IRIS F.A. VIS

Planning and Control Concepts for Material Handling Systems

Planning and Control Concepts for Material Handling Systems

Planning and Control Concepts for Material Handling Systems

Plannings- en besturingsconcepten voor
transport- en opslagsystemen

Proefschrift

ter verkrijging van de graad van doctor
aan de Erasmus Universiteit Rotterdam
op gezag van de Rector Magnificus
Prof.dr.ir. J.H. van Bemmel
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op
vrijdag 17 mei 2002 om 13.30 uur

door

Iris Frederieke Arianne Vis
geboren te Leidschendam

Promotiecommissie:
Promotoren:
Prof.dr. M.B.M. de Koster
Prof.dr.ir. R. Dekker

Overige leden:
Prof.dr. S.L. van de Velde
Dr.ir. K.I. Aardal
Prof.dr. M.W.P. Savelsbergh

ERIM Ph.D. Series Research in Management 14
TRAIL Thesis Series T2002/2, The Netherlands TRAIL Research School

Cover photo: container terminal of Salerno, Italy by I.F.A. Vis.
ISBN 90-5892-021-6

to my parents

Voorwoord

Een heel traject van opleiding en ontwikkeling ging vooraf aan het schrijven van dit proefschrift. Reeds op de lagere school had ik een meer dan gemiddelde belangstelling voor rekenen en denkpuzzels. De lessen wiskunde op de middelbare school spraken mij dan ook zeer aan. Aan het einde van mijn middelbareschooltijd twijfelde ik tussen de studies rechten en wiskunde. De heer A. Vink, mijn toenmalige leraar wiskunde, ben ik nog steeds dankbaar voor het vertrouwen en de stimulans die hij mij heeft gegeven om voor een studie wiskunde te kiezen. De studie wiskunde aan de Universiteit Leiden heeft mij een goede basis verschaft om dit proefschrift te schrijven. De denkwijze, het formuleren van bewijzen en het vertalen van een praktijkprobleem naar de theorie komen mij nog dagelijks van pas. Mijn afstudeerscriptie over kasvoorraadbeheer bij banken heb ik geschreven onder de uitstekende begeleiding van Lodewijk Kallenberg, Erwin Dekker en Marc Salomon. In december 1997 heb ik de AXA Leven scriptieprijs ontvangen voor dit afstudeeronderzoek.

In september 1997 ben ik begonnen als Assistent in Opleiding (AiO) aan de faculteit Bedrijfskunde van de Erasmus Universiteit Rotterdam. Samen met de andere eerstejaars AiO's begaf ik mij op het onderzoekspad. Pursey Heugens en Michael Mol bedank ik voor het feit dat zij mij hebben ingewijd in de wondere wereld van Bedrijfskunde. Samen met Leon Peeters heb ik verschillende cursussen van het Landelijk Netwerk Mathematische Besliskunde gevolgd. Het samen uitwerken van de opgaven leverde naast interessante ook gezellige discussies op. Ook de discussies die wij samen hebben gevoerd over hoofdstuk 3 van dit proefschrift waren zeer nuttig. Mijn jaargenoten bedank ik voor de gezelligheid in de gangen en de etentjes na werktijd. Ik hoop dat we, ook nu we allemaal een andere kant zijn opgegaan, nog af en toe bij elkaar kunnen komen om bij te praten. Samen met Patrick Meersmans heb ik gedurende het eerste jaar van de AiO-periode intensief gewerkt aan een simulatiestudie in het kader van het FAMAS-project. Deze samenwerking heeft geleid tot twee gezamenlijke publicaties, namelijk [189] en [272]. De vriendschappelijke contacten die we aan deze samenwerking hebben overgehouden waardeer ik zeker.

Mijn promotoren René de Koster en Rommert Dekker dank ik voor de gelegenheid die zij mij hebben geboden om dit onderzoeksproject uit te voeren en zelfstandig onderzoek te verrichten. Van oktober 1999 tot en met december 1999 ben ik Visiting Scholar aan het Georgia Institute of Technology in Atlanta (VS) geweest. Dit bezoek was financieel gezien mogelijk door de som geld verbonden aan mijn scriptieprijs en de financiële bijdragen van de faculteit Bedrijfskunde van de Erasmus Universiteit Rotterdam en de Vereniging Trustfonds Erasmus Universiteit Rotterdam. Ik dank deze instellingen voor deze ondersteuning. Martin Savelsbergh dank ik voor de gelegenheid die hij mij heeft geboden gedurende deze drie maanden intensief onderzoek te doen, voor zijn inhoudelijke bijdragen aan mijn onderzoek en de leerzame gesprekken. Jo van Nunen dank ik hartelijk voor zijn persoonlijke belangstelling gedurende mijn AiO periode.

