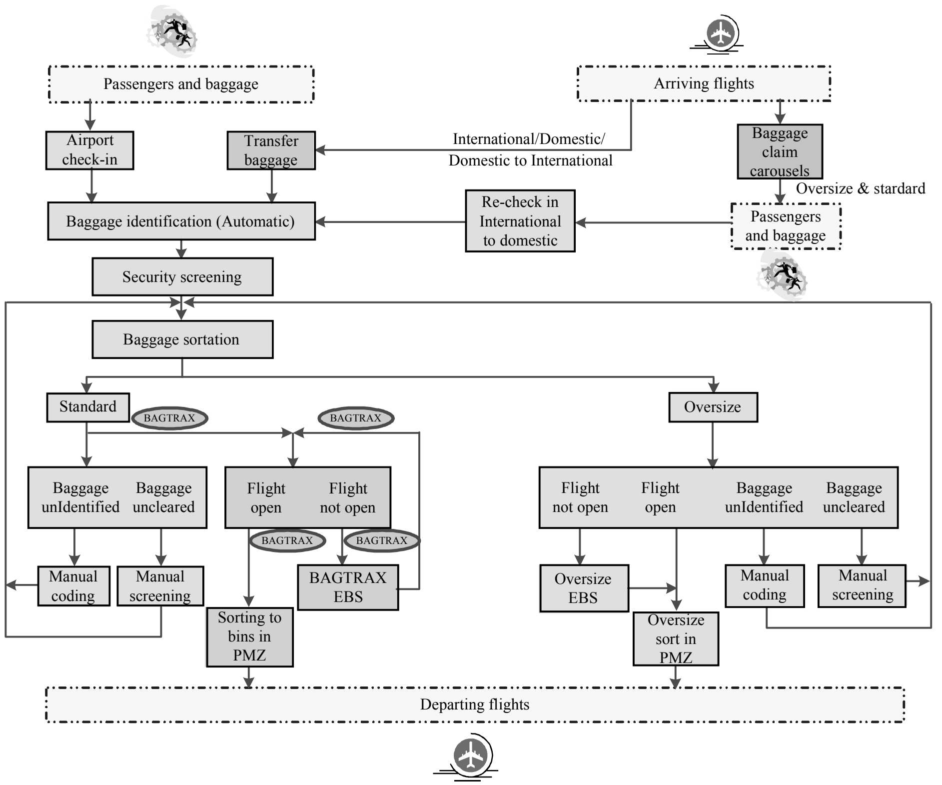


Figure 1.6 A baggage handling system (courtesy of Vanderlande Industries)

An airport terminal can be seen as a type of transshipment terminal in which people and baggage are transferred. Baggage is our main concern here. In an airport, baggage needs to be moved from check-in areas to air planes, transferred from one plane to others etcetera. *Baggage Handling Systems* (BHSs) take care of moving baggage (or luggage) between airport's internal locations. In BHSs, baggage has to be moved as quickly as possible.

Traditional BHSs use conveyor-like systems to transport baggage. Other BHSs at large airports use *Destination-Coded Vehicles* (DCVs) to transport baggage quickly over long distances. This type of BHS is also a type of VBIT system. A DCV can be considered as an automated guided vehicle which has a capacity of one piece of baggage (or bag) and can operate at a high speed (up to 10m/s). This type of BHS requires a higher investment than traditional ones. It costs around $10,000 per meter of track plus another $10,000 per vehicle (Neufville, 1994). Figure 1.6 gives a flow diagram of a baggage handling system using destination-coded vehicles at the Oslo Gardermoen airport. In Figure 1.6, BAGTRAX is the part of the BHS which uses DCVs to move baggage. BAGTRAX can only transport standard bags. Oversized or unidentified bags have to pass other systems for further processing.

In the next section, we define the VBIT system and discuss some key issues in design and control of such systems.

1.3 Vehicle-Based Internal Transport systems

A vehicle-based internal transport (VBIT) system can be defined as follows:

> "*A vehicle-based internal transport system is a transport system that uses (guided) vehicles as the means of transport. Vehicle(s) travel(s) on a closed network within physical boundaries, like the building of the warehouse or limited by physical guide-paths as in the airport baggage handling system. Vehicles are controlled and monitored by a central control system.*"

This research studies VBIT systems (or VBITSs) which use guided vehicles for transportation of loads. Guided vehicles can be automated guided vehicles or person-guided vehicles such as forklift trucks equipped with radio-frequency (RF) terminals. RF terminals provide communication between the central controller and vehicles' drivers.

1.3.1 Issues in design and control of VBIT systems

As discussed before, the main objective of the vehicle control problem in most VBIT systems is to move loads (pallets) as quickly as possible within facilities. In VBIT systems, there are several important issues which need to be taken into account in design and control. These issues include the choice of vehicle guidance and guide-path design, determining the number of vehicles needed, vehicle scheduling and routing, parking and battery management, vehicle collision prevention, deadlock prevention and resolution, safety and the facility information system (or the facility control system). These issues are discussed briefly in this section and are reviewed in more detail in the next chapter.

- **Vehicle guidance and communication**

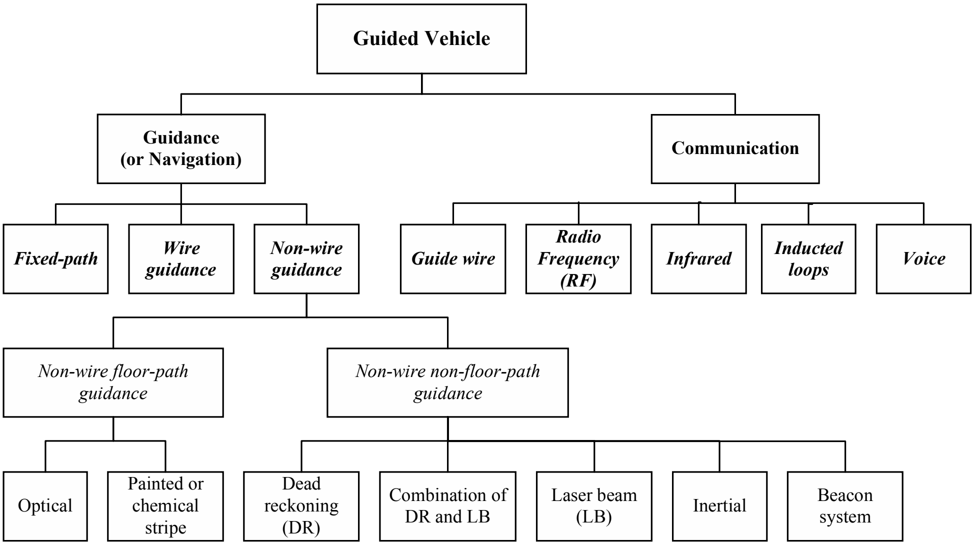


Figure 1.7 A classification of vehicle guidance and communication (derived from Jünemann and Schmidt, 2000 and Tompkins et al., 2003)

The vehicle guidance (or navigation) system finds the paths which a vehicle needs to follow to reach its destination. A communication system provides a means of transmitting information (data, commands, etc.) between vehicles and the control system. Figure 1.7 shows a classification of vehicle guidance and communication methods.

Vehicle guidance

According to Figure 1.7 there are three main types of vehicle-guidance: fixed-path, wire and non-wire guidance.

Fixed-path guidance: a vehicle operating in a fixed-path guidance system follows fixed-tracks (like rail-track) systems. This type of path-guidance is robust to environment interferences, but it is hard to change. It can be seen, for example, in airport baggage handling systems and in many FMSs.

Wire guidance: instead of using fixed-paths, electrical wires buried underground function as guide-paths for vehicles. This type of guidance has similar advantages and disadvantages as the fixed-path guidance and can be found in many environments like FMSs and warehouses.

Non-wire guidance: the non-wire guide-path is a type of virtual guide-paths. Advanced technologies permit to change vehicle guide-paths by updating the guide-path map in the control system and vehicles' controllers. More details on a specific type of vehicle guidance can be found in Jünemann and Schmidt (2000) and Tompkins et al. (2003). Figure 1.8 shows the basic of two modern guidance methods (laser and inertial guidance).

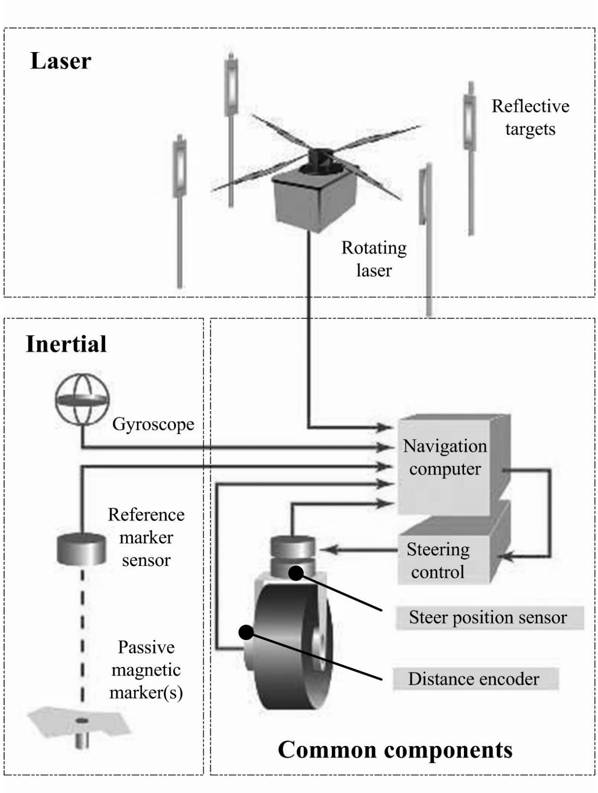


Figure 1.8 Laser and inertial guidance (courtesy of Siemens Dematic)

In laser guidance, a laser source is mounted on top of a vehicle. The vehicle uses a rotating laser from the laser source to locate its position by analyzing the reflective lights received from reflective "targets" attached to the columns, walls or other warehouse structures. A vehicle needs four reflective targets to find its position. Since a laser-guidance vehicle needs to read reflective lights, it is not possible to build high racks in warehouses. The inertial guidance uses floor-flush, magnetized "markers" to provide location reference to the on-board navigation computer. The inertial guidance obtains about the same accuracy as the laser guidance.

Vehicle communication

Figure 1.7 shows five communication types used by vehicles in internal transport environments. Radio Frequency or RF communication (a communication medium by which vehicles and a control system are directed by means of high and low frequency radio transmission directives) provides maximum flexibility in system control. It provides almost constant communication between vehicles and the control system. RF communication is very popular in warehouse environments. Infrared communication (another type of wireless communication) is highly reliable but cannot provide communication continuously. In a system using guide wire communication, data is transmitted through guide wire. This type of communication provides almost the same flexibility as RF. The guide wire communication is mostly suitable for systems using wire guidance. Inductive loops provide another means of point-to-point communication. Voice communication using voice recognition is suitable for person-guided vehicle systems.

The following sections describe some important issues in design and control of a vehicle-based internal transport system.

- **Vehicle guide-path design**

The guide-paths of a VBIT system are usually decided at early stage of the design process. Changing vehicle guide-paths is not an easy task; moreover it requires a significant investment. These make the guide-path design problem critical. In the case of non-wire guidance, the guide-paths can easily be changed by updating the guide-path map in the vehicle and system controllers. However, changing the vehicle guide-path system is still not a daily task. In addition, the guide-path system also affects strongly other processes in VBIT systems such as scheduling, routing, etc. Thus, deciding the guide-path system is a long term decision in the VBIT system design process.

- **Estimating the number of vehicles**

Guided vehicles are expensive, so determining a right number of vehicles are important. Vehicle characteristics such as the guidance method, speed, capacity, battery life, etc. are important factors which need to be taken into account when determining the number of vehicles. The guide-path system also affects the decision on the number of vehicles required.

- **Vehicle scheduling**

The vehicle scheduling system decides which vehicle should transport which load and when. This can be done by solving a complicated optimization model or by assigning vehicles to loads based on some intuitive dispatching (or assignment) rules. Dispatching is related to immediate decisions such as where a vehicle should be sent to at a specific moment. The main goal of most VBIT scheduling problems is to move loads (products, pallets or containers) from pick-up locations to drop-off locations as quickly as possible satisfying time-window constraints. Other criteria can be minimizing the maximum load waiting time, the maximum number of items in critical queues or meeting due times. The scheduling system may also perform the routing task which specifies which route a vehicle should take to perform its job. Normally, a vehicle should take the shortest path to its destination. However, in highly congested environments, vehicles may have to take alternative routes to avoid congestion and collision with other vehicles.

Vehicles can be dispatched centrally or decentrally. The main difference between them is that in a decentralized system, a vehicle operates as an independent agent based on local and limited information. While in a centralized control system, a system controller is responsible for dispatching vehicles using available information from all possible sources. In the perfect scenario, all information about load arrivals is known in advance for the whole planning period (e.g. a day) and the vehicle travel time is deterministic. In that case, the vehicle schedule can be determined offline in advance. In practice, we may know some information about load arrivals, but this information is incomplete or unreliable. In addition, travel times are not deterministic, vehicles can be broken down or delayed by some reasons. Therefore, the offline optimal schedule makes no sense in practice. Dynamic vehicle scheduling using a rolling horizon is a solution to cope with the stochastic nature of the environment. This approach schedules vehicles for a (short) fixed time horizon (T) and a new schedule is generated after a certain execution period (shorter than T) for the next planning horizon T. Another solution is using dispatching rules to control vehicles. Dispatching rules are simple and easy to implement in most cases. However, if advance load arrival information is known, they are outperformed by (dynamic) scheduling approaches.

The control mechanism can be different between automated guided vehicles and person-guided vehicles. In an automated guided vehicle system, the control system has full control of vehicles. The control system assigns loads to vehicles and also decides upon their routes. For person-guided vehicles such as forklift trucks equipped by RF terminals, the control system informs which load a vehicle needs to transport but the vehicle’s driver may decide the route that the vehicle should take to reach the load. This makes the driving time in person-guided vehicle systems uncertain.

The truck scheduling problem vs. the VBIT scheduling problem

The truck scheduling problem in external transport systems shares many similarities with VBIT scheduling problems. Similarly to guided vehicles in VBIT systems, trucks in external transport systems have to pick-up loads at some locations and deliver them at other locations satisfying loads' time-windows. However, these two problems are not exactly the same. The main differences between them are:

- The objectives of the two problems are different. Minimizing the average load waiting time is the most important objective of a VBIT scheduling problem while minimizing the vehicle travel distances and the number of required vehicles are more relevant for external transport systems.
- Travel distances (time) in VBIT environments are much shorter. This leaves little time for scheduling vehicles. Therefore, scheduling algorithms for VBIT systems s should perform quickly.
- Advance information about load arrival in VBIT systems is less reliable than in external transport systems. This leads to a shorter planning horizon and a higher rescheduling frequency.
- Operating layouts of VBIT systems are quite different from external transport systems. Working environments for vehicles in VBIT systems are condensed in comparison with those of external transport systems. In addition, unidirectional paths are popular in VBIT environments.
- In VBIT systems, the vehicle blocking and congestion possibility and their possible impacts on the objective are higher than in external transport systems.
- A much higher load arrival rate is normally encountered in VBIT systems.
- The battery-charging problem may have an impact on the VBIT scheduling problem.

▪ **Parking and battery management**

When a vehicle becomes idle and does not have any assigned task, it moves to a parking location. Parking locations need to be located efficiently to reduce the response time of vehicles to new coming jobs. The main criteria for the parking-location design problem are minimizing the average and maximum response times of vehicles to new load arrivals.

Vehicles cannot operate continuously without charging or swapping their batteries, as batteries have to be recharged after a certain operating period. The main problems to be considered for battery management are: where to locate battery charging stations and when vehicles should go for battery charging or swapping. Naturally, battery charging stations should be located at or next to parking locations. In practice, a company selects fixed locations in its facility for battery stations. A vehicle should go to a battery station when its battery nearly runs out. The decision that needs to be taken is when a vehicle should be sent to which charging or swapping station.

- **Vehicle collision, safety and system deadlock**

Vehicle collision and deadlock may cause serious problems for VBIT systems. Deadlock may happen in some situations such as when two vehicles arrive at a crossing point at the same time or when two vehicles block each other on a bidirectional path. In AGV systems where vehicles travel under control of a central controller without human interference, a minor deadlock between two vehicles may block the whole system. To avoid the deadlock situation, in general, zone control is implemented to prevent two or more vehicles enter one zone at the same time. Most AGVs are equipped with some type of safety sensors, e.g. laser sensors to detect the distance to objects in front of them. The safety sensors help avoiding collisions. A vehicle stops if it is too close to the vehicle in front of it or when it strays away from the guide-path. In person-guided vehicle systems, the driver can resolve most deadlock situations. Still, collisions sometimes happen.

- **Facility information (control) system**

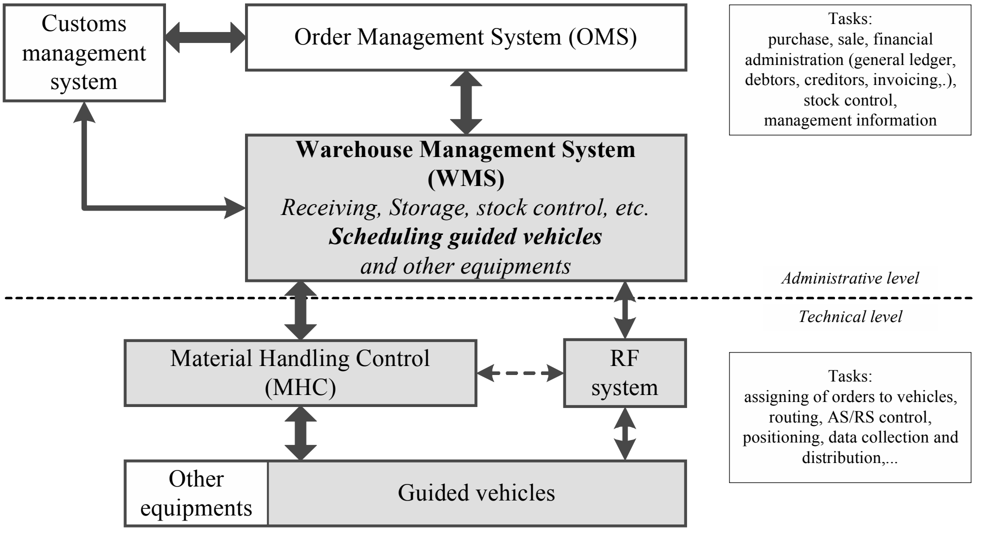


Figure 1.9 A hierarchical structure of a facility information system in a typical warehouse (adapted from De Koster and Neuteboom, 2001)

Figure 1.9 shows a typical information system structure of a distribution center in which a Warehouse Management System (WMS) controls the underlying mechanical systems including vehicles and the RF system. A WMS is a typical shop floor control (SFC) system, which controls the processes “on the floor”. The WMS also provides an interface with up-stream information systems such as an Order Management System (a part of higher-level Enterprise Resource Planning systems). The order management system (Figure 1.9) is mainly responsible for longer-term planning issues such as purchasing, sale, etc. The WMS controls all flows in a facility (e.g. a distribution center) and also performs some planning tasks such as stock control. The WMS determines what has to be transported and when (released time and due time). The WMS also does prioritizing and sequencing of jobs. The material handling device control system directly controls vehicles. MHC in connection with the WMS provides operational control of VBIT systems. The RF system provides communication between vehicles and the WMS.

In a modern VBIT system, the control system can monitor vehicle positions continuously using sensors and wireless communications (e.g. RF communication). In less advanced VBIT systems such as a person-guided vehicle system in a warehouse, vehicle positions cannot be monitored continuously. However, the system controller knows the next destination of a vehicle and can predict the vehicle’s arrival time at its destination. The driver informs the control system when (s)he reaches the destination, by scanning the location or confirming the assignment.

After briefly reviewing the most important issues in design and control of a vehicle-based internal transport system, section 1.4 gives some reasons why this research is important for both theory and practice.

1.4 Research motivation

This thesis concerns the design and control processes of vehicle-based internal transport systems. The scheduling and dispatching problems are the main focuses. Dispatching involves instantaneous decisions. A dispatching decision is made when (a) a vehicle drops off a load; (b) a vehicle reaches its parking location or (c) a new load arrives. Scheduling involves a longer planning horizon. At scheduling moments, the scheduling system makes a plan for all vehicles during the planning horizon. A scheduling plan includes the load sequences which vehicles should transport, the corresponding pick-up and delivery times and also the routes that vehicles should follow. Scheduling decisions are made less frequent than dispatching decisions. A dispatching system may be seen as a scheduling system with a zero planning horizon. From a practical point of view, it is interesting to know which vehicle-control approach (scheduling or dispatching) is more efficient and effective, why it is so and under which circumstances which approach is better.

In literature, the most popular type of VBIT environments studied is the manufacturing environment. Other environments such as warehouses are nearly forgotten. We also noticed that in literature most studied VBIT systems are simplified systems with unrealistic assumptions (Van der Meer, 2000; Le-Anh and De Koster, 2004a). They are not good representations of real-life systems. Real-life VBIT systems, particularly in warehouse environments (e.g. the ones studied in this thesis), are more complicated. In addition, there is no guarantee that good dispatching rules for (unrealistic) environments will also perform well for real-life environments. Thus, it is important to study the performance of vehicle dispatching rules in practice. It is interesting to see if the best dispatching rules in literature also perform well in real-life environments. It is also important to find robust dispatching rules which are applicable for different working conditions and different environments. We will cope with these challenges in this study.

Studies from literature also reveal that dispatching is the most popular vehicle control approach for VBIT systems (Le-Anh and De Koster, 2004a). There are some studies investigating the scheduling problem in VBIT systems, but only a few of them consider the dynamic scheduling problem. Therefore, it is very important to enrich the knowledge on the possible contributions of different dynamic vehicle scheduling approaches for VBIT systems.

This research has three main objectives. Firstly, we want to evaluate the performance of several well-known dispatching rules in literature for two real-word cases and rank them based on their performance. On the basis of this performance ranking, we can suggest dispatching rules for implementation in practice. Secondly, we aim at deriving some new and robust dispatching rules for practice and also for different types of environment. Finally, we propose several dynamic scheduling approaches for VBIT systems and compare their performance with the best dispatching rules in existence. We also investigate the impact of several factors such as the guide-path layout, the load arrival rate and variance and the amount of pre-arrival information on vehicle control (dispatching and scheduling) approaches. Based on experimental results, we give some suggestions on when and where we should apply a specific vehicle control approach. This work started as a follow-up research of Van der Meer (2000). In this research, we make a better classification of dispatching rules. We also introduce some intelligent dispatching rules and several dynamic vehicle scheduling approaches and compare their performances.

By fulfilling these main objectives, this thesis has following key contributions: (1) evaluating and ranking the performance of commonly used dispatching rules such as the nearest-vehicle-first rule for two real-world environments; (2) proposing some new and efficient dispatching rules; (3) adapting dispatching rules for a new environment (VBIT systems using a large number of vehicles); (4) proposing dynamic vehicle scheduling algorithms (dynamic scheduling approaches using a rolling horizon and a look-ahead dynamic assignment algorithm) and proving their superiority to dispatching rules in VBIT environments; (5) elaborating impacts of the load arrival rate and variance, the guide-path layout an the load pre-arrival information on the system performance. In addition to the main contributions described above, this thesis also provides a comprehensive review on design and control of VBIT systems. Most key related issues including guide-path design, estimating the number of vehicles required, vehicle scheduling, idle-vehicle positioning, battery management, vehicle routing, and deadlock resolution are discussed.

The simulation approach is chosen for dispatching-rule experiments. The main advantage of using simulation is that most complex real-world systems, which cannot be formulated as mathematical systems, can be modelled. However, comprehensive experiments are required to support simulation results. To describe two real-life cases as close as possible to the real-life situations, we selected the AutoMod™ (a simulation package specialized in modelling material handling systems) for the modelling purpose (Brooks Automation, 2002). Most of characteristics of vehicles and VBIT systems can be described in AutoMod™. This software contains some standard vehicle dispatching rules, which are used popularly. This software also provides a very good visual animation tool for debugging and verification of simulation models, and another tool for doing statistical analysis. Less flexibility in implementation of complex and non-standard vehicle control rules is the main disadvantage of AutoMod™. All dispatching rules, including case-specific dispatching rules, several good dispatching rules obtained from literature and some new dispatching rules have been implemented in simulation models. A combination approach (of simulation and optimization) has been selected for evaluating the performance of dynamic vehicle scheduling approaches.

1.5 Outline of the thesis

In this section, we provide an outline of this thesis and give some brief information about each chapter. Chapter 2 is based on Le-Anh and De Koster (2004a). This chapter presents a literature review on key issues on design and control of a VBIT system. In this literature review, we discuss and classify important models and results from key publications in literature on VBIT systems, including often-neglected areas, such as idle-vehicle positioning and battery management.

In chapter 3, we experiment with some simple and well-known dispatching rules and company-specific dispatching rules for two real-life cases. Simple dispatching rules, such as shortest-travel-distance-first or modified-first-come-first-serve rules can be implemented easily. This chapter briefly describes two real-life cases, simulation models and dispatching rules. Furthermore, we rank dispatching rules according to their performance (mainly based the average load waiting time). A sensitivity analysis is also provided to examine the behavior of dispatching rules under different vehicle-utilization levels. This chapter is partly based on De Koster et al. (2004).

Several more advanced (or complex) dispatching rules, such as dispatching rules with vehicle reassignment or multi-attribute dispatching rules are evaluated in chapter 4. This chapter uses the two simulation models in chapter 3. In this chapter, we rank complex dispatching rules and two benchmarking rules from the previous chapter. The main criterion is also minimizing the average load waiting time. This chapter is partly based on our research in Le Anh and De Koster (2004c) and Le-Anh and De Koster (2004b).

Chapter 5 is based on Le Anh and De Koster (2004e). In this chapter, we adapt some good dispatching rules from the previous two chapters and from literature for a specific type of VBIT environment: VBIT systems with many vehicles. We model two VBIT systems and dispatching rules in AutoMod^{TM}.

In chapter 6, we study several dynamic vehicle scheduling approaches for VBIT systems. Some good and quick heuristics are introduced for solving the offline scheduling problem. We solve the online (real-time) scheduling problem using dynamic scheduling approaches. We then evaluate the performance of these dynamic scheduling approaches for two typical warehouse layouts. This chapter is based on Le-Anh and de Koster (2004d).

In chapter 7, we summarise the main findings of this research. We also give general conclusions and suggest some directions for further research.

Chapter 2

Literature review

In this thesis, we study vehicle-based internal transport systems in facilities such as warehouses, manufacturing plants. Hence, it is interesting to have some knowledge on the history of VBIT systems in such facilities. The warehouse has a very long history. In early writings, man stored their excess food and kept animals for emergency surplus. At the early stage, warehouses were operated manually. During World War II, the forklift truck and wooden pallet were introduced for mechanized warehouses. The system using forklift trucks to move goods within a warehouse can be seen as a vehicle-based internal transport system. Automated guided vehicle systems were originally designed to support flexible manufacturing systems, which were introduced during the 1970s. The first major published works on AGV systems can be traced back to the early 1980's, starting with papers of Maxwell and Muckstadt (1982) and Egbelu and Tanchoco (1984), Egbelu and Tanchoco (1986). The literature on AGV systems has been enriched since by a huge number of publications. However, general VBIT systems have not been received much attention so far.

As mentioned in the previous chapter, VBIT systems using guided vehicles are the main focus of this thesis. VBIT systems using guided vehicles can be classified into two main types: AGV and person-guided vehicle systems. Most related studies on VBIT systems in literature concern automated guided vehicles, so in this chapter we mainly look at AGV systems and mention impacts of person-guided vehicles when necessary. In details, we discuss key issues related to design and control of VBIT systems. These issues include guide-path design, estimating the number of vehicles required, vehicle scheduling, idle-vehicle positioning, battery management, vehicle routing and deadlock resolution. They belong to different levels of the decision-making process. The guide-path design problem can be seen as a problem at strategic level involving the longest planning horizon. The decision at this stage has a strong impact on decisions at other levels. The choice of vehicles is also important and directly affects the guide-path design problem and the estimation of the number of vehicles required. The type of vehicles can be provided as data for the guide-path design problem or the two problems (selecting a type of vehicle and guide-path design) should be addressed at the same time. Issues at tactical level include estimating the number of vehicles, scheduling vehicle, positioning idle-vehicles and, managing the battery-charging scheme. Finally, vehicle routing, deadlock prevention and resolution problems are addressed at the operational level. The deadlock resolution problem can be put in a finer level: real-time control level. Many of these issues are mentioned in several review papers: Ashayeri (1989), Co and Tanchoco (1991), King and Wilson (1991), Sinriech (1995) and Qiu et al. (2002). During the design and implementation process, some interactions and iterations can be seen between steps. For example, the type of the guide-path system directly influences the number of vehicles required and the complexity of the vehicle scheduling system.

Traditional AGV systems use fixed guide-paths for vehicles. Modern AGV systems differ from the classic ones as described for instance in the books of Jünemann and Schmidt (2000) and Tompkins et al. (2003) in several aspects. Rather than using fixed paths, many modern AGVs are free-ranging, which means their preferred tracks are software programmed, and can be changed (relatively) easily when new stations or flows are added. A second difference is in the way they can be controlled. Agent technology allows decisions to be taken by these smart vehicles that in the past were taken by central controllers. This leads to adaptive, self-learning systems and is particularly appropriate for large, complex systems with many vehicles and much potential vehicle interference. These developments do not imply that the traditional decision-making problem has become obsolete. Rather, they lead to new challenges for research. VBIT systems using person-guided vehicles share some similarity with modern AGV systems, since person-guided vehicles are also free-ranging. In such system the driver may decide which path vehicle should take. The control system guides vehicles through a RF terminal. In this chapter, we both discuss the traditional AGV system decision-making problems and the impact of free-ranging guided vehicles (modern AGV and person-guided vehicles) on decision-making.

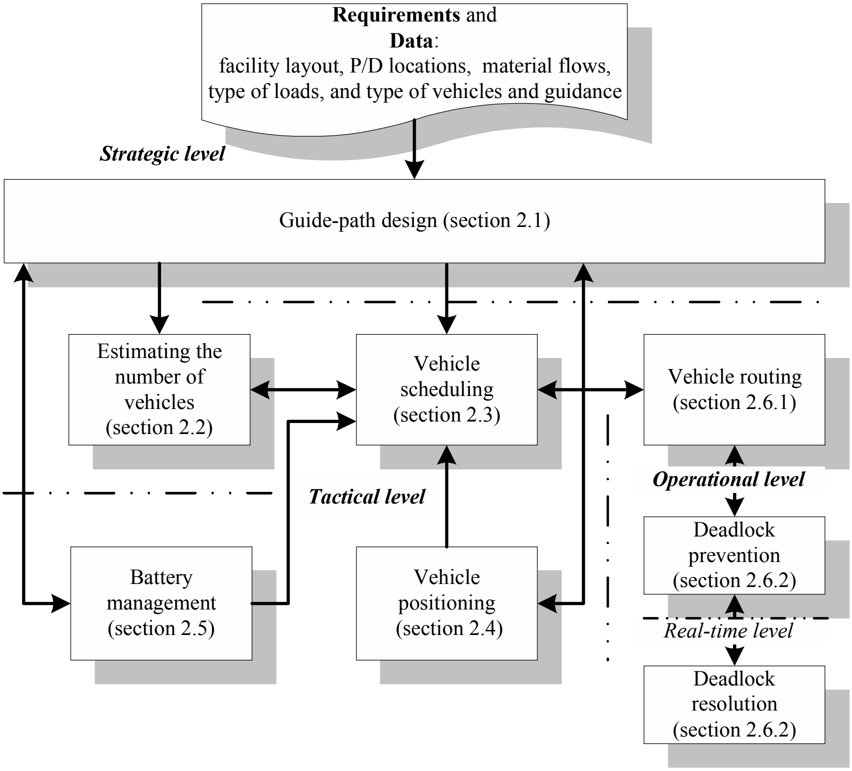


Figure 2.1 Issues in the VBIT system design and control and their interactions

Figure 2.1 shows key issues in the process of design and control of a VBIT system. This process assumes that the facility layout is given and the task is to design a VBIT system to support facility operations. In Figure 2.1, depending on the perspective from which we look at the design and control processes, a problem may belong to different levels. For example, the issue of selecting an appropriate scheduling system belongs to the tactical level, but the scheduling decision belongs to the operational level. Later in this chapter, we discuss key issues in VBITS design and control (Figure 2.1) in details. Obviously, the guide-path system directly affects other decisions at the tactical level. For example, if the tandem guide-path system is selected, the number of vehicles required equals the number of loops and some simple dispatching rules can be used for dispatching vehicles. In this case, an idle vehicle can park any where on its loop. In general, a battery station needs a location in the facility. The parking location and the location for battery-charging stations impose restrictions on the guide-path design problem. The vehicle requirement and the scheduling system less likely influence the guide-path system. When a deadlock-free routing algorithm is chosen, real-time deadlock resolution is not an issue. On the contrary, a system with a good real-time deadlock prevention system does not require a complicated and conflict-free routing algorithm. Simulation is an useful tool to evaluate the performance of designed systems (Ashayeri et al., 1985; Ashayeri and Gelders, 1987).

In the next section, we discuss the guide-path design problem. The issues and models for the facility layout design problem can be found in Askin and Standridge (1993) and Tompkins et al. (2003). Ashayeri and Gelders (1985) provide a review on warehouse design and optimization problem. The facility layout design problem is not a subject of this study.

2.1 Guide-path design

The design process of a VBIT system using guided vehicles (or a guided vehicle system) starts by choosing the right type of vehicles (if the type of vehicle is not given). A commonly used approach for selecting the vehicle type for a guided vehicle system is using a knowledge-based (or expert) system (Malmborg et al., 1987). The next step is to find an appropriate guide-path system for vehicles. The guide-path normally follows the existing aisles in the facility. The integrated problem of designing the facility layout and the guided vehicle system is too complicated to tackle, so the common approach is solving these two problems separately. To obtain better solutions, we may solve these two problems iteratively.

Most published works on the guide-path design problem assume that facility layout and locations of pick-up/ delivery (P/D) stations are given and fixed. The main problem is to decide the connections or guide-path segments to be included in the solution. In some cases, the number of parallel lanes of a connection is to be decided as well. This optimization problem also needs the material flows between departments in the facility. This information is used to construct a "from-to" flowchart which is necessary for the guide-path design problem. In a network flow model, vehicle guide-paths are usually represented such that aisle intersections; pick-up and delivery (P/D) locations can be considered as nodes on a graph connected by a set of arcs. The arcs describe the paths that vehicles can follow when moving from node to node. Directed arcs indicate directions of vehicle flows. Cost can be assigned to each arc representing the distance between the two end points of a segment or the time required by a vehicle to travel along the arc. The network-flow model can be translated to a 0-1 integer optimization model. The main objective of a guide-path design problem is minimizing the total vehicle travel distance. Information shortage is an important problem for guide-path design. For example, the flow of materials within a warehouse can be changed over time and it is difficult to estimate.

Guide-path systems can be classified roughly by the characteristics indicated in Table 2.1. The flow topology describes the complexity of the guide-path network. In the simplest case, the guide-path system consists of only one single loop. Several loops grouped together form a tandem configuration. A conventional topology is a complicated network with paths, crosses, shortcuts and junctions. A path segment in a network may contain only one lane or few parallel lanes. Vehicles can travel a lane in only one direction (unidirectional) or both directions (bidirectional).

Table 2.1 Characteristics of guide-path systems

Flow topology	Number of parallel lanes	Flow direction
Conventional	Single lane	Unidirectional flow
Single-loop	Multiple lanes	Bidirectional flow
Tandem		

Selecting an appropriate type of the guide-path system is important. Unfortunately, there is no guideline for it. The guide-path type is normally chosen based on the characteristics of a facility and the designer's experiences. An expert system can be useful to support the guide-path system selection process. After choosing an appropriate type of guide-path system, the designer can use a suitable (mathematical) model to obtain the best possible guide-path system. In practice, conventional guide-path systems can be seen regularly in warehouses and distribution centers (De Koster et al., 2004); single-loop systems are used, for example, in cross-dock centers. Tandem configuration may be more appropriate for manufacturing environments where workstations are grouped into manufacturing cells.

2.1.1 Performance criteria

Beamon (1998) describes several important criteria for designing guided vehicle systems such as vehicle travel time, vehicle utilization, queue length, and material handling cost. The most common performance criterion for guide-path design is minimizing the total vehicle travel distance corresponding to a given layout and flows (Gaskins and Tanchoco, 1987; Kaspi and Tanchoco, 1990). Kaspi et al. (2002) include both the vehicle loaded and empty travel times in the objective function. Lim et al. (2002) use the total vehicle travel time (including the loaded and empty vehicle travel times and waiting time caused by congestion or vehicle interferences) as the objective function. Several authors use multiple objectives. Kim and Tanchoco (1993) consider the travel cost and the cost of each path segment. Chen et al. (1999) use the total vehicle travel time and the trip failure rate as performance measures in their model. Talbot (2003) uses the required number of vehicles and the guide-path length to measure the system performance.

The next three sections review the design problem of the three most popular guide-path systems in literature.

2.1.2 Conventional guide-path system

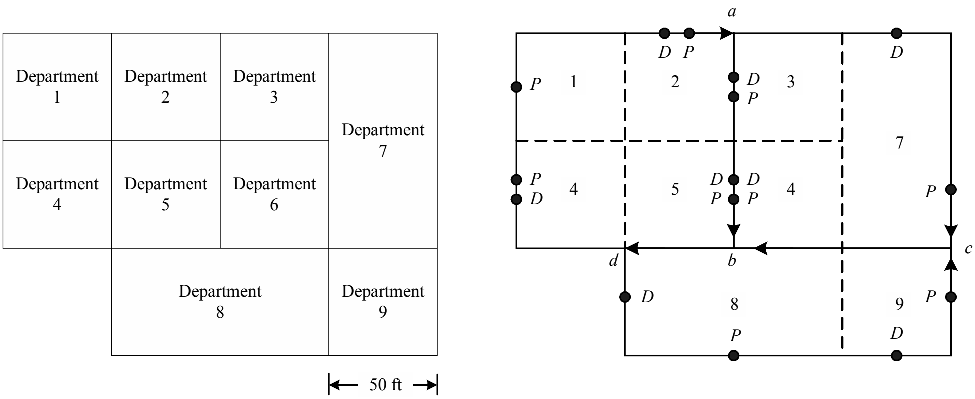
The conventional guide-path system can be divided into two main categories: unidirectional and bidirectional systems.

❑ **Unidirectional guide-path system**

Unidirectional conventional guide-path systems are popular in practice, particularly in warehouses or distribution centers (Figure 1.1). An example of a facility layout and the corresponding from-to chart are given in Figure 2.2 (left) and Table 2.2. Figure 2.2 (right) shows an alternative guide-path system corresponding to the layout on the left. In some cases, we do not know all information about material flow in the system. In this case, this information may need to be estimated or predicted.

Table 2.2 Interdepartmental flows (from-to chart)

From-To	1	2	3	4	5	6	7	8	9	*Sum*
1	—	40	25	30	10	10	20	5	10	150
2		—	40		30		10	10		90
3			—				50		10	60
4		5	10	—		10				25
5				100	—					100
6				60		—				60
7						40	—		40	80
8				10		5		—		15
9					60				—	60
Sum	0	45	75	200	100	65	80	15	60	640



(a) General facility layout *(b)* A path alternative

Figure 2.2 An example of a facility and the corresponding guide-path system (Askin and Standridge, 1993)

Gaskins and Tanchoco (1987) formulate the guide-path design as a 0-1 *integer programming* (IP) model. Their model aims at finding the guide-paths (or flow-paths) which minimize the total vehicle loaded travel time. Gaskins and Tanchoco (1987) use the path distance to describe the travel distance along a feasible path from one node to another. There are three main types of constraints in their model: (1) ensuring that not any node becomes a sink node; (2) ensuring that not any group of nodes becomes a sink; (3) ensuring the shortest path is taken (optional). The model of Gaskins et al. (1989) selects both the number of parallel paths (lanes) to include in the guide-path system and the paths' directions. The vehicle empty travel time can be incorporated into the models of Gaskins and Tanchoco (1987) and Gaskins et al. (1989) by modifying the corresponding from-to chat.

Goetz and Egbelu (1990), Kaspi and Tanchoco (1990), Sinriech and Tanchoco (1991), Kim and Tanchoco (1993) propose several improved 0-1 integer programming models for the guide-path design problem on the basic of the Gaskins and Tanchoco (1987) model. However, the size of the guide-path design 0-1 IP model still can be huge for practical problems. To speed up the solution procedure, Goetz and Egbelu (1990) focus on only the major flows between departments and Sinriech and Tanchoco (1991) consider only intersection nodes in their branch-and-bound algorithm.

Kaspi et al. (2002) propose an improved formulation for the guide-path design problem by explicitly incorporating the vehicle empty travel in the objective function and reducing the number of binary variables. Kaspi et al. (2002) solve this model using a branch-and-bound depth-first search algorithm. The empty vehicle flow is computed during the execution of the search algorithm by solving a transportation problem to distribute empty vehicles from the group of delivery stations to the group of pick-up stations to minimize the total distribution flow.

Besides empty vehicle travel time, time lost caused by vehicle interference also impacts the quality of solutions. Lim et al. (2002) consider total vehicle travel, including the empty and loaded vehicle travel times and time lost caused by congestion and vehicle interference in their design model. They estimate the total vehicle travel time using the Q-learning technique (a process of learning how to match states with actions in order to maximize a numerical reward). They show that their results are superior to those of Kim and Tanchoco (1993). Obviously, the quality of their solution depends heavily on the accuracy of travel times computed by the Q-learning process.

A more complicated problem is tackled by Johnson and Brandeau (1993, 1994), Al-Sultan and Bozer (1998). Their models select both path configuration and P/D stations at a same time. Al-Sultan and Bozer (1998) use a simulated annealing heuristic to solve the guide-path design problem. In their paper, they also note that the model of Gaskins and Tanchoco (1987) may generate infeasible or non-optimal solutions. Johnson and Brandeau (1993, 1994) use the benefit of an AGV system and fixed cost of setting a pick-up/delivery (P/D) station as the objective function instead of total vehicle travel distances. Johnson and Brandeau's (1993) model also determines the number of vehicles required to warrant a service level (expected time until a workstation is replenished from the central depot). The pool of vehicles is approximated by an M/G/c queuing system. They formulate the problem as a 0-1 IP model and solve it using a branch-and-bound algorithm.

In general, besides P/D stations, parking and battery-charging locations need to be taken into account in guide-path design models as well. Moreover, faster solution approaches are still needed.

- **Bidirectional guide-path system**

The conventional bidirectional guide-path system is not popular in material handling systems, although it can result in a higher productivity than the corresponding unidirectional one. The main reason is that the control problem in such systems becomes very complicated. This problem can be resolved by using dual unidirectional lanes. However, the dual lanes system needs more space and is more costly. In literature, there are only few studies on the conventional bidirectional guide-path system (Egbelu and

Tanchoco, 1986; Gaskins et al., 1989). Egbelu and Tanchoco (1986) provide a guideline for design of single-lane bidirectional guide-path systems. Gaskins et al. (1989) propose a model for a bidirectional guide-path system in which the travel distance and the number of lanes are minimized. The number and direction of lanes are decided in their model. Their model assumes that the capacity of each lane and the maximum number of parallel lanes are given and fixed.

Bidirectional guide-path systems are particularly used in systems where vehicle interference rarely happens such as in tandem guide-path systems (section 2.1.4).

2.1.3 Single-loop guide-path system

The main difference between the single-loop and the conventional guide-path system is that in the single-loop layout, vehicles travel in only one loop without any shortcut or alternative routes (e.g. the loop in the left part of Figure 2.3). The travel mode in the single-loop system is usually unidirectional. Bidirectional traveling is possible but, in this case, vehicle interference is likely to happen. Vehicles in single-loop systems can be controlled by simple dispatching rules such as first-encountered-first-serve (FEFS), implying that an empty vehicle should pick-up the first load it encounters. Tanchoco and Sinriech (1992) propose an optimal procedure to design a single-loop system. The main objective is to find "best" single-loop guide-paths and to locate P/D stations along the loop. Tanchoco and Sinriech's (1992) procedure consists of five components:

(1) An IP formulation is used to find an initial valid loop (a valid single-loop problem - VSLP) - a valid loop contains at least one segment for each department in the facility layout.
(2) A procedure (find all single loops - FASL) enumerates all possible valid single-loop guide-paths using a two-phase approach. The first phase creates new valid single-loops by expanding the initial loop. The second phase generates more valid loops by contracting the last loop in the previous phase.
(3) Loop-elimination rules are used to reduce the numbers of candidate loops (inferior loops are eliminated).
(4) A model determines locations of pick-up and delivery stations for each department along a single-loop path, by solving a mixed integer programming (MIP) problem (a single-loop station location problem - SLSLP). The objective is minimizing the total flow times in the system.

(5) A lower bound calculation procedure computes lower bounds for candidate loops. Instead of solving the SLSLP problem for each valid single-loop that is very time-consuming, this lower bound can help to eliminate some inferior loops quickly.

Their iterative algorithm involving of solving two 0-1 IP models is very time consuming for realistic problems.

Some other models and solution procedures for the single-loop guide-path system design are proposed in Sinriech and Tanchoco (1993), Chen et al. (1999), and Asef-Vaziri et al. (2000). Chen et al. (1999) present a *mixed*-IP (or MIP) model to design guide-paths for a single-loop dual rail (path) system (SLDR), a special class of the single-loop system. This system contains only one loop (single-loop), but vehicles use two parallel tracks. This model also captures the vehicle failure rate in the objective function, which is claimed to produce more reliable results. This SLDR problem was solved using CPLEX (an optimization package). An instance containing 13 P/D locations needs about 2hours of computation time using CPLEX on a SPARC station 2. Asef-Vaziri et al. (2000) propose an alternative formulation to Tanchoco and Sinriech's (1992) formulation that has a smaller number of binary variables and takes into account a larger set of feasible integer solutions. Their formulation takes the design of a unidirectional single-loop and the location of P/D stations into account at the same time.

The throughput of the single-loop system drops slightly compared with the throughput of the conventional system (Tanchoco and Sinriech, 1992). To obtain the same throughput with the conventional system, the single-loop system needs more vehicles. Obviously, the single-loop system eliminates the inference problem at intersections (this system has no intersection at all). However, with multiple vehicles operating in the same loop, vehicle interference is still possible, since vehicles may have different operating speeds.

2.1.4 Tandem guide-path system

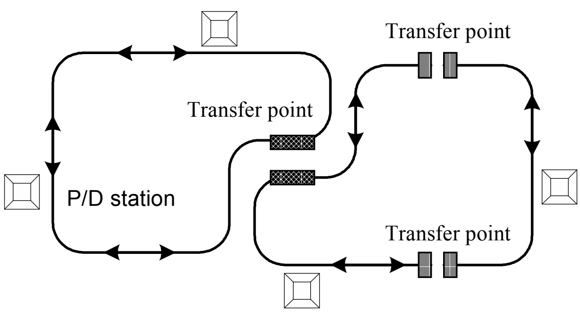


Figure 2.3 A tandem guide-path system with three zones (one loop and two segments)

The tandem guide-path system was first introduced by Bozer and Srinivasan (1991). The tandem guide-path system contains multiple zones. Only one vehicle serves each zone and transfer-stations are used to interface between zones (Figure 2.3). In case zones are loops, we have a tandem-loop configuration in which a number of non-overlapping single-loop paths provide transportation possibilities. In a tandem guide-path system, a job may require more than one vehicle to transport it to its destination. Vehicle blocking and interference problems are totally eliminated.

Bozer and Srinivasan (1992) introduce an algorithm based on a set-partitioning approach to decompose a system into non-overlapping, single-vehicle zones operating in tandem. The procedure starts by generating promising subsets of workstations. A vehicle must have enough capacity to serve a subset (with a workload ω). After that, the feasibility of each subset is checked. These subsets (columns) are fed into a set-partitioning model, which is used to identify the best tandem configuration. The set-partitioning model is given below:

Minimize z (2.1)

subject to

$$z - \omega_p x_p \geq 0 \qquad \forall p \tag{2.2}$$

$$\sum_p a_{ip} x_p = 1 \qquad \forall i \tag{2.3}$$

$$\sum_p x_p = L \tag{2.4}$$

where x_p = 1 if the column p is used in the final partition, = 0 otherwise; a_{ip} = 1 if workstation i is covered by column p and 0 otherwise; L- the desired number of zones (i.e. the number of vehicles) set by the analysis; ω_p: the workload factor of the p-th column obtained from the previous phase.