Naast het doen van onderzoek heb ik de mogelijkheid gebruikt om in de afgelopen vier jaar ook op onderwijsgebied actief te zijn. De resultaten van het door mij opgezette en gegeven projectvak 'Worden Magazijnen Overbodig?' hebben geleid tot vier publicaties in het Tijdschrift voor Inkoop en Logistiek (zie [24], [53], [107] en [278]). Tevens heb ik computerpractica begeleid in het verdiepingsblok van de major logistiek. De begeleiding van een afstudeerder op het gebied van intern transport in containerterminals was een uitdaging. Mede door deze onderwijsactiviteiten heb ik ontdekt, hoe leuk het is om mijn eigen logistieke kennis over te brengen op enthousiaste studenten. Kees Jan Roodbergen dank ik voor de goede samenwerking op het gebied van onderwijs en voor het lezen en bediscussiëren van verschillende hoofdstukken uit dit proefschrift.

AiO zijn in een multidisciplinaire vakgroep heeft vele voordelen. Mijn kennis op het gebied van logistiek, technologie, innovatie, standaardisatie en kwaliteitsmanagement is gegroeid door interessante discussies tijdens de lunches. De discussies over de informelere onderwerpen zoals de wasmachine, die op een timer werkt, toebehorende aan een van de mannelijke leden van de vakgroep, hebben mij altijd zeer geboeid. De leden en oud-leden van de vakgroep Management van Technologie en Innovatie en de andere medewerkers van de Erasmus Universiteit Rotterdam met wie ik contact heb gehad, bedank ik voor de prettige samenwerking.

Gedurende twee jaar ben ik achtereenvolgens vice-voorzitter en voorzitter van de AiO-raad Bedrijfskunde geweest. In dit kader heb ik veel contacten gehad met de medewerkers van de onderzoeksschool ERIM. Met name AnneMarie Stolkwijk bedank ik voor de goede samenwerking en haar persoonlijke belangstelling.

Parallel aan een leven vol opleiding en werk loopt een privéleven. Gedurende het promotietraject hebben vele vrienden en bekenden belangstelling getoond voor mijn wel en wee. Enkelen van hen noem ik hier graag bij naam: de heer W. van der Bijl, Anke Brockhoff, Moritz Fleischmann, mevrouw J.C. van Heijst-Steenbeek, Hanneke Mulder, mevrouw A.L.W. Samama-Polak, Birgit Schmude en Loes van der Veer-Stotijn. Myriam Mousset-Wisse en Bertien Renting dank ik daarnaast voor hun aandacht voor mijn verhalen en het mij bezorgen van de benodige ontspanning tijdens het schrijven van mijn proefschrift.

Mijn oma dank ik voor haar liefde en aandacht voor mij. Haar karakter, met name haar doorzettingsvermogen, is een voorbeeld voor mij. Ewout is een fantastische broer. Onze mooie reizen, het gezamenlijk musiceren, het bediscussiëren van theoretische en praktische problemen en de gezelligheid zijn fijn. Mijn ouders ben ik dankbaar voor hun onvoorwaardelijke steun, aandacht, belangstelling en stimulering. Mede dankzij alle kansen die zij mij hebben geboden heb ik me kunnen ontwikkelen tot de vrouw die ik nu ben. Mijn echtgenoot Kees Jan bedank ik voor zijn begrip, aandacht en liefde. Hij is er altijd voor me.

Rotterdam, 2002
Iris Vis

Contents

List of Figures

1
Introduction

Technological innovations never seem to stop. New ways of communicating are continuously being developed, like mobile telephone calls, SMS messages, email and WAP. As a result, various processes in society are subject to an ever increasing speed of change. For example, customers can order products via the electronical highway from their homes. Customers are gradually getting used to this new method of ordering, which is fast and convenient. As a consequence, they expect an equally fast and convenient delivery process of the ordered products. Customers might even prefer to indicate exactly the time at which the products are to be delivered.

Historically, the focus of many companies has been on improving processes in the company. More recently the awareness has grown that it is also necessary to improve the process of interactions between firms to keep up with changes in customer demand. According to Handfield and Nichols [102] the *supply chain* encompasses all activities associated with the flow and transformation of goods from the raw materials stage through to the end-user, as well as the associated information flows. Materials and information flow both up and down the supply chain. Flows from supplier to customer are called *downstream flows.* Materials also flow *upstream* through the supply chain from customers to suppliers. Reasons for the occurrence of the latter flows are, for example, recycling of products and money back guarantee for unsatisfied customers. Supply chains are networks of mutually related suppliers and customers. Each customer is a supplier for the next organisation downstream in the supply chain. Figure 1.1 represents a simplified layout of a supply chain.

In a supply chain different centres can be distinguished, like manufacturing centres, warehouses, cross-docking centres and transhipment terminals. These organisations can be considered as nodes in a network. *Manufacturing centres* produce products. For this production raw material and/or semi-manufactured products might be needed. Finished products can be shipped to retailers and their customers via *warehouses* or *terminals*. In Section 1.3.1 a more detailed description is given of these nodes. Supply chains can vary in length and contents. Furthermore, companies can be part of various supply chains. Because of this, the term *supply networks* is used as well.

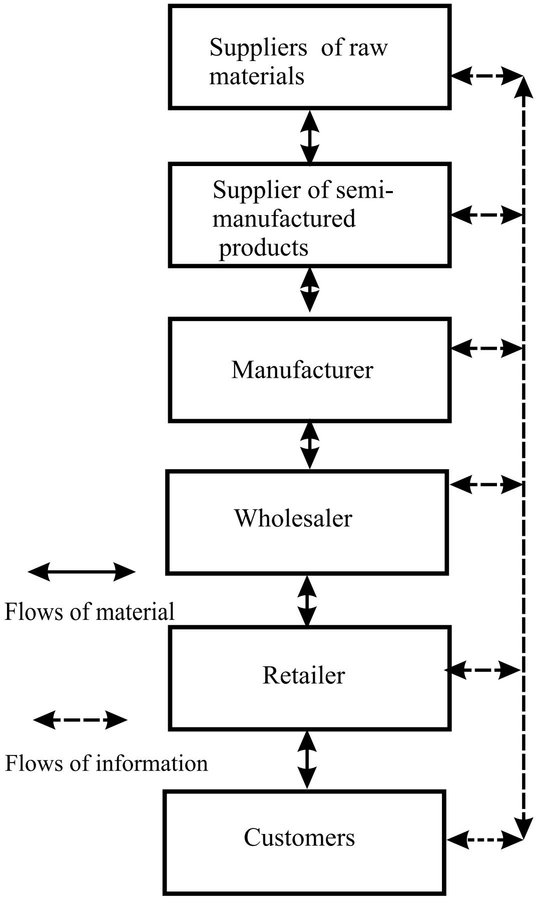

FIGURE 1.1. Simplified representation of a supply chain.

Activities of multiple companies in a supply chain can be adjusted to ensure an efficient supply chain instead of an efficient company in a potentially inefficient supply chain. As a result, involvement in the entire supply chain is a necessity to be able to meet, for example, tight delivery schedules and high service levels. *Supply chain management* can be defined as the efforts to integrate the processes in the supply chain. Cooperation, effective coordination and integration of materials and information and trust through out the supply chain might be necessary to obtain a valuable chain with satisfied customers. Simultaneously, attention can be paid to reduction of costs.

The costs due to the flow of materials through the supply chain can approach 75% of its budget (Handfield and Nichols [102]). This material flow should be managed accurately to ensure that the corresponding costs can be reduced, while meeting the service levels simultaneously. Therefore, attention can be paid to *logistics* activities.