The first constraint ensures the workload in any zone does not exceed z (maximum workload). The second constraint ensures that each workstation is assigned to only one column (zone) and the last constraint forces the resulting partition to have exactly L zones. The objective of the set-partitioning problem is to avoid generating bottleneck zones by evenly distributing the overall workload among the zones as much as possible. This model was solved using LINDO, however to solve big problems, a more efficient algorithm is proposed. In Bozer and Srinivasan (1992), a vehicle in the tandem guide-path system does not need to use FEFS, but it can use other dispatching rules as well.

Huang (1997) introduces a variation of tandem configurations with an additional transportation center. This particular tandem configuration serves transportation jobs quicker and needs smaller number transfer stations between loops. However, the transportation center needs more space and requires a higher investment. His procedure allocates a transfer point in each zone and constructs a transportation center to connect them for a given tandem system. Yu and Egbelu (2001) introduce a variable-path tandem system based on partitioning a conventional guide-path into non-overlapping tandem zones. A dedicated vehicle serves each partition (sub-network) and additional transfer points provide interfaces between adjacent networks. Each sub-network is not necessarily a loop, so the dedicated vehicle has a greater flexibility in routing.

Ross et al. (1996) compare the performance (AGV utilization, mean flow time, mean tardiness, mean percent tardy) of tandem and conventional systems for specific configurations. They show that the tandem system performs as efficient as the conventional system. In comparison with the conventional system, the tandem system is simpler to control, has no congestion problem and is easier to expand. However, the tandem system requires additional transfer buffers which are costly and increase the handling time. This disadvantage may reduce the system throughput. The tandem system also has other disadvantages such as less tolerance to system failures.

- **The segmented guide-path system (segmented flow topology - SFT)**

Sinriech and Tanchoco (1995) describe a specific type of guide-path systems: the segmented flow topology (Figure 2.3). The SFT system contains one or more zones, each of which is separated into non-overlapping segments served by a single vehicle. Transfer buffers are situated at both ends of each segment and serve as interface devices between the segments (Figure 2.3). A SFT system may be not fully connected, depending on logical material flow requirements. Considering the SFT system carefully, it appears to be very similar to a general tandem system. Even in the case when the SFT system is not fully connected, we may interpret it as a combination of several tandem sub-systems. Sinriech and Tanchoco (1995) propose a procedure to solve the segmented flow-path layout design problem. Their results show that the SFT system outperforms the conventional system according to many criteria. However, even a small problem requires a large amount of time to solve. In Sinriech and Tanchoco (1997), a more efficient procedure was developed for the SFT design problem, which can be implemented for real-life problems. The main disadvantage of the SFT system is that it requires additional transfer stations.

To summarize this guide-path system section we describe some characteristics of guide-path systems in Table 2.3 and Table 2.4.

Table 2.3 A comparison of guide-path systems

Features	Conventional	Single-loop	Tandem
Number of mutually exclusive zones	One zone, fully connected system	One zone, fully connected system	Split system which retains connectivity through transfer buffers, can be non-connected for a special case (SFT)
Number of vehicles per zones	Multiple	Multiple	Single
Operating with a bidirectional system	Difficult	Difficult	Simple
Traffic control	Difficult	Easy	Easy
Vehicle scheduling/ dispatching	Complex scheduling/ dispatching system	Simple	Simple
Congestion (probability)	High	Low	No
Intermediate buffers required (to transfer loads between loops or transfer points)	No	No	Yes

Table 2.4 Advantages and disadvantages of guide-path systems

Guide-path system	Advantages	Disadvantages
Conventional	- Flexible in routing - Efficiency achieved by utilizing alternative routes - Shorter travel distances - Tolerance to system failures	- Complicated to control - Congestion, interference problems are likely to happen - Difficulty of expansion
Single-loop	- Simplicity of control - Congestion, blocking, interference problems are reduced in comparison with the conventional system	- Less flexible in routing - Less tolerance to system failures - Vehicle blocking and interference are possible - Extra transport capacity needed - Longer travel for loads - Difficulty of expansion
Tandem	- No vehicle congestion and interference - Simplicity to control - Easy expansion - Effective use of the bidirectional path	- Additional transfer buffers are required - Restriction of one vehicle per zone - Less tolerance to system failures - Some loads are handled by more than one vehicles - Additional time is required to transfer loads at buffers

Many modern AGV systems do not use fixed guide-paths (induction tracks). The guide-paths may, for example, be computer-programmed and uploaded to the vehicles' controllers. These vehicles are free-ranging and find their way using optical (laser), magnetic, odometer, gyroscope, vision, or radio- frequency techniques (Tompkins et al., 2003). In order to make full use of the flexibility capabilities of such systems, smart AGVs (or self-guided vehicles) are needed. The flexibility of changing guide-paths demands a capability to adapt the guide-path system satisfying new system requirements. In this case, obtaining the optimal guide-path system as a first objective becomes less important, however the system's flexibility becomes crucial. This observation is also applied for manned guided vehicle systems.

Another emerging problem in the guide-path design is selecting an appropriate type of guide-path system, but a guideline to select a suitable guide-path system is not available yet.

2.2 Estimating the number of vehicles

2.2.1 Single-load capacity vehicles

The number of vehicles heavily influences the performance of AGV systems (Van der Meer, 2000). AGVs are usually expensive, so determining the type and the appropriate number of vehicles is important. For a tandem configuration, the required number of vehicles is equal to the number of zones, but for other guide-path systems this number has to be estimated. According to Egbelu (1987), there are three main factors affecting the required number of vehicles: (1) guide path layout, (2) locations of load transfer points and (3) vehicle dispatching strategies. Egbelu (1987) proposes four analytical models to calculate the number of vehicles based on several information sources such as the expected number of loaded trips between stations and the number of workstations in the facility. This model requires a distance matrix for all required locations in the system and the flow of materials as the input data. Model 4 of Egbelu (1987) estimates the number of vehicles as:

$$N = \left[\left(\sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} \middle/ V \right) + \left(\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \right) \times \left(t_u + t_l \right) \right] \Big/ \left(60T - t \right) \qquad (2.5)$$

with n- number of workstations; f_{ij}- expected number of loaded trips required between workstation i and workstation j during the period or shift; D_{ij}- the estimated empty and loaded travel distance between stations i and j; T- length of the period of shift during which the f_{ij} exchange occur; V- average vehicle travel speed; t_l- mean time to load a vehicle; t_u- mean time to unload a vehicle; t- expected lost time by each vehicle during a time period of T due to battery change.

Egbelu (1987) indicates that the first three models in his paper are normally over optimistic when estimating the number of vehicles. The fourth model (described above) provides a reasonably good estimation in most cases except when the shortest-travel-distance first vehicle dispatching rule is used. The main factors that lead to different results between models are the method of estimating the empty travel and time lost caused by blocking. Egbelu (1987) mentions the important role of dispatching rules in estimating the required number of vehicles, but he does not take it into account explicitly. Similar approaches are proposed by Maxwell and Muckstadt (1982), Mahadevan and Narendran (1993).

Since dispatching rules have an important role in estimating the required number of vehicles, methods explicitly considering dispatching rules in the estimation model normally provide better results. Srinivasan et al. (1994) analyze a system using only one vehicle traveling under the modified-FCFS rule. They extend the result for the multi-vehicle case, by approximating K-vehicles by a single vehicle traveling K-times faster. They gain a good approximation for vehicle statistics such as the fraction of time a vehicle travels empty and also for the required number of vehicles. The quality of their model deteriorates when the system requires a large number of vehicles. Shen and Kobza (1998) propose another analytical model to estimate the number of vehicles in light-traffic systems.

Other analytical approaches estimating the required number of vehicles include: queuing models (Tanchoco et al., 1987; Talbot, 2003); statistical approach (Arifin and Egbelu, 2000); multi-criteria decision modeling (Sinriech and Tanchoco, 1992a) and network-flow modeling (Vis et al., 2001). The queuing model of Talbot (2003) and Chevalier et al. (2002) estimates the number of vehicles to achieve a desired fill rate (the probability to have a vehicle available at a station to satisfy a request). The quality of estimation deteriorates in light-traffic systems, as it tends to overestimate the number of vehicles. Vis et al. (2001) propose a network-flow formulation for determining the number of AGVs required at a semi-automated container terminal in which a job is a node and an arc(i,j) with capacity of one corresponds to a vehicle that can execute both jobs (i and j) in sequence, satisfying certain time-window restrictions. The minimum number of vehicles equals the minimum number of directed paths such that each node in the network is included in exactly one path. They also develop a polynomial time minimum-flow algorithm to solve the problem.

Practically, the precise number of required vehicles can not be given by any models. The exact number can be smaller or higher than the estimated one depending on characteristics of the estimation model. This number needs to be adjusted by the designer, probably with an assistant of a simulation model.

2.2.2 Multi-load capacity vehicles

The use of multi-load capacity vehicles can reduce the number of vehicles needed or increase the throughput of a system. A multi-load vehicle can pick-up additional loads while transporting a previously assigned load. The use of multi-load vehicles can, therefore, reduce the amount of vehicles' empty trip time and also the total distance traveled is likely to reduce. Bilge and Tanchoco (1997) demonstrate the effectiveness of using multi-load vehicles compared to unit-load vehicles. After experimenting using simulation, they conclude that using multi-load AGVs increases the system throughput, especially in case of high transport demands. Using simulation, Van der Meer and De Koster (1999) also show that multi-load vehicles help to increase the system performance, particularly when multiple loads can be picked up at one location. A disadvantage is that a more complex scheduling system is required.

Using estimation models mentioned above, the estimated number of vehicles may considerably differ from the real vehicle requirements due to some impractical assumptions in the analytical models. Moreover, the number of vehicles is strongly affected by the dispatching rules used (Egbelu, 1987), traffic management, congestion and other factors. Therefore, the estimated number should be re-evaluated using a simulation model for specific operational conditions.

2.3 Vehicle scheduling

The vehicle scheduling system decides when, where and how a vehicle should act to perform tasks, including the routes it should take. If all tasks are known prior to the planning period, the scheduling problem can be solved offline. However in practice, exact information about jobs (tasks) is usually known at a very late instant. This makes offline scheduling hardly possible. Therefore, online scheduling or dispatching systems are needed to control vehicles. The input data for the scheduling problem includes a distance matrix of all locations, load arrival data (released and delivered locations, time windows), vehicle data (type, capacity, speed, etc.) and some optional data (e.g. parking policy).

2.3.1 Offline scheduling

In the offline case, all transportation requests are known in advance. The complete vehicle routes can be optimized and constructed before vehicles carry them out. However, a small change in job arrival time, a change in driving time (congestion), or failure of a vehicle may impact or even destroy the whole schedule. The scheduling problem in an AGV system is similar to a pick-up and delivery problem with time windows (PDPTW), which often has travel time minimization or minimizing the number of vehicles as objectives. In most guided vehicle system, pallet loads (single loads) are used, so a vehicle can only carry one (pallet) load at a time. This makes the vehicle scheduling problem in most guided vehicle systems resemble to a multiple traveling salesman problem with time windows (*m*-TSPTW). However, the vehicle scheduling problem in AGV systems has some characteristics which make them different from the PDPTW (and *m*-TSPTW). These characteristics are higher traffic density, shorter travel distances, shorter planning horizon due to stochastic load arrivals, vehicle interference and battery charging problems.

The PDPTW and *m*-TSPTW problems are known to be NP-hard, so it is unlikely that we can find an algorithm to solve this type of problem in polynomial time. Due to this reason heuristics are the most appropriate approach to cope with this type of problem. Dumas et al. (1991) develop an exact algorithm, which uses a column-generation scheme with a constrained shortest-path as a sub-problem, to solve the PDPTW. The objective is to minimize the sum of the total travel cost. Only homogeneous vehicles (that is of a single type) are considered. Savelsbergh and Sol (1995) provide a survey on the general pick-up and delivery problem (GPDP).

In manufacturing environments, the vehicle schedule is affected by the machine schedule. Hence, in this type of environment, the scheduling system needs the capability to deal with both systems at the same time. Bilge and Ulusoy (1995) formulate the integrated scheduling problem of machines and the AGV system as a MIP formulation. Since the problem is very difficult to solve by an exact method, they propose a heuristic for the solution. The heuristic solves two scheduling problems (machines and AGVs) iteratively until a sufficiently good result is obtained. Ulusoy et al. (1997) use a genetic algorithm to solve the integrated scheduling problem. Abdelmaguid et al. (2004) improve the integrated scheduling problem' solution by applying a hybrid genetic algorithm.

2.3.2 Online scheduling

In practice, environments are usually stochastic (job arrivals, the travel time, loading and unloading times fluctuate, vehicles can breakdown), so the schedule has to be adapted dynamically in time. Savelsbergh and Sol (1995) discuss several solution approaches for the dynamic PDPTW. The schedule of vehicles should be updated when new transportation request information arrives. An approach is to schedule vehicles using a rolling horizon in which vehicle routes are updated after a predetermined time period (time horizon). Savelsbergh and Sol (1998) develop a set-partitioning model for a real PDPTW of a Dutch parcel carrier. They develop a branch-and-price algorithm to solve this problem dynamically. Yang et al. (2004) propose several policies to schedule vehicles dynamically. Their model (the truck-load pick-up-and-delivery problem with time windows) is described below:

$$\text{Minimize} \sum_{k=1}^{K}\sum_{i=1}^{N} c_{0i}^{k} x_{k,K+i} + \sum_{i=1}^{N}\left(\alpha\omega_i x_{K+i,K+i} + \sum_{j=1,j\neq i}^{N} c_{ij} x_{K+i,K+j}\right) + p\sum_{i=1}^{N} t_i \qquad (2.6)$$

subject to:

$$\sum_{v=1}^{K+N} x_{uv} = 1 \quad \forall u = 1,..,K+N \qquad (2.7)$$

$$\sum_{v=1}^{K+N} x_{vu} = 1 \quad \forall u = 1,..,K+N \qquad (2.8)$$

$$x_{uv} = 0,1 \qquad \forall u,v = 1,..,K+N \qquad (2.9)$$

$$-\sum_{k=1}^{K}\left(d_{0i}^{k} + \theta_k\right) x_{k,K+i} + t_i \geq 0 \qquad \forall i = 1,..,N \qquad (2.10)$$

$$\left(\omega_i + d_{ij}\right) x_{K+i,K+i} - T x_{K+i,K+j} - t_i + t_j \geq -T + \omega_i + d_{ij} \quad \forall i,j = 1,..,N \qquad (2.11)$$

$$\tau_i^- \leq t_i \leq \tau_i^+ \qquad \forall i = 1,..,N \qquad (2.12)$$

where *K*- number of trucks (truck *k* is first available at time θ_k and at location o_k); *N*- number of known demands; demand *i* required to move from a origin a_i to a destination b_i between time windows $[\tau_i^-, \tau_i^+]$, t_i- service time at node *i*; *C(a,b)*- cost of empty travel between two points *a* and *b* = 1×*D(a,b)*; *D(a,b)*- distance between two points *a* and *b*; ω_i (= $D(a_i,b_i)$)- required loaded distance to serve demand *i*; *p*- penalty coefficient; $\alpha\omega_i$- the lost revenue for rejecting demand *i* (α- a positive constant); $d_{0i}^k = D(o_k, a_i)$, $c_{0i}^k = C(o_k, a_i)$, $k = 1..K, i = 1..N$ - distance and cost matrices for a truck *k* and a demand *i*; $d_{ij} = D(b_i, a_j)$, $c_{ij} = C(b_i, a_j)$, $j = 1..N, i = 1..N$ - distance and cost matrices for other demands; *T*- a large number. The binary variable $x_{k,K+i}$ is to indicate whether truck *k* first serves demand *i*;

$x_{K+i,K+j}$ is to indicate whether there is a truck that serves demand i and demand j consecutively; $x_{k,k}$ = 1 means that truck k serves no demand; and $x_{K+i,K+i}$ = 1 means that demand i is rejected.

The objective is to minimize the total cost of processing all demands, which is a combination of the cost of empty travel distance, of penalty for delay, and of lost revenues due to loads rejected. Due to the constraints (2.7)-(2.9), the solution will constitute a feasible assignment. The constraints (2.10)-(2.11) disallow any cycle without a truck. The constraint (2.12) is a time-window constraint. For a real-time situation, this problem is solved every time a new request for service is received (they call this the "OPTUN" policy) and as a result, new assignments are made. This problem was solved using CPLEX. Since this problem is hard to solve for big instances, the input for the problem is restricted to a few known jobs. Other policies in their paper assign a new load to a vehicle based on a specific criterion such as total cost. They show for a test problem, that OPTUN outperforms simple policies. The OPTUN policy has limited applications since CPLEX cannot solve big instances. OPTUN also uses some probabilistic information of future jobs which certainly improve solution quality.

The model of Yang et al. (2004) is a type of model for m-TSPTW which captures most characteristics of the scheduling problem in guided vehicle systems. However, the main objective of the guided vehicle scheduling problem, in most case, is minimizing the average load waiting time. The empty travel time is not the main concern for guided vehicle scheduling problems and loads (transport requests) in a guided vehicle system should not be rejected. Meersmans (2002) proposes a heuristic based on a beam-search algorithm to dynamically schedule AGVs at a container terminal. The quality of the schedule depends on the length of the planning horizon (the scheduling problem takes into account only known jobs during that time period) and the rescheduling frequency (the frequency at which the schedule is regenerated). He observes that a longer planning horizon and a higher rescheduling frequency lead to a better performance. Sabuncuoglu and Kizilisik (2003) evaluate several online (or dynamic) scheduling policies for a FMS. They have several similar observations to Meersmans (2002): (1) the performance of the system becomes better when the frequency of rescheduling increases and (2) a better offline algorithm leads to a better online performance. Other vehicles assignment strategies have been proposed by Cordeau et al. (2002); Powell et al. (2000).

The guided vehicle scheduling problem shares many similarities with the external transport scheduling problem. However, because of higher uncertainties in internal transport environments, a shorter planning horizon (and a higher planning frequency) is expected for this type of scheduling problem in comparison with that of the external transport problem.

We find only few studies which applied the dynamic scheduling approach for guided vehicle systems, despite their proven efficiency for the truck scheduling problem. Therefore, this is the research area which needs to be investigated further.

2.3.3 Vehicle dispatching

We may consider a dispatching system as a scheduling system with a zero planning horizon and a dispatching decision is made when (a) a vehicle drop off a load; (b) a vehicle reaches its parking location (c) a new load arrived. A dispatching system uses dispatching rules to control vehicles. Online dispatching rules are simple and can be easily adapted for automated guided vehicle management systems. The common objectives are minimizing load waiting time, maximizing system throughput, minimizing queue length, or guaranteeing a certain service level at stations. There are two main types of online dispatching systems: decentralized and centralized systems.

- **Decentralized system**

Decentralized control systems dispatch vehicles based on local information only. There is no system to coordinate between AGVs and the central control system. Traditionally, vehicle systems have been implemented and analyzed assuming that every vehicle is allowed to visit any P/D location in the system. One of the simplest implementations is one in which vehicles circulate in a unidirectional single-loop.

Bartholdi III and Platzman (1989) study a decentralized heuristic to control AGVs in a simple loop. In their research, an AGV, which can carry up to three loads, travels in a simple unidirectional loop and transports loads according to the FEFS rule. With the FEFS rule, the AGV circulates a loop continuously. Whenever the vehicle has space available, it picks up the first load encountered, which will then be delivered whenever the destination is reached. Sinriech and Tanchoco (1992b) provide another study that investigates the performance of single-loop systems.

The smart vehicles mentioned in the introduction are modern examples of decentralized control vehicles. Berman and Edan (2002) propose a hierarchical, fuzzy behavior-based methodology to control an AGV system. Central knowledge about the system’s state is not available. Agents, representing smart AGVs, collect the workstations’ statuses directly and dynamically decide their next task. Lindeijer (2003) also uses agent-technology to determine the best, deadlock-free route an AGV can take (see section 2.6.2). The agent-technology becomes more and more important to control AGVs in intelligent manufacturing systems (Shen and Norrie, 1999).

The main advantage of the decentralized control system is its simplicity, but its efficiency is low. The centralized control system is more complicated but can provide a better performance (De Koster and Van der Meer, 1998).

- **Centralized system**

In centralized control systems, a central controller keeps track of all movements regarding internal transport. All information related to vehicles such as pick-up and delivery locations, load-release times, vehicle positions and status, are stored in the controller's database. The controller assigns loads to vehicles (or vice versa) according to specified rules. The centralized controller continuously communicates with vehicles to guide them. Depending on the way in which transportation requests are assigned, the dispatching rules can be divided into two categories (Egbelu and Tanchoco, 1984): workstation-initiated (jobs at a workstation have the priority to claim vehicles) and vehicle-initiated dispatching rules (vehicles have the priority to claim jobs). In this paper, we classify vehicle-dispatching rules as single-attribute, multi-attribute, hierarchical, look-ahead and pre-emption dispatching rules.

<u>Single-attribute dispatching rules</u>

Single-attribute dispatching rules dispatch vehicles based on one parameter/criterion only. Parameters can be travel distance (distance-based), queue length (workload-based), load waiting time (time-based), or other criteria such as a rule based on vehicle availability (by Talbot, 2003).

Distance-based dispatching rules dispatch vehicles based on travel distances or travel times. This category includes rules such as shortest-travel-time(distance)-first (STT(D)F) or nearest-work-station-first (NWF), and nearest-vehicle-first (NVF). According to the shortest-travel-time-first (STTF) rule (Egbelu and Tanchoco, 1984), a vehicle is sent to the closest load to be transported. The closeness of a load can be defined in terms of travel time or distance. This rule leads to little empty travel time of vehicles, but is sensitive to the layout of load locations in the facility (Egbelu and Tanchoco, 1984; De Koster et al., 2004).

Workload-based dispatching rules take queue sizes (or workloads of workstations) into account. In Egbelu and Tanchoco (1984) and Sabuncuoglu (1998), several queue-size rules are introduced, such as the maximum-outgoing-queue-size (MOQS) rule and the minimum-remaining-outgoing-queue-space (MROQS) rule. The MOQS rule dispatches a vehicle to the workstation with the largest number of loads waiting to be picked up in its outgoing queue. MROQS dispatches vehicles to the workstation with the minimum remaining space in its outgoing queue. The aim of this rule is to reduce the possibility of queue overflowing or workstation blocking. In addition, several rules based on vehicle utilization (such as select the least utilized vehicle) are proposed by Egbelu and Tanchoco (1984) and Mahadevan and Narendran (1994).

Time-based dispatching rules dispatch vehicles based on jobs' waiting time. These rules include the first-come-first-served (FCFS) rule the modified first-come-first-served rule (MODFCFS) (see Egbelu and Tanchoco, 1984; Srinivasan et al., 1994). The MODFCFS attempts to reduce unnecessary empty travel time by allowing the vehicle to override the FCFS rule whenever it finds an unassigned move request at the destination point. Yamashita (2001) provides an analytical analysis of AGV systems using FCFS dispatching policies.

De Koster et al. (2004) carry out extensive simulation experiments with several commonly used dispatching rules such as NVF, for three real-life cases. They experiment with different operating conditions for AGV systems and point out that for environments where queue spaces are not critical, the distance-based dispatching rules (STDF, NVF) outperform other rules. However, when it is not the case, the time-based (MODFCFS) or workload-based dispatching rules might perform better.

Multi-attribute dispatching rules

Multi-attribute rules dispatch vehicles using more than one parameter (Klein and Kim, 1996; Hwang and Kim, 1998; Jeong and Randhawa, 2001). In general they outperform single-attribute dispatching rules. Klein and Kim (1996) propose several multi-attribute dispatching rules that are based on the multi-criteria decision making approach. Hwang and Kim (1998) propose a dispatching rule based on the bidding concept which is similar to a multi-attribute dispatching rule. The difference is that the dispatching function can take any form (also non-linear). Jeong and Randhawa (2001) propose multi-attribute dispatching rules that use three parameters: vehicle empty travel distance, remaining spaces in input buffers and remaining spaces in outgoing buffers to decide which load should be transported by a vehicle. They use an additive waiting model to compute weights for member parameters. A neural network is used to dynamically adjust the parameters' weights reflecting changes in the system. According to their results, a simple multi-attribute dispatching rule with a good set of weights might perform very well and is better in many cases than a multi-attribute dispatching rule with dynamically adjusted weights. Jeong and Randhawa (2001) have done a quite extensive simulation study, however only one layout was used in their experiments.

Hierarchical dispatching rules

This type of dispatching rules is typical for manufacturing systems where the added value of a part during the manufacturing process is taken into account when the dispatching decision has to be made. Sabuncuoglu and Hommertzheim (1992) use a dynamic dispatching algorithm for scheduling machines and AGVs in a flexible manufacturing system (FMS). In their algorithm, different decision criteria are applied sequentially to identify the most appropriate part and the machine to be served. They identify four hierarchical logic levels: push logic, buffer logic, pull logic and push-pull logic. At each logic level, some priority rules are applied to select the part and the machine. Their algorithm performs quite well compared with simpler dispatching rules. Similar approaches have been proposed for scheduling AGVs in FMSs by Yim and Linn (1993), Taghaboni (1997) and Tan and Tang (2001). Kim et al. (1999) introduce a hierarchical rule based on workload balancing. At the first level, the jobs are prioritized and at the second level a vehicle is assigned to the job with the highest priority. A complex priority index based on workload balancing among machines (dominant factor) and the urgency of jobs is defined.

Dispatching rules using a look-ahead period, or vehicle reassignment (pre-emption)

Bozer and Yen (1996) introduce two dispatching rules that consider reassignment of moving vehicles. These are modified shortest-travel-time-first (MOD STTF) and bidding-based device dispatching (B^2D^2). The MOD STTF rule is similar to the STTF rule in the sense that it assigns empty vehicles to move requests based on the proximity of the vehicle and the load location, and each vehicle has only one request at a time. The difference is that an empty vehicle may be reassigned to another move request or an empty vehicle may "release" another empty vehicle. If a vehicle travels "uncommitted" to its assigned destination, it may be reassigned to a new arrival request according to some specific conditions (Bozer and Yen, 1996). To some extent, the B^2D^2 rule is similar to the MOD STTF rule, but it is much more complicated. Using a quite extensive simulation study (four layouts and a large set of experimental conditions) Bozer and Yen (1996) show that MOD STTF and B^2D^2 outperform STTF.

Look-ahead dispatching rules use some advance information about loads to be available shortly to dispatch vehicles (Mantel and Landeweerd, 1995). De Koster et al. (2004) improve the AGVS performance using dispatching rules with prior information on the availability of loads. They experiment with several widely used dispatching rules (NVF, MODFCFS etc.) with and without pre-arrival information. According to their simulation experiments, a very short look-ahead period has significant positive effects on the system performance.

Table 2.5 A guideline for selecting an appropriate vehicle scheduling system

Criteria										
Guide-path system			*Ability of SFC[*] system to deal with complicated controllers*		*Degree of stochasticity of jobs' arrivals*				*Job density*	
Single loop	Tan-dem	Conven-tional	Yes	No	No	Low	Medium	High	Low	High
D	D	S/ D	S/ D	D	S(off)	S(on)	S(on)/ D	S(on)/D	S (r)/ D	S/D(r)

[*] *SFC: Shop floor control; D: dispatching; S: scheduling; on: online; off: offline; r: recommended.*

Table 2.5 presents a guideline for designers to choose a suitable vehicle scheduling system for implementation. Vehicles in simple guide-path systems (single-loop, tandem) can be dispatched using simple dispatching rules without reducing the system performance. In practice, the available SFC system may not have the capability to deal with a complicated controller. In this case, a scheduling system that requires more information and advanced monitoring systems may not be applicable. In highly stochastic environments, it is impossible to schedule vehicles over a long horizon, so dispatching rules might be a better option in this case. In case of a high job density, vehicles are busy most of the time so implementing a complicated scheduling system will not be very helpful.

Because of their simplicity, vehicle dispatching rules are easy to implement. However, as indicated by Meersmans (2002) and Sabuncuoglu and Kizilisik (2003), dynamic vehicle scheduling is often more efficient. Meersmans (2002) also indicates that dynamic vehicle scheduling has the capability of taking other factors such as co-ordination between different transportation means in facilities into account. Another important observation is that most dispatching rules are applied for unrealistic environments. The guided vehicle systems in warehouses have not received much attention. These environments have some specific characteristics such as larger operating areas, more complex guide-path system and queue spaces are not as critical as in manufacturing systems. Hence, it is important to find which kind of dispatching rules and scheduling algorithms are efficient and robust for such environments.

Another issue, which has to be considered in the scheduling, is vehicle parking. Most scheduling problems suppose that vehicles can stay at the load's pick-up/drop-off locations. However, this is not true in some AGV systems. Thus, vehicle parking problem should also be included in the vehicle scheduling problem.

2.4 Vehicle positioning

Vehicle idleness is unavoidable in automated guided vehicle systems. Rather than forcing vehicles to return to the vehicle depot, it is better to park vehicles at locations (vehicle home locations or dwell points) that are closer to load-release locations than the vehicle depot. Two main strategies for idle-vehicle positioning (parking) are static and dynamic strategies.

2.4.1 Static vehicle positioning strategy

Vehicle parking locations should be selected to minimize the vehicle response time to new movement requests or to evenly distribute idle vehicles over the network. Several positioning strategies are proposed in literature (Egbelu, 1993; Van der Meer, 2000). Four major approaches are:

Central-zone positioning rule: a certain parking area in the vehicle network has been designated for buffering idle vehicles. This area can be close to stations with a high probability of a load transport request, or at battery- recharge or fuel stations.

Circulatory-loop positioning rule: one or more cruising loops are defined for idle vehicles. When a vehicle becomes idle, it travels one of the loops until a transport order is received.

Drop-off point positioning rule: a vehicle remains at the point of the last delivery job until it is reassigned.

Distributed-positioning rule: a distributed-positioning rule employs multiple dwell points as opposed to a single point, as in the central zone case. When a vehicle becomes idle, it is routed to one of the dwell points.

Most literature that discusses dwell-point strategies for automated guided vehicle systems, involves selecting home locations of vehicles in a single-loop. A common approach in finding home locations of vehicles in a single-loop is translating the loop layout into a circular layout and after this step all following calculations are based on angular positions. Two other approaches use Markov chain theory (Kim and Kim, 1997) and network flow modeling (Hu and Egbelu, 2000). Egbelu (1993) uses the circular layout conversion to search for the best home locations for idle vehicles. He proposes four models and solution methods: for a single vehicle in a unidirectional loop and in a bidirectional loop, and for multiple vehicles in a unidirectional loop and in a bidirectional loop. The objective of the model is to minimize the maximum response time of the idle vehicles. The objective function of the simplest case (single vehicle, unidirectional loop) is stated as:

$$T = \min\left\{\max_{1\le i\le n}\left\{\frac{R}{V}\left[(\alpha_i-\beta)(1-X_i)+(360+\alpha_i-\beta)X_i\right]\right\}\right\} \qquad (2.13)$$

where α_i : angular location of the i_{th} workstation; β : angular location of the vehicle; $R = C/360^0$ (C: the total length or perimeter of a guide-path that describes the loop) ; V: the average speed of the vehicle; $X_i = 1$ if $\alpha_i \ge \beta$, = 0 otherwise.

According to Egbelu (1993), the optimal home location of a vehicle in a unidirectional loop coincides with the location of a workstation and the optimal home location of a vehicle in a bidirectional loop lies at the midpoint of an arc. Based on these characteristics and the traffic flow of the system, Egbelu proposes several algorithms to find the optimal locations for idle vehicles. He also indicates that it is extremely difficult to control multiple vehicles in the bidirectional loop. Kim (1995) proposes a similar approach to minimize the mean response time for a pick-up call and a single parking place policy is used. Gademann and Van de Velde (2000) consider the problem of positioning *m* AGVs in a loop layout with *n* stations. They provide an overview of time complexities for uni-directional and bidirectional flow systems and show that criteria like maximum response time and average response time can be minimized in polynomial time for any number of vehicles. Lee and Ventura (2001) propose a polynomial-time dynamic-programming algorithm that determines the optimal dwell-points of idle AGVs for both unidirectional and bidirectional loop layouts. The objective is minimizing the mean response time. Their algorithm decomposes the set of pick-up stations into subsets so that a single vehicle serves all stations in a subset.

Hu and Egbelu (2000) propose network-flow based models for selecting the optimal parking locations for idle vehicles in a unidirectional network (not necessarily to be a loop). The objectives are minimizing the maximum response time and minimizing the mean response time. A formulation with the non-linear objective function is presented below (Hu and Egbelu, 2000):

$$Minimize\ \left\{\max_{\forall i,j}\left\{d_{ij}y_{ij}\right\}\right\}\ (minimization\ of\ maximum\ system\ response\ time) \quad (2.14)$$

$$subject\ to:\quad \sum_{i=1}^{N} x_i = n \quad (2.15)$$

$$\sum_{i=1}^{N} y_{ij} = 1 \quad j = 1,2,...,m \quad (2.16)$$

$$\sum_{j=1}^{N} y_{ij} \le m x_i \quad i = 1,2,...,N \quad (2.17)$$

$$x_i, y_{ij} = 0 \text{ or } 1,\ \forall i,j \quad (2.18)$$

where x_i : = 1 if node i is selected as a dwell point, = 0 otherwise; y_{ij} = 1 if dwell point i serves pick-up station j, = 0 otherwise; m: the number of pick-up stations; n: the number of vehicles; N: the number of possible dwell points; d_{ij}: a shortest distance matrix of the modified network (obtained from the origin network after applying the network reconfiguration procedure) which is free from convergent nodes.

This formulation is then transformed into a MIP model by replacing the objective function with expressions: (*min Z*, *s.t.* $d_{ij}y_{ij} \le Z$). Two solution procedures, one exact and one heuristic, are proposed to solve the problem. The exact method solves a sequence of set covering problems. After finding the optimal locations for vehicles they present a six-steps procedure to distribute vehicles among dwell points. For minimizing the mean response time, a linear integer program is given. This model formulation is similar to the formulation of the *p*-median problem and solved using a branch-and bound-method. After finding the optimal dwell points, vehicles are distributed by the same procedure used for the previous case.

2.4.2 Dynamic vehicle positioning strategy

When pick-up demands at stations change over time, the home locations of vehicles may need to be changed. In order to adapt to this situation, some dynamic procedures for selecting dwell points are proposed. Kim (1995) adapts his static algorithms to cope with the dynamic situation. Kim’s algorithm bases on the calculation of the contribution of each segment on the circular layout to the mean angular travel distance. Chang and Egbelu (1996) propose dynamic algorithms to select the home location of a single vehicle in the unidirectional and bidirectional loop. They show that, in a very busy system, the performance is independent of the dwell point selection rules. Hu and Egbelu (2000) extend their algorithms to the situation where the pick-up demands change over time in a non-uniform manner. Their method requires an accurate update of load pick-ups remaining at each station at event times. When the ratio of remaining load pick-ups changes the idle-vehicle positioning problem has to be solved again. Their algorithm applies for minimizing both the maximum response time and the mean response time.

Most studies in this area focus on loop layouts only. The research of Hu and Egbelu (2000) is the only one that takes conventional layouts into account. In practice, companies may define fixed parking locations, because vehicles may or can only park in certain areas. These areas can be defined for safety reasons, to avoid congestion, to allow a change of drivers, to recharge the vehicle's battery, etc. and impose constraints on the idle vehicle positioning problem. Such practical issues are often overlooked or omitted in theoretical models.

2.5 Battery management

Although battery management is important for vehicle management, the battery management problem is usually omitted in research. Naturally, vehicles have to be charged after a certain operating period, but most research on guided vehicle systems assumes that the battery problem has little effect on performance. However, in reality there is a potential impact on performance as vehicles with nearly empty batteries are unavailable for the process, even if swap batteries are used. Battery swapping can only be carried out at specific locations, so vehicles are temporarily unavailable. This means that either additional vehicles are needed or load-waiting times increase.

According to McHaney (1995), the batteries' constraints can only be omitted under some circumstances: systems with naturally occurring breaks, or shift changes coinciding with battery swapping or charging, systems with ample amounts of idle time, and systems where charging can be regulated and insured to take place without impacting system operation. A modern and fully charged AGV may run for 6 hours or more without recharging its batteries. In facilities such as warehouses, vehicles may have naturally breaks (e.g. at lunch and coffee times), battery-charging may not be a problem. McHaney (1995) presents three types of charging schemes: (1) opportunity charging - uses the natural idle time in an AGV's cycle to replenish batteries, (2) automatic charging - an AGV runs until its battery is depleted to a certain level and then the scheduler assigns this AGV for recharging, (3) combination system - this is a combination of the previous two. Ebben (2001) suggests several heuristic rules for dispatching vehicles, which need to be recharged. It is possible to send vehicles to the nearest battery station, farthest reachable battery station on the current route, etc. In addition, we also have to consider the capacity of the battery charging stations (are there sufficient charging positions for vehicles), and the vehicle's next job, so we can add some other rules, such as sending vehicles to the battery station closest to the vehicle's next job.

The problem of estimating the number of batteries required is also important. The required number of batteries depends strongly on the chosen battery type (Ebben, 2001). Ebben (2001) also shows that the number of battery changes required largely depends on the net capacity of the battery and less on the number of battery stations. He also proposes a cost trade-off analysis to help the designer to choose the battery's type, the number and position of battery stations.

Another issue here is how to select locations for battery-charging stations. These stations have to be located to minimize battery-charging effects on the system operation. Battery stations may coincide with the vehicle parking locations to save spaces and to incorporate with opportunity charging. The vehicle's battery charging scheme should also be considered explicitly when vehicles are scheduled for operations. These issues are not considered in the literature at this stage.

2.6 Vehicle routing and deadlock resolution

At the operational decision level, the vehicle routing and deadlock resolution problems have to be addressed. Scheduling and routing vehicles in internal transport systems without deadlock are very important. A deadlock may cause the whole system to collapse or to become blocked. Deadlock may happen in some situations such as when two vehicles arrive at a crossing point at the same time or when two vehicles travel toward each others in different directions on a bidirectional path. There are several ways to avoid deadlock and collision in automated guided vehicle systems, for example using a better routing algorithm, using single-loop, tandem or SFT configurations; identification of imminent collision through forward sensing and consequently avoiding this through vehicle backtracking and/or rerouting; imposing zone control and extensive route pre-planning. Because of the complexity of scheduling algorithms and the stochasticity of guided vehicle environments, we do not think, in many cases, it is the good idea to incorporate free-routing into the scheduling algorithm. The main reason is that a free-routing schedule can be destroyed by a small uncertainty in vehicles' travel-time. And it certainly happens in person-guided vehicle systems. Thus, a solution is to schedule vehicles dynamically for short planning horizon or using vehicle dispatching rules. Deadlock and traffic problems should be taken care of by a real-time monitoring system. The deadlock free scheduling and routing algorithms might be more useful for automated guided vehicle systems.

2.6.1 Vehicle routing

Vehicle scheduling and routing problems are closely related and should be addressed concurrently. The vehicle routing problem decides the route a vehicle should take and the sequence of loads (or jobs) that this vehicle should visit. The scheduling problem also decides the times that a vehicle should pick-up (and delivery) loads. In tandem systems, the routing problem is very simple, but in conventional systems, it is more complicated. In the scheduling section (2.3), we supposed that a vehicle could reach its destination without deadlock. However, to avoid deadlock, the vehicle routing problem needs to be taken into account as well. Kim and Tanchoco (1991) propose an algorithm based on Dijkstra's shortest-path method to schedule vehicles based on the nodes' time windows. This approach produces a deadlock-free schedule. However, as noted by the authors, a small change in the schedule may destroy it completely. Taghaboni and Tanchoco (1995) introduce an incremental route planning and scheduling algorithm. Other approaches are introduced in Rajotia et al. (1998b) and Qiu and Hsu (2001). More details about the vehicle routing issue can be found in Qiu et al. (2002).

2.6.2 Deadlock resolution

Vehicle deadlock prevention and resolution problems are important in VBIT systems. Depending on the guide-path system and the vehicle control mechanism, these problems can be more serious in one system than in others. Deadlock prevention and resolving tasks are not issues in systems using tandem-loop guide-paths. However, they are important in other systems, particularly, systems using conventional guide-paths. In manufacturing systems, there are not many spaces available at input and output queues of workstations. Thus, using queues' spaces efficiently is an important issue.

- **Balancing the system workload**

When machines in a manufacturing system or the P/D locations in a distribution center have only little space for load buffering, the system might be blocked by buffers overflowing. Possible reasons include insufficient buffering capacity or using an inappropriate scheduling (dispatching) system. To cope with the first problem, a central buffer may be introduced to solve the temporary blocking problem. Kim et al. (1999) show that the central buffer has an important influence on the system performance. The second problem can be solved by applying workload-related dispatching rules (Egbelu and Tanchoco, 1984; Mahadevan and Narendran, 1994; Kim et al., 1999).

❑ Forward sensing

AGVs can be equipped with sensors (which may be used to detect if they are too close to other vehicles (Zeng et al., 1991)). A vehicle stops when the distance between it and the vehicle in front of it, is less than a threshold value. This technique, however, is not effective for systems with many curved guide-paths. This technique should be used in combination with other techniques. Zeng et al. (1991) use Petri nets to detect deadlocks in AGVSs. A Petri-net approach seems to be a promising direction to detect and prevent deadlocks in AGV systems (Zeng et al., 1991; Hsieh and Kang, 1998).

❑ Control the traffic at intersections

Routing vehicles through intersections is a key issue in the deadlock resolution. Egbelu and Tanchoco (1986) develop an algorithm to route vehicles through an intersection without deadlock. Their model assumes that all nodes have buffers, that the buffers are of infinite capacity and the time required for a vehicle to steer into and out of the buffer is small compared to the overall travel time required. Also in their paper, several types of buffering areas for vehicles in transit are designed. These include “loop”, “siding” and “spur” designs. The buffering areas provide spaces to avoid the blocking situation at intersections. The main idea behind the deadlock resolution at an intersection is to buffer selected vehicles and gradually resolve deadlocks. Evers and Koppers (1996) introduce the concept of “semaphore” as an abstraction of a traffic light to control vehicles at intersections. The number of vehicles controlled by a semaphore cannot exceed a specified maximum, which is called the capacity of the semaphore. A semaphore, in their paper, is a non-negative integer variable (S), with the interpretation of free capacity, on which two operations are defined: “Wait” and “Signal”. Operation “Wait” is executed when a vehicle arrives at the protected facility, whereas “Signal” is executed when the vehicle leaves the protected facility.

❑ Zone planning

Zone planning is an efficient method to avoid deadlock. There are two types of zoning systems: static zoning and dynamic zoning. In case of a static zoning, guide-paths are divided into several zones. When a vehicle arrives at a zone, the controller checks for the presence of another vehicle in this zone. If a vehicle is already traveling in this zone, then the vehicle intended to enter that zone has to wait until the other has passed. In case of a dynamic zoning strategy, zones are not fixed; they can be changed according to the traffic flow in the system. Ho (2000) presents a dynamic-zone strategy for vehicle-collision prevention. His method relies on two procedures: the Zone Adjustment Procedure and the Zone Assistance Procedure. With the Zone Adjustment Procedure, the area of each zone changes according to the current production demand. The Zone Assistance Procedure allows vehicles to help each other so that the workload of every vehicle is balanced over time. Reveliotis (2000) proposes a zone control strategy that determines vehicle routes incrementally, one zone at a time. Routing decisions are the results of a sequence of safety and performance considerations, with the former being primarily based on structural/ logical rather than timing aspects of the system behavior.

In general, the deadlock resolution task is much simpler in systems using smart AGVs or person-guided vehicles, since these vehicles can handle most parts of deadlock avoidance tasks.

2.7 Concluding remarks

In this chapter, most key issues in guided vehicle system design and control are discussed. These issues include guide-path design, estimating the required number of vehicles, vehicle scheduling, idle-vehicle positioning, battery management, vehicle routing and deadlock resolution. We have discussed most important models and solution approaches. We also indicate areas which are potentially improving the VBITS performance, such as VBIT scheduling problem or recommending specific dispatching rules to use in practice. It is impossible to tackle all challenges in one research project, in this study we focus on some of them.

In literature, most studies on dispatching vehicles in VBIT systems used unrealistic cases for experiments. Since there is no guarantee that good dispatching rules for unrealistic environments will perform well in real-life environments, we are interested in testing and ranking good dispatching rules from literature for real-word cases. We also aim at finding good dispatching rules for different type of environments. Another important area which has not received much attention from researchers is real-time scheduling of VBIT systems. Dispatching is currently the most popular approach in practice. However, a scheduling approach which can use of load pre-arrival information efficiently should lead to a better performance. Thus, another objective of this thesis is to investigate the potential contributions of the vehicle scheduling approach for VBIT systems.

In the next chapter, we implement several simple and good dispatching rules in literature for two simulation models of two real-life cases and rank these dispatching rules mainly according to the average load waiting time.

Chapter 3

Control of vehicle-based internal transport systems using simple dispatching rules

In the previous chapter, we learned that the literature on vehicle dispatching rules is very rich. However, we also found that most of the studies on dispatching rules are based on unrealistic assumptions and simplified layouts. In other environments in practice such as warehouses, guide-path layouts are more complicated than what has been studied in the literature. Moreover, the best dispatching rules in literature have not been applied to many real-world vehicle-based internal transport systems yet. Therefore, it is important to find which dispatching rules perform well for which environments in practice. In this chapter, we study the performance of different good and well-known dispatching rules in two different real-world environments. The first one is a distribution center for computer components. The second one is a production plant for packaging glass. VBIT systems are used to transport (pallet) loads within these facilities. The guided vehicles (GVs) are person-guided forklift trucks equipped with RF terminals. These GVs can be dispatched in the same ways as AGVs using dispatching rules. These two companies currently use customized, but not efficient dispatching rules to control vehicles. In both cases, a Warehouse Management System (or WMS) matches the GVs with loads or vice versa. These systems keep track of inventory and the movements of loads and vehicles. Due to the high degree of stochasticity (such as unreliable load arrival information) within each transport environment, the GVs are dispatched real-time.

The performance objectives in the different cases are fairly similar. In the distribution center, pallets have to be moved as quickly as possible to serve trucks at the receiving and shipping lanes. The main objective is therefore to minimize the average pallet waiting time, i.e., the time difference between the release of the pallet (load) until a vehicle picks up the pallet. To avoid the situation where some loads might be forgotten, the control system should be able to keep the maximum load waiting time at an acceptable level. Therefore, the dispatching rules which produce smaller values of the maximum load waiting time are preferred. Moreover, the number of loads in critical queues (queues that, upon becoming full, lead to propagating blocking effects in the system) should be small to avoid queue overflow (in cases where queue space is restricted). The production plant has similar objectives. When the glass has cooled down sufficiently and the inbound pallets are released for transport, they should be picked up from the conveyors and stored as quickly as possible (since there is limited buffering capacity). Furthermore, the trucks that come to pick-up a shipment of pallets should be served as soon as possible.

This chapter is based on two real-life cases which have been studied by Van der Meer (2000). In this study, he implemented several dispatching rules in literature such as the nearest-vehicle-first (NVF) rule for the two case studies. The major weakness of his study is that he used the output of only one run without applying batch means to decide the performance of dispatching rules. In this chapter (partly based on De Koster et al., 2004), we have produced more reliable results. We used the batch means methods to obtain the mean values of outputs and their 95% Confidence Interval (see section 3.1 for explanations on the statistical analysis). Moreover, we use the Tukey test to rank dispatching rules used. We introduce a variation the NVF rule which is NVF with time truncation (section 3.2.3) aiming at improving the NVF rule performance. We also consider more performance criteria (maximum load waiting time, vehicle utilization and maximum number of loads in critical queues) for evaluating dispatching-rule performances.

In this chapter, we investigate whether dispatching rules behave similarly in different environments or if the relative performances of the dispatching rules depend on the environment. Moreover, we will investigate the performance of the dispatching rules currently used by the companies and compare them to several standard rules described in the literature as well as to some new ones. Finally, we will investigate possible performance gains (reduction of load waiting times) when load pre-arrival information is available and whether this changes the ranking of the dispatching rules. As a result, we gain a better understanding of applicability and quality of dispatching rules in practice.