Logistics is defined as a part of the supply chain process. According to the *Council of Logistics Management,* logistics is the part of the supply chain process that plans, implements, and controls the efficient, effective flow and storage of goods, service and related information from the point of origin to the point of consumption in order to meet customer's requirements. Clearly, logistics activities occur in nodes in the supply chain and between several nodes in the supply chain. This includes, for example, the transportation of goods from one location to another, planning of activities at warehouses as well as inventory keeping in warehouses. Inventories are, among other reasons, held to balance fluctuations in production and demand, to combine products from several producers for delivery to common customers and to enable emergency deliveries of critical components. However, logistics is not just concerned with handling products. The information flows to ensure that the right product is at the right place at the right time in the right quantity are part of logistics as well.

Therefore, information is one of the flows through the supply chain. Customers just present their wishes to retailers. From the retailers information flows to distribution centres and thereafter to the manufacturer. Next, the manufacturer can fulfil the wishes of its customers. In the past information was generally passed through to other members of the supply chain on paper. This was a slow, error prone and expensive process. However, for successful management of supply chains, adequate information flows are of great importance. Accurate information flows can, for example, reduce inventories because of reduced uncertainty. Furthermore, to serve customers in the best way it is important to have information on orders, product availability, and so on, available instantaneously. Information should be available for all departments in one organisation and for multiple organisations across the supply chain. Therefore, a paperless environment should be created.

To measure the performance of a supply chain, the performance of an individual node, such as a warehouse, or the performance of the supply chain as a whole can be observed. Several performance measures can be defined for the supply chain. Firstly, the total lead time of the supply chain, which is the time from the stage of raw materials to the delivery of the product to the customer can be used as a performance measure. The response time, the changes in average inventory through the supply chain and the reliability of the quality of the product are examples of other potential performance measures.

Companies have to adjust their actions constantly in order to remain competitive. Many trends are impacting the performance of the supply chain. In Section 1.2 some of these trends and their impact on supply chain management are discussed. To deal with these trends, several adaptions in logistics might be required. Section 1.2 indicates how logistics activities can be adapted to each of these trends. In a supply chain various nodes exist. The trends discussed also impact the processes in these nodes. Some of these nodes will be considered in Section 1.3. To perform logistics activities in material handling centres, various types of equipments can be used. Section 1.3.2 gives a short description of several types of equipments. Furthermore, some operating and planning policies controlling the processes in material handling centres are treated.

Material handling systems consist of the equipments, personnel, information, materials and related planning and control systems. Improvements in the performance of material handling systems might result in improvements in the performance of the supply chain. This thesis focuses on the development of control and planning concepts for material handling systems used in nodes in a supply chain. The thesis consists of three parts concerning automated guided vehicles, storage and retrieval machines and a practical application. In the next section we discuss briefly the main objectives and contributions of each part of this thesis. In Section 1.4 a more detailed outline is given of the problems studied in this thesis.

1.1 Objectives and contributions of this thesis

The first part of this thesis discusses Automated Guided Vehicles (AGVs). These vehicles take care of the internal transport of loads in material handling centres. A literature review is given on decision problems of automated guided vehicle systems. A planning problem is the determination of the number of vehicles required to transport a set of loads. We formulated this problem as a directed network problem. Each node represents a load to be transported. Directed arcs connect loads which can be transported by the same AGV. A new minimum flow algorithm is developed. With this algorithm the minimum number of directed paths can be found such that each node is covered by exactly one path in the network. This number of directed paths corresponds to the minimum number of vehicles required. With the algorithm the problem of determining minimum vehicle requirements can be solved in polynomial time. An extension of this problem is the case in which each load has a time-window in which the transport should start. In the literature less attention has been paid to this type of problems. Firstly, an integer linear programming model is formulated. Alternatively, the problem can be solved by using a set partitioning approach.

Storage and Retrieval Machines (SRMs) take care of the storage and retrieval of loads in a storage area. Part II of this thesis starts with a literature review on sequencing storage and retrieval requests. From this review it can be concluded that the research question on scheduling storage and retrieval requests for a unit load storage and retrieval machine working in a number of aisles is still interesting to be studied. Each aisle in the storage area is assumed to have an input/output station at both ends where loads are transferred from the SRM to other types of equipments. A new algorithm is developed to obtain the smallest distance required for a unit load SRM to perform all storage and retrieval requests in one aisle. Furthermore, we developed a new dynamic programming algorithm to sequence requests for a single SRM in a number of parallel aisles with transfer stations at both ends of each aisle. This algorithm solves the problem in polynomial time.