The organization of this chapter is as follows: section 3.1 introduces the research approach; section 3.2 describes two real-world cases including simulation environments and dispatching rules; section 3.4 provides simulation results and analysis; section 3.5 presents a sensitivity analysis and in section 3.6 conclusions are drawn.

3.1 Simulation approach and statistical analysis

In this thesis, (discrete event) simulation has been used as an important tool for modeling and investigating behaviors of dispatching rules in two real-life cases. As indicated by Law and Kelton (2000), simulation is one of the main tools to study real-life systems. Another method is operation research including deterministic and stochastic optimization. The main advantage of the simulation approach is that most complex, real-word systems which cannot be accurately described by a mathematical model can be evaluated analytically. However, developing a simulation model can be expensive and time-consuming. Simulation results can be difficult to interpret. In a simulation study, simulation models have to be constructed and validated carefully. Since each run of a stochastic simulation model produces just an estimation of the model's characteristics for a particular set of input parameters, we need to apply some good statistical analysis techniques to guarantee the obtained conclusions are valid and reliable. There are two main issues which need to be taken into account in a simulation study. These issues are generating random numbers and output data analysis (including raking dispatching rules by comparing alternatives).

- **Generating random numbers**

Random number generation is an important issue for most simulation studies. In order to compare two alternatives (e.g. two dispatching rules for a simulation model), we need the same (common) random numbers for the alternatives. However, an alternative may behave favorably under a random number stream, we need to evaluate the performance of an alternative using several runs (decided to satisfy a certain confidence level, e.g. 95%) with different random number streams (or using batch means) to get a confidence interval of each performance measure. In the simulation study, we use multiple runs for each alternative and we use the same random number set for each corresponding pair of alternatives.

- **Output data analysis**

As mentioned before, in order to obtain a confidence interval for a simulation output, we need several runs (or replications) of a simulation model using different random numbers. The method which estimates a simulation output confidence interval using outputs from several different runs, is called "replication and deletion" (Law and Kelton, 2000). Besides the replication/deletion method there are several other methods to estimate the outputs' confidence intervals such as batch means method, autoregressive method etc. In practice, the replication/deletion method is the most popular method because of its simplicity. This is the main method used to compare different alternatives. Most modern simulation software uses the replication/deletion method as the only mean for statistical analysis.

Another popular method to obtain an output's confidence interval is the batch means method. Unlike the replication/deletion method, the batch means method uses the result of only one (long) run of the simulation model. This long run is then divided into smaller batches. We estimate the output's confidence interval based on results of different batches like results of different runs of the replication/deletion method. For the batch means method, it is crucial to make sure that these batches are independent. Law and Kelton (2000) suggest that a large enough batch size should lead to independent batches. The independence of batches can be verified by some techniques such as using the scatter plot of batches' means. In this chapter, we used the batch means method to analyze output data. We used this method because of limitations of the old version of AutoMod^{TM} (version 8.2) which was used to model the cases. Recently, we have converted our simulation models to a new version of AutoMod^{TM} (version 10) which have the capability to analyze data using the replication/deletion method, so we decided to apply the replication/deletion method for analyzing output data in the chapter 4. This might lead to small differences between results of the two chapters. However, the mean values of corresponding results (the average load waiting time of the same dispatching rule) in the two chapters lie in the 95% confidence intervals of each other. This means that the differences are acceptable.

- **Comparing alternatives (i.e. dispatching rules in this thesis)**

Comparing the expected responses of two alternatives

Law and Kelton (2000) indicate that, for stochastic simulation, comparing the responses of two alternatives based on only single run (without batch means) is not reliable. In addition, it is not possible to conclude that an alternative is better than another by simply comparing the average values of the corresponding responses (of the two alternatives). Here, the confidence intervals of responses play a role. According to Law and Kelton (2000), a paired-*t* confidence interval method can be used for comparison purpose. This method is applicable when the number of replications to collect data for each alternative is the same ($n_1 = n_2 = n$). We define X_{1j} and X_{2j} as the corresponding outputs of two alternatives and $Z_j = X_{1j} - X_{2j}$ is the difference between them. Assuming that Z_j's are IID random variables and $E(Z_j) = \zeta$, let $\overline{Z}(n) = \frac{\sum_{j=1}^{n} Z_j}{n}$; and $\hat{Var}[\overline{Z}(n)] = \frac{\sum_{j=1}^{n} \left[Z_j - \overline{Z}(n) \right]^2}{n(n-1)}$; the (approximate) $100 \times (1-\alpha)$ percent confidence interval (95% for α = 0.05) is $\overline{Z}(n) \pm t_{n-1,1-\alpha/2} \sqrt{\hat{Var}[\overline{Z}(n)]}$. This confidence interval is called the *paired-t confidence interval*. If the paired-*t* confidence interval does not contain zero, we can conclude (or failto-reject the hypothesis) that the two responses are different. By using the paired-*t* confidence interval, we reduce the problem of comparing two confidence intervals into inspecting the confidence interval of their difference. Law and Kelton (2000) point out that practically we do not have to assume that X_{1j} and X_{2j} are independent; nor do we have to assume that $\mathrm{Var}(X_{1j}) = \mathrm{Var}(X_{2j})$. This is very important, since allowing positive correlations between X_{1j} and X_{2j} leads to a smaller confidence interval (Law and Kelton, 2000).

Comparing the expected responses of more than two alternatives

To compare more than two alternatives, we can do all-pairwise comparisons of responses. In this case, the individual confidence levels have to be adjusted upward so that the overall confidence of all intervals' covering their respective target is at the desired level $(1-\alpha)$ (Law and Kelton, 2000). The all-pairwise comparisons for *k* responses requires *K* $(=k\times(k-1)/2)$ evaluations. According to Law and Kelton (2000), the individual confidence level should be $1-\alpha/[k\times(k-1)/2]$. All-pairwise simultaneous comparisons can be done in many ways (Stoline, 1981; Hsu ,1996). Stoline (1981) shows that the Tukey test is one of the best methods to perform all-pairwise comparisons. For the balanced cases (the number of replications for every alternative is the same and equals *n*), the $100\times(1-\alpha)$ percent simultaneous Tukey confidence intervals for *K* pairwise comparisons are $\bar{y}_i-\bar{y}_j\pm(q_{\alpha,k,\nu})\times s/\sqrt{n}$ in which $\bar{y}_i$ is the estimated value of μ_i (the mean value of response *i*), *n* is the number of replications (or the number of batches when the batch means method is used), $q_{\alpha,k,\upsilon}$ is the upper α point of the Studentized range distribution with parameter *k* and ν (degree of freedom = $k\times(n-1)$), $s=\left(\sum_{i=1}^{k}\sum_{j=1}^{n}\left(y_{ij}-\bar{y}_i\right)^2/\nu\right)^{1/2}$. For each pairwise comparison, if the corresponding confidence interval does not contain zero, the two involved alternatives' responses are different.

Besides all-pairwise comparisons (MCA), Hsu (1996) describes three other multiple comparison methods: all-contrast comparisons (ACC), multiple comparisons with the best (MCB) and multiple comparisons with the control (MCC). Hsu (1996) indicates that MCA, MCB and MCC are good methods to do multiple comparisons. To rank alternatives, all-pairwise comparisons are necessary. Therefore, for our ranking purpose, MCA is an appropriate selection. In this thesis, Tukey tests (95% confidence interval) are used to compare (and rank) dispatching rules. We use SPSS version 11 to perform Tukey tests. In a ranking table of dispatching rules according to a performance criterion (e.g. Table 3.7), values of the average load waiting times of dispatching rules with a same rank are not different with 95% confidence level. However, a rule which is placed above another rule in a ranking group performs slightly better than the lower one (its mean value is smaller than the mean value of the rule placed below it).

3.2 Case descriptions and dispatching rules

This section describes two real-life cases, which had been originally modeled by Van der Meer (Van der Meer, 2000). In this research, we adapted his models for experiments. We thank him for his efforts and assistance.

3.2.1 The European Distribution Center (EDC)

The first case concerns the transportation of pallet loads at the European distribution center of a computer hardware and software wholesaler. This wholesaler distributes computer products to different retail stores in Europe and determines how many to purchase and store to be able to comply with the demands of the retailers. Because computer products change quickly over time, it is necessary to keep inventory levels low and the storage times as short as possible. A large part of the incoming products are packed in cartons, stacked per product on pallets. Five forklift trucks (or guided vehicles) with vehicle-mounted terminals transport the pallets. A central WMS keeps track of inventory and the position of stored products.

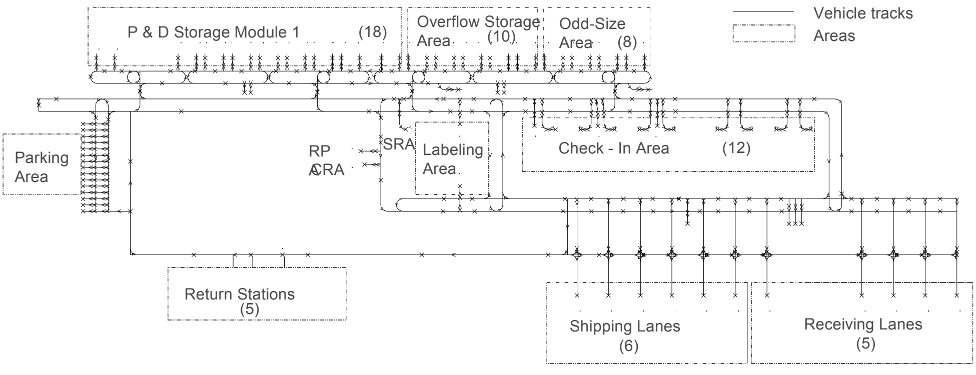


Figure 3.1 Guide-path layout of the EDC

The distribution center can be divided into several areas (see Figure 3.1) with a total guided-vehicle (GV) operating area of 40 by 140 meters. Each weekday, trucks arrive at the *Receiving Lanes* of the distribution center where the pallets (loads) are unloaded. In total there are five Receiving Lanes. If the cartons on the pallets contain returned or broken products they are manually transported to one of the five *Return stations*. The pallets are manually transported to one of twelve *Check-in Area* stations if the content of the cartons is unclear. At each of the previously mentioned stations, the pallets are labeled with a so-called license plate number (bar code). This license plate number contains information about the content of the cartons and the location the pallet should be brought to. At the moment the license plate is placed on the pallet, the pallet is entered into the WMS. If the cartons on the pallet are odd-shaped, or if the pallet is one of many with the same product, it will be transported to the *Odd-Size* or *Overflow Storage Area*. The Odd-Size Storage Area and the Overflow Storage Area have 10 and 8 Pick & Drop (P&D) locations. Otherwise the pallets go to one of the 18 P&D locations of *P&D Storage Module 1*. Within the storage modules, pallets are stored and orders are picked. From Storage Module 1, pallets can be transported to the Repalletization Area (*RPA*), the Shelf Replenishment Area (*SRA*), the Central Return Area (*CRA*), the *Shipping Lanes* and the *Labeling Area*. The Labeling Area has one delivery station and one pick-up station. RPA, CRA and SRA have one station each, and there are 6 shipping lanes in total (see Figure 3.1). From RPA, pallets move to Storage Module 1 or to CRA. At SRA the cartons of the pallets are placed on a conveyor belt, and will be transported to the shelf area where products are hand picked. Pallets at CRA always move to Storage Module 1. At the Labeling Area, pallets receive customer stickers and packing lists. The Shipping Lanes are the final stations. There, trucks arrive at dock doors to transport products to retail stores. The main flows are indicated in Figure 3.1.

Table 3.1 Total throughput in pallets per day (obtained from a six week period)

From / To		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	***Total***
1	Labeling Area	0	0	159	0	0	0	0	0	0	0	0	159
2	Check-in Area	0	0	0	0	0	0	22	0	0	0	0	22
3	Shipping Lanes	0	0	0	0	0	0	0	0	0	0	0	0
4	Receiving Lanes	0	0	0	0	0	0	109	2	2	0	0	113
5	SRA	0	0	0	0	0	0	0	0	0	0	0	0
6	RPA	0	0	0	0	0	0	9	0	0	0	1	10
7	P&D Storage Module 1	144	0	31	0	17	5	2	0	0	0	0	199
8	Overflow Storage Area	4	0	12	0	0	0	0	0	0	0	0	16
9	Odd-Size Area	11	0	40	0	0	1	0	0	0	0	0	52
10	Return Stations	0	0	0	0	0	0	6	0	0	0	0	6
11	CRA	0	0	0	0	0	0	4	0	0	0	0	4
Total		159	0	242	0	17	6	152	2	2	0	1	**581**

Table 3.1 shows the material flows in the facility (EDC). These flows have been measured for a period of six weeks.

Simulation Environment

To calculate the performance of each dispatching rule, the layout of the warehouse (Figure 3.1) and other relevant specifications of the warehouse and GVs have been modeled in the AutoMod™ (version 8.2). The data on load release times, origins and destinations come directly from the database of the WMS of the company. Other parameters such as vehicle speed, pick-up times come from careful measurements made at the distribution center.

All the parameters are kept the same for each dispatching scenario. These parameters include: the material flow, the number and locations of loads generated in the system, load generation instants, the speed of the vehicles, vehicle capacity, the paths via which the vehicles may travel, the load pick-up and set down time, the number of simulated days and the number of working hours per day. Table 3.2 gives a summary of some other values of the simulation model.

Table 3.2 Parameters of the EDC

Parameters	*Values*	*Parameters*	*Values*
GV speed	2 m/s	Loads generated per hour	77
Acceleration/deceleration	0.5 m/s^2	Average load transport time (sec.)	109
Pick-up time of a load	15 s	Current number of vehicles	5
Set down time of a load	15 s	Size of transport areas (m×m)	40×140
Vehicle capacity	1 load (pallet)	Nr. of different transport distances (approximately)	800
Number of working hours per day	7.5 hours	Min vehicle utilization (%)	47

The load release times and release locations have been measured for a period of six weeks. It appears that the requests for a certain transport depend highly on the time of day and can be modeled properly using Poisson distributions (this has been tested using a series of χ^2-tests). Each type of transport is independently exponentially generated at its own rate. Each day is in turn divided into four periods. Period 1: from the start of the day until the coffee break, period 2: from the coffee break until lunch, period 3: from lunch until the tea break, and period 4: from the tea break until the end of the working day. These periods are introduced to realistically represent the variation in the inter-arrival rates over the day. For example, in period 4 there are more loads transported to the shipping lanes than in period 1.

3.2.2 The glass production plant

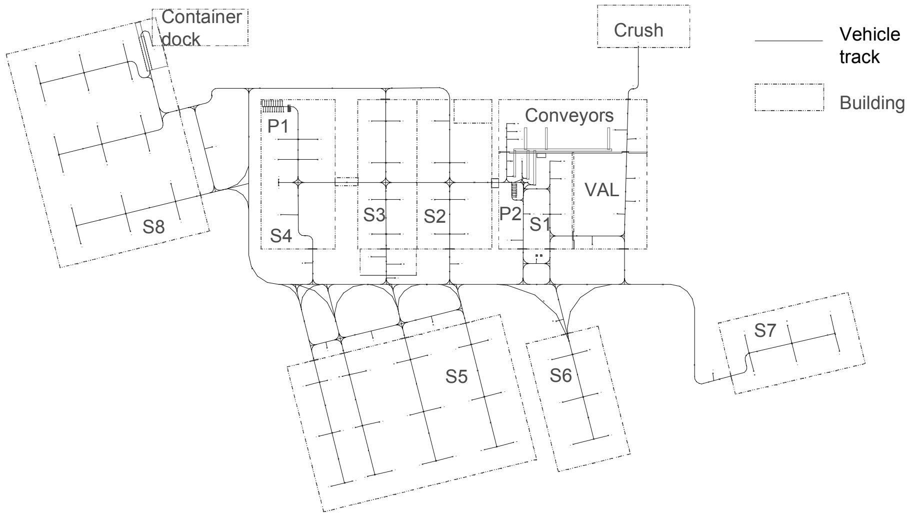


Figure 3.2 Guide-path layout and main flows of the Production Plant

The second case concerns the transportation of pallet loads at a production plant of packaging glass. The glassware is stored after production at the site until the clients (manufacturers that fill the glassware) collect the products for their own use. About 400 different glassware products, varying from jars to bottles, are produced. With three glass melting ovens and nine production lines, nine different glassware products are produced simultaneously, 24 hours per day, 365 days per year. The glassware is carefully stacked on pallets, which are then wrapped in plastic foil and finally moved by three *conveyors* to the 'landing' zone in one of the storage areas (see Figure 3.2). There are eight main storage areas (denoted by S1 through S8 in Figure 3.2) with a total of 55000 square meters of storage space. The dual load RF-guided forklift trucks (FLTs) move two pallets at a time, which arrive at the *conveyors* in pairs and are transported to one of the eight storage areas. The total operating area of the GVs is 315 by 540 meters. The pallets are always moved in pairs.

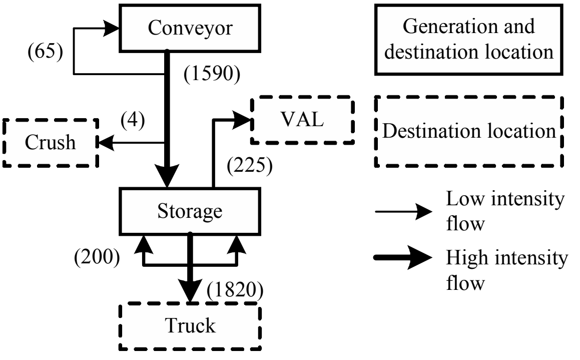


Figure 3.3 Average material flow between all locations of the glass production plant (weekday)

Figure 3.3 shows the main material flows within the production plant. On average between 1200 - 1400 production pallets, 200-250 Value Added Logistics (*VAL*) pallets and 60-70 'extra foil pallets' arrive per day at the landing zone. These inbound pallets are stored by product type in stows of 90-120 pallets. On average there are four pallets per day which have to go back to be crushed in the *Crush* area. Within the storage areas, about 200-250 pallets per day have to be moved in batches of 10 pallets to the *VAL* area (except in the weekends) and 200 pallets are reallocated within the storage areas for storage space optimization. Furthermore, on average 1820 outbound pallets have to be moved per day in batches of 28 pallets to 65 trucks which arrive just outside the storage areas between 6.00 am and 10.00 pm, except in the weekends. In 20 % of the cases, the trucks must visit two storage areas to be completely loaded. On average 10 % of all outbound pallets from S8 leave via the *container dock* instead of the main door of S8 since 10 % of the trucks arriving there can only be loaded from the back. Furthermore, there are peak arrivals of trucks during the day, since more trucks arrive in the morning and late afternoon compared to the early afternoon and the evening. The transport vehicles will park at the closest parking place (P1 or P2) when they have no task.

In total 11 transport vehicles are used 24 hours a day, 365 days per year. The vehicles are free to move anywhere on the paths of the defined operating area (see Figure 3.2) and can pass each other if necessary. However, there is room for only one FLT at a time at the pick-up and drop locations of the conveyors, trucks and stows in the storage buildings.

Simulation Environment

The layout of the production plant (Figure 3.2) and other relevant specifications of the environment and forklifts have been modeled in the AutoMod™ (version 9). The data on load release times, origins and destinations come directly from the database of the WMS of the company and expert judgments. Other parameters such as vehicle speed, pick-up time come from careful measurements made at the production plant. Since pallets are always moved in pairs, the case can be modeled as a uni-load environment with half the number of pallets to be transported. The number of generated pallets is approximated using uniform distribution. The uniform character of load generations is due to the uniform release of loads from the production lines. Table 3.3 gives a summary of some other values of the model.

Table 3.3 The parameters used for each scenario

Parameters		*Values*	*Parameters*	*Values*
Speed of loaded FLTs	in curves	2.5 m/s	Loads generated per hour	67 (81, 33)
	on straight paths	2 m/s	Average load transport time (sec.)	141
Speed of empty FLTs	in curves	3.5 m/s		
	on straight paths	3 m/s	Current number of vehicles	11
Pick-up time of a load		13 s	Size of transport areas (m×m)	315×540
Set down time of a load		14 s		
Vehicle capacity		1 dual-load (pallet)	Nr. of different transport distances (approximately)	2000
Number of working hours per day		24 hours	Min vehicle utilization (%)	24 (29, 12)

Table 3.3 shows three values for loads generated per hour. The first value (67) is the average value for the whole week (weekdays and weekend). The second one (81) is the number of generated loads per hour during weekdays. The number of generated loads per hour is significantly smaller during weekend (33). This leads to three values for the minimum vehicle utilization (as empty trips are not included in the calculation, we use the term minimum vehicle utilization).

3.2.3 Common dispatching rules for all cases

We made a selection of the most common simple dispatching rules described in literature (see section 2.3.3) which, at least in principle, could also be implemented at all the companies using their current vehicle dispatching systems. A dispatching rule assigns loads to vehicles on real-time basic as soon as a vehicle becomes available (empty, being waked up by a load or another vehicle). For all common and case-specific dispatching rules it holds that if there are no move requests in the system when the vehicle is looking for work, the vehicle will park at the nearest parking location and becomes idle until a move request becomes available. In the EDC case, there is only one parking area and there are two parking areas in the glass production plant. We did not implement any specific idle-vehicle positioning strategy.

(a) *Shortest Travel Distance First (STDF)*

Under this rule, a released or idle vehicle searches for the closest available (not yet assigned) load to transport. The closeness is measured in terms of travel distance. When a load arrived, it gives a signal about its availability. If there is a vehicle at the load arrival location, this vehicle will get the load; otherwise all idle vehicles in the systems will be awakened then. These vehicles will search for available loads. In general, a facility layout may contain a few remote stations. The stations not near a vehicle release point can therefore never qualify to receive a vehicle dispatch. This illustrates the major drawback of this rule; it is sensitive to the layout of the facilities.

(b) *Nearest Vehicle First (NVF)*

In a system using NVF, loads (or workstations) have the dispatching initiative. When a load enters the system (i.e. at a workstation) it places a move request; the shortest distance along the traveling paths to every available vehicle is then calculated. The idle vehicle, whose travel distance is the shortest, is dispatched to the point of request. However, when a vehicle becomes idle (and has not been claimed by any load), it searches for the closest load, i.e., at that point the dispatching initiative is at the vehicle and the rule used is STDF. The main difference between the NVF rule and the STDF rule is that with the NVF rule the load has the initiative to claim a vehicle when the load becomes available in the system.

(c) *Modified First-Come-First-Served (MODFCFS)*

A vehicle operating under MODFCFS, introduced by Srinivasan *et al*. (1994), delivering a load at the input queue of station *i*, first inspects the output queue of that station. This vehicle is then assigned to the oldest request (longest waiting load) at station *i* if one or more loads is found. However, if the output queue of station *i* is empty, this vehicle serves the oldest request in the entire system.

(d) Nearest Vehicle First with Time Priority (NVFTP)

To avoid the shortcoming of the NVF rule (remote areas may be ignored), we also introduce a new rule: nearest vehicle first with time priority (NVFTP). It is similar to the NVF rule, but we incorporate a time threshold for the load waiting time, in order to give a higher priority to loads that have to wait for a long time. When the load waiting time of a load reaches the threshold value (θ), this load has a higher priority for transportation. The threshold value should be decided for the specific situation. With the NVF rule, some loads which are not located in remote areas still have to wait rather long (about three times the average load waiting times or larger), so we should not take a low value of θ. That would make the performance of the NVF rule worse. If θ is very large, then the NVFTP rule becomes identical to the NVF rule. Therefore, we decided to examine the value of θ around four or five times the average load waiting time of the case when NVF is used. For each case, the best case-specific θ value has been chosen after extensive experimentation varying θ in small steps.

3.2.4 Case-specific dispatching rules for the European Distribution Center

(a) *Work-List-Dispatching (WLD)* (see Van der Meer, 2000 for more details)

This rule is the current rule at the EDC. According to this rule it is possible to give priorities to certain locations where loads are to be picked up. This will be illustrated with a general example. Suppose we are investigating a warehouse with three locations. When a vehicle becomes idle after dropping off a load at location 1, the central computer will search the work list of location 1 (see Table 3.4). First location 1 is checked for work by the central computer, because that location is on top of the work list. If a load is waiting there to be picked up, then the WMS instructs a vehicle to retrieve the load. If there is no work at location 1, then location 3 is checked, etcetera. When a vehicle becomes idle at location 3, only location 3 is checked with more priority before all other locations are checked in a random or a particular order.

Table 3.4 Example of work lists for centralized control

Location 1	Location 1	Location 3	Location 2
Location 2	Location 2	Location 3	Location 1
Location 3	Location 3	All	

In the case of the distribution center there are many work lists (defined by the company), a unique one for every drop-off location (for areas in Figure 3.1). There are eight drop-off locations: Shipping lane, SRA, RPA, CRA, Storage Module 1, Overflow area, Odd-size area and Labeling area. The work lists are constructed such, that the locations around the current position of the idle vehicle are checked for work first. Furthermore, the route the idle vehicle should follow next is consistent (in most cases) with the unidirectional flow of the paths. This reduces the probability of circulating around empty, to pick-up a load that has been made available just ‘behind’ the current location of the idle vehicle.

(b) *Load-List Dispatching (LLD)*

A Load-List is a list of locations, like a work-list, where a waiting load may find an empty vehicle to wake up. When a load is output to a pick-up point, the load-list at that location is scanned first for parking locations to wake an idle vehicle. The newly awakened vehicle then searches the work-list of the parking location. Since the vehicle scans the work-list, it may find a higher priority load than the load that woke it. With this rule the first dispatching initiative lies with the load, however, the vehicle will determine the move request.

(c) *Dispatching with Pre-arrival Information (DPI)*

This rule uses all the dispatching rules for the distribution center. The difference is that the load gives a signal x time units in prior to its actual release time. The time between the actual release, and the virtual release x time units before, can be interpreted as a forecast time. This gives the vehicle the opportunity to travel to the load before the load is physically ready for transport. The vehicle can therefore arrive just before (but also after) the load is ready for transport, thereby reducing load-waiting times. However, an increase in average load waiting time is also possible. For example, when the value of x is too large and the vehicle arrives before the actual release of the load. A vehicle is then idle while waiting for the actual release of the load instead of using this time to transport another load. In the case of the distribution center, the labeling station or cranes in the storage areas can trigger this pre-arrival information of loads about 5, 10 or 15 seconds in advance.

3.2.5 Case-specific dispatching rules for the glass production plant

(a) Dedicated Dispatching (DD)

This dispatching rule is currently used at the production plant. With this dispatching rule 11 GVs are used, 5 vehicles are dedicated to the inbound jobs, 2 vehicles are dedicated to all internal jobs (the reallocation moves for storage space optimization and the pallet moves to the VAL area) and the remaining 4 vehicles are dedicated to all outbound moves. Since there are no outbound jobs at night and in the weekends, the remaining 4 ‘outbound’ vehicles are free to do any other task. In all cases, all idle vehicles searching for a task will first claim the load closest to a vehicle within 100 meters. The idea is that vehicles will have less empty travel time. If there is no task closer than 100 meters the vehicle will claim the load that has been waiting longest in the entire system.

(b) C100FCFS

The C100FCFS operates as follows: the first available vehicle claims the nearest load within 100 meters around the vehicle. If the vehicle did not find any matched loads, it claims the oldest load in the system. This is still a special rule since it is a hybrid rule of distance and time.

(c) *Dispatching with Pre-arrival Information (DPI)*

This rule uses the previously described dispatching rules. Information on loads can be made available 30, 60, 90 or 120 seconds in advance. This pre-arrival information can, for example, already be triggered as soon as inbound loads are placed on the conveyors. Outbound loads can already be released when the trucks arrive at the gate. There is no pre-arrival information available for loads that are moved to VAL or for storage space optimization.

3.3 Experimental setup

Assumptions for the simulation models:

- Vehicles operate continuously without any breakdowns,
- All vehicles have uni-load capacity (which is true, apart from the production plant),
- Vehicles choose the shortest path to pick-up and deliver loads,
- Loads are generated in batches of one,
- Loads at each input queue are processed on a first come first serve basis,
- There is no operational time lost due to recharging vehicles,
- There is sufficient space for waiting loads.

The cases are modeled as close to the real-life situations as possible. Vehicles (in simulation) were modeled with dimensions of real-vehicles in practice and they behave similarly to real-vehicles. For examples, they have different speeds on straight and curve paths, and they need to accelerate before reaching the normal speed and to decelerate before stop. For both cases the system started idle and empty. Data has been gathered and analyzed after the system reached steady state. The length of the transient period for the EDC is about two days and that length for the glass production plant is about one week. The *batch means* method (see section 3.1) was applied to determine mean values of performance indicators. The corresponding values for the number of batches and batch lengths in the three cases are: eight of 75 hrs for the EDC and six of 14 days for the production plant. Cases are different in nature, so we select different batch lengths and different number of batches for each case in the balance of the expected statistical performance and the cost of simulation run. The batch sizes have been determined per case such that subsequent batches could be expected to be independent. By inspecting the scatter plots of their mean values, the independence of subsequent batches has been verified.

Table 3.5 Experiment design parameters

Models		*Distribution Center*	*Production Plant*
Dispatching rules	General	NVF, NVFTP STDF, MODFCFS	NVF, NVFTP STDF, MODFCFS
	Case specific	WLD, LLD	DD, C100FCFS
Pre-arrival information		0, 5, 10, 15 sec.	0, 30, 60, 90, 120 sec.
Number of vehicles		4, 5, 6	5, 7, 9, 11

Table 3.5 shows all experiment design parameters for the three cases. In comparison with Van der Meer (2000), we have done more extensive experiments. We introduced a new rule (NVFTP) and we also experimented with different number of vehicles. Van der Meer (2000) use only a fixed the number of vehicles (5 vehicles for the EDC and 11 vehicles for the glass production plant) for experiments in each case. The main performance criterion here is minimizing the average load waiting time. Because of their small scale, the vehicle travel distance in vehicle-based internal transport systems is not as important as in external transport systems. The most important objective of the VBIT systems is maximizing throughput and it can be achieved by minimizing the average load waiting time. In general, only the average load waiting time of the whole system is important. However, in some cases, we may have to consider the average load waiting times at some specific stations. We also take into consideration three additional criteria including the maximum load waiting time, vehicle utilization and the maximum number of loads in the critical queues. In the EDC, critical queues are queues at the labeling area and the P&D storage modules. In the glass production plant case, queues at the end of conveyors (connecting with the vehicle guide paths) are critical queues. These queues are critical, since there are only limited spaces for loads at these locations and overflowing these queues may lead to system blocking.

Statistical Analysis

In this chapter, we have used much more sound statistical analysis. We have ranked dispatching rules with a 95% confidence level using the Tukey test. The batch mean method is used to obtain data for analysis (see section 3.1).

Model validation and verification (see also Van der Meer, 2000)

For two models, the companies have provided the input data. Data that was not available in the database systems such as real vehicle speeds and handling times has been carefully measured in practice. The operation of the systems has been validated using 3-D animation together with the responsible manager of the companies. The results have been checked against reality for the current dispatching rule.

In the next section, we present the results of the two models in detail.

3.4 Results

In both cases, the dispatching rules are ranked primarily according to the average load waiting time. Besides the average load waiting times, the maximum load waiting time, the vehicle utilization and the maximum number of loads in critical queues (see section 3.3) are considered as well. Between two rules with the same value of the average load waiting time, we prefer the rule with smaller value of the maximum load waiting time, since with this rule it is less likely that some loads might be forgotten. The number of loads in critical queues (see section 3.3) should be small to avoid queue overflowing. However, we did not take it into account when dispatching vehicles.

3.4.1 Performance of dispatching rules in the distribution center case (EDC)

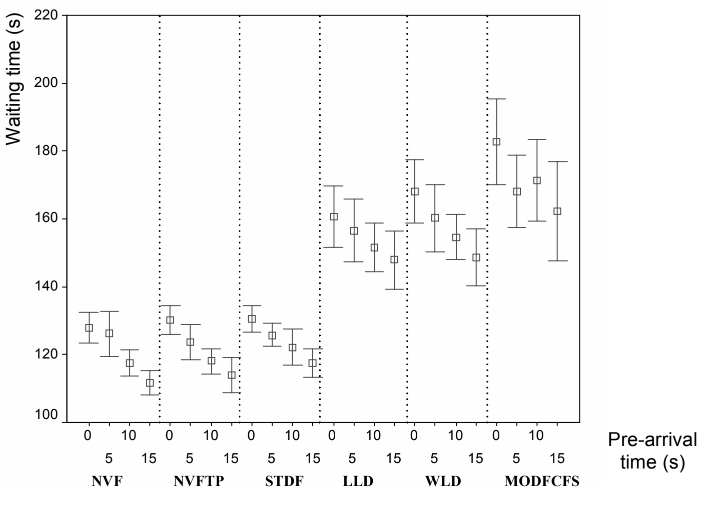


Figure 3.4 95% confidence interval graph for average load waiting times

Figure 3.4 shows the means and 95% level confidence intervals for the average load waiting times for the distribution center (with five vehicles). The threshold value for NVFTP (see section 3.6) is selected to be five times the average waiting time for the NVF rule. Regardless of the value of θ, no improvement in terms of average load waiting times was found compared to NVF. However, using NVTP leads to a decrease in the maximum load waiting time, since a load that has to wait long, obtains higher priority to be transported.

Table 3.6, Table 3.7 and Figure 3.4 give a summary of the results obtained. Table 3.6 shows that three rules (NVF, NVFTP and STDF) are the best rules in this case according to the average load waiting time criterion. Notice that two rules (NVF and STDF) are practically the same, except for the dispatch initiative. The performance of NVF is slightly better than that of STDF, since a load just entering the system can immediately claim an idle vehicle. The performance of the NVFTP rule is nearly identical. Since we selected a fairly large value for the time threshold (θ), only a small number of loads receives higher priority for transportation according to NVFTP.

Table 3.6 Summary of results of the various dispatch rules

PA (sec.)	*Performance indicators*	*Dispatching rules*					
		NVF	NVFTP	STDF	LLD	WLD	MODFCFS
	Ave-Wait sec.	127.9(±4.4)	130.2(±4.3)	130.5(±3.8)	160.6(±9.1)	168.1(±9.4)	182.8(±12.5)
0	Max-wait sec.	779.9	776.4	762.6	1559.6	1361.4	655.2
	Utilization %	74.93	74.85	76.1	77.6	78.64	78.6
	Max inCO	2	2	2	2	2	2
	Ave-Wait sec.	126.2(±6.7)	123.7(±5.2)	125.7(±3.5)	156.6(±9.2)	160.2(±10)	168.1(±10.8)
5	Max-wait sec.	1081.2	877.9	773.2	1371.4	1413.3	846.0
	Utilization %	75.4	75.2	76.1	77.7	78.7	78.8
	Max inCO	2	2	2	2	2	2
	Ave-Wait sec.	117.6(±3.9)	118.0(±3.8)	122.2(±5.4)	151.5(±7.1)	154.6(±6.7)	171.5(±12.0)
10	Max-wait sec.	949.3	726.4	703.3	2081.3	1062.2	791.2
	Utilization %	75.0	75.2	76.2	77.6	78.8	78.8
	Max inCO	2	2	2	2	2	2
	Ave-Wait sec.	111.7(±3.7)	113.9(±5.2)	117.6(±4.2)	147.9(±8.6)	148.7(±8.3)	162.3(±14.6)
15	Max-wait sec.	1249.0	781.3	703.9	1742.3	1744.6	714.1
	Utilization %	75.1	75.2	76.2	77.5	78.7	78.5
	Max_inCQ	2	2	2	2	2	2

PA: pre-arrival time; *Ave (Max)-Wait*: average (maximum) load waiting times; *Utilization %*: vehicle utilization; *Max_inCQ*: the maximum number of loads in critical queues; (± number): ± 95% confidence interval.

The next two rules, LLD and WLD are also very similar. Both make use of priority lists, but the claim initiative is different. Although both perform practically the same for all criteria, LLD has slightly more favorable waiting times. In any case both are ranked below NVF and STDF, which means that the performance with the current dispatching rule (WLD) used at the company could be increased. The difference of LLD and WLD can be explained similarly as the difference between NVF and STDF.

The rule ranked lowest is MODFCFS, although it is the simplest rule and has about the same vehicle utilization as WLD and LLD. However, the load waiting time is higher on average. This rule results in the smallest maximum load waiting time. Performance ranks are based on the Tukey test and indicated in Table 3.7.

Table 3.7 Rank of dispatching rules by average waiting times for the EDC (5 vehicles)

Dispatching rules	*Rank*		
NVFTP	1		
NVF	1		
STDF	1		
LLD		4	
WLD		4	
MODFCFS			6

Rules in the same subset have a comparatively equal mean and different subsets indicate that the mean values of these subsets are different (at 95% confidence interval).

Table 3.6 shows another remarkable result. MODFCFS using a pre-arrival time of five seconds reduces the average load waiting time by 14 seconds. Using a virtual release time five seconds before the physical release changes the allocation of vehicles in such a way that the mean waiting time decreases more than proportional. The reverse is also possible as is seen in Table 3.6. When the pre-arrival time is changed to 10 seconds, the vehicles are allocated unfavorably and the mean waiting time increases again with three seconds with respect to the waiting time with five seconds pre-arrival information (see section 3.2.2 for a possible explanation of this effect).

3.4.2 Performance of dispatching rules in the production plant case

In the production plant case, the current number of vehicles is 11. This number is also used in the comparison of dispatching rules. The time threshold value is set using the procedure explained in section 3.2.3 and in this case the optimal threshold value θ equals about five times the average load waiting times obtained for the NVF rule (≈ 900 seconds).

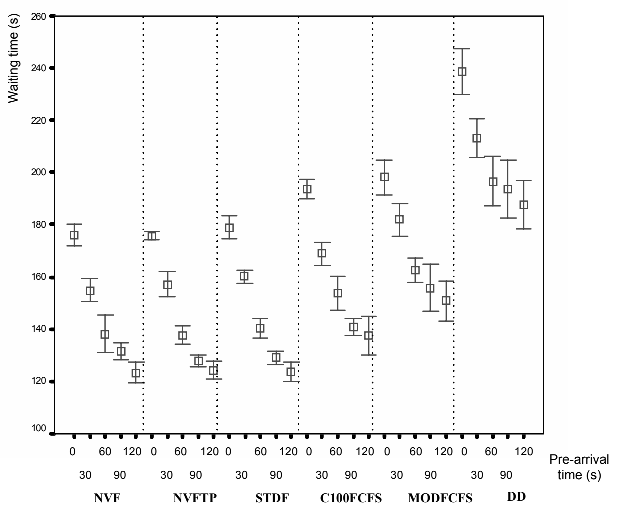


Figure 3.5 95% confidence interval graph for average load waiting times

Table 3.8 Summary of results of the various dispatch rules

PA (sec.)	*Performance indicators*	*Dispatching rules*					
		NVF	NVFTP	STDF	C100FCFS	MODFCFS	DD
0	Ave-Wait sec.	176.1(±4.1)	175.6(±1.7)	179.0(±4.5)	193.7(±3.7)	198.2(±6.8)	238.6(±8.7)
	Max-wait sec.	2228.7	1555.3	1645.2	1453.2	1500.5	3602.5
	Utilization %	42.0	41.7	42.7	43.2	43.9	31.7
	Max_inCQ	8	7	6	6	6	1
30	Ave-Wait sec.	154.9(±4.4)	157.2(±4.8)	160.1(±2.7)	168.9(±4.3)	181.9(±6.1)	213.3(±7.4)
	Max-wait sec.	1778.7	1479.4	1700.2	1416.4	1693.8	4546.6
	Utilization %	42.3	42.7	43.1	43.8	44.1	32.2
	Max_inCQ	6	6	9	7	9	1
60	Ave-Wait sec.	138.1(±7.1)	137.8(±3.3)	140.4(±3.8)	153.9(±6.5)	162.7(±4.5)	196.6(±9.5)
	Max-wait sec.	2110.6	1427.1	1740.4	1593.3	1389.3	4097.7
	Utilization %	44.0	43.9	44.2	45.1	45.8	34.0
	Max_inCQ	7	8	6	8	7	1
90	Ave-Wait sec.	131.5(±3.3)	127.9(±2.2)	129.0(±2.7)	140.9(±3.3)	155.8(±9.2)	193.6(±11.3)
	Max-wait sec.	1515.5	1581.9	2099.0	1313.5	1873.8	4147.6
	Utilization %	46.3	45.8	461.7	46.8	47.6	36.4
	Max_inCQ	6	5	7	7	11	1
120	Ave-Wait sec.	123.2(±3.9)	124.3(±3.4)	123.6(±3.7)	137.5(±7.3)	150.9(±7.7)	187.6(±9.3)
	Max-wait sec.	1565.0	1406.6	1750.8	1548.7	1819.9	4436.3
	Utilization %	48.3	48.8	48.8	49.6	50.2	38.8
	Max_inCQ	6	6	8	8	8	1

PA: pre-arrival time; *Ave (Max)-Wait*: average (maximum) load waiting times; *Utilization %*: vehicle utilization; *Max_inCQ*: the maximum number of loads in critical queues; (± number): ± 95% confidence interval.

Table 3.9 Rank of dispatching rules by average waiting times for the production plant (11 vehicles)

Dispatching rules	*Rank*		
NVFTP	1		
NVF	1		
STDF	1		
C100FCFS		4	
MODFCFS		4	
DD			6

Rules in the same subset have a comparatively equal mean and different subsets indicate that the mean values of these subsets are different (at 95% confidence level).

Table 3.8 gives a summary of the results obtained for the production plant. The best dispatching rules of the distribution center (see previous case) are also the best rules studied in the production environment. In this case, it is still difficult to say which of these rules is best, although the average load waiting times are slightly in favor of NVF and NVFTP in comparison with STDF. All three distance-based rules outperform the next groups of rules Table 3.9). The DD rule gives higher priority to the area in which critical queues are present (queues at the end of conveyors). Hence, the maximum number of items in the critical queues in this case is substantially smaller than in the other cases.

C100FCFS and MODFCFS perform rather similarly without the use of pre-arrival information ($x = 0$). The NVFTP rule performs slightly better than NVF in some cases. In this model the travel distances are long, so it is possible to have the effects of neglected remote areas. However, we can see that there are no significant differences in performance of the two rules in terms of average load waiting time. When remote areas are involved, the NVFTP can help to avoid neglect of loads at remote areas. In this case, using the NVFTP rule, the maximum load waiting time is significantly reduced at the cost of an extensive search for the best θ value.

The current dispatching rule (DD) is clearly outperformed and can easily be improved by relaxing the vehicle dedication constraints. In this way the dispatching rule changes to C100FCFS. If this initial distance is increased to the longest length between two locations on the premises, the rule changes to NVF. If the average load waiting time of the DD rule is satisfactory, the production plant should consider using NVF with fewer vehicles.

3.5 Sensitivity analysis

In the previous section we have seen that there is a clear performance ranking of dispatching rules, based on average waiting time. Although NVF and NVFTP consistently rank best, especially NVF has the drawback of relatively long maximum waiting times. In this performance ranking, the vehicle utilization rate may play a role, since the effect of unfavorable allocations will be larger in case of high utilizations. Therefore, in this section we analyze all two cases when the number of vehicles is a variable. Performance indicators are average load waiting time, maximum load waiting time and vehicle utilization. We concentrated on common dispatching rules NVF, STDF and MODFCFS. The NVF and STDF rules represent distance-based dispatching rules. These rules dispatch vehicles based on proximities of loads to these vehicles. The MODFCFS represents time-based dispatching rules, which dispatch vehicles based on residence times of loads in a system.

3.5.1 The distribution center case

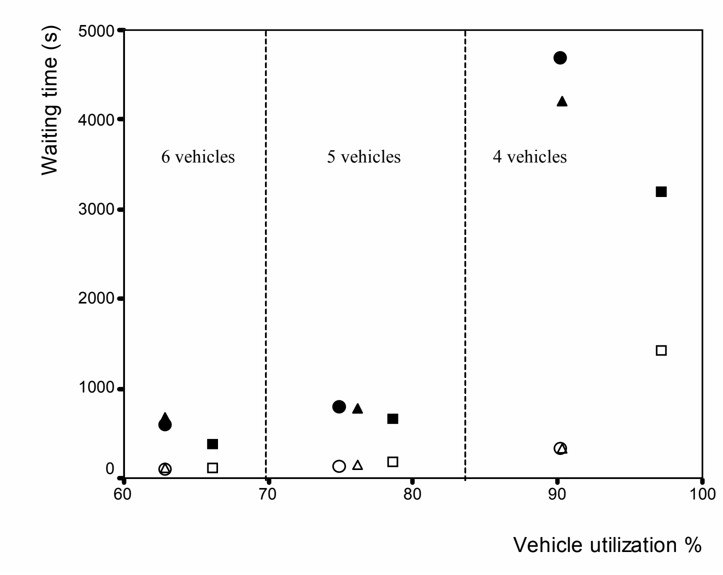


Figure 3.6 The scatter plot of average and max. load waiting times vs. vehicle utilization

● Maximum load waiting time for NVF; ○ Average load waiting time for NVF; ▲ Maximum load waiting time for STDF; △ Average load waiting time for STDF; ■ Maximum load waiting time for MODFCFS; □ Average load waiting time for MODFCFS.

Table 3.10 Results for the distribution center case

Disp. rules	NVF			STDF			MODFCFS		
# of vehicles	4	5	6	4	5	6	4	5	6
Ave-wait sec.	336.8	127.9	94.2	321.2	130.5	102.6	1418.0	182.8	111.3
Max-wait sec.	4689.8	799.9	589	4193.1	762.6	667.7	3202.9	655.2	374.4
Utilization %	90.2	74.9	62.9	90.2	76.1	62.9	97.2	78.6	66.2
Max_inCQ	2	2	2	2	2	2	2	2	2

From Table 3.10 and Figure 3.6, it is clear that the average load waiting time and the maximum load waiting time increase when the number of vehicles decreases. NVF and STDF always lead to smaller average load waiting times than the MODFCFS rule, but MODFCFS realizes much smaller maximum load waiting time, especially, when the vehicle utilization is high. This is not a surprising result since minimizing the maximum load waiting time is the nature of MODFCFS.

3.5.2 The production plant case

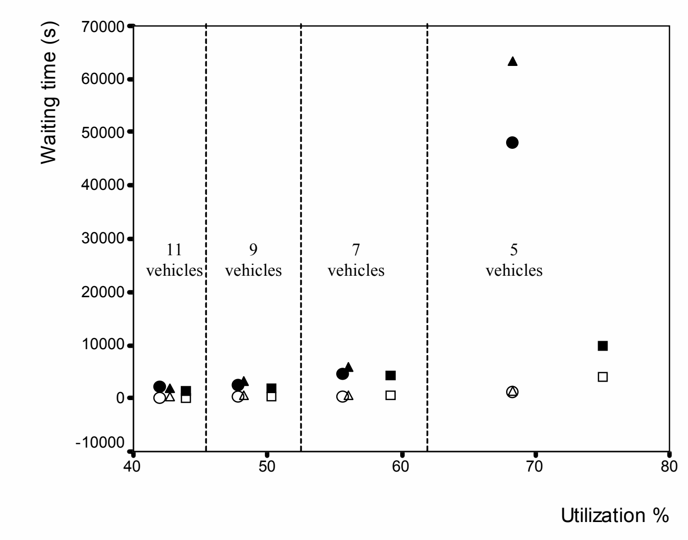


Figure 3.7 The scatter plot of average and max. load waiting times vs. vehicle utilization

● Maximum load waiting time for NVF; ○ Average load waiting time for NVF; ▲ Maximum load waiting time for STDF; △ Average load waiting time for STDF; ■ Maximum load waiting time for MODFCFS; □ Average load waiting time for MODFCFS.