In part III simulation studies are performed to test the new planning and control concepts developed in parts I and II in the context of a container terminal. Firstly, quantitative models from the literature are discussed concerning decision problems at container terminals. Two of these decision problems are discussed in more detail in the remaining chapters of this part. The algorithms of part I of this thesis for determining the minimum number of vehicle required are tested to gain insight in vehicle requirements at container terminals. Finally, the dynamic programming algorithm from part II is used to gain insight in travel distances of cranes executing storage and retrieval requests at container terminals.

1.2 Trends

Numerous trends potentially impact the performance of the supply chain. We discuss some of these trends briefly in this section.

According to Handfield and Nichols [102] only a small fraction of the lead time is used for executing real processes. The remainder of the time is spent on unproductive activities, like waiting. To remain competitive and to satisfy demands of customers the complete process should be executed effectively. Therefore, one of the activities in supply chain management is to make improvements in *lead time performance*. This increases the flexibility to react on changes in customer demand, it decreases the risk of obsolete stock and it reduces inventories.

To shorten the lead time of the supply chain, we can focus on the separate processes in the supply chain. The key processes to focus on are processes with the longest average throughput times and processes with the greatest variance in throughput times. Possible candidates for review are the storage and retrieval process at warehouses, planning and scheduling of materials, transportation of goods and the manufacturing processes. Important in this respect is also the just-in-time concept; for a customer it is of no importance how long it takes for a product to be created, stored, transported and delivered as long as it arrives exactly at the time she/he demands it.

One radical way to shorten the lead time is to design a supply network with a minimum of inventories. Products are then shipped directly or via cross-docking centres from the manufacturing centre to the customer. Lead times mainly decrease because products are no longer stored (i.e. waiting) in warehouses. To obtain a network with low inventories several things should be arranged well in the supply chain such as flexibility of the production, communication facilities between companies and reliability of production and distribution. A less rigorous way to shorten lead times is to reduce the number of stockkeeping points in the supply network. In the traditional form inventory is held at several places, like at the retailer and at the manufacturer. A reduction in the number of inventory points is possible, for example, by concentrating inventory at the manufacturer. As a result, retrieval from products at the warehouses of the distribution centre, transport to the warehouse of the retailer and storing the products in the warehouse of the retailer become superfluous activities, and savings in time and costs can be obtained.

Depending on the type of product and its market another possibility to reduce inventories might be to assemble products from standarised parts. The inventory consists of standard elements such that the final appearance of a product only has to be decided on after the order of a customer. The product is created after the wishes of the customer. This can be done at the plant or at the warehouse. This way of manufacturing products is already executed in the computer industry. A customer orders a computer and thereafter the computer is built with standard components present in the warehouse. As a consequence, products can be adapted to the wishes of customers by having, except for storage activities, value adding activities at warehouses. This process is indicated by the term *value added logistics*. As a result, a company can respond much faster to the orders of consumers and lead times of the supply chain, inventory and costs may be reduced. Brockmann [34] discusses value added logistics and other trends in warehousing.

A new category of products that are held in inventory are products that are returned by customers. Due to governmental rules, environmental laws and scarcity of raw materials, repairing and recycling of products, which are at the end of their life cycle, has become a necessity. In The Netherlands customers even have to pay a *disposal fee* when they buy products, like refrigerators. This fee is used for the recycling of used products. Furthermore, retailers have the obligation to accept such products at the end of the life cycle.

Used products have to be transformed into new products or products that can be thrown away without damaging the environment. Otherwise, products are repaired and sent back to the customer. Furthermore, it is getting more and more common to guarantee customers that products, that are ordered, for example, via the Internet and are not satisfactory, can be sent back. These products are also sent back to the warehouse. They need to be checked before they can be sold again. Consequently, *return flows* become more important. These return flows are managed by the receiver. Uncertainty occurs in return flows. To manage all return flows, organisations in the supply chain have to improve their logistics functions.

At banks, return flows are already common business for years. Customers bring cash money to the bank to deposit it. Furthermore, they withdraw money from the bank for daily expenditures. When deciding how much money to order from the central bank, banks take both flows of money into account. In Vis [279] a model is developed to decide how much money should be ordered at the central bank to meet a certain service level and to minimise associated costs taking into account both flows of money. For other businesses new logistics models can be developed to manage these increasing return flows. In Fleischmann [80] some new models for the handling of return flows are discussed.