Table 3.11 Results for the production plant case

Disp. Rules	NVF				STDF				MODFCFS			
# of vehicles	5	7	9	11	5	7	9	11	5	7	9	11
Ave-wait sec.	1076.7	262.0	199.1	176.1	1096.8	271.9	202.6	179.0	4076.1	588.0	256.5	182.8
Max-wait sec.	43339	6462.0	2365.0	2228.7	41043	5688.2	2857.4	1645.2	9894.3	4327.3	2021.4	1500.5
Utilization %	68.3	55.6	47.8	42.0	68.3	56.0	48.2	42.7	75.1	59.2	50.3	43.9
Max_inCQ	66	12	7	8	54	20	9	6	39	16	11	6

Analysis of the production plant case (Table 3.11 and Figure 3.7) leads to a similar result as in the previous case. A special characteristic of this case is that the travel distances are much longer, so loads may have to wait long, before they are transported when the NVF, STDF rules are used. MODFCFS results in a smaller value of the maximum load waiting time, particularly when only five vehicles are available. However, the maximum load waiting time is still very large, because of weekdays' high utilization and weekend's low utilization. When five vehicles are used, the maximum number of loads in critical queues is growing sharply. Note that, even with five vehicles the vehicle utilizations are still moderate. This is since transportation jobs have different distributions during the operating period. There are peaks during some periods (e.g. during the early morning and late afternoon of week days), and there may only be a small number of jobs to do in the weekend. An important reason for the sharp increase in maximum load waiting time when NVF and STDF are used in this case is the special structure of the production plant. There are some remote areas (see Figure 3.2) and load picks-up are not equally distributed over time. Vehicles are usually busy in some areas with high density of transportation jobs and may then ignore loads in remote areas. The second reason is the closeness of jobs. In this case we have three main pick-up queues, which are close to each other. However, two queues are closer to the track system than the third one and loads in this latter queue sometimes hardly qualify for transportation in comparison with loads in the other two. According to the results (Table 3.8 and Table 3.11), the resulting average load waiting time of NVF and STDF using nine vehicles are smaller than the corresponding value of DD using 11 vehicles. Thus, to obtain the performance of DD rule with 11 vehicles we can use the NVF or STDF with eight or nine vehicles. Actually, the company now uses only eight vehicles for internal transportation tasks.

3.6 Concluding remarks

In this chapter, we have investigated the performance of several well-known simple dispatching rules found in literature and some case-specific rules. We also studied the value of pre-arrival information and possible performance gains using a look-ahead policy when such pre-arrival information is available. Most of the dispatching rules came from Van der Meer (2000). However, in this chapter we have done more extensive experiments and have used more sound statistical analysis. These are crucial to guarantee reliable results. We used data of two real-life cases: a distribution center and a production plant. Our results are agreeable with Van der Meer's results for the current working conditions of the two cases. In addition, we extended his results for different working conditions (different numbers of vehicles used) and introduced another good dispatching rule (NVFTP).

After experimenting, several important conclusions can be drawn. The distance-based dispatching rules (NVF, STDF) perform significantly better with respect to average load-waiting time than the time-based dispatching rules (such as MODFCFS), regardless of vehicle utilization rates. Similar results are obtained in both cases, since in our models there is always enough space for loads in queues (implying no congestion and delay caused by overflowing queues). The need of preventing blocking effects (overflowing of queues) may lead to rules aiming at minimizing the maximum waiting time, such as MODFCFS. We observed that the relative ratio of the maximum load waiting time and the average load waiting time in the production plant is higher than the corresponding ratio in the EDC. The main reason can be the dispersion of the production plant layout.

According to the results of this chapter and also the results of De Koster et al. (2004), for the environments where queue space is not a restriction, the general ranking of the common dispatching rules based on average load waiting time appears to be: (1) NVF/NVFTP, (2) STDF, (3) case specific rules and (4) MODFCFS. According to Le-Anh and de Koster (2004b), for the EDC case, this raking is also true when another distribution (gamma) is used to generate loads. The loaded travel times are more or less constant, so reducing empty vehicle travel time is an important factor to minimize the average load waiting time. In our experiment environments (where queue space is not a restriction), distance-based dispatching rules (NVF, STDF) attempt to minimize empty vehicle travel time and they outperform the other rules. However, while minimizing the average load waiting time, the NVF and STDF rules also tend to maximize the maximum load waiting time. This is especially true when vehicle utilizations are high or in the presence of remote areas. The NVFTP rule (a truncation rule based on NVF) is designed to overcome this shortcoming. This rule helps to reduce the maximum load waiting time significantly. A drawback of this rule is that it is cumbersome to determine the best truncation parameter. Values of three to five times the average waiting time (in case of using NVF) appear to perform well.

In view of the different characteristics of the two models (three models in De Koster et al., 2004), the NVF (NVFTP) and STDF rules are likely to perform well in many real-life environments. Furthermore, using realistic pre-arrival information can significantly reduce the average load waiting time. Also rules where the load takes the initiative (NVF, LLD) perform slightly better than the rules where the vehicle takes the initiative (STDF, WLD). For the two companies studied, neither NVF nor STDF is used. The reason for this is that although these rules may seem simple, they are not yet available in standard warehouse management software. This suggests that there may be some room for improvement of this software.

One common characteristic of the two real-life cases is that queue space is not a restriction and we observed that distance-based dispatching rules outperform other rules (such as MODFCFS). However, as observed by Co and Tanchoco (1991), the guide-path layout and queue space restriction can influence dispatching rules' performances. Therefore, more experiments with different environments are still required in order to draw more general conclusions.

Since simple and good dispatching rules such as STDF or NVF perform very well for two real-world cases, we expect that more intelligent (or advanced) dispatching rules will improve the performance of the two real-life cases further. In the next chapter, we evaluate the performance of several more intelligent dispatching rules such as multi-attribute dispatching rules for the two cases described in this chapter.

Chapter 4

Control of vehicle-based internal transport systems using more intelligent dispatching rules

The dispatching rules used in the previous chapter are quite straightforward. Recent literature suggests that more advanced (or more intelligent) dispatching rules such as multi-attribute rules (Klein and Kim, 1996, see also 2.3.3) and pre-emptive rules (Bozer and Yen, 1996, see also 2.3.3) outperform simple dispatching rules. Thus, in this chapter, we extend the results of the previous chapter by comparing the performance of two representative rules (NVF, MODFCFS) in the previous chapter and that of several more advanced (or complex) dispatching rules. Two more advanced rules from literature are (1) the multi-attribute dispatching rule (Klein and Kim, 1996) and (2) the modified-shortest-travel-time-first rule (Bozer and Yen, 1996). We propose three new rules: (3) the nearest-vehicle-first with vehicle reassignment, (4) the nearest-vehicle-first with vehicle reassignment and time truncation, and (5) the combined dispatching rule which integrates multi-attribute dispatching with vehicle reassignment. Since these five rules are more advanced and more complicated than the dispatching rules in the previous chapter, in this chapter, we refer to them as the *complex* dispatching rules, since they are more complicated than single-attribute dispatching rules. Dispatching rules (2) - (4) are referred to as *reassignment* dispatching rules.

This chapter is partly based on the two papers: Le Anh and De Koster (2004c) and Le-Anh and de Koster (2004b). It is organized as follows: section 4.1 introduces the dispatching rules; section 4.2 describes experimental setups; section 4.3 evaluates the performance of dispatching rules for two real-life cases and finally conclusions are drawn in section 4.4.

4.1 Dispatching rules

In order to evaluate the complex dispatching rules' performance, we select two dispatching rules from the previous chapter for benchmarking. These rules are MODFCFS and NVF. NVF is among the best rules to minimize the average load waiting time and MODFCFS is the best rule to minimize the maximum load waiting time. Sections 4.1.1 and 4.1.2 describe these rules in more details.

4.1.1 Single-attribute dispatching rules

(a) Modified First-Come-First-Served (MODFCFS)
See section 3.2.3.

(b) *Nearest Vehicle First (NVF)*

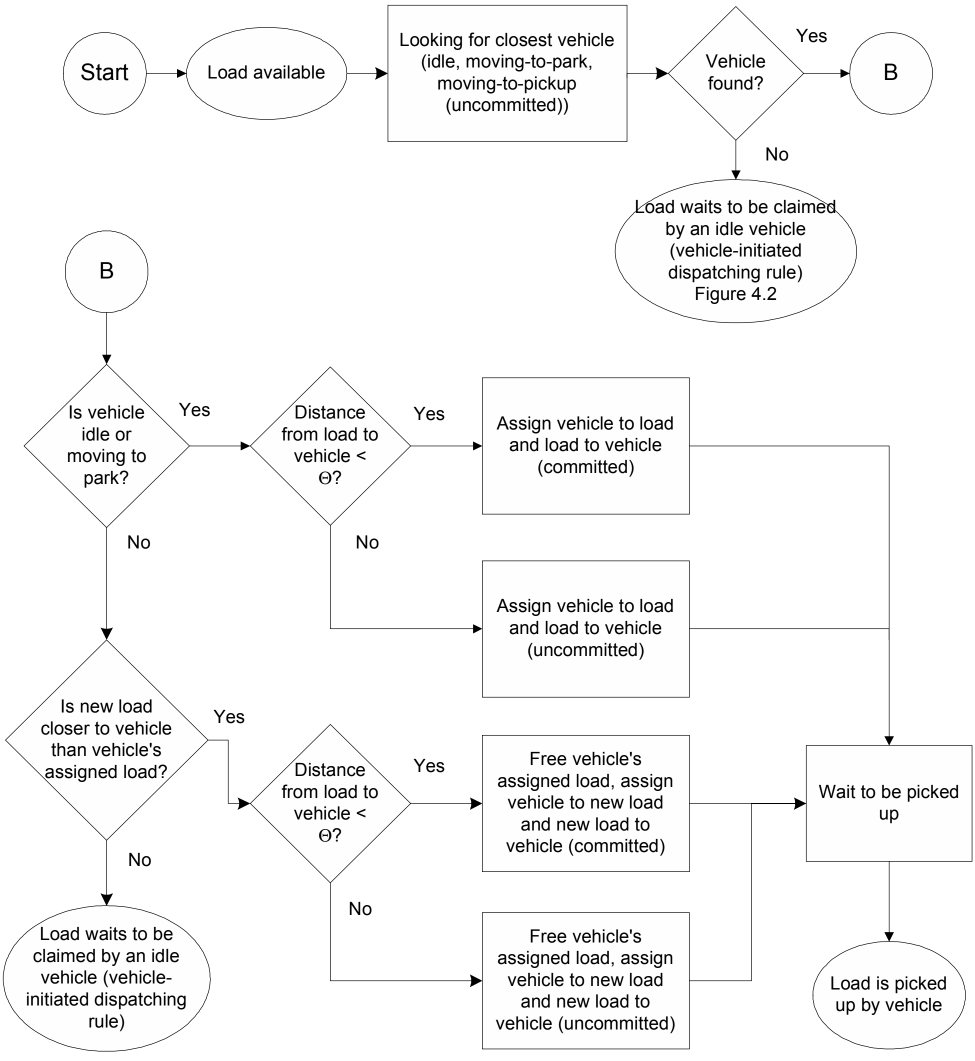
See section 3.2.3.

(c) Nearest Vehicle First with vehicle Re-assignment (NVF_R)

This rule operates similarly to NVF. The difference is that a load not only can claim idle vehicles, but also can claim moving-to-park vehicles. A just arrived load claims the closest (idle or moving-to-park) vehicle, if such vehicles are available. Otherwise this load waits at its released location until an idle vehicle claims it. When a vehicle becomes idle and is currently not claimed by any load, this vehicle searches for the closest load in the system (vehicle-initiated). This rule cannot reassign moving-to-pickup vehicles.

(d) Nearest Vehicle First with vehicle Re-assignment and Cancellation (NVF_RC)

This rule which is a simplification (to allow for implementation in AutoMod^{TM}) of the MOD STTF rule of Bozer and Yen (1996) can also reassign a moving-to-pickup vehicle (Figure 4.1). This rule differentiates from MOD STTF as follow: (1) a cancelled load becomes free and has to wait to be claimed by an idle vehicle; (2) the reassignment and cancellation procedure is invoked only when a new load arrived. NVF_RC is a load-initiated dispatching rule. When a load just enters the system, this load immediately searches for a vehicle as indicated in Figure 4.1. If this load cannot find any vehicle, it waits at its released location until being claimed by an idle vehicle. The main difference between NVF_R and NVF_RC is that in a system using NVF_R moving-to-pickup vehicles cannot be reassigned.



*(Vehicle status: **idle**: vehicle stay idle (has no job) at a parking location; **moving-to-park**: a vehicle has no job and is traveling to a parking location; **moving-to-pickup**: a vehicle is traveling to the vehicle's assigned load pick-up location; **committed**: means that the vehicle cannot be diverted to another destination, **uncommitted** otherwise.)*

Figure 4.1 The impact of the load behavior on dispatching rules with vehicle reassignment and cancellation.

In the system using NVF_RC, moving-to-pickup vehicles which travel uncommitted (see Figure 4.1) to pick-up a load can be reassigned to a new load. A load is committed to a vehicle if the vehicle claims the load and the travel distance from the vehicle to the load is smaller than a distance threshold Θ (chosen around the value of the average load transportation time). When a vehicle is moving-to-pickup a load, a new arriving load can claim this vehicle only if the load that this vehicle is going to pick-up, is not committed to this vehicle. When a vehicle becomes idle, this vehicle searches for a load as described in Figure 4.2.

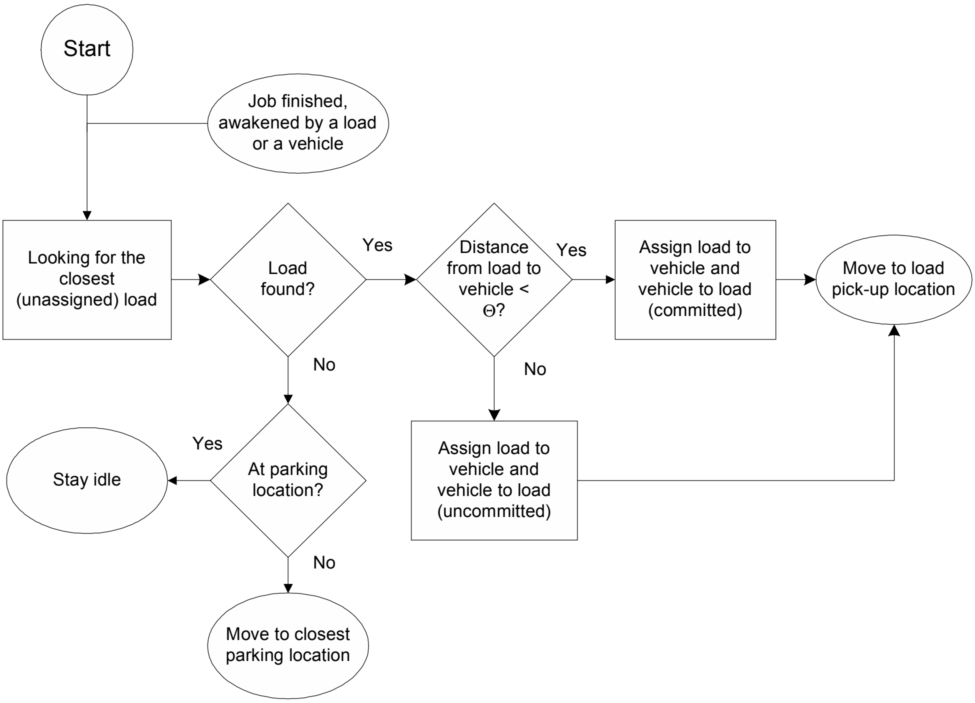


Figure 4.2 Vehicle-initiated dispatching.

4.1.2 Multi-attribute dispatching rules

(a) Nearest Vehicle First with Time Priority and vehicle Reassignment (NVFTP_R)

Under this rule, a load that just enters a system claims a vehicle in the same way as a load in a system using the NVF_R rule does. If this load cannot find a vehicle, it remains at its current location and waits until being claimed by an idle vehicle in a similar manner as STDF (see section 3.2.3). The difference is that in the case when waiting times of all loads in the system are smaller than a time threshold θ (load type I), an idle vehicle claims the closest one for transportation. If there are loads with load waiting times larger than θ (load type II), those loads have higher priority for transportation than loads type I. Among type II loads, a vehicle selects the nearest one. Loads (type I) are considered only when there are no loads of type II in the system. The time threshold (θ) is chosen around 4×(average load waiting time when the STDF rule is used). This value was found after several experiments for the two cases. If the vehicle under consideration cannot find a load to carry, this vehicle stays idle at its current location (if it is a parking location) or travels to the closest parking place.

(b) Multi-attribute dispatching rule (Multi-att)

A multi-attribute rule dispatches vehicles based on a multi-attribute dispatching function. This dispatching approach had been implemented by several authors such as Klein and Kim (1996) and Jeong and Randhawa (2001). Parameters of the dispatching function are selected depending on environments. In our experiments, capacities of queues are not the bottleneck in the system, so mainly vehicle travel distance and load waiting time affect the system performance. Therefore, we selected vehicle empty travel distance and load waiting time to be decision attributes. Let dis_{vi} denote the empty travel distance from the current vehicle (v) location to the pick-up location of load i and $wait_i$ denote the waiting time of load i. dis_{vi} and $wait_i$ are normalized to DIS_{vi} and $WAIT_i$ using the following expressions:

$$DIS_{vi} = \frac{dis_{vi} - \min_j dis_{vj}}{\max_j dis_{vj} - \min_j dis_{vj}}; \; WAIT_i = \frac{\max_j wait_j - wait_i}{\max_j wait_j - \min_j wait_j}$$

$\max_j dis_{vj}$, $\min_j dis_{vj}$ are the *max* and *min* travel distances from vehicle v to all loads in the system. $\max_j wait_j$, $\min_j wait_j$ are the *max* and *min* waiting times of all loads in the system. The attributes DIS_{vi} and $WAIT_i$ are used to compute the score function S_{vi}.

$$S_{vi} = w_1 \times DIS_{vi} + w_2 \times WAIT_i; \; w_1 + w_2 = 1$$

w_1, w_2 are weights of the vehicle empty travel distance and the load waiting time respectively.

The score function S_{vi} is then used to select the suitable load for a vehicle. When a vehicle becomes idle, this vehicle searches for a load to pick-up as follows:

- If this vehicle finds one or more loads in the system then:
  - + Values of the score function for all waiting loads in the system are calculated,
  - + A load that has the smallest value of the score function is chosen to be picked up,
- If this vehicle cannot find a job, it goes to the closets parking location and remains idle until being awakened by a load or by another vehicle.

Results of Jeong and Randhawa (2001) reveal that the additive multi-attribute rule performs better with a higher weight of the unloaded (or empty) vehicle travel distance. In addition, results of Van der Meer and De Koster (2000) show that distance-based dispatching rules perform better than time-based dispatching rules, so we give a higher weight to the vehicle empty travel distance attribute. Depending on the specific case, the best attribute weights can be found by experiments. In this case, we select the weights of travel distance and waiting time to be 0.8 and 0.2 respectively.

(c) The combined dispatching rule (Combi)

It is possible to improve multi-attribute dispatching rules by applying vehicle reassignment. Hence, we introduce a new combined rule (*Combi*), which uses vehicle reassignment in combination with multi-attribute dispatching. This rule is a load-initiated dispatching rule. When a new load enters the system, this load checks for an available vehicle (idle or moving-to-park) in the same manner with NVF_R. If this load finds a vehicle, it claims that vehicle, and the vehicle is redirected to pick-up the load. Otherwise, this load waits at its release location until an idle vehicle claims it. We do not use cancellation here (reassigning moving-to-pickup vehicles) since cancellation can eliminate the effect of multi-attribute dispatching. An idle vehicle selects a load to transport using the score function similar to the multi-attribute dispatching rule (*Multi-att*).

Table 4.1 summarizes characteristics of all dispatching rules used in this chapter. For all rules in Table 4.1, when a vehicle becomes idle (and has not been claimed by a load) and cannot find any load in the system for transportation, this vehicle will park at the closest parking location.

Table 4.1 Dispatching rules and their characteristics

	Vehicle-initiated	*Workstation-initiated*	*Time priority*	*Reassign-ment*	*Cancella-tion*	*Sources*
Single-attribute dispatching rules						
MODFCFS	✓					Srinivasan et al. (1994)
NVF	✓	✓				Egbelu and Tanchoco (1984)
NVF_R	✓	✓		✓		This thesis
NVF_RC	✓	✓		✓	✓	Similar to MOD STTF Bozer and Yen (1996)
Multi-attribute dispatching rules						
NVFTP_R	✓	✓	✓			This thesis
Multi-att	✓		✓			Klein and Kim (1996)
Combi	✓	✓	✓	✓		This thesis

4.2 Experimental environments

In this chapter, we use the same two real-world cases (the European distribution center and the glass production plant) in the previous chapter for experiments (see 3.2). These cases have been modeled using AutoMod^{TM} version 10. All assumptions for the simulation study in this chapter are kept the same as the previous chapter.

For each combination of experimental factors, we use a replication of ten runs to determine results. The lengths of one run are 75 hours and 14 days for the European distribution center and the glass production plant respectively. The replication/deletion approach (see section 3.1) was applied for determining mean values of performance indicators. Table 4.2 shows factors used in experiments.

Table 4.2. Experimental factors

Models	*The European Distribution Center*	*The glass Production Plant*
Dispatching rules	MODFCFS, NVF, NVF_R, NVF_RC, NVFTP_R, *Multi_att*, *Combi*	MODFCFS, NVF, NVF_R, NVF_RC, NVFTP_R, *Multi_att*, *Combi*
Number of vehicles	4, 5, 6	7, 9, 11

The performance criteria are also the same as the previous chapter. Minimizing the average load waiting time is the main performance criterion. Besides the average load waiting time, the maximum load waiting time, the vehicle utilization and the maximum number of loads in critical queues (see section 3.3) are considered as supplement.

The results of experiments have been analyzed and dispatching rules have been ranked (using the Tukey test with an overall confidence level of 95%, see section 3.1).

4.3 Performance evaluation

In this section, we evaluate the performance of the complex dispatching rules described in the previous section. The main performance criterion is minimizing the average load waiting time while the maximum vehicle waiting time is preferred to be as small as possible. The vehicle utilization is considered when they affect the performance of the dispatching rules. The rules' performances are evaluated with different numbers of vehicles in order to investigate the behavior of dispatching rules under different vehicle utilization levels. For example, the reassignment dispatching rules should perform well under low vehicle utilizations.

4.3.1 The European Distribution Center (EDC)

Performance evaluation for 4 vehicles case

Table 4.3 Experimental results for the distribution center (4 vehicles)

4 vehicles							
	MODFCFS	*NVF*	*NVF_R*	*NVFTP_R*	*NVF_RC*	*Multi_att*	*Combi*
Ave_wait *± 95% CI*	1144.62 ± 333.19	345.05 ± 86.53	320.03 ± 58.14	392.66 ± 84.79	319.26 ± 68.90	318.66 ± 54.66	310.63 ± 46.63
Max_wait	3953.71	5079.56	4421.63	2257.62	4372.24	2510.55	2682.03
Utilization%	97.86	92.72	90.63	91.57	> 81.12*	92.13	91.71
Max_inCQ	2	2	2	2	2	2	2

Ave_wait: average load waiting time (sec.); *Max_wait:* maximum load waiting time (sec.); *95% CI:* 95% confidence interval; *Utilization%:* vehicle utilization (%) = Percentage of [vehicle travel time (with a job) + vehicle's pick-up &set down time] / total vehicle available time}.

** We cannot get the exact number here since it is very difficult to separate "moving-to-pickup" time and empty traveling time in this case.*

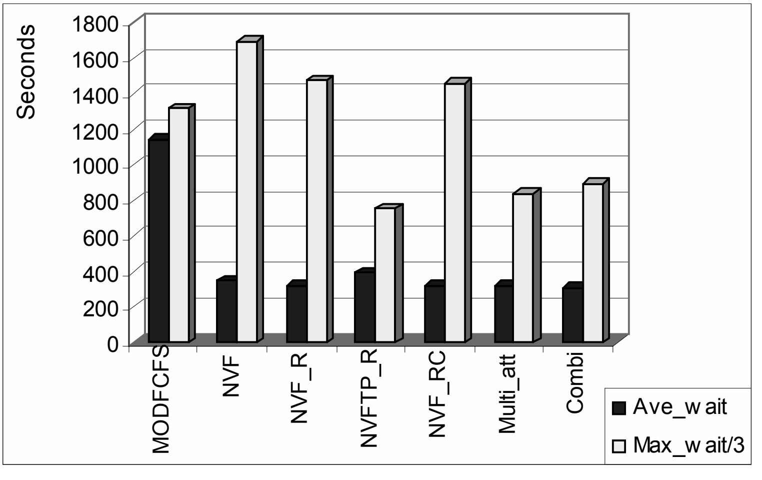


Figure 4.3 Performance of dispatching rules when 4 vehicles are used

Table 4.4 Rank of dispatching rules (Tukey - confidence interval 95%)

Average waiting time				Max waiting time		
RULES	*Rank*			*RULES*	*Rank*	
Combi	1			NVFTP_R	1	
Multi_att	1			Multi_att	1	
NVF_RC	1			Combi	1	
NVF_R	1			MODFCFS		4
NVF	1			NVF_RC		4
NVFTP_R	1			NVF_R		4
MODFCFS		7		NVF		4

Rules in different groups indicate that their mean values are significantly different (with 95% confidence interval).

Table 4.3 and Figure 4.3 show that MODFCFS performs worst according to the average waiting time and is also not good in terms of the maximum load waiting time. In spite of the fact that the performance of the six top rules in Table 4.4 (left) are not significantly different, the average load waiting time of the first rule (*Combi*) is about 20.9% smaller than the corresponding value of the sixth rule (NVFTP_R) (Table 4.3).

Results also show that the dispatching rules considering the load waiting time as one dispatching attribute (NVVFTP_R, *Multi_att*, *Combi*) perform well according to the maximum load waiting time criterion. In this case, the MODFCFS rule results in a very high value of the vehicle utilization (about 97.86%). This high value may lead to an unstable situation.

Table 4.4 shows two groups for average waiting times and two groups for max load waiting times. Three dispatching rules including NVFTP_R, *Multi_att* and *Combi* can be considered as the best rules in this case since these dispatching rules belong to the top group for two criteria (the average load waiting time and the max load waiting time). The *Combi* rule is preferred to NVFTP_R, since *Combi* results in a smaller value of the average load waiting time (20.9% smaller, see Table 4.3).

Performance evaluation for 5 vehicles case

Table 4.5 Experimental results for the distribution center (5 vehicles)

5 vehicles							
	MODFCFS	*NVF*	*NVF_R*	*NVFTP_R*	*NVF_RC*	*Multi_att*	*Combi*
Ave_wait *± 95% CI*	182.46 ± 25.79	131.76 ± 15.30	125.99 ± 9.89	130.85 ± 12.05	119.81 ± 10.30	134.54 ± 12.68	124.51 ± 10.21
Max_wait	704.01	1444.06	1154.44	675.39	1136.92	846.08	725.66
Utilization%	79.00	74.72	74.77	74.86	> 59.86	77.66	75.53
Max_inCQ	2	2	2	2	2	2	2

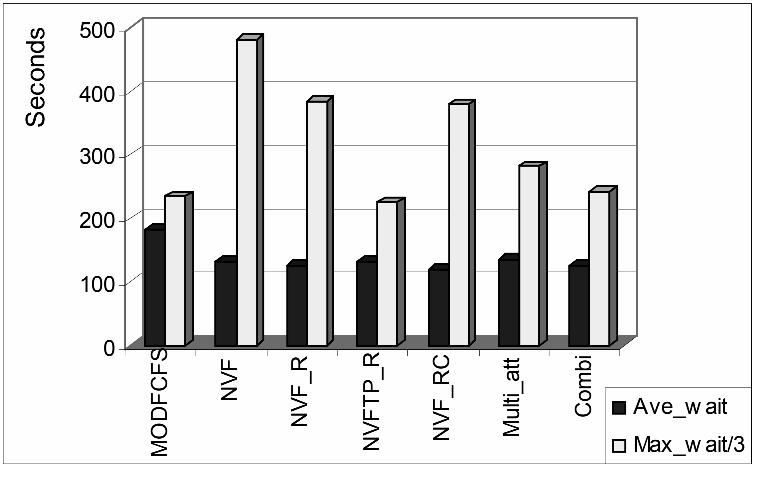


Figure 4.4 Performance of dispatching rules when 5 vehicles are used

When five vehicles are available, the vehicle utilization is not very high (about 75%), the reassignment dispatching rules perform very well according to the average load waiting time (Table 4.5 and Figure 4.4). In this case, vehicles spend a lot of time to go to their parking locations, so reassigning moving-to-park vehicles certainly saves unnecessary movements. The Tukey test (Table 4.6) reveals that NVFTP_R, *Multi_att* and *Combi* are the best rules in this case (in the top groups of two criteria). NVF_RC results in the smallest value of the average load waiting time, however it also leads to a high value of the maximum load waiting time (it belongs to the second group in Table 4.6 - right).

Table 4.6 Rank of dispatching rules (Tukey - confidence interval 95%)

Average waiting time				Max waiting time		
RULES	*Rank*			*RULES*	*Rank*	
NVF_RC	1			NVFTP_R	1	
Combi	1			MODFCFS	1	
NVF_R	1			Combi	1	
NVFTP_R	1			Multi_att	1	
NVF	1			NVF_RC		5
Multi_att	1			NVF_R		5
MODFCFS		7		NVF		5

The value of the maximum load waiting time corresponding to MODFCFS is relatively small (in the first group in Table 4.6 - right). However, MODFCFS is still the worst rule according to the average load waiting time.

Performance evaluation for 6 vehicles case

Table 4.7 Experimental results for the distribution center (6 vehicles)

6 vehicles							
	MODFCFS	*NVF*	*NVF_R*	*NVFTP_R*	*NVF_RC*	*Multi_att*	*Combi*
Ave_wait *± 95% CI*	113.64 ± 8.19	95.80 ± 3.66	92.50 ± 3.76	93.40 ± 4.55	88.26 ± 3.69	103.89 ± 3.70	92.70 ± 3.28
Max_wait	377.87	525.59	620.18	440.31	575.31	516.73	461.99
Utilization%	66.27	62.92	62.33	62.64	> 47.10	66.18	63.62
Max_inCQ	2	2	2	2	2	2	2

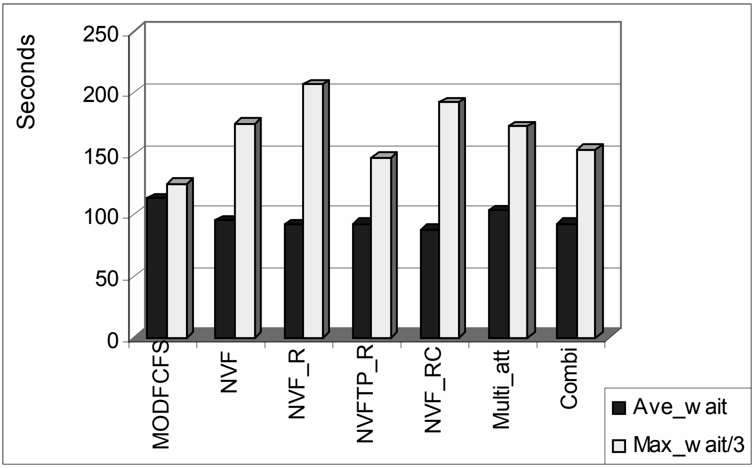


Figure 4.5 Performance of dispatching rules when 6 vehicles are used

We obtain a similar result as in the previous case. Rules using vehicle reassignment perform very well in terms of the average load waiting time. According to the average load waiting time criterion, NVF_RC performs significantly better than all other rules. Other reassignment rules also perform well (in the second group in Table 4.8 - left). Since the maximum load waiting times resulting from the complex dispatching rules in this case are only about 4 - 6 times of the average load waiting times, the maximum load waiting time criterion is not very important. Therefore, four dispatching rules including NVF_RC, NVF_R, NVFTP_R and *Combi* can be seen as the best rules in this case.

Table 4.8 Rank of dispatching rules (Tukey - confidence interval 95%)

Average waiting time						**Max waiting time**			
RULES	*Rank*					*RULES*	*Rank*		
NVF_RC	1					MODFCFS	1		
NVF_R		2				NVFTP_R	1		
Combi		2				Combi	1		
NVFTP_R		2				Multi_att		4	
NVF		2				NVF		4	
Multi_att			6			NVF_RC		4	
MODFCFS				7		NVF_R			7

We observe that under low vehicle utilization circumstances, reassigning moving vehicles has positive impact on the rules' performance. However, the multi-attribute dispatching has smaller impact.

4.3.2 The Glass Production Plant

In the previous chapter (section 3.5.2), we have seen that when only five vehicles are available for the glass production plant, the average and maximum load waiting times become too high. Therefore, in this chapter we only experiment with seven vehicles and more. In the glass production plant, the workload is quite different between weekdays and weekend. The vehicle utilization is high during weekdays, but is rather low at the weekends (the same number of vehicles is used). This leads to low overall vehicle utilization. If we decrease the number of vehicles used to less than seven, the average and max load waiting times become very high because of vehicle shortage during weekdays.

Performance evaluation for 7 vehicles case

Table 4.9 Experimental results for the glass production plant (7 vehicles)

7 vehicles							
	MODFCFS	*NVF*	*NVF_R*	*NVFTP_R*	*NVF_RC*	*Multi_att*	*Combi*
Ave_wait *± 95% CI*	611.90 ± 57.60	266.15 ± 7.84	270.73 ± 15.45	316.68 ± 28.11	257.22 ± 13.21	274.09 ± 10.35	265.61 ± 8.68
Max_wait	3480.55	4221.10	4818.29	3042.39	5365.26	3666.48	4457.12
Utilization%	59.02	55.63	55.25	56.17	> 40.94*	55.58	56.18
Max_inCQ	16	20	14	14	16	12	12

Ave_wait: average load waiting time (sec.); *Max_wait:* maximum load waiting time (sec.); *95% CI:* 95% confidence interval; *Utilization%:* vehicle utilization (%) = Percentage of [vehicle travel time (with a job) + vehicle's pick-up &set down time] / total vehicle available time}.

* *We cannot get the exact number here since it is very difficult to separate "moving-to-pickup" time and empty traveling time in this case.*

In this case, all dispatching rules except MODFCFS perform similarly. NVFTP_R performs slightly worse, but MODFCFS performs much worse.

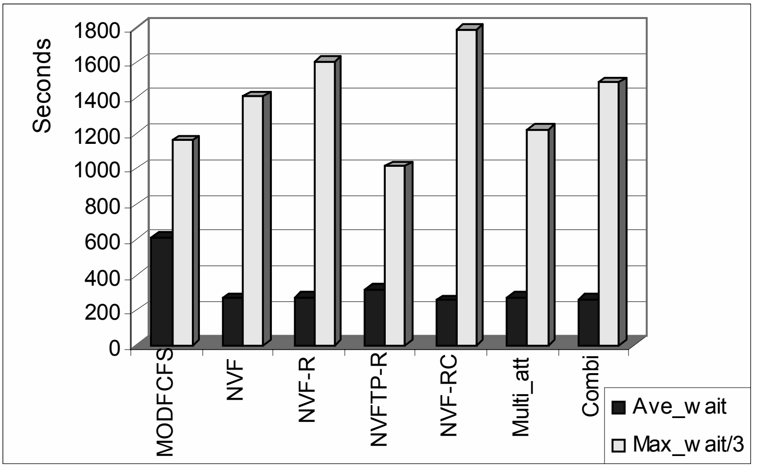


Figure 4.6 Performance of dispatching rules when 7 vehicles are used

Table 4.10 Rank of dispatching rules (Tukey - confidence interval 95%)

Average waiting time					Max waiting time		
RULES	*Rank*				*RULES*	*Rank*	
NVF_RC	1				NVFTP_R	1	
Combi	1				MODFCFS	1	
NVF	1				Multi_att	1	
NVF_R	1				NVF	1	
Multi_att	1				Combi	1	
NVFTP_R		6			NVF_R	1	
MODFCFS			7		NVF_RC		7

Rules in different groups indicate that their mean values are significantly different (with 95% confidence interval).

In Table 4.10, five dispatching rules (NVF_RC, NVF_R, NVF, *Multi_att* and *Combi*) are in the first groups for both criteria. NVF_RC results in the smallest value of the average load waiting time, but this value is not significantly smaller than the corresponding values of the four other rules (NVF_R, NVF, *Multi_att* and *Combi*). The MODFCFS rule is the worst one.

Performance evaluation for 9 vehicles case

Table 4.11 Experimental results for the glass production plant (9 vehicles)

9 vehicles							
	MODFCFS	*NVF*	*NVF_R*	*NVFTP_R*	*NVF_RC*	*Multi_att*	*Combi*
Ave_wait	263.32	197.80	197.21	216.26	185.90	203.57	199.91
± 95% CI	± 13.83	± 4.94	± 5.53	± 6.56	± 4.05	± 4.77	± 3.01
Max_wait	1808.67	2209.18	2473.18	1835.38	2066.98	2072.16	1757.88
Utilization%	50.25	47.73	47.44	48.55	> 31.41	48.23	48.08
Max_inCQ	11	9	10	11	14	8	8

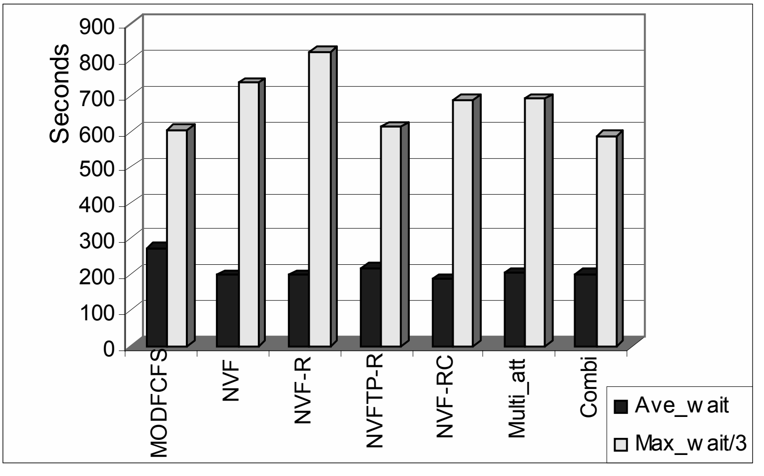


Figure 4.7 Performance of dispatching rules when 9 vehicles are used

Table 4.12 Rank of dispatching rules (Tukey - confidence interval 95%)

Average waiting time						Max waiting time	
RULES	*Rank*					*RULES*	*Rank*
NVF_RC	1					Combi	1
NVF_R	1					MODFCFS	1
NVF	1					NVFTP_R	1
Combi	1					NVF_RC	1
Multi_att		5				Multi_att	1
NVFTP_R			6			NVF	1
MODFCFS				7		NVF_R	1

All dispatching rules perform about the same according to the maximum load waiting time (Table 4.12), so the rules' ranks are purely based on the average load waiting time. In this case, four dispatching rules including NVF_RC, NVF_R, NVF and *Combi* are in the best group.

The dispatching rules considering the load waiting time as one (or only) decision attribute (NVFTP_R, *Multi_att*, *Combi* and MODFCFS) results in a bit smaller values of the maximum load waiting time in comparison with the other rules. However, the differences are not significant.

Performance evaluation for 11 vehicles case

Table 4.13 Experimental results for the glass production plant (11 vehicles)

11 vehicles							
	MODFCFS	*NVF*	*NVF_R*	*NVFTP_R*	*NVF_RC*	*Multi_att*	*Combi*
Ave_wait	189.10	176.26	175.16	178.20	167.10	178.99	178.85
± 95% CI	± 5.94	± 3.67	± 3.46	± 4.43	± 3.66	± 3.52	± 4.02
Max_wait	1343.28	1695.91	1737.66	1266.46	1679.05	1599.89	1578.53
Utilization%	44.05	42.39	41.51	42.55	25.15	42.26	42.46
Max_inCQ	6	6	6	6	7	5	5

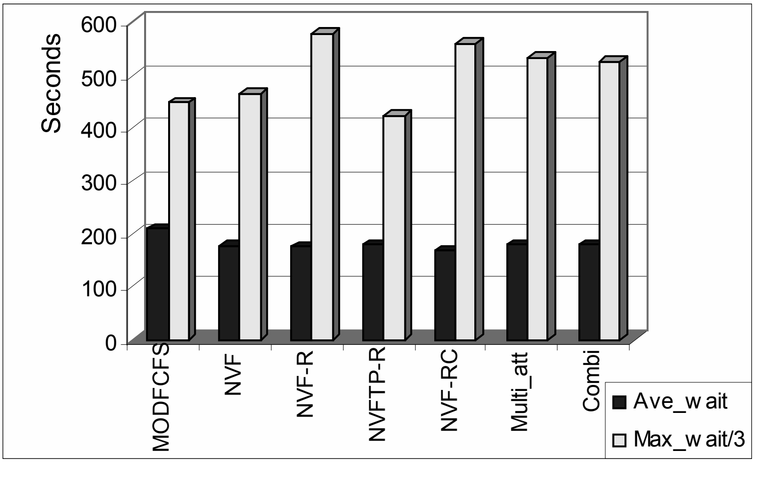


Figure 4.8 Performance of dispatching rules when 11 vehicles are used

Table 4.14 Rank of dispatching rules (Tukey - confidence interval 95%)

Average waiting time					Max waiting time	
RULES	*Rank*				*RULES*	*Rank*
NVF_RC	1				NVFTP_R	1
NVF_R		2			MODFCFS	1
NVF		2			Combi	1
NVFTP_R		2			Multi_att	1
Combi		2			NVF_RC	1
Multi_att		2			NVF	1
MODFCFS			7		NVF_R	1

The NVF_RC rule leads to the smallest value of the average load waiting time in this case. All dispatching rules perform similarly according to the maximum load waiting time. MODFCFS is the worst rule according to the average load waiting time.

Because of the special structure of this case (weekend and weekdays), we do not see the significant impact of using reassignment or multi-attribute dispatching rules. Reassigning moving vehicles or using a multi-attribute dispatching function still has some impacts.

4.3.3 Discussion

Experimental results show that MODFCFS is the worst rule for both cases and in all situations. This rule tends to minimize the maximum load waiting time, but in our experiments the truncation rule (NVFTP_R) and the rules considering the load waiting time as one decision factor (*Multi_att* and *Combi*) also result in low values of the maximum load waiting time.

Table 4.15 Ranking of dispatching rules according to the average load waiting time

<table>
<tr><th colspan="2">Low utilization</th><th colspan="2">High utilization</th></tr>
<tr><td>EDC</td><td>PP</td><td>EDC</td><td>PP</td></tr>
<tr><td>NVF_RC</td><td>NVF_RC</td><td rowspan="3">NVF_R, Combi, NVFTP_R, NVF, Multi_att, NVF_RC</td><td>Combi, NVF, NVF_R, NVF_RC</td></tr>
<tr><td>NVF_R, Combi, NVFTP_R, NVF</td><td rowspan="2">NVF_R, Combi, NVFTP_R, NVF, Multi_att</td><td>Multi_att</td></tr>
<tr><td>Multi_att</td><td>NVFTP_R</td></tr>
<tr><td>MODFCFS</td><td>MODFCFS</td><td>MODFCFS</td><td>MODFCFS</td></tr>
</table>

Table 4.16 Ranking of dispatching rules according to the maximum load waiting time

<table>
<tr><th colspan="2">Low utilization</th><th colspan="2">High utilization</th></tr>
<tr><td>EDC</td><td>PP</td><td>EDC</td><td>PP</td></tr>
<tr><td>MODFCFS, NVFTP_R, Combi</td><td rowspan="3">All rules</td><td rowspan="2">NVFTP_R, Combi, Multi_att, MODFCFS</td><td rowspan="2">Other rules</td></tr>
<tr><td>Multi_att, NVF, NVF_RC</td></tr>
<tr><td>NVF_R</td><td>NVF_RC, NVF_R, NVF, MODFCFS</td><td>NVF_RC</td></tr>
</table>

Table 4.15 and Table 4.16 summarize the ranking of dispatching rules according to the average and maximum load waiting times respectively. Results from both cases show that the system performance is strongly affected by the number of vehicles used (and the vehicle utilization). In the first case (EDC), the vehicle utilization has an important impact on different dispatching rules. The reassignment dispatching rules perform better under low vehicle utilizations and the multi-attribute rules perform better under high vehicle utilizations. A similar result is observed in the second case (the glass production plant), but the impact are less significant. The reason for the good performance of the reassignment dispatching rules under low vehicle utilizations is that, in this case vehicles can save a lot of unnecessary movements. For example, it is better to reassign a moving-to-park vehicle to pick-up a new load than letting this vehicle go to its parking location first. The multi-attribute rules (*Multi_att* and *Combi*) and the time truncation rule (NVFTP_R) can reduce the maximum load waiting time dramatically without significantly impact the average load waiting time.

The results of chapters 3 and 4 show that the MODFCFS rule is the worst rule in most cases. The main explanation for its bad performance is that this rule totally ignores vehicle empty travel distance when dispatching vehicles. Saving the vehicle empty travel distance certainly improves the system performance. In addition, in our experimental environments, queue space is not a restriction, so letting some loads wait for a long time does not cause any problem. This also explains why distance-based dispatching rules such as NVF perform well in our cases.

4.4 Concluding remarks

In this chapter, we have proposed several more intelligent dispatching rules (or complex dispatching rules) than rules in the previous chapter to apply in the two real-life cases (EDC and the glass production plant). The multi-attribute rule use two attributes (the vehicle empty travel distance and the load waiting time) to assign loads to vehicles. The idea is to reduce the maximum load waiting time resulting by pure distance-based dispatching rules such as STDF. The reassignment rules aim at saving unnecessary vehicle movements by redirecting moving-to-park (moving-to-pickup as well in case of NVF_RC) vehicles to pick-up new loads when they become available. NVF_RC also reassigns a moving-to-pickup vehicle to pick-up a new load which is closer to the vehicle current position than the vehicle assigned load. We also introduce the *Combi* rule which combines multi-attribute dispatching and vehicle reassignment. The *Combi* rule performs well and is robust to working conditions.

We have evaluated the performance of the complex dispatching rules (NVF_R, NVFTP_R, NVF_RC, *Multi-att* and *Combi*) for the two real-life cases. Results show that these rules are efficient and can reduce the average load waiting time in comparison to the NVF rule. A single best dispatching rule for all cases does not exist, but we can recommend specific types of rule for specific cases. The reassignment rules (NVF_R, NVFTP_R, NVF_RC, and *Combi*) are good for low vehicle utilization. The multi-attribute rules (*Multi-att* and *Combi*) are good for high vehicle utilizations. In general, the combined rule (*Combi*) is a good rule for most cases. According to above results, the relative ranking of the complex dispatching rules appears to be independent of the two environments. The main disadvantage of the *Combi* rule (and also Multi-att rule) is that this rule needs a good set of parameters' coefficients. Differently, NVFTP_R requires a good value of the time threshold.

The main findings of this chapter are introducing the two new dispatching rules (*Combi* and NVFTP_R) and showing that it is beneficial to reassign moving-to-park vehicles to pick-up closer loads. However, in a very busy system where vehicles have little free-time, reassigning moving-to-park vehicles is not useful. NVF_RC (Table 4.1) performs very well when vehicle utilization is not very high. However, the NVF_RC rule is sensitive to guide-path layout and is complicated to apply in practice. To apply this rule, the control system needs to monitor all vehicle positions precisely and continuously, which is not always possible in real-world VBIT systems. We introduce the NVF_R rule which is a variant of the NVF_RC rule. This rule is simpler and also performs well. The main purpose of introducing NVF_R is to examine impacts of reassigning moving-to-park vehicles in VBIT systems. Similar to Klein and Kim (1996) and Jeong and Randhawa

(2001), we found that multi-attribute rules are more robust to working conditions than single-attribute dispatching rules such as NVF. The performance of multi-attribute rules is dependent on selected parameters (for example, the vehicle empty travel time and the load waiting time) and their weights, so these parameters and their coefficients need to be selected carefully.

Table 4.17 summarizes the ranking of dispatching rules in this chapter. The dispatching rules in the first group are certainly better than NVF (one of the best dispatching rules of the previous chapter). In the first case (EDC), a vehicle utilization of 80% can be considered as the utilization threshold for dispatching rules' selection. It can be seen as a recommended threshold. In the second case (the glass production plant), this threshold is still appropriate, but only for weekdays.

Table 4.17 Ranking for dispatching rules

Rank	Dispatching rules
1	*Combi* – for all vehicle utilizations NVF_RC – for low vehicle utilizations Multi-att, NVFTP_R – for high vehicle utilizations
2	other rules
3	MODFCFS

In practice, when vehicles are controlled by human drivers, the rule with vehicle reassignment and cancellation may not be very attractive since a firm schedule is preferred by drivers and usually it takes some time for a driver to react to changes. Automated guided vehicles should not have any problem with the NVF_RC rule. However, reassigning vehicles too often is still not desirable even for AGVs.

In the next chapter, we study vehicle dispatching rules for a different type of environment: VBIT systems with many vehicles. This type of environment can be found in airport terminals (baggage handling systems) and has not received much attention from researchers.

Chapter 5

Control of vehicle-based internal transport systems using a large number of vehicles

In the two previous chapters, we have studied the performance of several dispatching rules for two real-world VBIT systems. The first one (EDC) is the warehouse of a retailer selling computer products. The second one is the production warehouse of a glass manufacturing plant. Although, these two systems have many differences, they still share some similarities. In this chapter (partly based on Le Anh and De Koster, 2004e), we study a totally different type of VBIT system which uses a large number of (automated) guided vehicles to transport loads within facilities. We call this an L-VBIT system or an L-VBITS. L-VBITSs can be found in modern airports in forms of baggage handling systems. L-VBITS usually follow unidirectional guide-paths. Although the literature on dispatching guided vehicles in VBIT systems is very rich, we cannot find many studies which particularly investigate the performance of vehicle dispatching rules in L-VBITSs (or similar environments). Therefore, in this chapter we aim at deriving robust and efficient dispatching rules for L-VBITSs. In an L-VBITS, loads from a station need to be moved to other stations as quickly as possible. Stations are normally far from each other and a large number of vehicles are required to serve loads.

This chapter is organized as follows: section 5.1 gives an introduction to L-VBITSs; section 5.2 introduces the experimental environments and dispatching rules; section 5.3 describes experimental setups; section 5.4 provides the performance evaluation of the used dispatching rules; and section 5.5 summarizes the findings of this chapter.

5.1 Introduction

Applications of L-VBITSs are continuously growing in numbers today. Baggage handling systems (BHSs) mentioned in section 1.2.4 is a type of L-VBITSs. Modern BHSs in large airport terminals use destination-coded vehicles (DCVs) to transport baggage. A DCV is a metal cart with wheels on the bottom and a plastic tub on top. Its only electronic device is a passive radio-frequency circuit that broadcasts a unique number identifying that particular car. DCVs are propelled by linear induction motors mounted to the tracks and can load and unload bags quickly (Figure 5.1). DCVs can operate at a high speed of 10 m/s. A DCV is also a type of guided vehicle.

Figure 5.1 A Destination-Coded Vehicle (courtesy of Vanderlande Industries)

Traditional BHSs in which conveyor-like systems are responsible for bag transportation are still popular in practice (Neufville, 1994). The main advantages of modern BHSs using DCVs over traditional ones are the capability of moving bags quickly over large distances and the sorting capability (a DCV can be sent to a specific destination). The important disadvantage of such BHSs is a high investment required for DCVs and their guide-path system. Figure 5.2 gives an example of a guide-path system for a BHS using destination-coded vehicles. This system uses rail-track type guide-paths connecting airport terminals. It contains few stations and operates with a large number of vehicles.

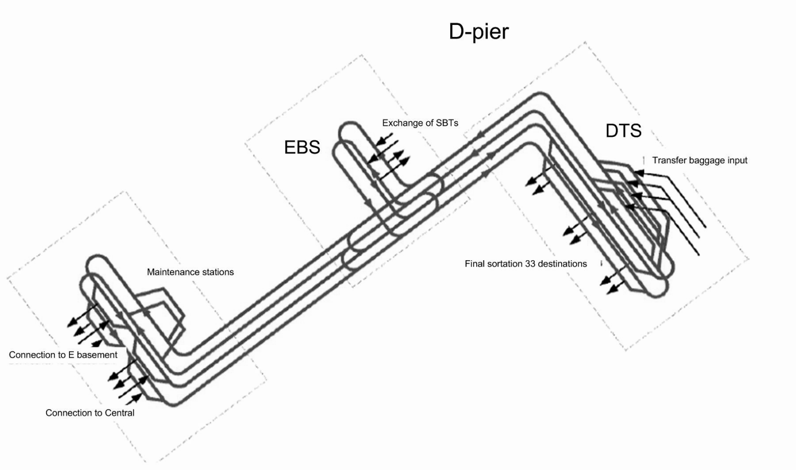


Figure 5.2 An example of a guide-path system for a BHS using DCVs (courtesy of Vanderlande Industries B.V.)

The literature on L-VBITSs is not abundant. We have found only few studies on similar systems (Chevalier et al., 2002; Talbot, 2003). Chevalier et al. (2002) and Talbot (2003) tackle the problem of estimating the number of vehicles needed in L-VBITSs satisfying a given service level. They also derived some good dispatching rules for L-VBITSs. There are many vehicles in L-VBITSs, so one of the tasks here is to manage the empty vehicles efficiently. This type of task was also tackled in Van der Heijden et al. (2002). They introduced some solution algorithms to allocate empty vehicles among terminals.

In this chapter, we derive some good dispatching rules for L-VBITSs. We propose two new multi-attribute dispatching rules (section 5.2.2). The two multi-attribute rules introduced in this chapter belong to the class of multi-attribute dispatching rules studied in the previous chapter, but use different parameters. We compare these rules with three good dispatching rules known from literature. These three dispatching rules include the modified-shortest-travel-distance-first rule (which is an adaptation of the STDF rule) and two vehicle control rules from Talbot (2003), which are the *Entrance Control* (EC) rule and the *Entrance Control rule with additional Assignment* (EC_A). Using simulation, we evaluate their performance for two experimental L-VBIT systems (adapted from Talbot, 2003).

In the next section, we describe experimental environments and introduce dispatching rules.

5.2 Experimental environments and dispatching rules

5.2.1 Experimental environments

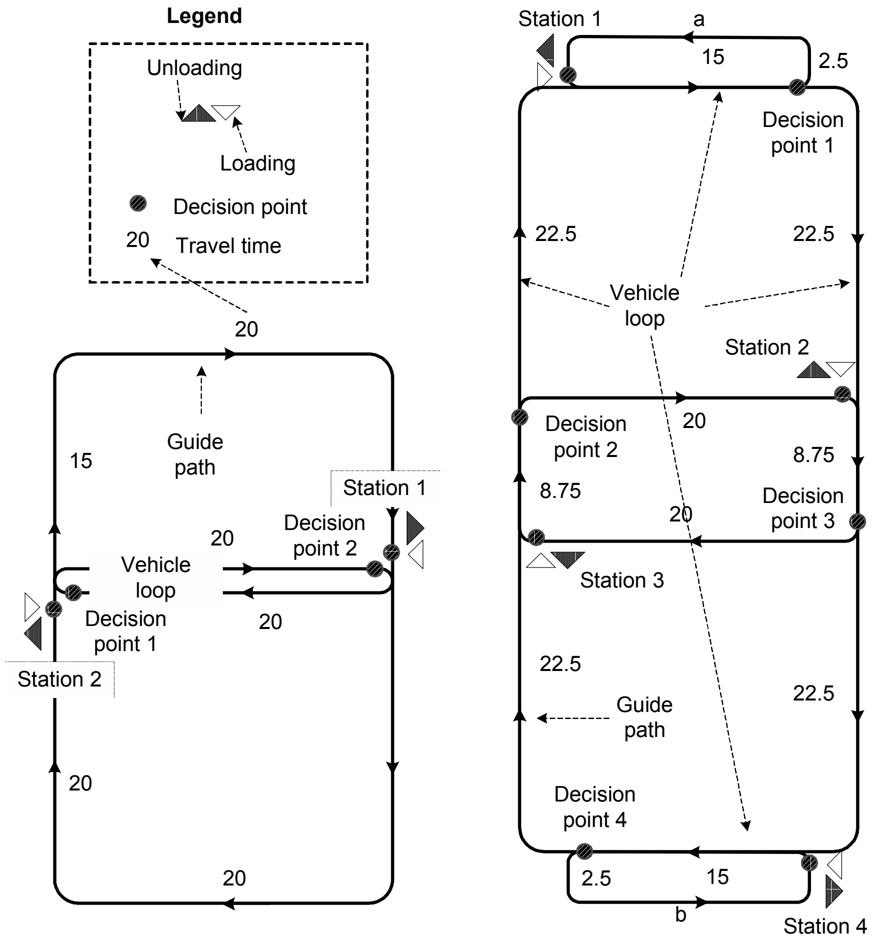


Figure 5.3 The experimental L-VBITSs (2 stations - left and 4 stations - right)

In this study, we have selected two layouts for L-VBITSs (Talbot, 2003). These layouts can be considered as simple cases of BHSs. Travel times for vehicles on each segment (in second) of both layouts are given in Figure 5.3. All path segments are unidirectional. One layout contains two stations and another contains four stations. Loads arrive at stations and need to be transported to other stations. A station in Figure 5.3 includes a loading station and an unloading station. A loaded vehicle arriving at a station can be loaded immediately after releasing its current load. A vehicle has dimensions so it is impossible to have too many vehicles on a short track. In the two above layouts, a track with the length of 20 can hold up to 200 vehicles. Since there are many vehicles in the systems and there are not enough parking places, many vehicles travel continuously on the vehicle loops (the loops which do not contain any station in Figure 5.3). Vehicles cannot bypass each other. Decision points and stations are locations where the next destination of a vehicle is decided. *Decision points* are located at junctions where vehicles have to decide which branch they should take. At a decision point *i* corresponding to station *i*, a vehicle decides that it should travel to the station *i* or takes other directions. In this chapter, for all dispatching rules, when an empty vehicle passes a station without being loaded, it travels directly to the vehicle loop. A station only decides the next destination of an empty vehicle when this station released a vehicle from its internal store according to the entrance rule with additional vehicle reassignment (see section 5.2.2).

For experiments, we use two cases from Talbot (2003). However, small modifications are required to allow implementation of these cases in AutoMod™. In our models, we created physical parking locations for vehicles at stations, so the released time of an empty vehicle to the main loop from a station's storage area is not zero (very small). In our implementation a released vehicle from a station might be blocked by a loading vehicle at this station.

In these two cases, load arrival rates at stations can be different. When load arrival rates at all stations are the same, we have a balanced system. Otherwise, this system is unbalanced. We have also adapted the load generation pattern of the original two cases. In the original models, a station can send loads to itself, but it is not the case in our models. We model these L-VBITSs in AutoMod™ version 10. All assumptions for the simulation study in this chapter are kept the same as in the chapter 3. For BHSs, we do not need the assumption on battery charging since DCVs operate without batteries. In our L-VBITSs, the shortest path from a vehicle to a station does not contain any other station on the path. In these models, a loading vehicle at a station may block other vehicles. This problem can be solved by creating bypass tracks for vehicles at loading stations.

5.2.2 Vehicle dispatching rules

Talbot (2003) indicates that (1) balancing the system workload (the most urgent station has the highest dispatching priority) and (2) increasing availability of vehicles at stations play important roles to improve the system performance. Chapter 3 and 4 show that (3) reducing the vehicle (empty) travel distances (time) is another important factor affecting the performance of VBIT systems. For L-VBITSs, we consider (1), (2) and (3) as three most important criteria. Other criterion such as meeting the load due time might be important as well.

(a) Modified shortest-travel-distance-first rule (MSTDF)

According to the STDF rule (chapter 3), a released or idle vehicle searches for the closest available load to pick-up. The closeness is measured in terms of travel distance. Results of chapter 4 suggest that reassigning (empty) vehicles to pick-up closer loads when they are available has a positive impact on the performance of dispatching rules. In general, travel distances between any two stations in BHSs (a typical type of L-VBITSs) are normally longer and the corresponding distances in traditional VBIT systems; therefore, it can be helpful to use vehicle reassignment (and cancellation) in BHSs (and also L-VBITSs).

Therefore, we propose the MSTDF rule (Modified-STDF) which works as follows:

- An idle vehicle searches for a new load when it reaches a decision point. If this vehicle finds the closest load, it will travel to the load's pick-up location.
- On the way to the assigned load pick-up location, if the vehicle passes another decision point and finds another load at a closer station, it will be redirected to the new station.

The MSTDF rule is actually a type of STDF with vehicle reassignment and cancellation.

(b) Entrance Control dispatching rule (EC)

EC works best for the balanced working condition (Talbot, 2003). The balanced working condition means that load arrival rates at stations are the same. This rule dispatches vehicles based on the net-stock of vehicles at stations and aims at increasing availability of vehicles at stations. The net-stock of vehicles ($s_i(t)$) at station i and time t is calculated as follows:

$$s_i(t) = x_i(t) + y_i(t) - c_i(t)$$

in which:

- $x_i(t)$: number of vehicles in the storage area of station i at time t,
- $y_i(t)$: number of vehicles (loaded or empty) traveling on the link between the decision point of station i and the corresponding station at time t. For example, in Figure 5.3 the four-station case, $y_1(t)$ is the number of vehicles on the link from the decision point 1 to the station 1 (path segment: decision point 1 -> a -> station 1) at time t.
- $c_i(t)$: number of loads waiting at station i at time t.

The net-stocks of vehicles are computed each time when a vehicle reaches a decision point.

The framework of the EC rule

- At the decision point *i* (Figure 5.3), a vehicle takes the direction of station *i* if $s_i(t) < S_i$ (a threshold value),
- If the net-stock of vehicles at a station reaches a threshold value S_i, this station releases a vehicle from its internal storage to the vehicle loop (Figure 5.3).

Talbot (2003) estimated the number of required vehicles and the threshold values (S_i) using a queuing approach. In this research, we use the estimated numbers (S_i) from her study and adjust them when necessary using simulation. A specific set of threshold values (S_i) might be only suitable for a specific situation. Therefore, when the load arrival pattern or the load arrival rate changes, we may need to adapt these values accordingly.

(c) Entrance control with additional assignment rule (EC_A)

EC_A operates in a similar manner with the EC rule. However, according to this rule, when a station releases a vehicle (at the second step of the EC rule's framework), this station sends this vehicle to the most urgent station (*additional assignment*). The most urgent station has the smallest value of the net-stock of vehicles.

The EC and EC_A rules in this chapter are originated from Talbot (2003).

(d) The multi-attribute dispatching rule (Multi-Att)

This rule dispatches vehicles based on a dispatching function associated with two parameters: the vehicle requirement at a specific station and the travel distance from the current vehicle position to the corresponding workstation. This rule aims at both reducing the vehicle empty travel time and balancing the workload among stations. In the two previous chapters, we have found that saving the vehicle empty travel distance (time), practically, is an efficient method to improve the system throughput. Therefore, we selected the travel distance as one term in the decision function. In addition, balancing the vehicle requirement (or workload) is another important criterion for L-VBITSs (Talbot, 2003). Thus we included this term in the decision function as well. We use the term "vehicle requirement" instead of "net-stock of vehicle" since it better reflects the characteristic of the decision factor in the dispatching function.

The dispatching function is defined as:

$$f_{vi}(d,s) = \alpha \times d_{vi} + \beta \times s_i$$

$$s_i = \frac{s_i(t) - \min s_i(t)}{\max_i s_i(t) - \min_i s_i(t)}; \; d_{vi} = \frac{d_{vi}(t) - \min d_{vi}(t)}{\max_i d_{vi}(t) - \min_i d_{vi}(t)}$$

- $s_i(t)$: the vehicle requirement of a station i at decision moment (t). This value is calculated in the same way as the net-stock of vehicles in Talbot (2003),
- $d_{vi}(t)$: the distance from the vehicle v to the station i at decision moment (t),
- $\max_i$, $\min_i$ $s_i(t)$: the max and min values of $s_i(t)$ for all stations i at decision moment (t),
- $\max_i$, $\min_i$ $d_{vi}(t)$: the max and min values of $d_{vi}(t)$ for all stations i at decision moment (t),
- s_i : the normalized value of $s_i(t)$ ($0 \le s_i \le 1$),
- d_{vi} : the normalized value of $d_{vi}(t)$ ($0 \le d_{vi} \le 1$),
- α, β: weights of the vehicle empty travel distance and the vehicle requirement respectively ($\alpha + \beta = 1$). Several values of α and β had been tested using simulation. We find that (α, β) = (0.5, 0.5) appears to be a good set of values.

The framework of the Multi-Att rule

- At a decision point (DC_i), a vehicle chooses the destination station based on the value of the decision function $f_{vi}(d,s)$ at the decision moment. The station with the smallest value of $f_{vi}(d,s)$ will be selected.
- If on the way to the destination station, the assigned vehicle passes another decision point (DC_j), this vehicle might be reassigned (to another station) based on new values of the decision function at DC_j.

The Multi-Att rule belongs to multi-attribute dispatching rules, but it uses different parameters in comparison with the Multi-att rule in the previous chapter.

(e) The modified multi-attribute dispatching rule (Multi-Mod)

We modify the dispatching function of the Multi-Att rule to obtain a new dispatching rule: the modified multi-attribute dispatching rule (Multi-Mod). The dispatching function is described as follows:

$$f_{vi}(d,s) = \alpha \times d_{vi} + \beta \times (s_i)^{\gamma}$$

- γ: power coefficient obtained by experiments ($\gamma = 4$ is a good value in our experiments).
- Other parameters are kept the same as for Mutli-Att. For this rule (α, β) = (0.5, 0.5) is also a good set of coefficients.

γ taking the value of 4 decreases the impact of the vehicle requirement in the dispatching function (since $0 \le s_i \le 1$). The main concern here is to test the behavior of a non-linear dispatching function.

5.3 Experimental setups

This section describes models' parameters and experiment setups.

Load pick-up and set down times

The loading (pick-up) and unloading (set down) times of a vehicle are 2.5 and 0 seconds respectively.

Load arrival rates

Two-station case

- The load inter-arrival distribution at the two stations is exponential and the load inter-arrival times (τ) at stations 1 and 2 are 3.5 and 5 seconds respectively,
- The probabilities that a load is sent from a station i to a station j are p_{ij} ($p_{11} = 0$, $p_{12} = 1$, $p_{21} = 1$, $p_{22} = 0$).

Four-station case

Two load arrival scenarios are selected:

- Balanced case: the load inter-arrival times are exponentially distributed with inter-arrival times (τ_1, τ_2, τ_3, τ_4) equal (12.2, 12.2, 12.2, 12.2) seconds.
- Unbalanced case: the load inter-arrival times are exponentially distributed with inter-arrival times (τ_1, τ_2, τ_3, τ_4) equal (4.5, 6, 9, 18) seconds.
- For both scenarios, the probabilities that a load is sent from a station i to a station j are p_{ij} ($p_{ii} = 0$ for all i, $p_{ij} = 1/3$ for all i, j and $i \ne j$).

The number of vehicles

- Two-station case: 3 levels have been used: 60, 65 and 70.
- Four-station case:
  + Balanced scenario: 4 levels have been used: 60, 70, 85 and 100,
  + Unbalanced scenario: 3 levels have been used: 70, 85 and 100.

Since the average load arrival rate is higher for the unbalanced scenario, we select an additional vehicle level (60) for the balanced scenario.

Vehicle dispatching rules

Five vehicle dispatching rules (MSTDF, EC, EC_A, Multi-Att and Multi-Mod) have been used. Threshold values for EC are (S_1 = 26, S_2 = 20) for the two-station case and are (S_i = 7, $\forall i$ = 1…4) and (S_1 = 10, S_2 = 8, S_3 = 6, S_4 = 4) for the four-station cases (balanced and unbalanced scenarios respectively). The S_i values have initially been taken from Talbot (2003). Since our models slightly deviate from Talbot's models, simulation experiments have been used to improve the values of S_i.

Performance criteria

Similar to the two previous chapter (chapter 3 and 4), the main criterion is minimizing the average load waiting time. We use other performance indicators (the maximum load waiting time and the maximum number of loads in queues) as supplement.

Simulation runs

For each scenario, a replication of ten runs of 120 minutes (about 3366 loads for the two-station case, 2360 loads for the balanced scenario four-station case, 4000 loads for the unbalanced scenario four-station case to be transported) has been used to gather data of performance indicators.

Statistical Analysis

The replication/deletion approach (see Law and Kelton, 2000) is used to determine values of performance indicators. Tukey's tests (see section 3.1) with 95% confidence interval (95%CI) are used to rank dispatching rules under various experimental conditions.

5.4 Performance evaluation

5.4.1 The two-station case

Since, in the two-station case, Multi_Mod performs the same as Multi_Att, we selected only one rule (Multi_Att) for evaluation. In this case, there are only two stations and loads are sent from one station to the other. Thus, the EC_A rule does not make much different here.

Table 5.1 Results for the two-station case

No. Vehs	Disp. rules	Ave_wait ± 95%CI (sec.)	Max_wait (sec.)	Max_inQ
60	*MSTDF*	9.66 ± 0.60	48.68	19
	Multi_Att	6.90 ± 0.21	37.72	15
	EC	6.76 ± 0.20	39.51	15
65	*MSTDF*	9.46 ± 0.30	49.22	19
	Multi_Att	6.62 ± 0.41	36.51	14
	EC	6.46 ± 0.36	37.35	15
70	*MSTDF*	8.55 ± 0.42	47.74	18
	Multi_Att	6.25 ± 0.37	35.90	14
	EC	6.45 ± 0.29	33.30	13
	EC'	6.31 ± 0.20	34.49	13

Ave_wait, *Max_wait*: average and maximum load waiting times; *95% CI*: the 95% confidence interval of the average load waiting time; *Max_inQ*: the maximum number of loads in queues; *No. Vehs*: number of vehicles; *Disp. rules*: dispatching rules. *EC'*: The EC rule using with another set of the threshold values (S_1 = 28, S_2 = 20).

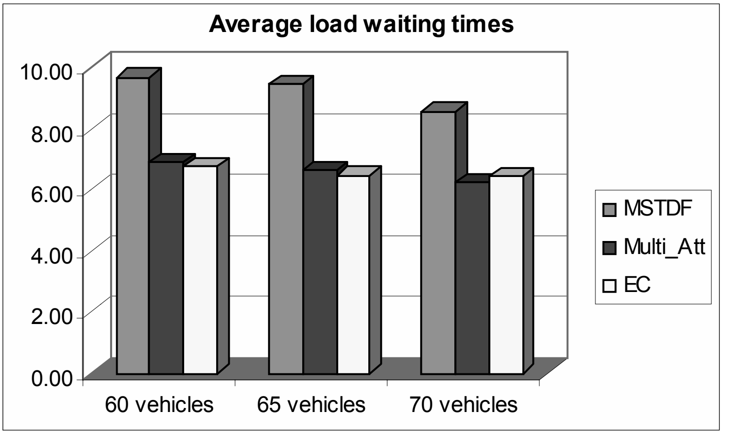


Figure 5.4 The average load waiting times for dispatching rules

Table 5.1 and Figure 5.4 show that the two dispatching rules (Multi_Att and EC) perform similarly. These rules outperform MSTDF for all three vehicle levels. These observations are also true for the maximum load waiting time criterion. In this case, the difference between two consecutive levels of the number of experimental vehicles is small (5 vehicles), we observed that the performance differences are not significant for all three dispatching rules. Results indicate that the EC rule is less sensitive to small changes in the number of vehicles. One reason is that we use the same threshold values for different numbers of vehicles.

The EC rule performance is, of course, dependent on the threshold values. However, it is not obvious to decide when we should change the EC rule's threshold values. Table 5.1 suggests that we should change these values when increasing the number of vehicles does not result in any clear improvement on the system performance (the average load waiting time in this chapter). Changing the threshold values (EC' rule), we can actually reduce the average load waiting time resulting by EC.

Table 5.2 The ranking of dispatching rules for the two-station case (Tukey's test 95%CI)

Disp. rules	*60, 65, 70 vehicles*	
Multi-Att	1	
EC	1	
MSTDF		3

Table 5.2 indicates that the difference between the average load waiting times resulting from the two rules (Multi_Att and EC) is not significant according to Tukey's test with a 95% confidence level.

5.4.2 The four-station cases

In the four-station cases, the two multi-attribute rules (Multi_Att, Multi_Mod) perform differently. EC and EC_A also behave differently under the two scenarios (balanced and unbalanced).

Balanced scenario

Table 5.3 Results for the four-station case, balanced scenario

No. Vehs	Disp. rules	Ave_wait ± 95%CI (sec.)	Max_wait (sec.)	Max_inQ
60	*MSTDF*	15.29 ± 1.20	121.90	18
	Multi_Att	17.76 ± 1.64	118.10	16
	Multi_Mod	17.69 ± 1.82	123.20	19
	EC	21.93 ± 2.43	177.20	25
	EC_A	19.82 ± 1.23	113.50	18
70	*MSTDF*	9.61 ± 0.49	81.60	14
	Multi_Att	6.09 ± 0.46	68.50	11
	Multi_Mod	5.35 ± 0.54	65.30	12
	EC	5.46 ± 0.83	77.10	12
	EC_A	10.25 ± 0.58	75.20	13
85	*MSTDF*	6.88 ± 0.41	68.80	12
	Multi_Att	2.95 ± 0.22	38.00	8
	Multi_Mod	2.60 ± 0.17	36.10	7
	EC	2.67 ± 0.16	32.10	8
	EC_A	5.45 ± 0.37	59.50	11
100	*MSTDF*	5.50 ± 0.17	62.40	12
	Multi_Att	2.09 ± 0.07	27.50	7
	Multi_Mod	1.94 ± 0.06	25.00	7
	EC	2.23 ± 0.07	26.70	7
	EC_A	3.63 ± 0.19	41.00	9

No. Vehs: number of vehicles; *Avewait*, *Maxwait*: average and maximum load waiting times; *Max_inQ*: the maximum number of loads in queues; *95% CI*: the 95% confidence interval of the average load waiting time; *Disp. rules*: dispatching rules; *Max_Q*: the maximum number of loads in queues.

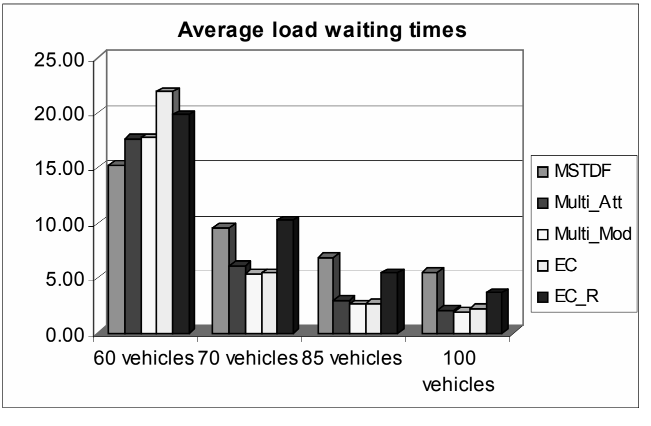


Figure 5.5 The average load waiting times for dispatching rules (balanced scenario)

From Table 5.3 and Figure 5.5, we find that the MSTDF rule is the worst rule in the four-station case when 70 vehicles or more are available. EC_A does not perform well either. This rule performs similarly to MSTDF when 70 vehicles are used. EC_A outperforms MSTDF when more vehicles are available. Three other rules (EC, Multi_Att and Multi_Mod) perform significantly better than MSTDF and EC_A. However, the ranking for these three dispatching rules can be different when different numbers of vehicles are used. In the balanced scenario, the Multi_Mod rule performs slightly better than Multi_Att and EC. The average waiting times obtained by these three dispatching rules are not significantly different when 70 and 85 vehicles are used (Table 5.4). In the case where 100 vehicles are available, the two dispatching rules (Multi_Att and Multi_Mod) perform significantly better than EC (Table 5.4). Surprisingly, when only a smaller number of vehicles (60) is used, MSTDF performs better than the other rules. In this case, MSTDF is the top rule in the first group in the ranking table (Table 5.4). A possible reason is that when only a small number of vehicles is available, saving vehicle travel time is more important than balancing the vehicle requirements. The EC rule performs badly in this case (60 vehicles). This is a sign that we may have to change the threshold values (S_i). In this case, changing the threshold S_i from 7 to 6 reduces the average load waiting time resulting by EC from 21.93 to 18.12.

Table 5.4 The ranking of dispatching rules for the four-station case, balanced scenario (Tukey's test 95%CI)

Rules	*60 vehicles*		*Rules*	*70 vehicles*		*Rules*	*85 vehicles*			*Rules*	*100 vehicles*			
MSTDF	1		**Multi-Mod**	1		**Multi-Mod**	1			**Multi-Mod**	1			
Multi-Att	1		**EC**	1		**EC**	1			**Multi-Att**	1			
Multi-Mod	1		**Multi-Att**	1		**Multi-Att**	1			**EC**		3		
EC_A		4	**EC_A**		4	**EC_A**		4		**EC_A**			4	
EC		4	**MSTDF**		4	**MSTDF**			5	**MSTDF**				5

Unbalanced scenario

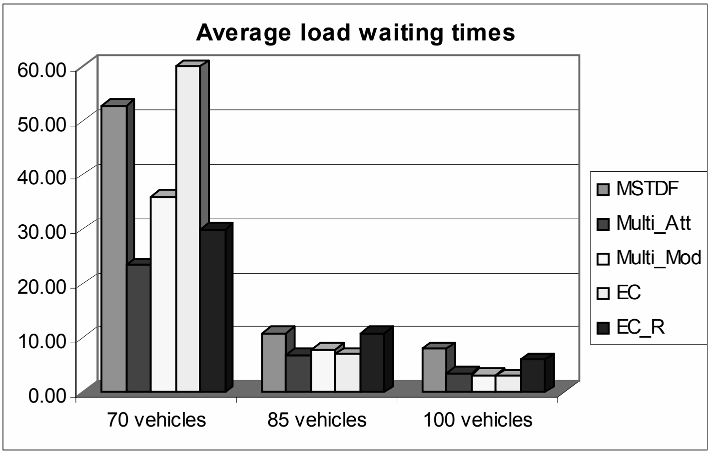
In the unbalanced scenario, when only 70 vehicles are available, the EC rule performs badly and is even worse than MSTDF. The EC_A rule performs quite well in this case. The two multi-attribute dispatching rules (Multi_Att and Multi_Mod) also perform well. MSTDF does not perform well in comparison with the two multi-attribute rules and EC_A.

Under the balanced scenario, we observe that saving the vehicle empty travel time seems to have more positive effects when less vehicles is available. It is also true for the unbalanced scenario: when 70 vehicles are used Multi-Mod performs worse than Multi-Att. As discussed before (section 5.2.2), the vehicle requirement has a stronger influence on Multi-Mod than on Multi-Att. It means that the empty vehicle travel time has a bigger impact on Multi-Att than on Multi-Mod. However, MSTDF still performs badly, since MSTDF cannot balance the vehicle requirement among stations well in the unbalanced situation.

Table 5.5 Results for the four-station case, unbalanced scenario

No. Vehs	**rules**	**Ave_wait ± 95%CI (sec.)**	**Max_wait (sec.)**	**Max_inQ**
70	*MSTDF*	52.80 ± 14.09	271.40	65
	Multi_Att	23.51 ± 5.49	123.50	31
	Multi_Mod	35.97 ± 3.92	176.00	41
	EC	190.52 ± 26.67	1073.00	247
	EC_A	29.94 ± 4.25	124.60	32
85	*MSTDF*	11.00 ± 0.84	88.80	23
	Multi_Att	6.95 ± 0.86	65.30	19
	Multi_Mod	7.92 ± 0.76	72.90	20
	EC	7.14 ± 1.44	79.20	22
	EC_A	10.89 ± 1.17	75.80	22
100	*MSTDF*	8.13 ± 0.87	70.10	19
	Multi_Att	3.55 ± 0.16	40.90	13
	Multi_Mod	3.28 ± 0.20	47.10	14
	EC	3.02 ± 0.15	32.90	12
	EC_A	6.09 ± 0.37	56.20	17

No. Vehs: number of vehicles; *Avewait*, *Maxwait*: average and maximum load waiting times; *Max_inQ*: the maximum number of loads in queues; *95% CI*: the 95% confidence interval of the average load waiting time; *Disp. rules*: dispatching rules; *Max_Q*: the maximum number of loads in queues.



**The average load waiting time resulted by EC (70 vehicles) exceeds the limit of the graph.*

Figure 5.6 The average load waiting times for dispatching rules (unbalanced scenario)

Table 5.5 shows that when higher numbers of vehicles (85, 100) are available, the two multi-attribute rules (Multi_Att and Multi_Mod) and the EC rule perform very well. EC performs slightly better than the two multi-attribute rules when 100 vehicles are used, but the differences are not significant (Table 5.6). In contrast, the EC_A and MSTDF rules perform badly. EC_A leads to a better performance than that of MSTDF when large numbers of vehicles (85, 100) are available.

Table 5.6 The ranking of dispatching rules for the four-station case, unbalanced scenario (Tukey's test 95%CI)

Disp. rules	*70 vehicles*			*Disp. rules*	*85 vehicles*		*Disp. rules*	*100 vehicles*		
EC_A	1			**Multi-Att**	1		**EC**	1		
Multi-Att	1			**EC**	1		**Multi-Mod**	1		
Multi-Mod	1			**Multi-Mod**	1		**Multi-Att**	1		
MSTDF		4		**EC_A**		4	**EC_A**		4	
EC			5	**MSTDF**		4	**MSTDF**			5

The two multi-attribute dispatching rules (Multi_Att and Multi_Mod) again obtain a very good performance in this case. They are more robust than the EC rule. From Table 5.6, we can consider the two multi-attribute dispatching rules (Multi_Att and Multi_Mod) as best for the unbalanced scenario.

Discussions

We found that the MSTDF rule does not perform well except for the balanced scenario when only a small number of vehicles (about less than 65, however we need more experiments to find a threshold number) is available. The MSTDF rule dispatches vehicles solely based on travel distance (time). Thus, it may result in some stations having too many vehicles while other stations might be forgotten and increases the average and maximum load waiting times. This observation is similar to the observation in the previous two chapters. As taking only the vehicle empty travel distance into account, the MSTDF rule cannot balance the workloads in the unbalanced situation. Two multi-attribute dispatching rules take into account the vehicle requirement at stations as well, so they can avoid the shortcoming of MSTDF. In our experiments, the two multi-attribute rules (Multi_Att and Multi_Mod) perform well under various working conditions. It is also similar to a conclusion in the previous chapter: multi-attribute rules are robust to working conditions.

The EC rule is similar to a type of decentralized dispatching rule. However, EC is still not a decentralized dispatching rule, since this rule requires some system information, such as the number of vehicles (loaded or empty) traveling on the link between the decision point and station *i* at the decision moment. The EC rule considers only information at a specific station at a decision moment, so it might cause unbalancing of the system workload. This is the reason why the EC rule may perform badly under the unbalanced working condition. By releasing empty vehicles to the most urgent station in the system, EC_A results in a better workload-balance. However, it may cause vehicles to travel excessive distances and therefore leads to a bad performance in many cases.

5.5 Concluding remarks

In this chapter, we have proposed two new multi-attribute dispatching rules (Multi-Att and Multi_Mod) which perform consistently well for the two experimental L-VBITSs. These dispatching rules are robust to different working conditions. We find that the EC dispatching rule from Talbot (2003) performs well for L-VBITSs, but this rule is not robust under unbalanced working conditions. The two multi-attribute dispatching rules (Multi-Att and Multi_Mod) normally work well with a given set of the parameters (α, β, γ) which is the same for all working conditions (balanced and unbalanced). This characteristic makes them easier to apply in practice.

In this chapter, we also find that the MSTDF rule (an adaptation of STDF) which should perform well in environments where queue-space is not a restriction (chapters 3, 4) does not do well in L-VBITSs. We should also note that the two multi-attribute rules in this chapter also use vehicle reassignment and cancellation. However, it is not exactly the same as in the previous chapter since vehicles in the two experimental L-VBITSs in this chapter do not have central parking locations. Using both multi-attribute dispatching and vehicle reassignment, the two multi-attribute rules in this chapter can be seen as a type of combined dispatching rule. Results from this chapter and the previous chapter indicate that multi-attribute and combined dispatching rules result in good performances for various working conditions and environments in general. In practice, if travel distances are significantly long, it might be useful to set up additional distributed parking locations for idle vehicles. In such a case, the guide-path system and dispatching policies have to be adapted accordingly.

The results of the chapters (3, 4 and 5) show that the system performance can be improved using more intelligent dispatching rules. However, studies from literature indicate that the scheduling approach performs significantly better in external transport than the dispatching approach. Therefore, we expect the scheduling approach will do the same for VBIT systems. In the next chapter, we devote our attention to the performance of various dynamic scheduling approaches for VBIT systems.

Chapter 6

Scheduling of vehicle-based internal transport systems

As discussed in the literature review chapter (section 2.3), the scheduling approach for vehicle-based internal transport system has not received much attention from researchers. In this chapter, we devote our attention to (dynamic) scheduling approaches for VBIT systems. This chapter is based on Le-Anh and de Koster (2004d).

In practice, control of vehicles using dispatching rules is the most popular strategy in VIBITSs. Generally, system controllers dispatch vehicles (or AGVs) using simple and intuitive dispatching rules such as the nearest-vehicle-first rule. An important practical reason for selecting simple vehicle dispatching rules is that they are easy to adapt to warehouse management systems (WMSs) or shop-floor control systems (SFCs). Also, the dynamic and stochastic environments in which vehicles have to work and the relatively short travel times make a vehicle dispatching approach more obvious than a scheduling approach. The main characteristics of scheduling problems in real-life VBIT systems are high traffic density, short planning horizon due to stochastic load arrivals and many possibilities of vehicle interferences. These characteristics make offline schedules useless. However, a vehicle scheduling approach with a rolling horizon and frequent rescheduling might lead to a better overall system performance than a dispatching approach. Since the scheduling approach has been used efficiently for external transport, we develop such scheduling strategy for VBIT systems in this chapter. The main purposes are to investigate the potential contribution of the scheduling approach for VBIT systems.

This chapter is organized as follows. Section 6.1 gives a definition for VBIT scheduling problems; section 6.2 formulates the vehicle scheduling problem mathematically and describes its characteristics; section 6.3 discusses the literature related to vehicle scheduling problems; section 6.4 presents the experimental layouts; in section 6.5, we propose solution approaches for static and real-time scheduling problems, and we also provide an empirical (average-case) performance evaluation for the proposed heuristics; in section 6.6, we describe experimental environments and parameters and evaluate performance of the proposed dynamic scheduling approaches and two vehicle dispatching rules; finally in section 6.8, we give some conclusion remarks.

6.1 Problem definition

In general, a VBIT scheduling problem involves assigning a set of vehicles to transport a given set of loads. Similar to the chapters 3 and 4, the main objective of the scheduling problem is minimizing the average load waiting time. In this chapter, all assumptions are kept the same as in the chapter 3, and we have two additional assumptions: (a) there are no traffic problems (congestion, deadlock, etc.); (b) vehicles can stay at their drop-off (or pick-up) locations. Although these assumptions hold for many VBIT systems in practice, problems such as vehicle congestion might have an impact. Mathematically, the scheduling problem of a VBIT system can be formulated as a pick-up and delivery problem with time windows (PDPTW), in which a vehicle picks-up loads at several locations and delivers them to their destinations satisfying certain time-window restrictions. However, the assumptions of this chapter permit us to reformulate the VBIT scheduling problem as a multiple traveling salesman problem with time windows (*m*-TSPTW) (section 6.2).

Since the *m*-TSPTW is an NP-Hard problem (Desrochers et al., 1988), even a small instance can be very difficult to solve to optimality. We therefore propose three heuristics for solving static (offline) instances of the scheduling problem, which are later applied with rolling horizons. We also propose a look-ahead dynamic assignment algorithm for the VBIT scheduling problem, which is based on Fleischmann et al. (2004). The heuristics and the dynamic solution approaches are described in greater detail in section 6.5.

We then evaluate the performance of the proposed dynamic (or real-time) scheduling approaches and compare their performance with the two best-performing dispatching rules in the pervious chapters (*NVF* and *NVF* with a look-ahead period - *NVF_LA*) for two experimental environments. In chapter 4, we identified the multi-attribute rule (*Multi-att*) and the *NVF_RC* as the two best dispatching rules. However, for the two experimental systems in this chapter, *Multi-att* and *NVF* perform similarly. We assume that vehicles can always park at their drop-off locations and also because of the simple layouts in this chapter reassigning vehicles makes a little difference here. Thus, we selected *NVF* for evaluation. In chapter 3, we have shown that load pre-arrival information may improve the system performance substantially, so we choose *NVF_LA* as the other rule for benchmarking.

6.2 Mathematical formulation for the static case

For offline VBIT scheduling, we define a set of available vehicles (K) and a set of jobs (N) which need to be picked-up within time-windows $[e_p, l_p]$ ($p \in N$) and dropped off at their delivery locations. The scheduling problem for VBIT systems can be formulated as a PDPTW. However, we reformulate this problem as an m-TSPTW by projecting time-windows at delivery locations to the corresponding pick-up locations (assuming a deterministic transport time) and logically considering a pick-up and a corresponding delivery job as a single job-node. If the time-window at the pick-up location is $[e_p, l_p]$, and at the delivery location is $[e_d, l_d]$, and the travel time between the two locations is t_{pd}, the time-window of the job-node will be $[e_n, l_n]$ with $e_n = e_p$, $l_n = \min(l_p, l_d - t_{pd})$. We suppose that the time-window projection for job-nodes is always feasible ($[e_n, l_n] \neq \emptyset$). In many VBIT systems, only one-sided time-windows are present at pick-up locations (load release times, or r_p) and no time-windows are present at delivery locations, so $[e_n, l_n]$ is always $\neq \emptyset$. The travel time from job-node i to job-node j (t_{ij}) equals the travel time from the origin of job i (i^+) to the destination of i (i^-) ($t_{i^+i^-}$) plus the travel time from the destination of i to the origin of j ($t_{i^-j^+}$).

The m-TSPTW can be seen as a graph $G = (V, A)$, in which V is a set of vertices and A is a set of arcs. $V = \{0\} \cup N \cup \{n+1\}$, where $\{0\}(\{n+1\})$ denotes the depot (end depot) and $N = \{1, ..., n\}$ is the set of (job-)nodes. $A = \{0\} \times N \cup I \cup N \times \{n+1\}$, where $I \subset N \times N$ is the set of arcs connecting job-nodes. $\{0\} \times N$ contains the arcs from the depot to job-nodes and $N \times \{n+1\}$ contains the arcs from job-nodes to end depot (which is the same physical location as the depot in our computations). For each arc $(i,j) \in A$, there is an associated travel time (distance) t_{ij} and for each job-node i there is an associated time-window $[e_i, l_i]$. In the following, K is the number of vehicles and B is a big number.

Decision variables are:

- x_{ij}^{k} ($(i,j)\in A$, $k\in K$) that take the value 1 if arc (i,j) is covered by vehicle k, and 0 otherwise,
- D_i ($i\in N$) indicates the service start time of (job-)node i,
- D_0^k, D_{n+1}^k are the starting time of vehicle k at the depot and the arrival time of vehicle k at the end depot.

As discussed in previous chapters, in VBIT systems, minimizing the average load waiting time is in practice the most important objective of the VBIT scheduling problem.

The model formulation becomes then:

$$\textit{Minimize } \frac{1}{|N|}\sum_{i\in N}\left(D_i - e_i\right) \tag{6.1}$$

subject to

$$\sum_{k\in K}\sum_{j\in N} x_{ij}^{k} = 1 \qquad \forall i\in N \tag{6.2}$$

$$\sum_{j\in V} x_{ij}^{k} - \sum_{j\in V} x_{ji}^{k} = 0 \quad \forall i\in N, \forall k\in K \tag{6.3}$$

$$\sum_{j\in N} x_{0j}^{k} = 1 \qquad \forall k\in K \tag{6.4}$$

$$\sum_{i\in N} x_{i,n+1}^{k} = 1 \qquad \forall k\in K \tag{6.5}$$

$$D_i + t_{ij} - D_j \le B\left(1 - x_{ij}^{k}\right) \qquad \forall i,j\in N, \forall k\in K \tag{6.6}$$

$$D_0^k + t_{0j} - D_j \le B\left(1 - x_{0j}^{k}\right) \quad \forall j\in N, \forall k\in K \tag{6.7}$$

$$D_i + t_{i,n+1} - D_{n+1}^k \le B\left(1 - x_{i,n+1}^{k}\right) \ \forall i\in N, \forall k\in K \tag{6.8}$$

$$e_i \le D_i \le l_i \qquad \forall i\in V \tag{6.9}$$

$$x_{ij}^{k} \quad \textit{binary} \qquad \forall i,j\in V, \forall k\in K \tag{6.10}$$

Constraints (6.2)-(6.5) form a multi-commodity flow formulation. The constraint (6.6) indicates that if a vehicle k serves node j after node i, the constraint $D_i + t_{ij} \le D_j$ must be satisfied. Constraints (6.6)-(6.8) ensure feasibility of the schedule. Equations (6.9) and (6.10) are time-window and binary constraints. The number of binary and linear variables in this formulation are $K\times(N+2)\times(N+2)$ and $N + 2\times K$ respectively.

6.3 Literature overview on the scheduling problem solutions

In the literature review chapter, we have already discussed some characteristics and solution approaches for the VBIT scheduling problem. In this section, we review solution approaches for the vehicle scheduling problem in greater detail. In the literature, the PDPTW, *m*-TSPTW and vehicle routing problems with time windows (VRPTW) have been studied extensively (Desrochers et al., 1988; Savelsbergh and Sol, 1995). Desrochers et al. (1988) provide a review of vehicle routing with time windows including PDPTW and *m*-TSPTW and solutions approaches. Savelsbergh and Sol (1995) focus on PDPTW (referred to as general pick-up and delivery problems – GPDPs) and their dynamic versions. In their paper, the *m*-TSPTW is referred to as the full truckload PDPTW. The dial-a-ride problem is another important variation of vehicle routing problems. In a dial-a-ride problem, a vehicle may pick-up multiple-loads, which is not possible in *m*-TSPTW.

Desrochers et al. (1988) mention two main types of optimization algorithms for VRPTW: dynamic programming and branch-and-bound. Both methods are very time consuming and cannot solve practical problems within an acceptable time limit. Dumas et al. (1991) introduce an exact algorithm to solve PDPTW using a column-generation scheme. The sub-problem (or pricing problem) is a constrained shortest-path problem. Their algorithm can handle multiple depots and different vehicle types. Desaulniers et al. (1998) propose a similar approach to solve multi-depot vehicle scheduling problems with time windows and waiting costs. In order to solve practical-size problems, they also propose a heuristic to speed up the branch-and-bound process. Savelsbergh and Sol (1998) and Xu et al. (2003) propose some speed-ups of the column-generation algorithm. They use several heuristics to generate columns with negative reduced costs and eliminate unattractive columns by sophisticated column management schemes. Besides set-partitioning and column-generation approaches, several other heuristics have been proposed for the VRPTW, such as saving heuristics (Kindervater and Savelsbergh, 1992; Laporte et al., 2000; Cordeau et al., 2002).

Psaraftis (1988) provides a survey on solution approaches for dynamic vehicle routing problems. Two main approaches include an adaptation of the static solution and an implementation of static algorithms under a rolling horizon. Savelsbergh and Sol (1998) use the rolling horizon approach to solve a dynamic PDPTW. Several authors adapt the Tabu search approach which is used for the static problem to dispatch vehicles (trucks) dynamically (Rego and Roucairol, 1995; Gendreau et al., 1999). Gendreau et al. (1999) implement the Tabu search approach on a parallel platform to speed up the solution algorithm. Powell and Carvalho (1998) use a logistics queuing network to solve a dynamic fleet management problem. Ichoua et al. (2000) present another strategy to schedule vehicles in real-time. According to their strategy, the current destinations of vehicles at the decision moment can be changed by the re-optimization procedure. In their research, they assume that the re-optimization procedure takes δt time to perform. Hence, dummy points are used to represent the vehicle positions at the finishing time of the re-optimization procedure. These dummy points are actually used by the re-optimization algorithm to represent the vehicle "current" positions. They also introduced some rules for estimating value of δt. Most studies on the real-time vehicle scheduling do not take δt into account. However, if δt is short enough, it should not affect the quality of real-time scheduling solutions.