In determining inventory levels, information is of great importance. To share information through the entire supply chain the use of *e-commerce* can offer a big advantage. E-commerce is concerned with activities related to sales of products using electronic techniques. *Electronic Data Interchange* (EDI), Internet and e-mail are examples of these kinds of techniques. EDI is one of the systems that can be used to enable faster and more reliable delivery of orders to warehouses and goods to the customers. With EDI standard messages on, for example, orders of customers, are exchanged between computers. All parts of the supply chain have to agree with the chosen standard forms. With EDI information can be exchanged, in contrast to paper based systems, within a few minutes. Due to this faster way of communicating and sharing information fewer errors occur. This results in a more effective inventory management and a better service. Besides EDI, the Internet is used more and more.

On the *Internet* suppliers can offer their products, like books, computers and food, against fixed prices. Another way, to buy or sell products on the Internet is to participate in an auction. At the Internet all aspects of the traditional purchase process can be executed. The physical flows are the transport and delivery of the item to the customer and potentially in a later stage the transport and delivery of some spare parts and return flows (see Hagdorn-van der Meijden and Van Nunen [98]).

For retailers selling products in shops and via the Internet one of the consequences is the fact that the size of orders becomes smaller while the number of orders grow simultaneously. As a result, the order processing, the inventory management, the layout of the warehouse and so on, may have to be changed for efficiency. The delivery times should become smaller and more flexible. The supply chain is directed straight by customers. Furthermore, 'power relations' in the supply chain will shift. For example, the power of the manufacturer increases and the power of the retailer decreases if products are delivered directly to the customers by the manufacturer.

By using e-commerce, supply chains can become more competitive. For example, costs can be reduced and information can easily be distributed through the supply chain (see Fraser *et al.* [82]).

Instead of performing all processes in the own organisation, it can be decided to *outsource* some of the activities. Each firm should decide whether all activities can be executed or that only key activities should be performed by the firm itself. The remaining activities can be outsourced to third parties. The relationship between firms and third party logistics providers features mutual trust, respect and openness. This is necessary to obtain strategic competitive advantages. Warehousing and transport are two of the main logistics activities, that can be outsourced. In 1996, 50% of all transport activities in The Netherlands was executed by third party logistics providers. Furthermore, 15% of the warehousing activities was outsourced (see Van der Baan [270]).

Advantages of outsourcing side activities are, for example, reduction of costs, better service, larger flexibility and efficiency and less investments in logistics and skilled staff. On the other hand, the direct contact between the firm and the customer disappears. Furthermore, confidential information has to be shared with the logistics provider. This might be a disadvantage, if the provider also works for the competitor.

To ensure that the supply chain is managed efficiently, the outsourcing process should be well implemented. The third party logistics provider should operate effectively as part of the supply chain. Nowadays, *fourth party logistics providers* exist as well. They tune third party logistics providers to each other.

Supply chains are no longer concentrated in one country, but they are spreading out all over the world and flows of material are crossing borders. Supply chain managers nowadays have to deal with the uncertainty and complexity of *global networks*. Numerous differences exist between national and global supply networks. We will discuss a few of them (see Dornier *et al.* [59]).

In global networks geographic distances will be larger and time differences will occur. Firms are dealing with this problem by keeping larger inventories. Delivery times will be larger. A larger variance in delivery times will also occur due to unexpected delays, like customs regulations. Firms face a great challenge in implementing just-in-time production while suppliers are at a great distance.

Secondly, forecasting demand in foreign countries is more difficult than in the home country. Firms are operating in environments with different languages, different cultures and habits. As a result, large safety stocks are held. Another influence on the performance of the global supply chain is the effect varying exchange rates. They influence costs, prices and the amount of products sold.

If firms are starting operations abroad, they have to deal with problems concerning infrastructure. In, for example, developing countries there is a lack of sources, like transportation networks, telecommunication networks, skilled employees and materials. Furthermore, firms should be aware of (changes in) foreign regulations.