Yang et al. (2004) study a dynamic truckload PDP. They propose several benchmark local policies that are actually similar to vehicle online dispatching rules in VBIT systems. They also propose two re-optimization policies (MYOPT and OPTUN) to solve the problem dynamically. The MYOPT policy solves a static instance at every step (when information about a new job arrival is received). OPTUN differs from MYOPT by including some opportunity costs which are based on probabilistic knowledge of future requests in the optimization model. The probabilistic knowledge of future requests helps to improve the solution quality. They prove that two re-optimization policies outperform local policies. Fleischmann et al. (2004) use a dynamic assignment algorithm to assign jobs to vehicles. The main objectives are minimizing the total order delays and vehicle empty travel time. They show that their approach is superior to assignment rules and some insertion algorithms. Kim and Bae (2004) propose a look-ahead dispatching method to dispatch AGVs at a container terminal, in which tasks must be carried out according to a fixed order. The main objective is to minimize the delays times of container cranes. They formulate the dispatching problem as a mixed-integer programming problem and propose a heuristic to solve it. They apply this heuristic dynamically to schedule AGVs. The dispatching heuristic is invoked each time an AGV becomes free. The dispatching procedure takes only limited tasks into consideration. Using simulation, they show that their look-ahead dispatching methods outperform the shortest-travel-distance first, earliest-due-date and revised shortest imminent operation dispatching rules.

After studying the literature on the dynamic (or real-time) vehicle scheduling, we find that most studies concern external transport systems. The problem of dynamic scheduling of VBIT systems has not attracted many researchers. This problem is similar to the vehicle scheduling problem for external transport. However, it also has some differences (see section 1.3.1). In most external transport problems, a large part of scheduling data is known in advance, so a tentative schedule for vehicles can be derived. Unknown jobs arrived will be incorporated in the current vehicle schedule. In VBIT systems, normally, we only know a limited (small) number of jobs in advance. Therefore, in VBIT systems we have to schedule with a shorter horizon and reschedule more often than in external transport systems. In this chapter, we systematically compare the performance of different real-time scheduling approaches and dispatching rules for two experimental environments under various working conditions.

6.4 Experimental layouts

In this section, we describe the two layouts used in this chapter. More experimental data such as load arrival rates is given in section 6.6.

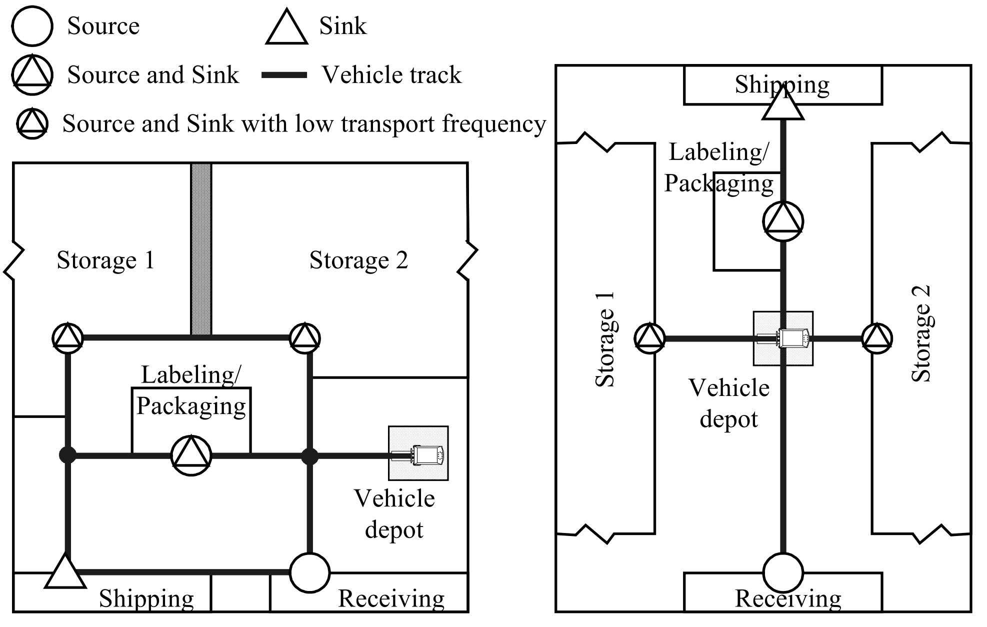


Figure 6.1 U-layout (left) and I-layout (right) used in experiments

We select two warehouse environments for experiments. Depending on function, several basic warehouse layout types exist (Tompkins et al., 2003). We select U- and I-layout types warehouses, which are very common in practice (Tompkins et al., 2003; Van der Meer, 2000). In U-layout warehouses, storage is a main function. I-layouts are used when transshipment is an important function and it is possible for trucks to arrive at different sides of the warehouse. In both layouts, loads needing transportation are generated at receiving, labeling and storage areas. Three load flows (from receiving to the storage areas, from the storage areas to labeling and from labeling to shipping) are kept identical in the simulation experiments. In the U-layout, locations with transportation requests are more concentrated than in the I-layout (see Figure 6.1). In the latter layout, the receiving area is located further from the other areas. Since loads generation patterns are identical, it may occur that a load pick-up position is located quite far from the vehicles' positions, which may negatively impact the performance of rules like *NVF* (called the remote-area effect in De Koster et al., 2004). The travel distance matrices for both layouts are given below:

Table 6.1 The distance matrices for the U- and I- layouts

U layout										**I layout**					
Location		*0*	*1*	*2*	*3*	*4*	*5*			*0*	*1*	*2*	*3*	*4*	*5*
Depot	*0*	0	10	20	10	10	20		*0*	0	10	6	4	5	10
Receiving	*1*	10	0	20	10	10	10		*1*	10	0	16	14	15	20
Storage 1	*2*	20	20	0	10	10	10		*2*	6	16	0	10	11	16
Storage 2	*3*	10	10	10	0	10	20		*3*	4	14	10	0	9	14
Labeling	*4*	10	10	10	10	0	10		*4*	5	15	11	9	0	5
Shipping	*5*	20	10	10	20	10	0		*5*	10	20	16	14	5	0

6.5 Solution approaches

6.5.1 The static scheduling problem

In the section 6.2, we formulated the static (or offline) scheduling problem in a VBIT system as an m-TSPTW. In principle, we can use general-purpose optimization packages such as CPLEX to solve the m-TSPTW. However, such software can only solve small instances of the m-TSPTW, which makes them unusable for practical problems. We used CPLEX 7.1 to solve some instances of our problems (2 vehicles, 12 loads). Typically, CPLEX needed more than 30 minutes, and sometimes even a few hours to solve many instances. This is unacceptable in the real-time scheduling. In this section, we describe several heuristics which will be used later to cope with realistic m-TSPTW. Some of them have been introduced originally for the TSP and VRP, but they are useful for our research as well. We also propose a column-generation and a combined heuristic (a combination of existing heuristics designed to suit our problems) where we define the *cost* of a vehicle tour as the average load waiting time of the loads served in this tour.

- **Insertion heuristic**

The insertion heuristic (Van der Meer, 2000; Laporte et al., 2000) is frequently used for real-time dynamic scheduling problems. The main advantages of the insertion algorithm are its simplicity and speed of calculation.

The pseudo code of the insertion algorithm (Insertion) is given as follow:

- *Step 0*: Initialize all vehicle routes at the depot node $\{0\}$, let the set S contain all (job-) nodes arranged in increasing order of the load (job) release times ($S \neq \varnothing$), set all tours' costs to zero.
- *Step 1*: Remove the first node from the set S and insert it into a specific tour with least cost, respecting the time-window constraints (5) - (8). By doing this, we expand vehicle tours gradually.
- *Step 2*: Repeat step 1 until $S = \varnothing$, compute total cost, stop.

- **Combined heuristic**

This heuristic starts with an initial solution created by the insertion heuristic and applies several improvement algorithms sequentially to improve the solution. Three improvement algorithms are used in this chapter and are *Re-insertion*, *Exchange* and *Relocation* (Kindervater and Savelsbergh, 1992; Laporte et al., 2000). We only apply these improvement algorithms and not other more complicated ones, since for the dynamic scheduling approach in internal transport, it does not pay off to take many loads (jobs) into account at once. At each step, we schedule up to about four loads for each vehicle, so other more complicated and time-consuming improvement heuristics such as 3-opt, will not be very useful. Among the three improvement algorithms, *Re-insertion* belongs to the class of route improvement heuristics and the two others belong to the class of assignment improvement heuristics. Figure 6.2 illustrates the three improvement heuristics.

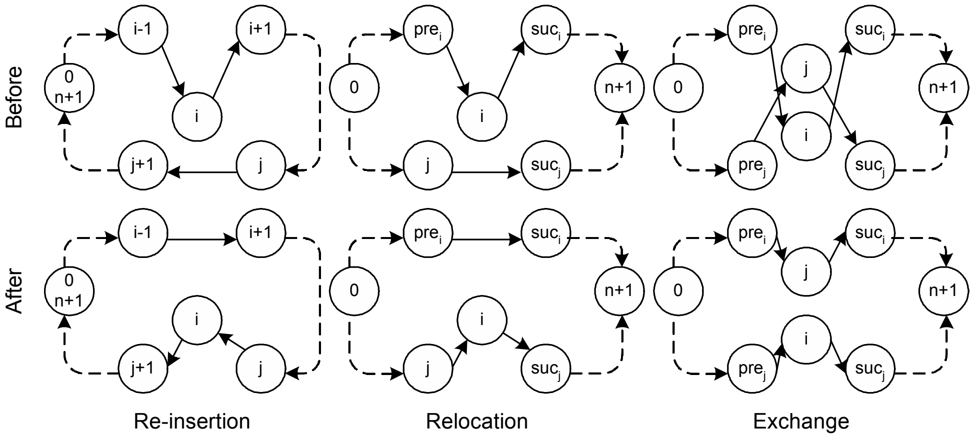


Figure 6.2 Improvement heuristic illustrations (Kindervater and Savelsbergh, 1992)

Re-insertion: The *Re-insertion* (or forward *Or-exchange*) algorithm works as follow:

- *Step 0*: set *it* (iteration index) to 1 (0 is the depot node).
- *Step 1*: remove the node at position *it* and search for the best insert position while respecting the constraints from node *it* + 1 to the end of the route.
- *Step 2*: if a cost reduction is found, then insert this node into the best insertion position, otherwise increase *it* by 1.
- *Step 3*: if node *it* is the last node in the route, stop. Otherwise go to *Step 1*.

Relocation:

- *Step 0*: set it_1 to 1 (node index for route 1), set *previous total cost* to total cost of route 1 and route 2.
- *Step 1*: find the best insert position of node it_1 in route 2.
- *Step 2*: if a cost reduction is found (*total cost of two new routes* < *previous total cost*), insert node it_1 of route 1 at the best insertion position in route 2.
- *Step 3*: increase it_1 by 1, re-compute total route cost, set *previous total cost* to the new total route cost.
- *Step 4*: if all nodes in route 1 have been investigated, stop. Otherwise go to *Step 1*.

Exchange:

- *Step 0*: set it_1 to 1 (node index for route 1), set *previous total cost* to total cost of route 1 and route 2.
- *Step 1*: find the best exchange position of node it_1 and a node in route 2.

- *Step 2*: if a cost reduction is found (*total cost of two new routes < previous total cost*), exchange node it_1 of route 1 with the best exchange node in route 2.
- *Step 3*: increase it_1 by 1, re-compute total route cost, set *previous total cost* to the new total route cost.
- *Step 4*: if all nodes in route 1 have been investigated, stop. Otherwise go to *Step 1*.

General framework for the combined heuristic

We propose a combined heuristic combining insertion and the improvement algorithms into one combined heuristic. The general-framework for the combined heuristic is given below:

- *Step 0*: create initial (vehicle) routes using the *Insertion* algorithm,
- *Step 1*: applying *Re-insertion* algorithm for initial routes,
- *Step 2*: applying *Exchange* algorithm for every pair of routes of the previous step,
- *Step 3*: applying *Relocation* algorithm for every pair of routes of the previous step,
- *Step 4*: applying *Re-insertion* algorithm again for all routes of the previous step.
  STOP.

It is clear that we should improve individual vehicle route at step 1 and step 4. We select the sequence *Exchange* (step 2) -> *Relocation* (step 3), since this sequence gives us a better solution (on average) than the reversed sequence.

Complexity of the combined heuristic

According to Van der Meer (2000), the complexity of the *Insertion* algorithm is $O(n^2)$ (n is the number of loads). Kindervater and Savelsbergh (1992) show that the complexity of the three improvement algorithms are $O(m_{\max}^2)$ ($m_{\max} \leq n$ is the maximum number of loads served by any vehicle route), which is $O(n^2)$ when $m_{\max} = n$. In the above framework, we apply *Re-insertion* for all routes, so the complexity of the re-insertion algorithm is $O(k\, m_{\max}^2)$ (k is the number of vehicles). Two other improvement algorithms are applied for all pair of routes. The number of route pairs equals $k(k-1)/2$, so the complexity of each assignment improvement algorithm applying for all pairs of routes is $O(k^2 m_{\max}^2)$. In conclusion, the overall complexity of the combined algorithm is $max\{O(n^2), O(k^2 m_{\max}^2)\}$. However in our scheduling problems, every vehicle serves about the same number of loads ($m_{\max} \cong n/k$), so the worst-case running time of the combined heuristic can be expected to be $O(n^2)$ as well.

- **Column generation (heuristic)**

The number of columns for our *m*-TSPTW can be huge ($O(k \times n!)$), hence it is impossible to enumerate all columns in a reasonable time. Thus, we used the column generation approach to generate only 'good' columns. The column-generation approach has been used by many authors for solving the PDPTW (Dumas et al., 1991; Savelsbergh and Sol, 1998). Savelsbergh and Sol (1998) use the column generation approach successfully to schedule vehicles in real-time for a Dutch parcel courier and concluded that it is a very promising approach. In this study, we apply this approach to solve the *m*-TSPTW. In order to apply the column-generation heuristic we re-formulate the *m*-TSPTW as a set-partitioning problem. This heuristic includes two steps: (1) generating columns for the master problem and (2) obtaining an integer solution.

Generating columns for the restricted master problem

The master problem (*set-partitioning problem*)

$$Minimize \sum_{k \in K} \sum_{r \in S_k} c_r^k z_r^k \tag{6.11}$$

subject to

$$\sum_{k \in K} \sum_{r \in S_k} \delta_{ir}^k z_r^k = 1 \quad \forall i \in N \tag{6.12}$$

$$\sum_{r \in S_k} z_r^k = 1 \quad \forall k \in K \tag{6.13}$$

$$z_r^k = 0 \text{ or } 1 \quad \forall k \in K, \forall r \in S_k \tag{6.14}$$

where: K: set of vehicles; S_k : set of routes for vehicle k; $z_r^k = 1$ if route $r \in S_k$ is selected, 0 otherwise; $\delta_{ir}^k = 1$ if job i is served on route $r \in S_k$, 0 otherwise; c_r^k : cost of route r served by vehicle k. A vehicle route starts at the depot (or at the vehicle's drop-off location in the dynamic case) visiting some nodes (each node exactly once) within their time-windows and finishes at the end depot.

The set-partitioning model selects routes covering all nodes, each node exactly once, with minimal cost. The linear relaxation of this problem (binary constraint set (6.14) is replaced by $z_r^k \geq 0$) is called the *restricted master problem* (*RMP*). The optimal solution of the restricted master problem is a lower bound on the objective value of the integer master problem. To get an initial feasible solution for the restricted master problem, we introduce artificial variables $y_i \geq 0$ ($i \in N$) and modify the restricted master problem as follows (Savelsbergh and Sol, 1998):

$$\textit{Minimize} \sum_{k \in K} \sum_{r \in S_k} c_r^k z_r^k + \sum_{i \in N} p y_i \tag{6.15}$$

in which $p > \max_{k \in K, r \in S_k} c_r^k$ is a high penalty cost

subject to

$$\sum_{k \in K} \sum_{r \in S_k} \delta_{ir}^k z_r^k + y_i = 1 \quad \forall i \in N \tag{6.16}$$

$$\sum_{r \in S_k} z_r^k \leq 1 \qquad \forall k \in K \tag{6.17}$$

$$z_r^k \geq 0 \qquad \forall k \in K, \forall r \in S_k \tag{6.18}$$

$$y_i \geq 0 \qquad \forall i \in N \tag{6.19}$$

An obvious feasible solution is $y_i = 1$ for all $i \in N$ and all other variables are zero. We call this formulation the *modified restricted-master problem* (*RMP'*).

The pricing problem (*shortest-path problem with time-windows*)

Suppose that the restricted master problem has a feasible solution z. Let u_i ($i \in N$) be dual variables corresponding to the constraint set (6.16), and v_k ($k \in K$) be dual variables corresponding to the constraint set (6.17). According to the linear programming duality (Ahuja et al., 1993), z is optimal for the restricted master problem if and only if for all $k \in K$ and $r \in S_k$ the reduced cost d_r^k is nonnegative, i.e. $d_r^k = c_r^k - \sum_{i \in N} \delta_{ir}^k u_i - v_k \geq 0$ for all $k \in K$ and $r \in S_k$.

The pricing problem is $\min \left\{ c_r^k - \sum_{i \in N} \delta_{ir}^k u_i - v_k \mid k \in K, r \in S_k \right\}$, in which the cost of route $r \in S_k$ is $c_r^k = \sum_{i \in N} (D_{ir} - e_i) \delta_{ir}^k$ (D_{ir}: the service time of node i in the route $r \in S_k$). The vehicle travel distance is not present in the route cost function, however it is reflected in the service time at nodes (D_{ir}). Therefore this problem is a *shortest-path problem with time-windows* (SPPTW). If the solution value of the pricing problem (z) is non-negative, then z is an optimal solution to the restricted master problem and we are done. If min $d_r^k < 0$, we then use an interactive scheme (column-generation) to generate a set of good columns for the integer master problem. We also get a good lower bound for the integer master problem. We have solved the SPPTW using the *generalized permanent labeling* (*GPL*) algorithm (Desrochers and Soumis, 1988) with *bucket* implementation (Dernado and Fox, 1979).

In many VBIT systems, there are only one-sided time-windows at pick-up locations and no time-windows are required at delivery locations. In that case, we add artificial time-windows for nodes, since the *GPL* algorithm needs two-sided time-windows to perform. Adding too long time-windows dramatically slows down the *GPL* algorithm. In contrast to this, too short time-windows may cut off the optimal solution. Generally, the *GPL* algorithm works best for cases where time-windows at nodes are tight. In cases where very wide time-windows exist at pick-up locations, the running-time of the *GPL* algorithm and therefore the column-generation algorithm may increase dramatically along with the problem size.

Column-generation scheme

- *Step 0*: solve the modified restricted master problem by the simplex algorithm (CPLEX),
- *Step 1*: get dual variables (u_i and v_k),
- *Step 2*: solve the pricing problem using the *GPL* algorithm. If the pricing problem's objective value ≥ 0, STOP. Otherwise, add the newly generated column into the (modified) restricted master problem and go to Step 0.

Obtaining an integer solution

The algorithm in the previous column-generation step provides a set of columns for the restricted master problem, which is now used to calculate an integer solution. We find that when a limited number of loads (about four loads per vehicle) is considered for scheduling, we obtain a very good solution by solving the integer master problem with this set of columns. We may then improve the solution using improvement algorithms. In our implementation, we replaced the set of set-partitioning constraints (6.16) by a set of set-covering constraints ($\sum_{k\in K}\sum_{r\in S_k}\delta_{ir}^k z_r^k + y_i \geq 1$ (6.20)), since we found in the experiments that using a set-covering formulation leads to better overall solutions. The formulation (6.15) - (6.19) with (6.20) replacing (6.16) is called the *modified' restricted-master problem* (*RMP"*).

Framework for column-generation heuristic

The framework of the column-generation heuristic is given in Figure 6.3.

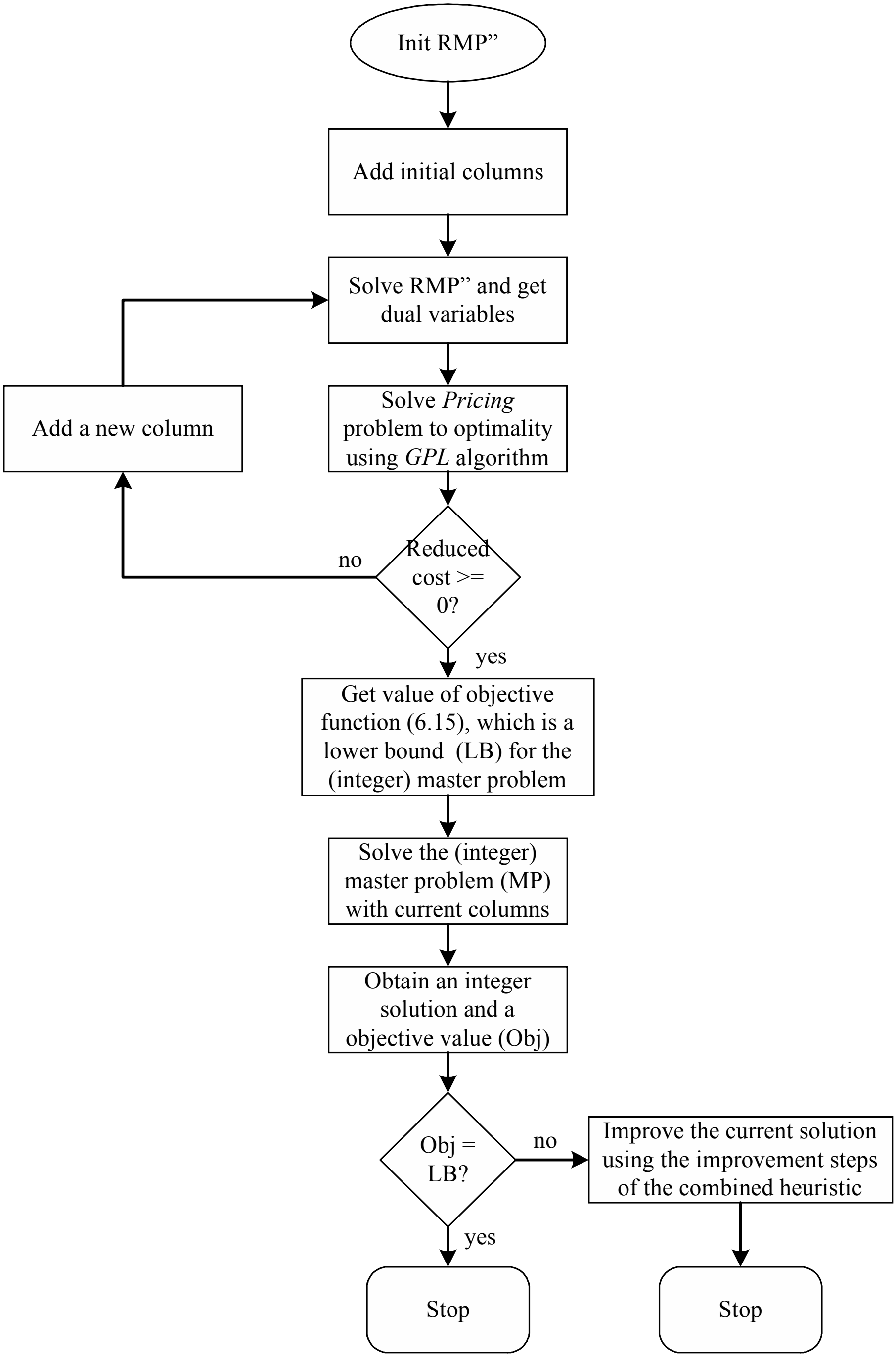


Figure 6.3 The framework of the column-generation heuristic

6.5.2 Computational results for the static case

Experimental environments (U and I layouts) and parameters are described in section 6.4. The three heuristics were coded in C++. For solving the set-covering problem, we use CPLEX 7.1 from ILOG. All experiments ran on a Toshiba Satellite Pro 2100 notebook (CPU: Mobile Intel Pentium 2GHz, 256MB ram). Input data has been generated using ten different seeds (for random numbers) corresponding to ten runs.

Table 6.2 Computational results (total waiting times) for the static case (U-layout)

U layout			*Run*										*performance*		
IA	*Dist*	*Alg*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*avg*	*gap%*	*RT(s)*
2 vehicles, 12 loads															
8	*Uni*	*ins*	34	73	137	103	95	176	59	98	41	173	**98.9**	**13.7**	<0.1
		com	34	73	128	103	95	164	51	63	41	173	**92.5**	**7.7**	0.1
		col	34	69	117	83	95	156	51	59	30	163	**85.7**	**0.4**	1.5
		LB	34	69	117	83	92	156	51	59	30	163	**85.4**		
	Exp	*ins*	61	68	169	162	119	202	110	149	101	183	**132.4**	**20.6**	<0.1
		com	61	68	153	102	113	182	78	72	101	183	**111.3**	**5.5**	0.1
		col	61	68	153	102	98	175	78	64	89	173	**106.1**	**0.9**	1.2
		LB	59	68	146	102	98	175	78	64	89	173	**105.2**		
6 vehicles, 36 loads															
3	*Uni*	*ins*	311	79	157	295	155	145	107	275	245	161	**193.0**	**37.3**	<0.1
		com	220	51	120	268	97	130	86	219	205	128	**152.4**	**20.6**	0.2
		col	213	40	96	265	92	130	54	142	162	112	**130.6**	**7.3**	45
		LB	204	40	93	215	68	128	54	141	160	108	**121.1**		
	Exp	*ins*	420	29	103	199	189	138	315	163	523	327	**240.6**	**33.9**	<0.1
		com	350	18	84	154	92	117	236	127	405	301	**188.4**	**15.6**	0.2
		col	350	18	84	110	68	115	191	102	381	248	**166.7**	**4.6**	35
		LB	326	18	84	106	62	114	187	101	353	239	**159.0**		

IA, Dist: load inter-arrival time mean value (time units) and distribution; *Uni, Exp*: uniform, exponential distributions; *Alg*: algorithm; *ins, com, col*: insertion, combined and column generation heuristics; *LB*: lower bound originated from the column-generation algorithm; *avg*: average of total waiting time (time units); *gap%*: gap with lower bound; *RT*: running time (CPU time - seconds).

Table 6.3 Computational results (total waiting times) for the static case (I-layout)

I layout			*Run*										*performance*		
IA	*Dist*	*Alg*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*avg*	*gap%*	*RT(s)*
2 vehicles, 12 loads															
8	*Uni*	*ins*	46	116	169	93	59	208	47	206	58	189	**119.1**	**23.8**	<0.1
		com	46	115	135	93	56	194	47	142	24	125	**97.7**	**7.2**	0.1
		col	46	90	128	71	56	194	47	142	10	125	**90.9**	**0.2**	1.3
		LB	46	90	126	71	56	194	47	142	10	125	**90.7**		
	Exp	*ins*	67	131	203	90	86	242	110	183	50	183	**134.5**	**17.2**	<0.1
		com	67	112	180	86	72	223	110	181	33	157	**122.1**	**8.8**	0.1
		col	67	112	138	84	72	213	110	153	21	157	**112.7**	**1.2**	1.6
		LB	67	112	134	84	72	213	110	145	21	156	**111.4**		
6 vehicles, 36 loads															
3	*Uni*	*ins*	342	69	183	335	190	167	175	315	301	205	**228.2**	**44.2**	<0.1
		com	255	43	145	227	134	120	133	298	186	131	**167.2**	**23.8**	0.2
		col	261	33	73	219	98	82	77	273	176	110	**140.2**	**9.1**	56
		LB	243	31	71	208	86	75	77	215	167	101	**127.4**		
	Exp	*ins*	489	24	122	190	181	99	311	228	381	374	**239.9**	**32.4**	<0.1
		com	421	16	80	159	96	66	233	167	320	278	**183.6**	**11.6**	0.2
		col	418	16	66	135	57	62	206	166	312	278	**171.6**	**5.5**	49
		LB	407	15	58	108	56	62	201	140	306	270	**162.2**		

Table 6.2 and Table 6.3 show that the combined heuristic gains significant improvements in comparison with the insertion heuristic without increasing running-times significantly. The column-generation heuristic obtains better results overall (obtaining optimal solutions in many cases when 2 vehicles are used). However, when the number of vehicles increases to 15 or more, this heuristic will take a considerable amount of time (half an hour or more depending on the problem) to run and may not satisfy real-time scheduling requirements.

6.5.3 The real-time scheduling problem

- **Dynamic scheduling using rolling horizons**

In VBIT systems, we may know information about load arrivals during a time period T in advance. This information may be not hundred percent reliable. Based on this information we propose two rolling-horizon strategies including rolling by time and rolling by the number of loads. When a vehicle starts to serve a load, it has to finish its jobs. Cancellation of jobs is not allowed.

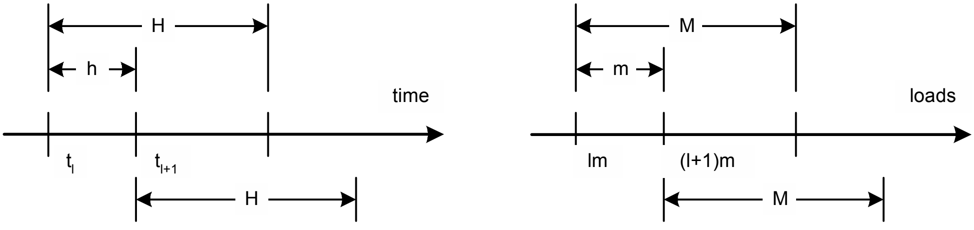


Figure 6.4 Rolling horizon illustration (by time - left and by the number of loads - right)

Rolling by time horizon *(see Figure 6.4)*

For this rolling horizon policy (Psaraftis, 1988), we schedule all (known) loads during a time period H ($0 < H \leq T$) using the proposed heuristics (section 6.5.1). Depending on load arrival rates and load inter-arrival distributions during the operating period, the number of scheduled loads can differ significantly for the time horizon H. However, vehicles only follow the resulting schedule during a time period $h = aH$ ($a < 1$, normally 0.4 – 0.6). After the time period h the system invokes the scheduling algorithm again to schedule all known loads (excluding those in transport and for which vehicles are on their way for pick-up) in the period $[h, h + H]$. The process stops when all loads have been transported.

Updating the problem formulation

For the offline (or static) problem, we assume that all vehicles start at the facility's central parking location (depot). However, during the execution of the algorithm, a vehicle may start at any load's drop-off location. Therefore, we need to modify the formulation (6.1) - (6.10) to reflect this by replacing (6.4) and (6.7) by the following constraints:

$$\sum_{j \in N} x_{0_k j}^k = 1 \qquad \forall k \in K \tag{6.21}$$

$$D_{0_k}^k + t_{0_k j} - D_j \leq B\left(1 - x_{0_k j}^k\right) \qquad \forall j \in N, \forall k \in K \tag{6.22}$$

The set N now contains loads which have not been served during the period $t_l + h$ and loads that have release times satisfying: $t_l + H < e_j \leq t_{l+1} + H$. A vehicle k becomes available at its last drop-off location (0_k) and at time $D_{0_k}^k$, which is the maximum of $t_l + h$ and the drop-off time of the last load served by vehicle k in the previous schedule.

***Rolling by the number of loads** (see Figure 6.4)*

As described in the time horizon policy, the number of scheduled loads at each step can differ significantly. When too many loads are taken into account, the running time of the scheduling algorithms may increase significantly and may not catch up with real-time events. A solution is to reduce the length of the time horizon. However, this may lead to insufficient loads available for scheduling, which limits the quality of the algorithm. Therefore, we propose a second rolling horizon policy - rolling by the number of loads. Suppose that during time period T, we know at least L loads in advance. This policy works as follows:

- Schedule M loads which are known in advance ($0 < M \leq L$) using the proposed heuristics,
- Re-schedule vehicles after the m^{th} load ($m = \lceil a*M \rceil$, $a < 1$) has been picked up by solving the scheduling problem again for the next following M loads,
- Repeat this process until all loads have been transported. STOP.

With this policy, we can always monitor the running time of the scheduling algorithm and keep it at an acceptable level.

Updating the problem formulation

For this type of the rolling horizon, we update the original formulation similar to the rolling by time approach. However, the set N now contains loads which have not been served in the current schedule execution ($M - m$ loads) and the next m loads.

Combined rolling horizon

Practically, we may combine the two rolling horizon policies into a combined one. When the number of loads known in advance is sufficient ($L \geq M$), we apply the rolling by the number of loads method, otherwise the time rolling horizon is used.

- **Dynamic scheduling using assignment algorithm**

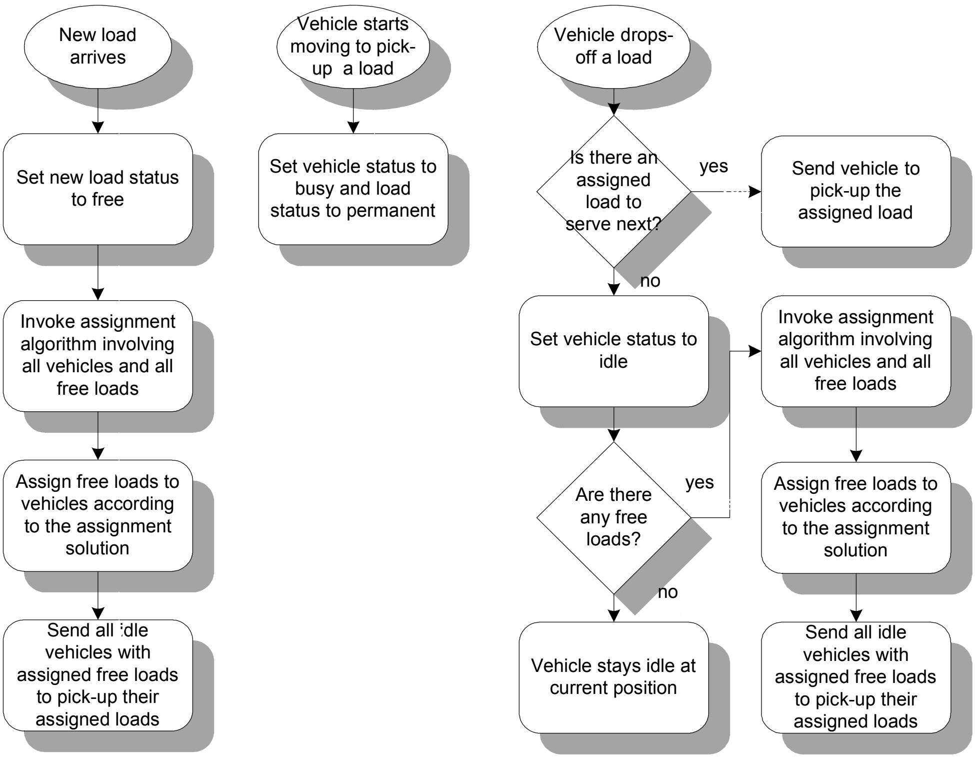
Dynamic assignment scheduling (DAS)

An intuitive scheduling approach is assigning loads to all vehicles, using an assignment algorithm. Fleischmann et al. (2004) use this approach to dynamically solve the full-truckload dispatching problem of a courier service. The main objectives in Fleischmann et al. (2004) include minimizing the order delay and the vehicle empty travel time. These are not relevant in our case, as we focus on minimizing the average load waiting time, so we adopt new cost functions in our implementation. We use the assignment algorithm of Jonker and Volgenant (1987) to solve the assignment problem. Dummy loads and dummy vehicles (as in Fleischmann et al., 2004) are introduced to balance the number of loads and vehicles for the assignment algorithm. We distinguish four types of involved costs as follows:

- The cost of assigning a real vehicle to a real load (f_{main}) equals $C_{empt} \times Travtime$ plus $C_{wait} \times (Lwaittime)^{\alpha}$, in which *Travtime* is the vehicle travel time from its available location (current location for an idle vehicle and the vehicle's current load drop-off location for a busy vehicle) to a load release location and *Lwaittime* is the estimated waiting time of corresponding load.
- The cost of assigning a real vehicle to a dummy load is the unattractiveness cost of a location (vehicle waits at it current location) is $C_{loc} \times 1$,
- The cost of assigning a dummy vehicle to a real load (load waits and remains unassigned at its release location) ($f_{urgency}$) equals $C_{urg}/(\textit{load release time} + \textit{time window size} - \textit{current time})^{\beta}$ if (*load release time* + *time window size*) > (*current time*) and equals ∞ otherwise,
- The cost of assigning a dummy vehicle to a dummy load (irrelevant cost) is 0.

The values of the cost coefficients in our implementation are $C_{empt} = 10$, $C_{wait} = 2$, $C_{loc} = 5 \times 10^3$, $C_{urg} = 2 \times 10^7$, $\alpha = 2$, $\beta = 1$ or 2 (for I- and U-layout respectively – section 6.6). Several of the cost coefficients are taken from Fleischmann et al. (2004) (C_{loc}, C_{urg}, α). Other good cost coefficients are obtained from experiments. In our problem, we have only one-sided time-windows for loads and the cost function f_{main} is in favor of loads with smaller waiting times. This may lead to a very high value of the maximum load waiting time, so we introduce an artificial time-window for loads to guarantee an acceptable value of the maximum load waiting time. The general operating framework for the scheduling approach using the dynamic algorithm is illustrated in Figure 6.5 (adapted from Fleischmann et al., 2004).

Since in our systems, loads have only release times (no time windows for delivery times), to limit the maximum load waiting time resulting of *DAS*, we need introduce an artificial time fence (or time window size T_W). A T_W equaling about the value of the maximum load waiting time when *NVF* is used appears to perform quite well.



Free load: a load already arrived but not assigned to any vehicle or the assigned vehicle is still busy serving another load. A *busy vehicle* will be available at its current load drop-off location at drop-off time.

Figure 6.5 The general framework for the dynamic assignment algorithm

Look-ahead dynamic assignment algorithm (LAS)

Obviously, the assignment algorithm works best for the case where we can assign about one load to each vehicle, but normally, with the implementation of Figure 6.5, we do not have enough loads to assign to all vehicles. In addition, we may know some information about future load arrivals, which we could use to improve *DAS*. Ichoua et al. (2000) and De Koster et al. (2004) also use this idea in their studies. Therefore, we introduce a look-ahead dynamic assignment algorithm (*LAS*). *LAS* schedules vehicles using the same approach as *DAS*, however besides free loads the assignment algorithm also takes into account loads which are known to arrive during a look-ahead period T_L. A good length for T_L is the period during which about K (the number of vehicles) loads are known to arrive ($T_L = K\times\tau$, τ is the load inter-arrival time). We can consider *LAS* a special case of the rolling by time policy in which H equals $K\times\tau$ and h equals min\{time that a new load arrives, time until the first vehicle drops-off its load\} from current time.

▪ **Vehicle dispatching rules**

For the VBIT systems in this chapter, it is assumed that vehicles can park at their drop-off locations. This assumption makes the dispatching rules using vehicle reassignment not relevant here. The simplicity of the guide-path layouts in this chapter also makes multi-attribute dispatching rules less attractive. In a primarily testing, we found that there are no significant difference between the performance of *NVF* and a multi-attribute dispatching rule which dispatches vehicles based on the vehicle empty travel distance and the load waiting time. In the U-layout, no difference is observed. However, in the I-layout a multi-attribute dispatching rule might perform better than *NVF* under very high vehicle utilization since the I-layout is more dispersive than the U-layout. In this chapter, we select two dispatching rules (*NVF* and *NVF* with look-ahead or *NVF_LA*) for experiments.

Nearest-Vehicle-First (NVF)

See section 3.2.3.

Nearest-Vehicle-First with look-ahead (NVF_LA)

NVF_LA operates similarly to *NVF*. The difference is that the load gives a signal Δ time units prior to its actual release time. The time between the actual release, and the virtual release Δ time units before, can be interpreted as a look-ahead time. This gives the vehicle the opportunity to travel to the load before the load is physically ready for transport. The vehicle can therefore arrive just before or after the load is ready for transport, thereby reducing load-waiting times.

6.6 Experiment setups

▪ **Performance criteria and influenced factors**

Performance criteria

In VBIT systems, the crucial performance criterion is minimizing the average load waiting time (*Avg_wait*). In this study, we consider minimizing the average load waiting time as the main performance criterion. We also use other performance indicators: the maximum load waiting time (*Max_wait*), vehicle utilization (*Util%*), and the maximum number of loads in queues (*Max_inQ*) as side criteria. To rank the performance of the dispatching rules and the scheduling approaches we use the Tukey test (Hsu, 1996) with 95% confidence interval (95% CI).

Influenced factors

Considering the main performance criterion (the average load waiting time), we can see several factors which might affect it directly. They are guide-path layout, vehicle utilization, load arrival rate, load arrival-rate variance, vehicle control policy, number of vehicles and amount of load pre-arrival information. Three factors including vehicle utilization, load arrival rate (or the load inter-arrival time), and number of vehicles are inter-related. Practically, reducing the number of vehicle leads to a similar effect as increasing the load arrival rate and increasing the load arrival rate also means increasing the vehicle utilization.

The vehicle utilization can be considered as an indirect factor. In this study we also consider it as a supplement performance criterion. For VBIT systems, we expect that the performance gaps between the dispatching rules and the scheduling approaches become larger under low vehicle utilization. In the experiments, we also use two load inter-arrival distributions (exponential and uniform) with different variances to analyze the sensitivity of the vehicle control policies. The experimental layout is expected to have an important impact on the system performance, particularly on the dispatching rules.

- **Experimental parameters**

We use the two layouts (Figure 6.1) to test the performance of different vehicle control methods. In our experiments, we assume that vehicles can park at their pick-up/ drop-off locations and vehicle loading and unloading times are negligible. Since varying the load inter-arrival time and the number of vehicles has similar effects, we vary only the load inter-arrival time.

All important experimental factors and their values are described below:

- Experimental layouts (*Lay*): 2 (U and I-layouts),
- Number of vehicles (*K*): 6 (typical number in warehouses),
- Load inter-arrival distributions (*Dist*): 2 (uniform, exponential),
- Load inter-arrival time (mean value τ): 2 levels ($\tau = 3, 3.6$). This implies a variance of τ^2 for exponential and $\tau^2/3$ for uniform distributions,
- Scheduling algorithms and dispatching rules:
    - Two dispatching rules (*Disp. Rules*): *NVF* and *NVF_LA*. The best length of the look-ahead period (T_L) is taken. This value is estimated using simulation experiments (section 6.7).
    - Two assignment algorithms (*Assign. Algs*): *DAS* and *LAS* ($T_L = K \times \tau$),

  - Three heuristics including insertion (*Insertion*), combined (*Com-Heur*) and column-generation (*Column-Heur*) heuristics under two rolling horizon policies: by time (*T*) and by the number of loads (*M*),
- Rolling horizon parameters:
  - Rolling by the number of loads: $M = K \times 4$, $m = K \times 2$ ($M = 24$, $m = 12$).
  - Rolling by time: $H = K \times 4 \times \tau$, $h = K \times 2 \times \tau$ ($H = 72$ and 86.4, $h = 36$ and 43.2 corresponding to $\tau = 3$ and 3.6).
- For each combination of experimental factors, we use ten replications ($N_R = 10$). The lengths of the planning horizons (simulation periods) are 900 ($\tau = 3$) and 1080 ($\tau = 3.6$) time units.

For all dynamic scheduling strategies, we set a time-window of 50 time units (e.g. seconds) for all job-nodes.

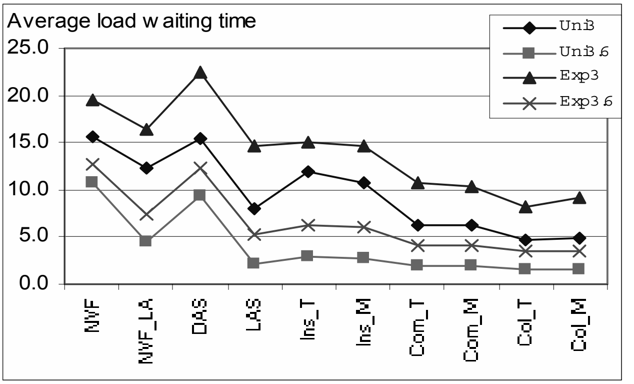
6.7 Performance evaluation

- **Performance evaluation for the U-layout**

Table 6.4 Experimental results for the U-layout

Dist	τ	perfor. measure	Disp. Rules		Scheduling algorithms									
					Assign. Algs		*Insertion*		*Com_Heur*		*Column_Heur*	
			NVF	*NVF_LA*	*DAS*	*LAS*	*T*	*M*	*T*	*M*	*T*	*M*
Uni	3	***Avg_wait***	**15.70**	**12.25**	**15.36**	**8.09**	**11.96**	**10.66**	**6.33**	**6.16**	**4.74**	**4.91**
		Max_wait	49.30	52.70	38.50	30.60	45.90	45.80	39.70	39.40	41.00	41.80
		Max_inQ	7	8	6	8	6	6	5	5	4	4
		Util%	95.99	92.19	92.65	98.68	94.74	94.86	93.08	93.09	91.23	92.04
	3.6	***Avg_wait***	**10.74**	**4.42**	**9.42**	**2.14**	**2.96**	**2.79**	**1.99**	**1.89**	**1.49**	**1.48**
		Max_wait	32.60	31.50	25.70	17.30	21.20	20.90	27.80	20.00	24.50	23.30
		Max_inQ	5	5	5	7	4	3	3	3	3	3
		Util%	86.65	86.21	79.22	96.83	84.25	84.25	82.63	82.83	81.91	81.93
Exp	3	***Avg_wait***	**19.51**	**16.48**	**22.52**	**14.58**	**14.98**	**14.55**	**10.70**	**10.37**	**8.17**	**9.14**
		Max_wait	68.20	68.70	53.00	43.70	47.40	48.70	46.90	46.30	47.40	46.90
		Max_inQ	9	10	8	9	7	8	7	6	6	6
		Util%	93.81	91.24	91.69	97.33	93.27	93.28	91.57	91.52	86.83	90.84
	3.6	***Avg_wait***	**12.72**	**7.34**	**12.39**	**5.20**	**6.18**	**5.97**	**4.17**	**4.12**	**3.46**	**3.57**
		Max_wait	43.50	46.80	35.90	27.40	37.50	36.40	37.60	34.80	35.90	37.80
		Max_inQ	6	7	6	8	5	5	4	4	4	4
		Util%	83.18	82.55	78.75	94.44	82.84	83.03	81.26	80.92	78.70	80.32

Dist: the load generation distribution; *τ*: the load inter-arrival time; *Avg_wait*, *Max_wait*: the average and max load waiting time (time units); *Max_inQ:* the maximum number of loads in queues; *Util%*: the vehicle utilization; *NVF*, *NVF_LA*: the nearest-vehicle-first rules without and with look-ahead; *DAS*, *LAS*: the dynamic assignment algorithms without and with look-ahead; Insertion: the (dynamic) insertion algorithm; *Com_Heur*, *Column_Heur*: the (dynamic) combined and column-generation heuristics; *T*, *M*: the two rolling schemes (by time and by the number of loads).



Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) distribution and the load inter-arrival time is 3 (time units); *Ins*, *Com*, *Col_T*, *_M*: insertion, combined and column-generation heuristics under two rolling horizon policies.

Figure 6.6 Average waiting times – U-layout

Table 6.4 and Figure 6.6 indicate clearly that the average load waiting time reduces dramatically when we schedule vehicles using dynamic scheduling strategies. Best results are obtained when we apply the column-generation heuristic to solve static instances of real-time scheduling problems. The largest improvement of the average waiting time of *Column_Heur* over *NVF* is 86.2% (uniform distribution, τ = 3.6). The reduction of the average waiting time when we compare the performance of *NVF* and *NVF_LA* is 59.6%. In order to rank the different scheduling policies, we used the Tukey test with 95% confidence intervals. For all inter-arrival distributions tested, results can be found in Table 6.5. Since the two rolling horizon policies (by *T* and *M*) perform quite similarly (see Table 6.4 and also by Tukey test), we use only one entry to represent both of them in Table 6.5. For example, the entry "column generation" represents both rolling horizon policies (by *T* and *M*) using column-generation heuristic. The *NVF_LA* and *LAS* perform significantly better than *NVF* and *DAS* (Table 6.5). Dynamic scheduling strategies are also favorable to dispatching rules considering the maximum load waiting time.

Table 6.5 Ranking of different scheduling policies for the U-layout (Tukey test with 95 % confidence interval)

Dist	*Uniform*		*Exponential*	
τ	3	3.6	3	3.6
Column generation	1	1	1	1
Combined heuristic	1	2	1	2
LAS	3	2	3	2
Insertion	3	4	3	2
NVF_ LA	3	5	3	2
DAS	6	6	6	6
NVF	6	7	6	6

Scheduling approaches are ranked from high to low according to the average load waiting time. The average load waiting times of scheduling approaches in the same number block are not significantly different.

DAS performs a little better than *NVF* in general, but not significantly (Table 6.4, Table 6.5). *LAS* performs very well and is as good as *Com_Heur* (Table 6.5), particularly in the high load inter-arrival time cases (τ = 3.6). The combined scheduling heuristic performs much better than insertion, for which the largest improvement is 42.2%. We also notice that the column-generation heuristic performs better than the combined heuristic. However for large real problems, the running-time of the column-generation heuristic grows rapidly, so it is only suitable for small and medium-sized cases (less than 15 vehicles).

- **Performance evaluation for the I-layout**

Table 6.6 Experimental results for the I-layout

			Disp. Rules		**Scheduling algorithms**							
					Assign. Algs		*Insertion*		*Com_Heur*		*Column_Heur*	
Dist	**τ**	**perfor. measure**	*NVF*	*NVF_LA*	*DAS*	*LAS*	*T*	*M*	*T*	*M*	*T*	*M*
Uni	3	***Avg_wait***	**40.10**	**36.11**	**27.71**	**17.73**	**19.20**	**18.47**	**12.80**	**12.45**	**10.57**	**10.40**
		Max_wait	204.2	189.4	59.30	49.10	49.30	49.30	49.20	49.50	49.20	49.50
		Max_inQ	19	18	9	10	8	8	7	7	6	7
		Util%	96.74	96.43	94.89	97.94	95.98	96.05	95.69	95.65	93.73	95.02
	3.6	***Avg_wait***	**14.73**	**10.64**	**13.27**	**3.29**	**4.87**	**4.91**	**3.04**	**3.04**	**2.46**	**2.46**
		Max_wait	66.50	70.10	34.00	22.00	32.50	28.80	35.00	33.00	34.40	33.40
		Max_inQ	7	8	6	7	4	4	4	4	4	4
		Util%	89.05	87.98	82.61	95.24	86.40	86.23	84.95	85.25	84.45	84.56
Exp	3	***Avg_wait***	**44.19**	**42.25**	**34.76**	**25.42**	**19.45**	**18.73**	**14.14**	**14.40**	**13.81**	**12.66**
		Max_wait	214.0	213.5	74.40	66.10	50.00	49.80	49.50	49.70	51.70	48.60
		Max_inQ	21	20	10	11	8	8	7	8	7	7
		Util%	95.89	95.68	93.48	96.89	94.31	93.93	94.04	94.06	93.57	93.51
	3.6	***Avg_wait***	**18.73**	**16.05**	**17.02**	**7.33**	**8.74**	**8.57**	**6.07**	**6.08**	**5.50**	**5.55**
		Max_wait	93.90	91.10	48.70	38.40	43.50	43.50	44.50	44.50	43.30	43.40
		Max_inQ	10	10	7	8	6	6	5	5	5	5
		Util%	87.03	86.73	81.74	93.40	85.32	84.73	83.50	83.81	83.10	83.29

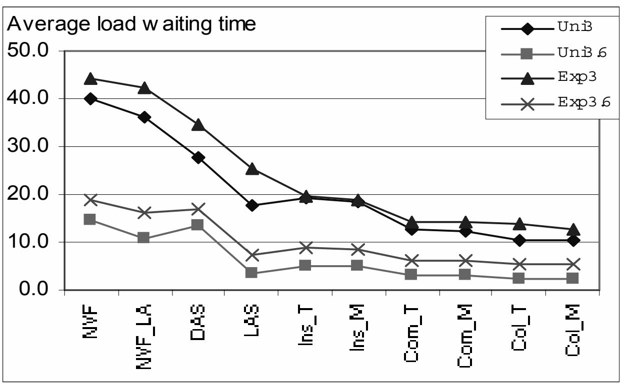


Figure 6.7 Average waiting times – I-layout

We observe similar effects of using different dynamic scheduling and dispatching strategies for the I-layout. However, in this layout improvements are smaller than in the U-layout. The largest improvement of the average waiting time of *Column_Heur* over *NVF* is

83.3% (uniform distribution, $\tau = 3.6$). In this layout, the performance of the *NVF_LA* rule is less impressive than in the U-layout. The improvement of the average waiting time of *NVF_LA* compared with that of *NVF* is 27.7%. We also observe that for both layouts bigger improvements are obtained for lower load arrival rates (or larger load inter-arrival time), corresponding to lower vehicle utilization rates. This is similar to the findings of Yang et al. (2004) and fairly obvious, since in highly utilized systems there is little gain in prematurely sending vehicles to pick-up locations as there are often loads in the neighborhood to be picked up. Table 6.6 and Table 6.7 show that *DAS* performs better than *NVF* but not significantly. *LAS,* instead, performs more impressively (in the top group in half of the cases).

Table 6.7 Ranking of different scheduling policies for the I-layout (Tukey test with 95 % confidence interval)

Dist	*Uniform*		*Exponential*	
τ	3	3.6	3	3.6
Column generation	1	1	1	1
Combined heuristic	1	1	2	1
LAS	3	1	2	1
Insertion	3	1	2	1
NVF_ LA	5	5	5	5
DAS	6	6	5	5
NVF	6	6	5	5

Table 6.7 clearly indicates that the three dynamic scheduling heuristics and *LAS* perform significantly better than dispatching rules and the simple dynamic assignment algorithm (*DAS*). Pre-arrival information has still a positive influence on the performance of *NVF*.

For both layouts, scheduling and dispatching strategies perform better when the load inter-arrival distribution is uniform instead of exponential. This can be explained by the fact that the variance of the uniform distribution used in our experiments is three-times lower than the variance of the exponential distribution. Another observation is that the average load waiting time in the U-layout is smaller than the corresponding value in the I-layout. Considering other performance criteria (max load waiting time, max number of loads in queues, vehicle utilization), we also find that scheduling algorithms perform better than vehicle dispatching rules. Comparing *LAS* with other scheduling approaches using rolling horizons, *LAS* performs worse in terms of the maximum number of loads in queues. *LAS* also results in a very high value of vehicle utilization. This is explained by *LAS* being a more local policy, implying that vehicles may travel longer distances (similar to Kim and Bae's observation) than in the other scheduling approaches.

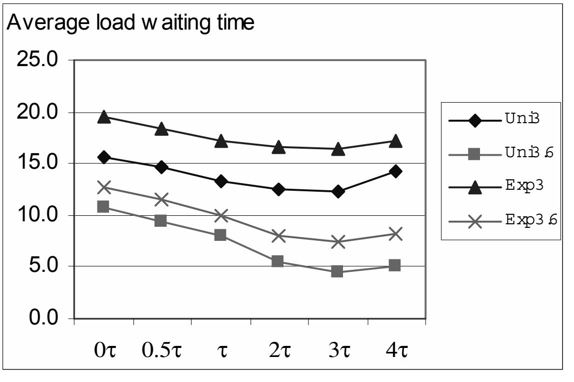
In the next two sections, we carry out experiments with look-ahead periods and rolling horizon policies to see how this may affect the performance of dispatching rules and scheduling algorithms.

- **Influences of look-ahead periods and of rolling horizon lengths**

In this section, we have experimented with six vehicles, two distributions (uniform and exponential), two load inter-arrival levels and have observe the influence of both the look-ahead period length and the rolling horizon period length for two layouts and two load inter-arrival levels. We express the length of the look-ahead period in terms of the load inter-arrival time.

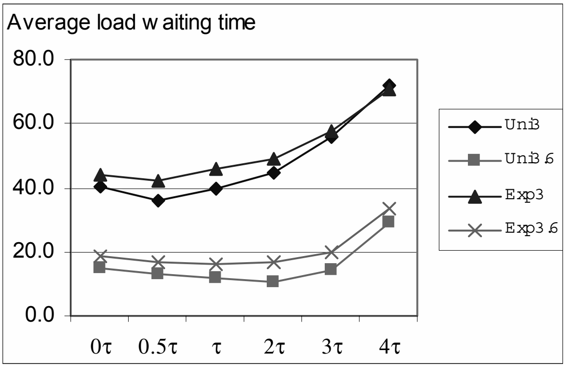
Influences of the look-ahead period

Influences of the look-ahead period on *NVF_LA*



Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) and the load inter-arrival time is 3 (time units).

Figure 6.8 Impacts of the look-head period the *NVF_LA* rule for the U-layout



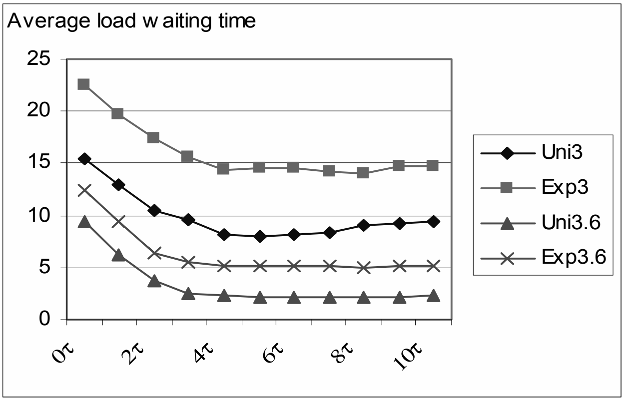
Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) and the load inter-arrival time is 3 (time units).

Figure 6.9 Impacts of the look-head period on the *NVF_LA* rule for the I-layout

In Figure 6.8, the best value for the look-ahead period for *NVF_LA* is similar for both distributions and load inter-arrival levels. It is about three times the average load inter-arrival time. Figure 6.9 shows different effects. For the larger load inter-arrival time (3.6), the best value for the look-ahead period is about two times the load inter-arrival time (7.2). For smaller load inter-arrival time (3), the best look-ahead time equals half the load inter-arrival time (1.5).

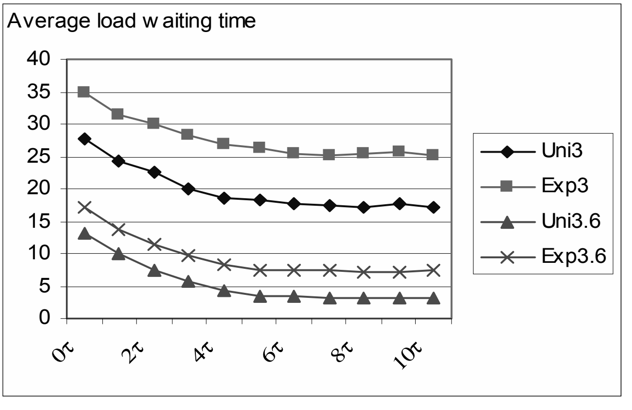
Different behaviors of the look-ahead period in two layouts under different operating conditions do not permit us to recommend a specific value for the best length of the look-ahead period. Good values can only be obtained by experiments. However, the experiments indicate that it can be fairly small.

Influences of the look-ahead period on *LAS*



Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) and the load inter-arrival time is 3 (time units).

Figure 6.10 Impacts of the look-head period on *LAS* for the U-layout



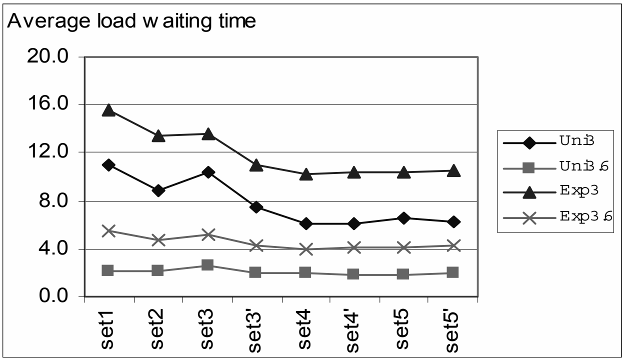
Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) and the load inter-arrival time is 3 (time units).

Figure 6.11 Impacts of the look-head period on *LAS* for the I-layout

In the experimental setup, we selected $K\times\tau$ ($6\times\tau$) to be the length of the look-ahead period (T_L) for *LAS*. This value is reasonable since, for the assignment algorithm, it is logic to assign one load for each vehicle. Figure 6.10 and Figure 6.11 show that this is a good value for the look-ahead period. However, the length of the look-ahead period has a slightly different impact on the two layouts. Look-ahead too far in advance cannot reduce the average load waiting time resulting by *LAS*. The reduction of the average waiting time resulting by *LAS* seems to be saturated beyond $K\times\tau$ time units, which is due to the fact that the assignment algorithm can only plan one load ahead for each vehicle.

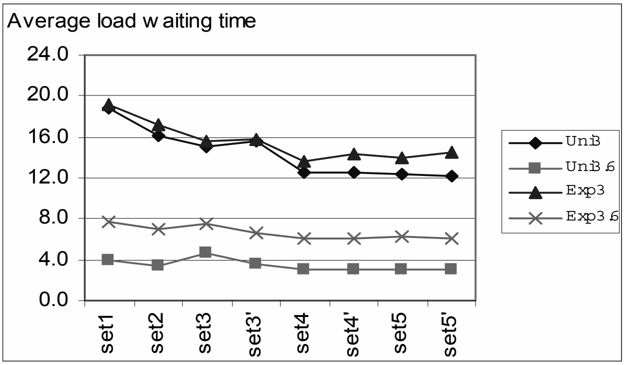
Influences of rolling horizon lengths

The performance of the two rolling horizon policies is very similar for both layouts under various conditions, so we have experimented with only the rolling by the number of loads policy. There are six sets of rolling horizon parameters (M, m): set 1 (3, 2); set 2 (6, 5); set 3 (12, 6); set 3' (12, 8); set 4 (24, 10); set 4' (24, 12); set 5 (36, 12); set 5' (36, 18). Since all three dynamic scheduling heuristics behave similarly, we selected only the combined heuristic for experiments.



Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) and the load inter-arrival time is 3 (time units).

Figure 6.12 Impacts of rolling horizon policies using the combined heuristic on the U-layout



Uni3 (*Exp3*): the load inter-arrival distribution is uniform (exponential) and the load inter-arrival time is 3 (time units).

Figure 6.13 Impacts of rolling horizon policies using the combined heuristic on the I-layout

Using set 3 (12, 6), we schedule about 2 loads in advance for each vehicle and let each vehicle execute about one load before re-scheduling. Figure 6.12 and Figure 6.13 show that significant improvements start when we schedule vehicles using set 4 (24, 10). With this set, we schedule about four loads per vehicle and let each vehicle execute about 1.7 loads. Taking more information into account (set 5, set 5'), we cannot obtain a further significant improvement. However, we are not interested in scheduling vehicles too far in advance. In general, we gain significant improvements when we schedule more than about three loads per vehicle and each vehicle should transport about two loads before re-scheduling.

- **Value of information and discussion**

Table 6.8 and Table 6.9 show that when load pre-arrival information is available, the rolling horizon approach always leads to better results in comparison to *NVF_LA* and *LAS*. Among *NVF_LA* and *LAS*, *LAS* is the better control policy. The rolling horizon approach and *LAS* result in a more substantial waiting time reduction (compared with *NVF_LA*) in the I-layout than in the U-layout.

Table 6.8 The average load waiting times resulting of the three approaches with a look-ahead period (U-layout)

LA_per	*Util%*	*0τ*	*0.5τ*	*τ*	*2τ*	*3τ*	*4τ*	*6τ*	*8τ*	*10τ*	*24τ*	*36τ*
NVF_LA												
Uni3	96.0	15.7	14.6	13.3	12.5	12.3	14.3					
Exp3	93.8	19.5	18.4	17.1	16.5	16.5	17.2			**		
Uni3.6	86.7	10.7	9.4	7.9	5.5	4.4	5.1					
Exp3.6	83.2	12.7	11.5	10.0	8.0	7.3	8.2					
LAS												
Uni3	92.7	15.4		12.9	10.5	9.5	8.2	8.1	9.1	9.3		
Exp3	91.7	22.5		19.7	17.3	15.6	14.4	14.6	14.1	14.8	***	
Uni3.6	79.2	9.4		6.3	3.7	2.5	2.3	2.1	2.2	2.2		
Exp3.6	78.8	12.4		9.3	6.5	5.4	5.1	5.2	5.0	5.2		
Rolling horizon (rolling by time using the combined heuristic)												
Uni3	93.1	*		10.4		9.3		8.0		7.4	6.2	6.5
Exp3	91.6	*		15.2		13.9		12.7		11.4	10.4	10.4
Uni3.6	82.6	*		2.2		2.0		2.0		2.0	1.9	1.9
Exp3.6	81.3	*		5.3		4.7		4.7		4.4	4.1	4.1

LA_per: length of the look-ahead time; τ: load inter-arrival time; *Uni3*: uniform load inter-arrival time with $\tau = 3$; (*): the rolling horizon approaches do not work if pre-arrival information of loads is not available; (**), (***): no further improvements found.

Table 6.9 The average load waiting times resulting of the three approaches with a look-ahead period (I-layout)

LA_per	*Util%*	0τ	0.5τ	τ	2τ	3τ	4τ	6τ	8τ	10τ	24τ	36τ
NVF_LA												
Uni3	96.7	40.1	36.1	39.6	44.7	55.6	72.1					
Exp3	95.9	44.2	42.3	45.9	49.0	57.4	70.8			**		
Uni3.6	89.1	14.7	12.9	11.7	10.6	14.2	29.3					
Exp3.6	87.0	18.7	16.7	16.1	17.0	19.7	33.8					
LAS												
Uni3	94.9	27.7		24.3	22.7	20.0	18.6	17.7	17.2	17.2		
Exp3	93.5	34.8		31.5	29.9	28.3	27.0	25.4	25.6	25.2	***	
Uni3.6	82.6	13.3		9.9	7.3	5.7	4.3	3.3	3.2	3.2		
Exp3.6	81.7	17.0		13.8	11.4	9.7	8.2	7.3	7.2	7.4		
Rolling horizon (rolling by time using the combined heuristic)												
Uni3	95.7	*		17.7		16.3		16.0		14.4	12.4	12.4
Exp3	94.0	*		18.3		17.1		17.0		15.6	14.4	14.0
Uni3.6	85.0	*		4.1		3.5		3.5		3.3	3.0	3.0
Exp3.6	83.5	*		7.4		7.1		7.0		6.5	6.1	6.2

In both layouts (Figure 6.1), the receiving area is the main load generation source and the shipping area is the main sink. At the shipping area, vehicles become available after dropping off their loads. It can be considered as a main vehicle source. Since vehicles at the receiving area only pick-up loads, this area needs vehicle dispatches from other areas. In the I-layout, the receiving area is the area farthest from the shipping area. Therefore, this area may sometimes have difficulty qualifying for a vehicle dispatch from the shipping area (particularly when using *NVF* or *NVF_LA*). This may lead to a vehicle shortage at the receiving area and explains the poor performance of the vehicle dispatching rules in the I-layout. De Koster et al. (2004) call this the 'remote-area' phenomenon, in which *NVF*-based rules perform poorly. Multi-attribute dispatching rules might overcome this. The influence of main factors found in this section are summarized in Table 6.10.

Table 6.10 The influence of main factors on the vehicle control (dispatching and scheduling) policies

Factors	**Impacts**
Load arrival rate ↑ (*Vehicle utilization* ↑)	- The performance gaps between the dispatching rules and the scheduling approaches ↓
Load arrival rate's variance ↑	- All vehicle control policies' performances ↓
Layouts with remote areas	- The performance of *NVF*-based rules reduce significantly
Horizon of pre-arrival information ↑	- The dispatching rules' performances ↑ (until a certain limit depending on the layout and the load arrival flow ($\leq 3\times\tau$ for the cases in this chapter) - LAS performance ↑ (saturated beyond about $K\times\tau$) - The rolling horizon approaches' performances ↑

6.8 Concluding remarks

In this chapter, we study real-time vehicle scheduling in internal transport systems. These systems can be characterized by a high degree of uncertainty, short travel times and, often, high vehicle utilization rates. This implies that dispatching vehicles is the common approach in practice. Literature on external transport has shown that scheduling vehicles may lead to better performance than dispatching them. Applying this to internal transport does not automatically lead to similar results, as the objectives of external transport and the circumstances are often quite different. Furthermore, optimal vehicle scheduling is time-consuming, if not infeasible. This study is one of the first to investigate the potential contribution of scheduling methods for internal transport.

We proceed by proposing three heuristics for the static vehicle scheduling problem (which can be formulated as an *m*-TSPTW). We use Insertion as a straightforward benchmark, *Com-Heur* as an improvement-based heuristic and *Column-Heur* as a column-generation based construction and improvement heuristic. We apply these static-situation heuristics dynamically by using them repeatedly under a rolling horizon (for which we use two variants). These results are compared with one of the best-performing dispatching rules, applied with and without look-ahead information (*NVF* and *NVF-LA*, respectively). We also modify an easy-to-implement assignment method introduced by Fleischmann et al. (2004), *DAS*, or with look-ahead information: *LAS*.

Using simulation, we systematically compare the performance (measured by average waiting time) of these seven methods (two dispatching and five scheduling, of which three are used with two different rolling horizon methods), by varying the following parameters:

- Load arrival rate (2 values, implying different vehicle utilizations)
- Load arrival variance (2 values – corresponding to the two distributions)
- Layout (2 variants, of which one contains a remote area: the I-layout)

Results show that the scheduling approaches perform significantly better than the dispatching rules. Depending on layouts and working conditions, improvements can be about 90%. However, for certain layouts (such as the U-layout where locations with transportation requests are concentrated), when the vehicle utilization is very high (> 95%) and little load pre-arrival information is available the performance gaps between the vehicle scheduling approaches and the dispatching rules are small (see Table 6.4). Table 6.4 indicates that the dispatching rules may also outperform the scheduling approaches. Taking the complexity and computation speed of the scheduling approaches into account, the dispatching rules have their advantages. Experimental results suggest that *NVF* may perform badly in layouts with remote areas (Table 6.10). Table 6.10 indicates that under very high vehicle utilization, using a scheduling approach cannot improve the system performance significantly compared with using a dispatching rule. Table 6.10 also shows that vehicle scheduling methods use load pre-arrival information more efficiently than dispatching rules.

We find that the two rolling horizon methods (rolling by time and by the number of scheduled loads) perform similarly. When sufficient load pre-arrival information is available (about two to four loads per vehicle), the rolling horizon approaches perform significantly better than *LAS* and *NVF_LA*. When we know only about one load per vehicle in advance, *LAS* performs significantly better compared to *NVF_LA*.

Chapter 7

Conclusions and further research

This research studies the operational control problem of vehicle-based internal transport systems. VBIT systems form an integral part of many industrial facilities such as warehouses, distribution centers, production plants, airport and transshipment terminals. These facilities can be very different in nature, but they share one common feature: a VBIT system takes care of internal transports.

The number of industrial vehicles used in practice has increased steadily over the years in comparison with other material handling equipment (MHIA, 2004). Furthermore, the number of new applications using industrial vehicles is growing as well. For example, modern baggage handling systems use destination-coded vehicles to move baggage. Destination-coded vehicles are used to replace conveyor-like equipment (or tilt-tray sorter). An important advantage of industrial vehicles over fixed-position equipment is their flexibility. The increasing number of new applications and vehicles used and also the requirement of increasing the efficiency of VBIT systems in practice motivate this research.

In this chapter, we summarize the main findings of this thesis and give some directions for further research.

7.1 Conclusions

There are many types of VBIT systems; however, in this research we limit our scope to VBIT systems using guided vehicles. We have provided a structured and detailed review on key issues related to design and control of a VBIT system using guided vehicles in chapter 2. This review covers both automated guided and person-guided vehicle systems. In this thesis, we particularly focus on the vehicle dispatching and scheduling problems in VBIT systems. We summarized and discussed these contributions in the next few sections.

- **Applying and ranking dispatching rules for real-world environments (chapters 3 and 4)**

Chapter 3 focuses on testing the performance (mainly measured by the average load waiting time) of several simple dispatching rules for two real-world environments. The first one is a retailer's warehouse for computer products and another one is the warehouse of a glassware production plant. The currently used dispatching rules in the two cases are based on priority lists for locations. Since these currently used dispatching rules are not evaluated in literature, it is interesting to see if the best rules known from literature perform well in these cases. Using simulation, several well-known dispatching rules are compared, including the modified-first-come-first-serve (MODFCFS), shortest-travel-distance-first (STDF), nearest-vehicle-first (NVF) rules and a variation of the NVF rule (NVF with time truncation, or NVFTP). We also study the value of pre-arrival information and possible performance gains of using a look-ahead policy when such pre-arrival information is available. Results show that the distance-based dispatching rules (NVF, STDF) perform significantly better with respect to average load-waiting time than the time-based dispatching rules (such as MODFCFS) and the specific dispatching rules used by the companies, regardless of vehicle utilization rates. The main draw back of the distance-based dispatching rules (NVF, STDF) is that while minimizing the average load waiting time, these rules also tend to maximize the maximum load waiting time. This is especially true when vehicle utilization is high or in the presence of remote areas. The NVFTP rule is designed to overcome this shortcoming. This rule performs well regarding both criteria (minimizing the average and maximum load waiting times). In general, using realistic (short term) pre-arrival information can significantly reduce the average load waiting time for any dispatching rule used. Based on two real-world environments evaluated in chapter 3, the general ranking of the common dispatching rules based on the average load waiting time appears to be: (1) NVF/NVFTP, (2) STDF, (3) case specific rules and (4) MODFCFS. This ranking should hold for VBITSs where queue space is not a restriction.

In chapter 4, we introduce some more intelligent dispatching rules (or complex dispatching rules) for the two cases introduced in chapter 3. The main goal of introducing the complex dispatching rules is to search for more efficient and robust dispatching rules for VBIT environments. A "*robust*" dispatching rule maintains its good performance in different situations (i.e. different load arrival rates and variances, different number of vehicles, etc.). The complex rules include: (1) the multi-attribute rule (Multi-att) which dispatches vehicle based on a multi-attribute dispatching function; (2) the nearest-vehicle-first with vehicle reassignment (NVF_R); (3) the nearest-vehicle-first with vehicle reassignment and time truncation (NVFTP_R); (4) the nearest-vehicle-first with vehicle reassignment and cancellation (NVF_RC); and (5) the combined dispatching rule which integrates multi-attribute dispatching and vehicle reassignment (Combi).

Impacts of reassigning moving vehicles

Results of chapter 4 indicate that reassigning moving vehicles has a positive impact on reducing the average load waiting time. This impact is more significant under low vehicle utilizations. The main reason is that in case of low utilization, vehicles spend a lot of time to go to parking locations when some loads might be available for picking-up. Reassigning moving-to-park vehicles to pick-up new loads can save vehicles' unnecessary movements and therefore reduces the average load waiting time. The rule implementing both vehicle reassignment and cancellation (NVF_RC) can reduce the average load-waiting time even further. However, reassignment dispatching rules lead to a high value of the maximum load waiting time. In addition, to implement the reassignment rule particularly NVF_RC, the control system needs to monitor vehicles' positions continuously. This is difficult or may even be impossible in many VBIT systems, particularly, in person-guided vehicle systems. Using estimated values for vehicles' travel distances or predicting vehicles' locations can be a solution.

Impacts of considering a multi-attribute dispatching function to control vehicles

This type of rule cannot reduce the average load waiting time in the two cases studied. However, they can reduce the maximum load waiting time significantly. This helps to avoid the situation where some loads might be forgotten and makes these types of rules robust. These rules might be more useful in other environments, like manufacturing, where queue spaces are very limited. In that case, queue spaces should be a decision attribute.

The combined impact

Combining reassigning moving vehicles and multi-attribute dispatching leads to an efficient and robust dispatching rule (*Combi*) in most cases, which is proven by the results of chapter 4.

Recommendations for choosing appropriate dispatching rules

The results from the chapters 3 and 4 suggest that when the maximum load-waiting time is not important, NVF_RC is the best solution. Otherwise, the *Combi* rule (see chapter 4) is the best one. However, in order to implement these dispatching rules, an advanced control system is required for monitoring all vehicles and loads continuously. For most real-world systems, simple rules such as NVF, NVFTP can be implemented without much difficulty. These rules also provide very good performance for the average load waiting time. The NVF_RC rule is not preferred in person-guided vehicle systems, since changing the vehicle destination frequently is not desirable for the vehicle's driver. The load pre-arrival information has a very positive effect to reduce the average load waiting time. Therefore, when such information is available, it should be used to improve the performance of dispatching rules.

- **Implementation of dispatching rules for a different type of environment (chapter 5)**

In the chapters 3 and 4, we have examined several dispatching rules for two types of warehouse environments. However, it is important to investigate other types of VBIT environments and to see how dispatching rules perform in such environments. In chapter 5, we study a totally different type of VBIT system (denoted by L-VBIT system or L-VBITS) which uses a large number of (automated) guided vehicles to transport loads. We have found that MSTDF (an adaptation of the STDF rule) does not perform well here. In contrary, dispatching rules such as EC and EC_A (see chapter 5) which are specially designed for this type of environment work well.

We observed that the EC rule performs well, but may perform badly under unbalanced working conditions. The EC_A rule, in general, performs worse than EC. Aiming at deriving good and robust dispatching rules for L-VBITSs, we introduce two new dispatching rules namely Mulit-Att and Multi-Mod which are both multi-attribute rules. These rules perform well for L-VBITSs under various working conditions. They can be good dispatching solutions to apply in practice.

Recommendations for selection of parameters for multi-attribute dispatching rules

The results from chapters 4 and 5 also show that the multi-attribute dispatching rule is, in general, robust to different working conditions and performing well in many types of environment. The main problems are selecting right parameters (or attributes) and assigning them appropriate weights. Thus, we suggest a selection framework to choose parameters (or attributes) for multi-attribute dispatching rules.

There are several factors that we should look at. Firstly, we need to start with the system performance criteria (or objectives). By looking at the performance criteria, we can derive directly some important factors. Secondly, we need to inspect other factors which are not implied directly by the performance criteria, but may have significant impact on the performance criteria. Table 7.1 summarizes several popular performance criteria and factors which we should look at.

Table 7.1 A recommendation for parameter selection

	Performance criteria	**Direct factors**	**Indirect factors**
Minimizing	Average load waiting time	Load waiting time	Vehicle empty travel time
	Max load waiting time	Load waiting time	Vehicle empty travel time
	Travel distance (time)	Vehicle empty travel time	
	Max. no. of loads in queues	No. of loads in queues	Load waiting time
Balancing workload		No. of loads in queues	Vehicle requirement
Preventing queues' overflow		Remaining spaces in queues	Vehicle requirement
Meeting time-windows		Time left from the current time until the load' latest pick-up time	Load waiting time

In Table 7.1, we listed only the most important direct and indirect factors which may have some impacts on the corresponding criteria. Since, in this thesis, we consider only single-load vehicles, the vehicle loaded travel time is normally fixed. Thus, the vehicle empty travel time is the only factor to look at when we look at the vehicle total travel time. The vehicle requirement can be the number of vehicles needed at a workstation. In chapter 5, this factor is defined as the vehicle net-stock. After identifying the key parameters for multi-attribute dispatching rules, we need to specify suitable weights for them. This can be done by inspecting the importance of factors and by experiments.

- **Dynamic scheduling approaches for VBIT systems (chapter 6)**

Since literature on external transport has shown that scheduling vehicles leads to better performance than dispatching them, in chapter 6, we study real-time scheduling approaches for vehicle-based internal transport systems. In this thesis, we formulate a VBIT scheduling problem as an *m*-TSPTW. To solve the static version of this problem, we propose three heuristics: Insertion, Combined (insertion and improvement heuristics) and Column-generation heuristics. In order to solve the real-time scheduling problem, we propose two rolling horizon approaches: dynamic rolling by time and rolling by the number of scheduled loads. We also propose another good dynamic scheduling strategy: the look-ahead dynamic assignment algorithm (*LAS*). These dynamic scheduling strategies are compared with two of the best vehicle dispatching rules (NVF and NVF_LA) from previous chapters for two experimental layouts (U- and I-layouts). Table 7.2 summarizes the advantages and disadvantages of different vehicle control (dispatching and scheduling) policies.

Table 7.2 Advantages and disadvantages of control policies

Vehicle control policies	Advantages	Disadvantages
Dispatching rules	- Simple - Easy to implement - Quick	- Limited performance - Sensitive to layout (depending on dispatching rules)
Dynamic Assignment Algorithm	- Perform better than dispatching rules - Quick - Less sensitive to guide-path layout	- Finding appropriate values for the parameters' coefficients (layout dependent) - More complicated in comparison with dispatching rules - Requiring advanced control systems
Dynamic rolling horizon	- Perform significantly better than both dispatching rules and the dynamic assignment algorithm - Less sensitive to guide-path layout	- Complicated - Requiring load pre-arrival information - Can be slow (may require significant time for computation) - Requiring advanced control systems

Dynamic scheduling approaches vs. dispatching rules

We found that dynamic scheduling strategies consistently outperform vehicle dispatching rules in two experimental environments and under different operating conditions. This observation is similar to that of the full truck-load scheduling problem in external transport environments. Improvements are remarkable when applying combined and column-generation to solve static instances. However, significant improvements are only possible when we know sufficient (about four loads per vehicle) information about future load arrivals.

Rolling horizon approaches vs. Dynamic assignment algorithms

According to results of chapter 6, the rolling horizon approaches outperform the dynamic assignment algorithms (with and without look ahead information), when the system can provide sufficient information about future loads' arrivals. The right amount of information is about two to four loads per vehicle or more. If we know in advance less than two loads per vehicle, the dynamic assignment algorithms are competitive with the rolling horizon approaches.

Dynamic assignment algorithms vs. Dispatching rules

The dynamic assignment algorithms (*DAS*, *LAS*) outperform dispatching rules, particularly when load pre-arrival information is available. The dynamic assignment algorithms are also fast and can therefore be applied in practice. The look-ahead dynamic algorithm (*LAS*) performs significantly better than dispatching rules and the simple dynamic assignment algorithm (*DAS*). The main disadvantage of dynamic assignment algorithms is that their performance depends on a cost function and its parameters. Depending on applications we have to select the right cost functions and tune cost parameters carefully.

Value of information

Results of chapter 6 show that load pre-arrival information plays an important role in improving the system performance. The scheduling (dispatching) method which can make use of more information should lead to a better performance. According to the results of this chapter we can make some recommendations for selecting a scheduling or a dispatching algorithm based on the amount of load pre-arrival information (Table 7.3). These recommendations are purely based on the quality of different vehicle control approaches. If we take the calculation time and the simplicity of control approaches into account as well, dispatching rules have some advantages.

Table 7.3 A recommendation for selection of scheduling strategies

Information	Scheduling strategies
No pre-arrival information available	(1) Dynamic assignment algorithm (DAS), (2) Dispatching rules,
Little pre-arrival information available (about 1 load per vehicle)	(1) Look-ahead DAS (or LAS), (2) Look-ahead dispatching rules,
More pre-arrival information available (more than 1 load per vehicle)	(1) Dynamic scheduling using a rolling horizon approach, (2) Look-ahead DAS (or LAS), (3) Look-ahead dispatching rules.

** A smaller number indicates a higher priority.*

7.2 Further research

The research in this thesis still has some limitations. These limitations suggest some directions for further research.

In literature, we cannot find any dispatching rule which is the best for all environments. This observation is also true in this research. One encouraging point is that the multi-attribute rule can be used for different environments, but we have to find the right parameters and assign the right weights to them. In practice, it is important to have a selection framework for dispatching rules based on the environments' characteristics. To do so, we need to do more experiments. Therefore, characterizing dispatching rules and environments is important.

The best dispatching rule in chapter 3 (NVFTP) requires a threshold for the time truncation. We suggested a method to select this value, but a better method is desirable. Also, better and more systematic methods can be useful to assign coefficients for parameters of the multi-attribute dispatching rules in chapter 4 and 5. Artificial intelligence techniques such as neural networks and fuzzy logic may be useful approaches here. It is also interesting to experiment with dispatching rules in chapter 5 for real-word baggage handling systems or similar systems in practice. For L-VBITSs (chapter 5), intelligent positioning of idle-vehicles may improve the system performance.

Some new VBIT systems use multi-load vehicles for transportation of loads. In practice, at some container terminals, double-load AGVs or multi-load trailers are used. For such systems, we need good dispatching rules for multi-load vehicles. In the case where multi-load vehicles are used, the scheduling problem is not an m-TSPTW anymore. This problem now becomes to pick-up and delivery problem with time-windows (PDPTW). In this case, we have to adapt the solution approaches to suit new situations. The dynamic assignment algorithms do not work for the multi-load vehicle scheduling problem, so adaptations are also required.

In chapter 6, we assume that vehicles can park at their drop off (or pick-up) locations (which is true for many cases). However, if this is not the case, we have to incorporate the vehicle parking problem into solution approaches. This makes the scheduling problem much more complicated. In some cases, the vehicle congestion problem may need to be taken into account as well.

Since the main concern of chapter 6 is deriving efficient approaches to schedule VBIT systems dynamically, we did not focus on getting the best solutions for the offline scheduling problem. Practically, some methods can be used to speed up and improve the quality of the column-generation heuristic. However, obtaining the optimal solution is not necessarily desirable because of the many uncertainties in VBIT environments. Some methods which can be used to speed up the column generation algorithm are: (1) using a better column management scheme by keeping only 'good' columns; (2) generating good columns by fast heuristics; (3) applying a Lagrangean relaxation approach to generate new columns; (4) applying some better heuristics to fix variables. Besides speeding up the column generation heuristic, other heuristics such as genetic algorithm and Tabu search can also be used to solve the offline scheduling problem. Moreover, in chapter 6, we did not explicitly investigate the combined rolling horizon policy. This may provide a better result than applying a single rolling horizon policy in systems with high variation of load arrivals. Another interesting research direction is to evaluate the performance of VBIT systems when the vehicle travel time is not deterministic.

Appendix

List of abbreviations

AGV(s)	= Automated Guided Vehicle(s)
AGVS(s)	= Automated Guided Vehicle System(s)
AMH	= Automated Material Handling
ASC(s)	= Automated Stacking Crane(s)
AS/RS	= Automatic Storage and Retrieval System
B^2D^2	= Bidding-Based Device Dispatching
BHS(s)	= Baggage Handling System(s)
C100FCFS	= Closer than 100 m First-Come-First-Served
Column_Heur	= Column-generation Heuristic
Com_Heur	= Combined Heuristic
DAS	= Dynamic ASsignment algorithm
DCV(s)	= Destination-Coded Vehicle(s)
DD	= Dedicated Dispatching
DPI	= Dispatching with Pre-arrival Information
EC	= Entrance Control
EC_A	= Entrance Control with additional Assignment
EDC	= European Distribution Center
FCFS	= First-Come-First-Served
FEFS	= First-Encountered-First-Served
FLT(s)	= Forklift Truck(s)
FMS(s)	= Flexible Manufacturing System(s)
GV(s)	= Guided Vehicle(s)
IP	= Integer Programming
L-VBITS(s)	= VBIT system(s) using a large number of vehicles
LAS	= Look-ahead dynamic ASsignment algorithm
LLD	= Load-List Dispatching
MH	= Material Handling
MHC	= Material Handling Control
MHS(s)	= Material Handling System(s)
MIP	= Mixed Integer Programming

MODFCFS = MODified First-Come-First-Served
MOD STTF = MODified Shortest-Travel-Time-First
MOQS = Maximum-Outgoing-Queue-Size
MROQS = Minimum-Remaining-Outgoing-Queue-Space
MSTDF = Modified Shortest-Travel-Distance-First
m-TSPTW = Multi Traveling Salesman Problem with Time Windows
Multi-att = Multi-attribute
Multi-Att = Multi-Attribute
Multi-Mod = Modified Multi-attribute
NVF = Nearest-Vehicles-First
NVF_LA = Nearest-Vehicles-First with Look-Ahead
NVF_R = Nearest-Vehicles-First with vehicle Reassignment
NVF_RC = Nearest-Vehicles-First with vehicle Reassignment and Cancellation
NVFTP = Nearest-Vehicles-First with Time Priority
NVFTP_R = Nearest-Vehicles-First with Time Priority and vehicle Reassignment
OMS = Order Management System
PDPTW = Pick-up and Delivery Problem with Time Windows
P (&)/D = Pick-up (&)/ Delivery
QC(s) = Quay Crane(s)
RF = Radio Frequency
SC(s) = Stacking Crane(s)
SFC = Shop Floor Control
SFT = Segmented Flow Topology
STD(T)F = Shortest-Travel-Distance (Time)-First
TSPTW = Traveling Salesman Problem with Time Windows
VAL = Value Added Logistics
VBIT = Vehicle-Based Internal Transport
VBITS(s) = Vehicle-Based Internal Transport System(s)
WLD = Work-list Dispatching
WMS(s) = Warehouse Management System(s)

References

Abdelmaguid, T.F., Nassef, A.O., Kamal, B.A. and Hassan, M.F. 2004. A hybridGA/heuristic approach to siumultaneous scheduling of machines and automated guided vehicles. International Journal of Production Research, 42 (2), 267-281.

Ahuja, K., Magnanti, T.L. and Orlin, J.B. 1993. Network flows: Theory, algorithms, and applications. Prentice Hall, Inc.

Al-Sultan, K.S. and Bozer, Y.A. 1998. Network configuration and machine layout in fixed-path material handling systems, Annals of Operations Research 76 21-54.

Arifin, R. and Egbelu, P.J. 2000. Determination of vehicle requirements in automated guided vehicle systems: a statistical approach, Production Planning & Control 11 258-270.

Asef-Vaziri, A., Laporte, G. and Sriskandarajah, C. 2000. The block layout shortest loop design problem, IIE Transactions 32 727-734.

Ashayeri, J. 1989. Analytical approaches in AGVS flow path design. Proc. 5th Int. Conf. Simulation in Manufacturing, 95-107.

Ashayeri, J. and Gelders, L.F. 1985. Warehouse design optimalization, European Journal of Operational Research 21 285-294.

Ashayeri, J. and Gelders, L.F. 1987. Interactive GPSS-PC program generator for automated material handling systems. The International Journal of Advanced Manufacturing Technology, 2 (4), 63-77.

Ashayeri, J., Gelders, L.F. and Van Looy, P.M. 1985. Micro-computer simulation in design of automated guided vehicle systems. Material Flow, 2 , 37-48.

Askin, R.R. and Standridge, C.R. 1993. Modeling and analysis of manufacturing systems. John Wiley & Sons, Inc., pp. 307-323.

Bartholdi III, J.J. and Platzman, L.K. 1989. Decentralized control of automated guided vehicles on a simple loop, IIE Transactions 21 (1) 76-81.

Beamon, B.M. 1998. Performance, reliability, and performability of material handling systems, International Journal of Production Research 36 (2) 377-393.

Berman, S. and Edan, Y. 2002. Decentralized autonomous AGV system for material handling, International Journal of Production Research 40 (15) 3995-4006.

Bilge, Ü. and Tanchoco, J.M.A. 1997. AGV systems with multi-load carriers: Basic issues and potential benefits, Journal of Manufacturing Systems 16 (3) 159-174.

Bilge, Ü. and Ulusoy, G. 1995. A time window approach to simultaneous scheduling of machines and material handling system in an FMS, Operations Research 43 (6) 1058-1070.

Bozer, Y.A. and Srinivasan, M.M. 1991. Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops, IIE Transactions 23 (1) 72-82.

Bozer, Y.A. and Srinivasan, M.M. 1992. Tandem AGV systems: a partitioning algorithm and performance comparison with conventional AGV systems, European Journal of Operational Research 63 173-191.

Bozer, Y.A. and Yen, C. 1996. Intelligent dispatching rules for trip-based material handling systems, Journal of Manufacturing Systems 15 (4) 226-239.

Brooks Automation, Inc. 2002. AutoMod™: User's guide. Brooks Automation, Inc.

Chang, S. and Egbelu, P.J. 1996. Dynamic relative positioning of AGVs in a loop layout to minimize mean system response time, International Journal of Production Research 34 (6) 1655-1673.

Chen, M.-T., McGinnis, L. and Zhou, C. 1999. Design and operation of single-loop dual-rail inter-bay material handling system, International Journal of Production Research 37 (10) 2217-2237.

Chevalier, P., Pochet, Y. and Talbot, L. 2002. Design of a 2-stations automated guided vehicle system. In: *Quantitative Approaches to Distribution Logistics and Supply Chain Management* (eds A. Klose, M. G. Speranza, and L. N. Van Wassenhove). Springer-Verlag, pp. 309-329.

Co, C.G. and Tanchoco, J.M.A. 1991. A review of research on AGVS vehicle management, Engineering Cost and Production Economics 21 35-42.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y. and Semet, F. 2002. A guide to vehicle routing heuristics, Journal of the Operational Research Society 53 512-522.

De Koster, R. 1995. Intern transportmaterieel. In: *Praktijkboek Magazijnen / Distributiecentra* (ed J.P. Duijker et al.). Kluwer Bedrijfswetenschappen, Deventer, pp. Ch. 3.7, pp. 1-4.

De Koster, R., Le Anh, T. and Van der Meer, J.R. 2004. Testing and classifying vehicle dispatching rules in three real-world settings, Journal of Operations Management 22 (4) 369-386.

De Koster, R. and Neuteboom, A.J. 2001. The logistics of supermarket chains: a comparision of seven chains in The Netherlands. The Netherlands, Elsevier.

De Koster, R. and Van der Meer, J.R. 1998. Centralized versus decentralized control of internal transport. In: *Advances in Distribution Logistics* (eds B. Fleischmann, J. A. E. E. van Nunen, M. G. Speranza, and P. Stähly). Springer Berlin, pp. 403-420.

Dernado, E.V. and Fox, B.L. 1979. Shortest-route methods: reaching, pruning, and buckets, Operations Research 27 (1) 161-186.

Desaulniers, G., Lavigne, J. and Soumis, F. 1998. Multi-depot vehicle scheduling problems with time windows and waiting costs, European Journal of Operational Research 111 479-494.

Desrochers, J., Lenstra, J.K., Savelsbergh, M.W.P. and Soumis, F. 1988. Vehicle routing with time windows: optimization and approximation. In: *Vehicle Routing: Methods and Studies* (ed B.L.Golden and A.A.Assad (Eds.)). Elsevier Science Publishers, pp. 65-84.

Desrochers, M. and Soumis, F. 1988. A generalized permanent labeling algorithm for the shortest path problem with time windows, INFOR 26 (3) 191-212.

Dumas, Y., Desrosiers, J. and Soumis, F. 1991. The pickup and delivery problem with time windows, European Journal of Operational Research 54 7-22.

Ebben, M., 2001. Logistic control in automated transportation networks, Ph. D Thesis, University of Twente.

Egbelu, P.J. 1987. The Use of non-simulation approaches in estimating vehicle requirements in an automated guided vehicle based transport system, Material Flow 4 17-32.

Egbelu, P.J. 1993. Positioning of automated guided vehicles in a loop layout to improve response time, European Journal of Operational Research 71 (1) 32-44.

Egbelu, P.J. and Tanchoco, J.M.A. 1984. Characterization of automated guided vehicle dispatching rules, International Journal of Production Research 22 (3) 359-374.

Egbelu, P.J. and Tanchoco, J.M.A. 1986. Potentials for bi-directional guide-path for automated guided vehicle based systems, International Journal of Production Research 24 (5) 1075-1097.

Evers, J.J.M. and Koppers, S.A.J. 1996. Automatic guided vehicle traffic control at a container terminal, Transportation Research A 30 (1) 21-34.

Fleischmann, B., Gnutzmann, S. and Sandvoß, E. 2004. Dynamic vehicle routing based on online traffic information. Transportation Science, 38 (4), 420-443.

Gademann, N. and Van de Velde, S. 2000. Positioning automated guided vehicles in a loop layout, European Journal of Operational Research 127 565-573.

Gaskins, R.J. and Tanchoco, J.M.A. 1987. Flow path design for automated guided vehicle systems, International Journal of Production Research 25 (5) 667-676.

Gaskins, R.J., Tanchoco, J.M.A. and Taghaboni, F. 1989. Virtual flow paths for free-ranging automated guided vehicle systems, International Journal of Production Research 27 (1) 91-100.

Gendreau, M., Guertin, F., Potvin, J.-Y. and Taillard, E. 1999. Parallel Tabu search for real-time vehicle routing and dispatching, Transportation Science 33 (4) 381-390.

Goetz, W.G. and Egbelu, P.J. 1990. Guide path design and location of load pick-up/drop-off points for an automated guide vehicle system, International Journal of Production Research 28 (5) 927-941.

Handfield, R.B. and Nichols, Jr.E.L. Introduction to supply chain management. 1999. New Jersey, Prentice Hall.

Ho, Y.-C. 2000. A dynamic-zone strategy for vehicle-collision prevention and load balancing in an AGV system with a single-loop guide path, Computers in Industry 42 159-176.

Hsu, J.C. 1996. Multiple comparisons: theory and methods. UK, Chapman & Hall.

Hsieh, S. and Kang, M.-Y. 1998. Developing AGVS Petri net control models from flowpath nets, Journal of Manufacturing Systems 17 (4) 237-250.