As a result, supply chain managers should be aware of local regulations, conditions, habits and so on when implementing a global network. Still, in global networks it is important to reduce inventories, to reduce the lead time of supply chains and to pay attention to all other trends mentioned.

More detailed information on trends impacting the performance of national and international supply chains can be found in, for example, Carter *et al.* [36]. In that paper trends are examined and forecasts are made for the behaviour of these trends in supply chains in 2008. In Vis and Roodbergen [285] six supply chains from real life are studied and it is examined whether the lead time of each supply chain can be reduced. Furthermore, some of the trends discussed so far can be observed in the supply chains studied.

To respond efficiently and effectively to these and other trends, adaptions in several of the nodes and their related processes in supply chain networks might be required.

Therefore, we discuss several of these nodes in supply chain networks in more detail in the next section.

1.3 Nodes in the supply chain network

Organisations in a supply chain can be considered as nodes in a network. Between these nodes logistics activities, like transport of products occur. in nodes, logistics activities are also performed. Examples of these kinds of activities are the internal transport of products and the storage and retrieval of products. From the view point of material flow through the supply chain, material handling is highly interrelated with logistics. According to the Material Handling Industry of America *material handling* is the movement, storage, control and protection of material, goods and products throughout the process of manufacturing, distribution, consumption and disposal. The focus is on the methods, mechanical equipments, systems and related controls used to achieve these functions.

The movement, storage, control and protection of materials occurs in and between the various nodes of a supply chain. We refer to these nodes as *material handling centres*. Examples of these centres are warehouses, cross-docking centres, terminals and manufacturing centres. More detailed descriptions of these centres are given in Section 1.3.1. According to Tompkins *et al.* [265] a considerable amount of the total costs of a product consist of material handling costs. Consequently, to reduce costs in supply chains, managers can focus on material handling centres and their corresponding activities and equipments.

We can distinguish between four planning and control levels in making decisions to obtain efficient processes and operations in material handling centres, namely the strategic level, the tactical level, the operational level and the real time level. At the strategic level it is, for example, decided how many centres are used and at which locations. The time-horizon of decisions at this level covers one to several years. These decisions lead to the definition of a set of constraints under which the decisions at the tactical and operational level have to be made. At the tactical level, it is, for example, decided which types of equipments are used and which centre layout is chosen. The time-horizon of these decisions covers a week to a year. At the operational level decisions are made of which the time-horizon covers a day to a week. One of these decisions is the determination of the number of vehicles required to ensure an efficient transportation process. In Chapters 3 and 4 methods are developed to deal with this problem. Another problem is the determination of the sequence in which storage and retrieval requests should be handled by a storage and retrieval system such that travel distances are minimised. This problem is studied in Chapter 6. Finally, at the real-time level, real-time decisions are made each minute on, for example, the operation of the vehicle or crane.

Logistic activities that are potential candidates for improvement in material handling centres are among other things: receiving and shipment of materials, internal transport of materials, storage and retrieval of materials. To perform these activities material handling equipments, like internal transport equipments and storage and retrieval systems can be used. Due to technological innovations, new types of equipments have been developed during the past decades, namely automated material handling equipment. Without human intervention, automated cranes and vehicles perform their tasks. Automated control technology is used for controlling automated material handling equipment. Automated material handling equipment demands high investments but little labour costs. For manned equipment, labour costs form a high percentage of the total costs. The difficulty of operational control problems can increase if the level of automation increases. Besides automated control technologies, information technology can be used in material handling centres, for example, for recognition of pallets, products and containers.

The choice for a certain type of equipment should be made by performing a feasibility and economic analysis on various types of equipments. Fisher *et al.* [76] discuss heuristic rules for selecting equipment types which are both technically and economically appropriate for the movement of specified loads over specified types of paths. Furthermore, the choice of equipment depends on the design of the material handling centre. Wide vehicles cannot travel through narrow lanes and the height of the vehicles should be lower than the height of the building in which internal transport occurs. The determination of the layout and the types of equipments used are interrelated issues. For example, Welgama and Gibson [290] propose an integrated methodology for determining both aspects simultaneously.