Hu, C.-H. and Egbelu, P.J. 2000. A framework for the selection of idle vehicle home locations in an automated guided vehicle system, International Journal of Production Research 38 (3) 543-562.

Huang, C. 1997. Design of material transportation system for tandem automated guided vehicle systems, International Journal of Production Research 35 (4) 943-953.

Hwang, H. and Kim, S.H. 1998. Development of dispatching rules for automated guided vehicle systems, Journal of Manufacturing Systems 17 (2) 137-143.

Ichoua, S., Gendreau, M. and Potvin, J.-Y. 2000. Diversion issues in real-time vehicle dispatching, Transportation Science 34 (4) 426-438.

Jeong, B.H. and Randhawa, S.U. 2001. A multi-attribute dispatching rule for automated guide vehicle systems, International Journal of Production Research 39 (13) 2817-2832.

Johnson, M.E. and Brandeau, M.L. 1993. An analytic model for design of a multivehicle automated guided vehicle system, Management Science 39 (12) 1477-1489.

Johnson, M.E. and Brandeau, M.L. 1994. An analytic model for design and analysis of single-vehicle asynchronous material handling systems, Transportation Science 28 (4) 337-353.

Jonker, R. and Volgenant, T. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems, Computing 38 325-340.

Jünemann, R. and Schmidt, T. 2000. Materialflußsysteme. Springer.

Kasilingam, R.G. 1998. Logistics and transportation: design and planning. Great Britan, The Kluwer Academic Publishers.

Kaspi, M., Kesselman, U. and Tanchoco, J.M.A. 2002. Optimal solution for the flow path design problem of a blanced unidirectional AGV system, International Journal of Production Research 40 (2) 389-401.

Kaspi, M. and Tanchoco, J.M.A. 1990. Optimal flow path design of unidirectional AGV systems, International Journal of Production Research 28 (6) 1023-1030.

Kim, C.W. and Tanchoco, J.M.A. 1993. Operational control of a bidirectional automated guided vehicle system, International Journal of Production Research 31 (9) 2123-2138.

Kim, C.W., Tanchoco, J.M.A. and Koo, P.H. 1999. AGV dispatching based on workload balancing, International Journal of Production Research 37 (17) 4053-4066.

Kim, C.W. and Tanchoco, J.M.A. 1991. Conflict-free shortest-time bidirectional AGV routeing, International Journal of Production Research 29 (12) 2377-2391.

Kim, K.H. 1995. Positioning of automated guided vehicles in a loop layout to minimize the mean vehicle response time, International Journal of Production Economics 39 (3) 201-214.

Kim, K.H. and Bae, J.W. 2004. A look-ahead dispatching method for automated guided vehicles in automated port container terminals. Transportation Science, 38 (2), 224-234.

Kim, K.H. and Kim, J.Y. 1997. Estimating mean response time and positioning idle vehicles of automated guided vehicle systems in loop layout, Computers and Industrial Engineering 33 (3-4) 669-672.

Kim, K.H. and Tanchoco, J.M.A. 1993. Economical design of material flow paths, International Journal of Production Research 31 (6) 1387-1407.

Kindervater, G.A.P. and Savelsbergh, M.W.P. 1992. Local search in physical distribution management, Report EUR-CS-92-05.

King, R.E. and Wilson, C. 1991. A review of automated guided-vehicle sytems design and scheduling, Production Planning & Control 2 (1) 44-51.

Klein, C.M. and Kim, J. 1996. AGV dispatching, International Journal of Production Research 34 (1) 95-110.

Laporte, G., Gendreau, M., Potvin, J.-Y. and Semet, F. 2000. Classical and modern heuristics for the vehicle routing problem, Intl. Trans. in Op. Res. 7 285-300.

Law, A.M. and Kelton, W.D. Simulation modeling & analysis. 2000. McGraw-Hill.

Le-Anh, T. and De Koster, R. 2004a. A review of design and control of automated guided vehicle systems, Technical Report ERS-2004-030-LIS, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam. To appear in the European Journal of Operational Research.

Le-Anh, T. and de Koster, R. 2004b. Online dispatching rules for vehicle-based internal transport systems, Technical Report ERS-2004-031-LIS, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam. To appear in the International Journal of Production Research.

Le Anh, T. and De Koster, R. 2004c. Vehicle on-line dispatching rules in practice. In: *Distribution Logistics: Advanced Solutions to Practical Problems* (ed Bernhard

Fleischmann and Andreas Klose). Springer, pp. 159-184.

Le-Anh, T. and de Koster, R. 2004d. Real-time scheduling approaches for vehicle-based internal transport systems, Technical Report ERS-2004-056-LIS, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam.

Le Anh, T. and De Koster, R. 2004e. Multi-attribute dispatching rules for AGV systems with many vehicles, Technical Report ERS-2004-077-LIS, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam. In revision for the International Journal of Production Economics.

Lee, C. and Ventura, J.A. 2001. Optimal dwell point location of automated guided vehicles to minimize mean response time in a loop layout, International Journal of Production Research 39 (17) 4013-4031.

Lim, J.K., Lim, J.M., Yoshimoto, K., Kim, K.H. and Takahashi, T. 2002. A construction algorithm for designing guide paths of automated guided vehicle systems, International Journal of Production Research 40 (15) 3981-3994.

Lindeijer, D.G., 2003. Controlling automated traffic agents, Ph.D Thesis, Technical University of Delft.

Mahadevan, B. and Narendran, T.T. 1993. Estimation of number of AGVs for an FMS: an analytical model, International Journal of Production Research 31 (7) 1655-1670.

Mahadevan, B. and Narendran, T.T. 1994. A hybrid modelling approach to the design of an AGV-based material handling system, International Journal of Production Research 32 (9) 2015-2030.

Malmborg, C.J., Agee, M.H., Simons, G.R. and Choudry, J.V. 1987. A prototype expert system for industrial truck type selection, Industrial Engineering 19 (3) 58-65.

Mantel, R.J. and Landeweerd, H.R.A. 1995. Design and operational control of an AGV system, International Journal of Production Economics 41 (1-3) 257-266.

Maxwell, W.L. and Muckstadt, J.A. 1982. Design of automated guided vehicle systems, IIE Transactions 14 (2) 114-124.

McHaney, R. 1995. Modelling battery constraints in discrete event automated guided vehicles simulations, International Journal of Production Research 33 3023-3040.

Meersmans, P., 2002. Optimization of container handling systems, Ph.D Thesis, Erasmus University Rotterdam.

Meersmans, P.J.M. and Dekker, R. 2001. Operations research supports container handling, Econometric Institute Report EI 2001-22.

MHIA, 2004a. Glossary, http://www.mhia.org/keywords/

MHIA ,2004b. Material handling equipment manufacturing, http://www.mhia.org/pdfs/mhem_segment_brief.pdf.

Neufville, R. 1994. The baggage system at Denver: prospects and lessons, Journal of Air transport management 1 (4) 229-236.

Powell, W.B. and Carvalho, T.A. 1998. Dynamic control of logistics queueing networks for large-scale fleet management, Transportation Science 32 (2) 90-109.

Powell, W.B., Towns, T.T. and Marar, A. 2000. On the value of optimal solutions for dynamic routing and scheduling in the presence of user noncompliance, Transportation Science 34 (1) 67-85.

Psaraftis, H.N. 1988. Dynamic vehicle routing problems. In: *Vehicle Routing: Methods and Studies* (ed B.L.Golden and A.A.Assad (Eds.)). Elsevier Science Publishers, pp. 233-248.

Qiu, L. and Hsu, W. 2001. A bi-directional path layout for conflict-free routing of AGVs, International Journal of Production Research 39 (10) 2177-2195.

Qiu, L., Hsu, W., Huang, S. and Wang, H. 2002. Scheduling and routing algorithms for AGVs: a survey, International Journal of Production Research 40 (3) 745-760.

Rajotia, S., Shanker, K. and Batra, J.L. 1998b. A semi-dynamic time window constrained routeing strategy in an AGV system, International Journal of Production Research 36 (1) 35-50.

Rego, C. and roucairol, C. 1995. Using Tabu search for solving a dynamic multi-terminal truck dispatching problem, European Journal of Operational Research 83 411-429.

Reveliotis, S.A. 2000. Conflict resolution in AGV systems, IIE Transactions 32 647-659.

Ross, E.A., Mahmoodi, F. and Mosier, C.T. 1996. Tandem configuration automated guided vehicle systems: A comparative study, Decision Sciences 27 (1) 81-102.

Sabuncuoglu, I. 1998. A study of scheduling rules of flexible manufacturing systems: a simulation approach, International Journal of Production Research 36 (2) 527-546.

Sabuncuoglu, I. and Hommertzheim, D.L. 1992. Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system, International Journal of Production Research 30 (5) 1059-1079.

Sabuncuoglu, I. and Kizilisik, O.B. 2003. Reactive scheduling in a dynamic and stochastic FMS environment, International Journal of Production Research 41 (17) 4211-4231.

Savelsbergh, M.W.P. and Sol, M. 1995. The general pickup and delivery problem, Transportation Science 29 (1) 17-29.

Savelsbergh, M. and Sol, M. 1998. DRIVE: Dynamic routing of independent vehicles, Operations Research 46 (4) 474-490.

Shen, W. and Norrie, D.H. 1999. Agent-based systems for intelligent manufacturing: a state-of-the-art survey, Knowledge and Information Systems, an International Journal 1 (2) 129-156.

Shen, Y.-C. and Kobza, J.E. 1998. A dispatching-rule-based algorithm for automated guided vehicle systems design, Production Planning & Control 9 (1) 47-59.

Sinriech, D. and Tanchoco, J.M.A. 1991. Intersection graph method for AGV flow path design, Intenational Journal of Production Research 29 (9) 1725-1732.

Sinriech, D. and Tanchoco, J.M.A. 1992a. An economic model for determining AGV fleet size, International Journal of Production Research 30 (6) 1255-1268.

Sinriech, D. and Tanchoco, J.M.A. 1992b. Impact of empty vehicle flow on performance of single-loop AGV systems, International Journal of Production Research 30 (10) 2237-2252.

Sinriech, D. and Tanchoco, J.M.A. 1993. Solution methods for the mathematical models of single-loop AGV systems, International Journal of Production Research 31 705-725.

Sinriech, D. and Tanchoco, J.M.A. 1995. An introduction to segmented flow approach for discrete material flow systems, International Journal of Production Research 33 (12) 3381-3410.

Sinriech, D. and Tanchoco, J.M.A. 1997. Design procedures and implemetation of the segmented flow topology (SFT) for discrete material flow systems, IIE Transactions 29 323-335.

Srinivasan, M.M., Bozer, Y.A. and Cho, M. 1994. Trip-based material handling systems: Throughput capacity analysis, IIE Transactions 26 (1) 70-89.

Stoline, M.R. 1981. The status of multiple comparisons: simultaneous estimation of all pairwise comparisons in one-way ANOVA designs. The American Statistician, 35 (3), 134-140.

Taghaboni, F. 1997. A value-added approach for automated guided vehicle task assignment, Journal of Manufacturing Systems 16 (1) 24-34.

Taghaboni, F. and Tanchoco, J.M.A. 1995. Comparison of dynamic routing techniques for automated guided vehicle systems, International Journal of Production Research 33 2653-2669.

Talbot, L., 2003. Design and performance analysis of multistation automated guided vehicle systems, Ph.D Thesis Universite Catholique de Louvain.

Tan, K.K. and Tang, K.Z. 2001. Vehicle dispatching system based on Taguchi-tuned fuzzy rules, European Journal of Operational Research 128 545-557.

Tanchoco, J.M.A., Egbelu, P.J. and Taghaboni, F. 1987. Determination of the total number of vehicles in an AGV-based material transport system, Material Flow 4 33-51.

Tanchoco, J.M.A. and Sinriech, D. 1992. OSL - optimal single-loop guide paths for AGVS, International Journal of Production Research 30 (3) 665-681.

Tompkins, J.A., White, J.A., Bozer, Y.A. and Tanchoco, J.M.A. Facilities planning. 2003. John Wiley & Sons, Inc.

Ulusoy, G., Sivrikaya-Serifoglu§, F. and Bilge, Ü. 1997. A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles, Computers Operations Research 24 (4) 335-351.

Van der Heijden, M., Ebben, M., Gademann, N. and van Harten, A. 2002. Scheduling vehicles in automated transportation systems. *OR Spektrum,* **24**, 31-58.

Van der Meer, J.R., 2000. Operational control of internal transport system, Ph.D Thesis Erasmus University Rotterdam.

Van der Meer, J.R. and De Koster, R. 1999. Using multiple load vehicles for internal transport with batch arrivals of loads. In: *New Trends in Distribution Logistics* (ed M.Grazia Speranza and P.Stähly). Springer Berlin, pp. 197-214.

Van der Meer, J.R. and De Koster, R. 2000. Centralized internal transport systems analyzed and classified. In: Progress in Material Handling Research. Ann Arbor Michigan.

Vis, I.F.A. and De Koster, R. 2003. Transshipment of containers at a container terminal: An overview, European Journal of Operational Research 147 1-16.

Vis, I.F.A., de Koster, R., Roodbergen, K.J. and Peeters, L.W.P. 2001. Determination of the number of automated guided vehicles required at a semi-automated container terminal, Journal of Operational Research society 52 (4) 409-417.

Xu, H., Chen, Z.-L., Rajagopal, S. and Arunapuram, S. 2003. Solving a practical pickup and delivery problem, Transportation Science 37 (3) 347-364.

Yamashita, H. 2001. Analysis of dispatching rules of AGV systems with multiple vehicles, IIE Transactions 33 889-895.

Yang, J., Jaillet, P. and Mahmassani, H. 2004. Real-Time multi-vehicle truckload pickup and delivery problems, Transportation Science 38 (2) 135-148.

Yim, D.S. and Linn, R.J. 1993. Push and pull rules for dispatching automated guided vehicles in a flexible manufacturing system, International Journal of Production Research 31 (1) 43-57.

Yu, W. and Egbelu, P.J. 2001. Design of a variable path tandem layout for automated guided vehicle systems, Journal of Manufacturing Systems 20 (5) 305-319.

Zeng, L., Wang, H.P.B. and Jin, S. 1991. Conflict detection of automated guided vehicles: a Petri net approach, International Journal of Production Research 29 (5) 865-879.

Samenvatting (Summary in Dutch)

Material handling omvat het transport en de distributie van producten en grondstoffen binnen de verschillende faciliteiten in de supply chain. Deze betreffen productiefaciliteiten, magazijnen, distributiecentra, luchthavens en overslagterminals. Hoewel deze material handling activiteiten niet direct zichtbaar zijn in eindproducten, hebben ze een groot aandeel in de waarde van een product. Dit aandeel wordt geschat op 15% tot 70% van de totale kosten van een eindproduct (Tompkins et al., 2003). Material handling kan daarom worden gezien als een middel om de totale productiekosten (of servicekosten) te reduceren door een efficiëntere doorstroming, lagere voorraden, hogere operationele efficiency en toegenomen veiligheid. Het moet worden gezien als een middel om een competitief handelsvoordeel te behalen. Material handling kan als volgt worden gedefinieerd:

> "*Material handling* = alle apparatuur en technologie die materialen en producten binnen een faciliteit verplaatsen, opslaan, beheren en beschermen" (MHIA)

Interne voertuigtransportsystemen (VBIT-systemen) zijn waarschijnlijk het meest voorkomende type material handling systeem. In dit onderzoek stellen wij ons tot doel de doelmatigheid van zulke VBIT-systemen in faciliteiten als magazijnen, distributiecentra, overslagterminals en luchthavens te verbeteren. De nadruk ligt daarbij op voertuigaansturingsactiviteiten: *scheduling*, het maken van een gedetailleerd plan voor een bepaalde periode, de planningshorizon, waarin is besloten welke voertuigen welke ladingen in welke volgorde en via welke route moeten vervoeren, met de bijbehorende ophaal- en aflevertijden, en *dispatching*, het nemen van een beslissing omtrent het toewijzen van één bepaald voertuig aan een op een gegeven tijdstip en locatie op te halen lading. Een scheduling-plan kan worden gemaakt met behulp van een gecompliceerd optimalisatiemodel of door het toewijzen van voertuigen aan ladingen volgens enkele intuïtieve dispatching-regels. Terwijl scheduling een planningshorizon vereist, heeft dispatching betrekking op onmiddellijke beslissingen. Scheduling gebeurt dan ook minder frequent dan dispatching. Een dispatching-beslissing wordt gemaakt wanneer (a) een voertuig een lading afgeeft; (b) een voertuig zijn parkeerlocatie bereikt of (c) een nieuwe lading arriveert. Vanuit praktisch oogpunt is het interessant om te weten welke voertuigaansturingsstrategie (scheduling of dispatching) efficiënter en effectiever is en waarom, en welke aanpak onder welke omstandigheden beter is.

Dit onderzoek heeft drie hoofddoelen. Ten eerste evalueren we de prestaties van verschillende dispatching-regels uit de literatuur door ze toe te passen op realistische praktijkproblemen en ze te rangschikken op basis van hun prestaties. Op basis van deze rangschikking stellen we dispatching-regels voor om te implementeren in de praktijk. Ten tweede willen we enkele nieuwe en robuuste dispatching-regels ontwikkelen voor de praktijk en tevens voor andere soorten omgevingen. Tenslotte stellen we enkele dynamische scheduling-strategieën voor voor VBIT-systemen en vergelijken hun prestaties met de beste bekende dispatching-regels. Tevens bevelen we aan wanneer en waar (onder welke omstandigheden) een specifieke voertuigaansturingsstrategie moet worden toegepast.

Hoofdstuk 1 van dit proefschrift introduceert het onderzoeksgebied en de doelstellingen van het proefschrift. Hoofdstuk 2 geeft een overzicht van de literatuur op het gebied van het ontwerp en de aansturing van interne voertuigtransportsystemen. In dit hoofdstuk geven we aan wat er ontbreekt in de literatuur en mogelijke onderzoeksrichtingen. Dit proefschrift vult enkele van deze gebreken in de literatuur in.

Hoofdstuk 3 richt zich op het testen van de prestatie (hoofdzakelijk gemeten door de gemiddelde wachttijd van te vervoeren ladingen) van enkele eenvoudige dispatching-regels in twee praktijksituaties. De eerste is een detailhandelsmagazijn voor computerproducten en de tweede is een magazijn van een glaswarenfabriek. De dispatching-regels die momenteel worden gebruikt in deze twee situaties zijn gebaseerd op prioriteitslijsten voor locaties. Omdat deze huidige dispatching-regels niet worden geëvalueerd in de literatuur, is het interessant om te bekijken of de beste regels uit de literatuur goed presteren in deze situaties. Met behulp van simulatie hebben we verschillende bekende dispatching-regels vergeleken, waaronder de aangepaste-wie-het-eerst-komt-die-het-eerst-maalt-regel (MODFCFS), de kleinste-afstand-eerst-regel (STDF) en de dichtstbijzijnd-voertuig-eerst-regel (NVF) en een variatie op de NVF-regel (NVF met wachttijdbeperking, of NVFTP). We onderzoeken ook de waarde van informatie omtrent aankomsttijdstippen en locaties van toekomstige ladingen en mogelijke prestatieverbeteringen die gerealiseerd kunnen worden door vooruit te kijken wanneer dergelijke informatie vroegtijdig beschikbaar is. Resultaten laten zien dat op afstand gebaseerde dispatching-regels (NVF, STDF) significant beter presteren met betrekking tot de gemiddelde wachttijd van een lading dan dispatching-regels die op tijd zijn gebaseerd (MODFCFS) en de specifieke dispatching-regels die gehanteerd worden door de twee bedrijven, ongeacht de bezettingsgraden van voertuigen. Het belangrijkste nadeel van de op afstand gebaseerde regels (NVF, STDF) is dat ze, terwijl ze de gemiddelde wachttijd van een lading minimaliseren, tevens de maximale wachttijd neigen te maximaliseren. Dit is met name het geval als de bezettingsgraad van voertuigen hoog is of bij aanwezigheid van afgelegen afzetgebieden. De NVFTP-regel is ontworpen om deze tekortkoming teniet te doen. Deze regel presteert goed met betrekking tot beide criteria (het minimaliseren van de gemiddelde en maximale wachttijd van ladingen). In het algemeen kan het gebruik in een dispatching-regel van vroegtijdig beschikbare en realistische informatie omtrent aankomsttijdstippen en locaties van toekomstige ladingen de gemiddelde wachttijd van een lading aanzienlijk terugdringen. Gebaseerd op de twee praktijksituaties die zijn bestudeerd in dit hoofdstuk, is de rangschikking van de dispatching-regels gebaseerd op de gemiddelde wachttijd van een lading als volgt: (1) NVF/NVFTP, (2) STDF, (3) situatie-specifieke regels en (4) MODFCFS.

In Hoofdstuk 4 introduceren we enkele complexere dispatching-regels voor de twee praktijksituaties uit Hoofdstuk 3. Het voornaamste doel van het bestuderen van complexe dispatching-regels is het zoeken van efficiëntere en robuustere dispatching-regels voor VBIT-omgevingen. Een "robuuste" dispatching-regel blijft goed presteren in verschillende situaties (d.w.z. verschillende aankomsttijden van ladingen en varianties in die aankomsttijden, verschillende aantallen beschikbare voertuigen, etc.). De complexe regels zijn: (1) de multi-factor regel (Multi-att) die voertuigen aanstuurt op basis van een functie die meerdere factoren in aanmerking neemt; (2) de dichtstbijzijnd-voertuig-eerst-regel met voertuig hertoewijzing (het opnieuw inschakelen van een voertuig dat op weg is naar zijn parkeerlocatie om een nieuwe lading op te halen) (NVF_R); (3) de dichtstbijzijnd-voertuig-eerst-regel met voertuig hertoewijzing en wachttijdbeperking (NVFTP_R); (4) de dichtstbijzijnd-voertuig-eerst-regel met voertuig hertoewijzing en afzegging (het opnieuw toewijzen van een voertuig dat onderweg is om een lading op te halen aan een andere lading die dichterbij, waarna de huidige taak vervalt) (NVF_RC); en (5) de gecombineerde dispatching-regel die multi-factor dispatching combineert met voertuig hertoewijzing (Combi). De resultaten van Hoofdstuk 4 geven aan dat het toewijzen van een nieuwe taak aan voertuigen die reeds onderweg zijn een positieve invloed kan hebben op het reduceren van de gemiddelde wachttijd van een lading. Deze invloed is groter als de bezettingsgraad van de voertuigen laag is. De regel die zowel hertoewijzing als afzegging implementeert (NVF_RC) kan de gemiddelde wachttijd van een lading nog verder reduceren. Dispatching-regels die hertoewijzing gebruiken leiden echter tot een hoge waarde van de maximale wachttijd van een lading. Bovendien moet het besturingssysteem, om de hertoewijzingsregel (met name NVF_RC) te implementeren, de posities van voertuigen continu monitoren. Dit is moeilijk of zelfs onmogelijk in veel VBIT-systemen, met name in systemen met door personen bestuurde voertuigen. Het gebruik van benaderingen voor reisafstanden van voertuigen of het voorspellen van de locaties van voertuigen kan een oplossing zijn. Multi-factor dispatching-regels kunnen de maximale wachttijd van een lading aanzienlijk terugdringen. De resultaten van Hoofdstuk 4 tonen aan dat, in de meeste gevallen, het combineren van multi-factor dispatching en het opnieuw toewijzen van rijdende voertuigen in één regel leidt tot een efficiënte en robuuste dispatching-regel (Combi).

In Hoofdstuk 5 bestuderen we een heel ander type voertuigbesturingssysteem (aangeduid met L-VBIT-systeem), dat een groot aantal (automatisch) aangestuurde voertuigen gebruikt voor het vervoeren van ladingen. Het blijkt dat het rechtstreeks implementeren van goed presterende dispatching-regels uit de vorige twee hoofdstukken niet werkt in zo'n omgeving. Ook de STDF-regel, aangepast aan L-VBIT-systemen, presteert niet erg goed. Daarom introduceren we twee nieuwe dispatching-regels, te weten Multi-Att en Multi-Mod, beide multi-factorregels. Deze regels presteren goed voor L-VBIT-systemen onder verscheidene omstandigheden en zijn geschikt om te implementeren in de praktijk.

De literatuur op het gebied van extern transport laat zien dat scheduling van voertuigen betere prestaties kan opleveren dan dispatching. Daarom bestuderen we in Hoofdstuk 6 scheduling-strategieën voor interne voertuigtransportsystemen waarbij voertuigen worden aangestuurd op basis van de op ieder willekeurig moment tijdens de planningshorizon beschikbare informatie (zogenaamde real-time scheduling). In dit proefschrift formuleren we een VBIT-schedulingsysteem als een handelsreizigerprobleem met tijdsvensters en *m* voertuigen (*m*-TSPTW). Om de statische versie van dit probleem op te lossen, stellen we twee aanpakken voor, waarbij de planningshorizon voortschrijdt op basis van tijd ofwel op basis van het aantal ingeplande ladingen. We stellen tevens een andere goede dynamische schedulingstrategie voor, het vooruitkijkende dynamische toewijzingsalgoritme (LAS). Deze dynamische schedulingstrategieën worden vergeleken met twee van de beste dispatching-regels (NVF en NVF_LA) uit vorige hoofdstukken voor twee experimentele layouts (een U- en een I-layout). De resultaten van Hoofdstuk 6 laten zien dat dynamische schedulingstrategieën consequent beter presteren dan dispatchingregels in beide experimentele omgevingen en onder verschillende operationele omstandigheden. Deze bevinding is vergelijkbaar met die van het volledige vrachtwagenlading scheduling-probleem in extern transport. Opmerkelijke verbeteringen ontstaan als gecombineerde en kolom-generatie heuristieken worden toegepast voor het oplossen van statische probleeminstanties. In de dynamische situatie zijn significante verbeteringen met betrekking tot dispatching alleen mogelijk als er voldoende informatie beschikbaar is aangaande toekomstige ladingen (ongeveer twee tot 4 ladingen per voertuig). Tevens blijkt uit de resultaten van Hoofdstuk 6 dat strategieën met een voortschrijdende horizon beter presteren dan de dynamische toewijzingsalgoritmes (met en zonder vooruitkijken), wanneer het systeem voldoende informatie kan verschaffen ten aanzien van de aankomsttijdstippen en locaties van toekomstige ladingen. Als we minder dan twee ladingen per voertuig vooruit kennen, zijn de dynamische toewijzingsalgoritmes competitief vergeleken met de strategieën met voortschrijdende horizon. De dynamische toewijzingsalgoritmes (DAS, LAS) presteren beter dan dispatching-regels, met name als informatie omtrent ladingen reeds vantevoren beschikbaar is. Deze dynamische algoritmes zijn ook snel en kunnen daarom goed toegepast worden in de praktijk. Het vooruitkijkende dynamische toewijzingsalgoritme (LAS) presteert significant beter dan dispatching-regels en het eenvoudige dynamische toewijzingsalgoritme (DAS). Het voornaamste nadeel van dynamische toewijzingsalgoritmes is dat hun succes afhankelijk is van hun kostenfuncties en -parameters. Afhankelijk van de toepassing, dient de juiste kostenfunctie te worden geselecteerd en dienen de kostenparameters zorgvuldig te worden ingesteld.

De resultaten van dit proefschrift laten zien dat de systeemlayout een grote invloed heeft op de voertuigaansturingsstrategieën (dispatching en scheduling). Dispatching-regels zijn gevoeliger voor verschillende typen layout (Hoofdstuk 6). Het belangrijkste voordeel van dispatching-regels is hun eenvoud. Schedulingstrategieën daarentegen zijn veel complexer. Het is niet eenvoudig om een voertuigschedulingstrategie te integreren in standaard *shop floor* besturingssoftware die gebruikt wordt in de praktijk. De schedulingstrategieën presteren vooral goed bij een lage voertuigbezettingsgraad. Het verschil in prestatie tussen dispatching- en schedulingstrategieën neemt af als deze bezettingsgraad toeneemt. De dispatching- en schedulingstrategieën presteren ook beter bij een kleine dan bij een grote variatie in aankomstfrequenties van ladingen. In systemen met een groot aantal voertuigen (zoals die in Hoofdstuk 5), lijkt het gebruik van een dispatching-regel meer voor de hand te liggen dan het gebruik van een schedulingstrategie. Dit proefschrift laat tevens zien dat de hoeveelheid informatie die vroegtijdig beschikbaar is ook een rol speelt in de selectie van een geschikte voertuigaansturingsstrategie. In een zeer stochastische omgeving (geen vroegtijdige informatie) zijn dispatching-regels en het DAS-algoritme geschikt. DAS presteert doorgaans beter dan dispatching-regels, maar is complexer. Als er enige vroegtijdige informatie beschikbaar is (ongeveer één lading per voertuig), dan dienen het vooruitkijkende dynamische toewijzingsalgoritme en dispatching-regels te worden toegepast. Als er meer informatie omtrent toekomstige ladingen beschikbaar is (meer dan één lading per voertuig), dan wordt een dynamische schedulingstrategie met een voortschrijdende horizon aanbevolen.

Tóm tắt (Summary in Vietnamese)

Nghiên cứu “*Điều khiển thông minh các hệ thống giao thông nội bộ*” tập trung nâng cao hiệu quả của hệ thống vận tải nội bộ trong các cơ sở như nhà kho, phân xưởng sản xuất, cảng đường thuỷ và cảng hàng không. Trong nghiên cứu này, chúng tôi tập trung vào phương pháp điều hành (hay điều độ vận tải) xe công nghiệp (*industrial vehicle*). Hai phương pháp chính trong điều độ vận tải gồm có: điều độ tức thời (*dispatching*) và điều độ theo kế hoạch (*scheduling*). Một phần của nghiên cứu này tập trung nâng cao chất lượng của điều độ tức thời trong thực tế. Để thực hiện mục tiêu này, chúng tôi đánh giá hiệu quả của một số phương pháp điều độ tức thời cho hai nhà kho trong thực tế bằng phương pháp mô phỏng. Nghiên cứu chỉ ra rằng trong các môi trường nơi có đủ chỗ cho sản phẩm (hàng hoá) đợi trước khi được chuyển đi, các phương pháp điều độ tức thời dựa trên việc giảm quãng đường xe phải đi có hiệu quả cao hơn các phương pháp điều độ dựa trên việc giảm thời gian đợi của sản phẩm. Bên cạnh việc đánh giá chất lượng, chúng tôi cũng đề xuất một số phương pháp điều độ tức thời mới nhằm nâng cao hơn hiệu quả của hệ thống. Thực nghiệm với các phương pháp điều độ mới này chỉ ra rằng các phương pháp điều độ tính đến nhiều yếu tố khác nhau khi điều khiển xe sẽ ổn định hơn. Hơn nữa, việc dùng các xe đang đi về điểm đỗ ngay vào việc vận chuyển hàng cũng có tác dụng tích cực trong việc nâng cao hiệu năng của các phương pháp điều độ vận tải. Cũng trong nghiên cứu này, chúng tôi đề xuất một số biện pháp điều độ tức thời cho một loại hệ thống vận chuyển dùng xe công nghiệp cần một số lượng lớn xe. Hệ thống vận chuyển hành lý trong sân bay là một trong những điển hình về loại hệ thống này. Trong các hệ thống này, luật điều độ tức thời sử dụng thông tin về quãng đường đi không tải của xe kết hợp với yêu cầu về xe ở một vị trí (hoặc một thiết bị) nhất định hoạt động rất tốt.

Ngoài việc nghiên cứu sử dụng nguyên tắc điều độ tức thời, chúng tôi còn nghiên cứu việc sử dụng phương pháp điều độ theo kế hoạch để điều hành xe công nghiệp. Dùng phương pháp mô phỏng kết hợp với tối ưu hoá, chúng tôi cũng chỉ ra rằng phương pháp điều độ theo kế hoạch vận hành tốt hơn phương pháp điều độ tức thời khi ta biết đủ thông tin về các sản phẩm sẽ đến. Kết quả nghiên cứu cũng cho thấy rằng, các yếu tố như hệ thống

đường đi của xe, tốc độ đến của sản phẩm, cũng như lượng thông tin mà ta có thể biết trước về các sản phẩm sẽ đến có ảnh hưởng quan trọng đến hiệu quả của các phương pháp điều độ vận tải.

Curriculum Vitae

Tuan Le Anh was born in Hanoi, Vietnam (1973). He studied electrical engineering at the Hanoi University of Technology and received his bachelor and master degrees in electrical engineering in 1995 and 1997. In 1998, he got his postgraduate diploma in industrial engineering from the Asian Institute of Technology. In 2000, he received his master degree (with great distinction) in industrial management at the Catholic University of Leuven with the thesis on applying heuristics to solve multi-item single-level capacitated lot-sizing problems. From 2001, he started his Ph.D research at the RSM Erasmus University (formerly Rotterdam School of Management/ Erasmus University Rotterdam). His Ph.D research focuses on the operational control of vehicle-based internal transport systems, which has resulted in a number of published (or forthcoming) articles in international scientific journals such as Journal of Operations Management, International Journal of Production Research and European Journal of Operational Research. He has also given presentations on his research in several international conferences in both Europe and North America.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES
RESEARCH IN MANAGEMENT

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

Appelman, J.H., *Governance of Global Interorganizational Tourism Networks; Changing Forms of Co-ordination between the Travel Agency and Aviation Sector*, Promotors: Prof. dr. F.M. Go & Prof. dr. B. Nooteboom, EPS-2004-036-MKT, ISBN 90-5892-060-7, http://hdl.handle.net/1765/1199

Berens, G., *Corporate Branding: The Development of Corporate Associations and their Influence on Stakeholder Reactions*, Promotor: Prof. dr. C. B. M. van Riel, EPS-2004-039-ORG, ISBN 90 –5892–065–8, http://hdl.handle.net/1765/1273

Berghe, D.A.F., *Working Across Borders: Multinational Enterprises and the Internationalization of Employment*, Promotors: Prof. dr. R.J.M. van Tulder & Prof. dr. E.J.J. Schenk, EPS-2003-029-ORG, ISBN 90-5892-05-34, http://hdl.handle.net/1765/1041

Bijman, W.J.J., *Essays on Agricultural Co-operatives; Governance Structure in Fruit and Vegetable Chains*, Promotor: Prof. dr. G.W.J. Hendrikse, EPS-2002-015-ORG, ISBN: 90-5892-024-0, http://hdl.handle.net/1765/867

Brito, M.P. de, *Managing Reverse Logistics or Reversing Logistics Management?* Promotors: Prof. dr. ir. R. Dekker & Prof. dr. M. B. M. de Koster, EPS-2004-035-LIS, ISBN: 90-5892-058-5, http://hdl.handle.net/1765/1132

Campbell, R.A.J., *Rethinking Risk in International Financial Markets*, Promotor: Prof. dr. C.G. Koedijk, EPS-2001-005-F&A, ISBN: 90-5892-008-9, http://hdl.handle.net/1765/306

Chen, Y., *Labour Flexibility in China's Companies: An Empirical Study*, Promotors: Prof. dr. A. Buitendam & Prof. dr. B. Krug, EPS-2001-006-ORG, ISBN: 90-5892-012-7, http://hdl.handle.net/1765/307

Daniševská, P., *Empirical Studies on Financial Intermediation and Corporate Policies*, Promotor: Prof. dr. C.G. Koedijk, EPS-2004-044-F&A, ISBN 90–5892–070–4, http://hdl.handle.net/1765/1518

Delporte-Vermeiren, D.J.E., *Improving the Flexibility and Profitability of ICT-enabled Business Networks: An Assessment Method and Tool*, Promotors: Prof.mr. dr. P.H.M. Vervest & Prof. dr. ir. H.W.G.M. van Heck, EPS-2003-020-LIS, ISBN: 90-5892-040-2, http://hdl.handle.net/1765/359

Dijksterhuis, M., *Organizational Dynamics of Cognition and Action in the Changing Dutch and US Banking Industries*, Promotors: Prof. dr. ing. F.A.J. van den Bosch & Prof. dr. H.W. Volberda, EPS-2003-026-STR, ISBN: 90-5892-048-8, http://hdl.handle.net/1765/1037

Fenema, P.C. van, *Coordination and Control of Globally Distributed Software Projects*, Promotor: Prof. dr. K. Kumar, EPS-2002-019-LIS, ISBN: 90-5892-030-5, http://hdl.handle.net/1765/360

Fleischmann, M., *Quantitative Models for Reverse Logistics*, Promoters: Prof. dr. ir. J.A.E.E. van Nunen & Prof. dr. ir. R. Dekker, EPS-2000-002-LIS, ISBN: 3540 417 117, http://hdl.handle.net/1765/1044

Flier, B., *Strategic Renewal of European Financial Incumbents: Coevolution of Environmental Selection, Institutional Effects, and Managerial Intentionality*, Promotors: Prof. dr. ing. F.A.J. van den Bosch & Prof. dr. H.W. Volberda,
EPS-2003-033-STR, ISBN: 90–5892–055–0, http://hdl.handle.net/1765/1071

Fok, D., *Advanced Econometric Marketing Models*, Promotor: Prof. dr. P.H.B.F. Franses, EPS-2003-027-MKT, ISBN: 90-5892-049-6, http://hdl.handle.net/1765/1035

Ganzaroli , A., *Creating Trust between Local and Global Systems*, Promotors: Prof. dr. K. Kumar & Prof. dr. R.M. Lee, EPS-2002-018-LIS, ISBN: 90-5892-031-3, http://hdl.handle.net/1765/361

Gilsing, V.A., *Exploration, Exploitation and Co-evolution in Innovation Networks*, Promotors: Prof. dr. B. Nooteboom & Prof. dr. J.P.M. Groenewegen, EPS-2003-032-ORG, ISBN 90-5892-05-42, http://hdl.handle.net/1765/1040

Graaf, G. de, *Tractable Morality: Customer Discourses of Bankers, Veterinarians and Charity Workers*, Promotors: Prof. dr. F. Leijnse & Prof. dr. T. van Willigenburg, EPS-2003-031-ORG, ISBN: 90–5892–051–8, http://hdl.handle.net/1765/1038

Hermans. J.M., *ICT in Information Services, Use and deployment of the Dutch securities trade, 1860-1970.* Promotor: Prof. dr. drs. F.H.A. Janszen, EPS-2004-046-ORG, ISBN 90-5892-072-0, http://hdl.handle.net/1765/1793

Heugens, P.M.A.R., *Strategic Issues Management: Implications for Corporate Performance*, Promotors: Prof. dr. ing. F.A.J. van den Bosch & Prof. dr. C.B.M. van Riel, EPS-2001-007-STR, ISBN: 90-5892-009-7,
http://hdl.handle.net/1765/358

Hooghiemstra, R., *The Construction of Reality*, Promotors: Prof. dr. L.G. van der Tas RA & Prof. dr. A.Th.H. Pruyn, EPS-2003-025-F&A, ISBN: 90-5892-047-X, http://hdl.handle.net/1765/871

Jong, C. de, *Dealing with Derivatives: Studies on the Role, Informational Content and Pricing of Financial Derivatives*, Promotor: Prof. dr. C.G. Koedijk, EPS-2003-023-F&A, ISBN: 90-5892-043-7, http://hdl.handle.net/1765/1043

Kippers, J., *Empirical Studies on Cash Payments*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2004-043-F&A. ISBN 90-5892-069-0, http://hdl.handle.net/1765/1520

Koppius, O.R., *Information Architecture and Electronic Market Performance*, Promotors: Prof. dr. P.H.M. Vervest & Prof. dr. ir. H.W.G.M. van Heck, EPS-2002-013-LIS, ISBN: 90-5892-023-2, http://hdl.handle.net/1765/921

Langen, P.W. de, *The Performance of Seaport Clusters; A Framework to Analyze Cluster Performance and an Application to the Seaport Clusters of Durban, Rotterdam and the Lower Mississippi*, Promotors: Prof. dr. B. Nooteboom & Prof. drs. H.W.H. Welters, EPS-2004-034-LIS, ISBN: 90-5892-056-9, http://hdl.handle.net/1765/1133

Liang, G., *New Competition; Foreign Direct Investment And Industrial Development In China,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-047-ORG, ISBN 90–5892–073–9, http://hdl.handle.net/1765/1795.

Loef, J., *Incongruity between Ads and Consumer Expectations of Advertising*, Promotors: Prof. dr. W.F. van Raaij & Prof. dr. G. Antonides, EPS-2002-017-MKT, ISBN: 90-5892-028-3, http://hdl.handle.net/1765/869

Mandele, L.M., van der, *Leadership and the Inflection Point: A Longitudinal Perspective*, Promotors: Prof. dr. H.W. Volberda, Prof. dr. H.R. Commandeur, EPS-2004-042-STR, ISBN 90–5892–067–4, http://hdl.handle.net/1765/1302

Meer, J.R. van der, *Operational Control of Internal Transport*, Promotors: Prof. dr. M.B.M. de Koster & Prof. dr. ir. R. Dekker, EPS-2000-001-LIS, ISBN:90-5892-004-6, http://hdl.handle.net/1765/859

Miltenburg, P.R., *Effects of Modular Sourcing on Manufacturing Flexibility in the Automotive Industry: A Study among German OEMs*, Promotors: Prof. dr. J. Paauwe & Prof. dr. H.R. Commandeur, EPS-2003-030-ORG, ISBN 90-5892-052-6, http://hdl.handle.net/1765/1039

Mol, M.M., *Outsourcing, Supplier-relations and Internationalisation: Global Source Strategy as a Chinese Puzzle*, Promotor: Prof. dr. R.J.M. van Tulder, EPS-2001-010-ORG, ISBN: 90-5892-014-3, http://hdl.handle.net/1765/355

Mulder, A., *Government Dilemmas in the Private Provision of Public Goods*, Promotor: Prof. dr. R.J.M. van Tulder, EPS-2004-045-ORG, ISBN: 90-5892- 071-2, http://hdl.handle.net/1765

Muller, A.R., *The Rise of Regionalism: Core Company Strategies Under The Second Wave of Integration*, Promotor: Prof. dr. R.J.M. van Tulder, EPS-2004-038-ORG, ISBN 90–5892–062–3, http://hdl.handle.net/1765/1272

Oosterhout, J. van, *The Quest for Legitimacy: On Authority and Responsibility in Governance*, Promotors: Prof. dr. T. van Willigenburg & Prof.mr. H.R. van Gunsteren, EPS-2002-012-ORG, ISBN: 90-5892-022-4, http://hdl.handle.net/1765/362

Peeters, L.W.P., *Cyclic Railway Timetable Optimization*, Promotors: Prof. dr. L.G. Kroon & Prof. dr. ir. J.A.E.E. van Nunen, EPS-2003-022-LIS, ISBN: 90-5892-042-9, http://hdl.handle.net/1765/429

Popova. V., *Knowledge Discovery and Monotonicity*, Promotor: Prof. dr. A. de Bruin, EPS-2004-037-LIS, ISBN 90-5892-061-5, http://hdl.handle.net/1765/1201

Puvanasvari Ratnasingam, P., *Interorganizational Trust in Business to Business E-Commerce*, Promotors: Prof. dr. K. Kumar & Prof. dr. H.G. van Dissel, EPS-2001-009-LIS, ISBN: 90-5892-017-8, http://hdl.handle.net/1765/356

Romero Morales, D., *Optimization Problems in Supply Chain Management*, Promotors: Prof. dr. ir. J.A.E.E. van Nunen & Dr. H.E. Romeijn, EPS-2000-003-LIS, ISBN: 90-9014078-6, http://hdl.handle.net/1765/865

Roodbergen , K.J., *Layout and Routing Methods for Warehouses*, Promotors: Prof. dr. M.B.M. de Koster & Prof. dr. ir. J.A.E.E. van Nunen, EPS-2001-004-LIS, ISBN: 90-5892-005-4, http://hdl.handle.net/1765/861

Schweizer, T.S., *An Individual Psychology of Novelty-Seeking, Creativity and Innovation,* Promotor: Prof.dr. R.J.M. van Tulder. EPS-2004-048-ORG, ISBN: 90-5892-07-71, http://hdl.handle.net/1765/1818

Six, F.E., *Trust and Trouble: Building Interpersonal Trust Within Organizations*, Promotors: Prof. dr. B. Nooteboom & Prof. dr. A.M. Sorge, EPS-2004-040-ORG, ISBN 90–5892–064–X, http://hdl.handle.net/1765/1271

Slager, A.M.H., *Banking across Borders*, Promotors: Prof. dr. D.M.N. van Wensveen & Prof. dr. R.J.M. van Tulder, EPS-2004-041-ORG, ISBN 90-5892-066–6, http://hdl.handle.net/1765/1301

Speklé, R.F., *Beyond Generics: A closer look at Hybrid and Hierarchical Governance*, Promotor: Prof. dr. M.A. van Hoepen RA, EPS-2001-008-F&A, ISBN: 90-5892-011-9, http://hdl.handle.net/1765/357

Teunter, L.H., *Analysis of Sales Promotion Effects on Household Purchase Behavior*, Promotors: Prof. dr. ir. B. Wierenga & Prof. dr. T. Kloek, EPS-2002-016-ORG, ISBN: 90-5892-029-1, http://hdl.handle.net/1765/868

Vis, I.F.A., *Planning and Control Concepts for Material Handling Systems*, Promotors: Prof. dr. M.B.M. de Koster & Prof. dr. ir. R. Dekker, EPS-2002-014-LIS, ISBN: 90-5892-021-6, http://hdl.handle.net/1765/866

Vliet, P. van, Downside Risk and Empirical Asset Pricing, Promotor: Prof. dr. G.T. Post, EPS-2004-049-F&A ISBN 90-5892-07-55, http://hdl.handle.net/1765/1819

Waal, T. de, *Processing of Erroneous and Unsafe Data*, Promotor: Prof. dr. ir. R. Dekker, EPS-2003-024-LIS, ISBN: 90-5892-045-3, http://hdl.handle.net/1765/870

Wielemaker, M.W., *Managing Initiatives: A Synthesis of the Conditioning and Knowledge-Creating View*, Promotors: Prof. dr. H.W. Volberda & Prof. dr. C.W.F. Baden-Fuller, EPS-2003-28-STR, ISBN 90-5892-050-X, http://hdl.handle.net/1765/1036

Wijk, R.A.J.L. van, *Organizing Knowledge in Internal Networks: A Multilevel Study*, Promotor: Prof. dr. ing. F.A.J. van den Bosch, EPS-2003-021-STR, ISBN: 90-5892-039-9, http://hdl.handle.net/1765/347

Wolters, M.J.J., *The Business of Modularity and the Modularity of Business*, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof. dr. ir. H.W.G.M. van Heck, EPS-2002-011-LIS, ISBN: 90-5892-020-8, http://hdl.handle.net/1765/920

Intelligent Control of Vehicle-Based Internal Transport Systems

"Intelligent control of vehicle-based internal transport (VBIT) systems" copes with real-time dispatching and scheduling of internal-transport vehicles, such as forklifts and guided vehicles. VBIT systems can be found in warehouses, distribution centers, manufacturing plants, airport and transshipment terminals. Using simulation of two real-world environments, dispatching rules described in literature and several newly introduced rules are compared on performance. The performance evaluation suggests that in environments where queue space is not a restriction, distance-based dispatching rules such as shortest-travel-distance-first outperform time-based dispatching rules such as modified-first-come-first-served and using load pre-arrival information has a significant positive impact on reducing the average load waiting time. Experimental results also reveal that multi-attribute dispatching rules combining distance and time aspects of vehicles and loads are robust to variations in working conditions. In addition, multi-attribute rules which take vehicle empty travel distance and vehicle requirement at a station into account perform very well in heavy-traffic VBIT systems such as baggage handling systems. Besides dispatching rules, the potential contribution of dynamic vehicle scheduling for VBIT systems is investigated. Experiments using simulation in combination with optimization show that when sufficient pre-arrival information is available a dynamic scheduling approach outperforms the dispatching approach. This thesis also evaluates the impact of guide-path layout, load arrival rate and variance, and the amount of load pre-arrival information on different vehicle control approaches (scheduling and dispatching). Based on experimental results, recommendations for selecting appropriate vehicle control approaches for specific situations are presented.

ERIM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are RSM Erasmus University and the Erasmus School of Economics. ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focussed on the management of the firm in its environment, its intra- and inter-firm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced graduate program in Research in Management. Within ERIM, over two hundred senior researchers and Ph.D. candidates are active in the different research programs. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

www.erim.eur.nl

ISBN 90-5892-079-8