Consequently, the design of a new facility or improvement of the design of an existing facility is also important in this context. Firstly, it should be decided at which location(s) the material handling centre(s) will be built. In this context, it is important to have information about the geographical locations of suppliers and customers, the types of equipments used for the transport of incoming and outgoing goods, the situation on the local labour market and so on. Taking into account the answers to these questions, the number of facilities and their locations can be determined by applying analytical and heuristic methods.

The performance of the centre and equipment highly depends on its design. Many aspects should be taken into account when designing a material handling centre or improving an existing one. According to Tompkins *et al.* [265], the basic questions of why, what, where, when, how, who and which should be asked constantly. Examples of these questions are: Why is material received and shipped as it is? What is to be moved and what should be done manually or automatically? Where is material handling required? When should material be moved? How should it be moved? Who should be handling material? and Which operations are necessary? By designing a material handling centre costs and travel distances should be minimised and delays and congestion should be avoided by making efficient use of space, staff and equipment.

Material handling systems consist of the material handling equipments, the personnel, information, materials and the related planning and control systems. Manda and Palekar [184] present an overview of literature concerning the design and control of material handling systems. To deal with current trends (see Section 1.2) the performance of material handling centres and the supply chain need to be improved. To obtain efficient material handling centres it might be required to improve the efficiency of material handling systems. Therefore, new planning and control concepts for material handling systems can be developed. Matson and White [186] indicate that the use of Operations Research is useful for problems arising in material handling systems. In this thesis, new planning and control concepts are developed by using techniques from Operations Research. For a better understanding of the various environments in which the methods and algorithms developed in this thesis can be used, we discuss four material handling centres in more detail. The material handling centres we consider are warehouses, cross-docking centres, containers terminals and manufacturing centres. For each material handling centre we discuss the centre itself and its logistics activities. Thereafter, various types of material handling equipments used in these different centres are treated. Finally, related planning and control problems are studied.

1.3.1 Material handling centres

We discuss four different types of material handling centres in more detail, namely warehouses, cross-docking centres, container terminals and manufacturing centres.

Warehouses

In supply chains *warehouses* are used to store inventories of raw materials, semi-finished products and finished products. Two types of warehouses can be distinguished, namely *distribution centres* and *production warehouses.* in distribution centres, products from several suppliers are stored before delivery to customers. Products and materials in a production environment are stored in production warehouses. Inventories are held to respond quickly to fluctuations in demand and production and to ensure same day delivery. Furthermore, in warehouses products from several producers can be combined for delivery to joint customers. The finishing touch to products is also performed in warehouses. Products are made customer specific by adding labels and so on.

A warehouse consists of various areas in which different kinds of logistics activities are performed. A short impression of these areas and related activities will be given. In the *receiving and control area* arriving trucks are emptied and arrived products are checked on amount and quality. Products can be transported directly to the *shipping area* (cross-docking) or they can be stored in the *storage area*. Products can be stored on, for example, pallets. These pallets are stored on the ground, on top of each other, or in racks. Racks are placed in rows with aisles in between. The storage area is divided in a number of blocks with subaisles. Cross aisles are positioned at the front and back of the storage area and between blocks. They are used to change aisles. Figure 1.2 represents an example of a storage area consisting of one block. A similar kind of layout is used in Chapter 6 of this thesis. Each square indicates a storage location. Aisles can be changed at the front and back of the storage area.

After a certain period of time products are retrieved from storage such that they can be transported to customers. This process is called *order picking*. Complete pallets, cases or just individual items are collected. Order pickers walk through aisles to collect the items from the order. To pick pallets storage and retrieval machines can be used, which travel through aisles to collect orders.

Commonly, a number of individual items (which locations are indicated by black squares in Figure 1.2) are retrieved by an order picker in a route starting and ending at the depot. The time required to pick an order consisting of a number of items consists of several parts, namely walking or driving between indicated locations, collecting products, collecting information about locations and returning the full vehicle to the location agreed upon. Pallets are usually retrieved by a unit load system. In this case a storage request is in practice usually combined with a retrieval request in one trip of the system. The pallet is stored at its location. Thereafter, the storage and retrieval machine travels to the location a pallet need to be retrieved. The trip ends the moment the pallet is delivered to its destination.

It might occur that the collection of products have to be regrouped or split after the orderpicking process such that they match with the orders. This is done in a so-called *accumulation and sorting area.* in the *shipping area*, the orders are loaded into

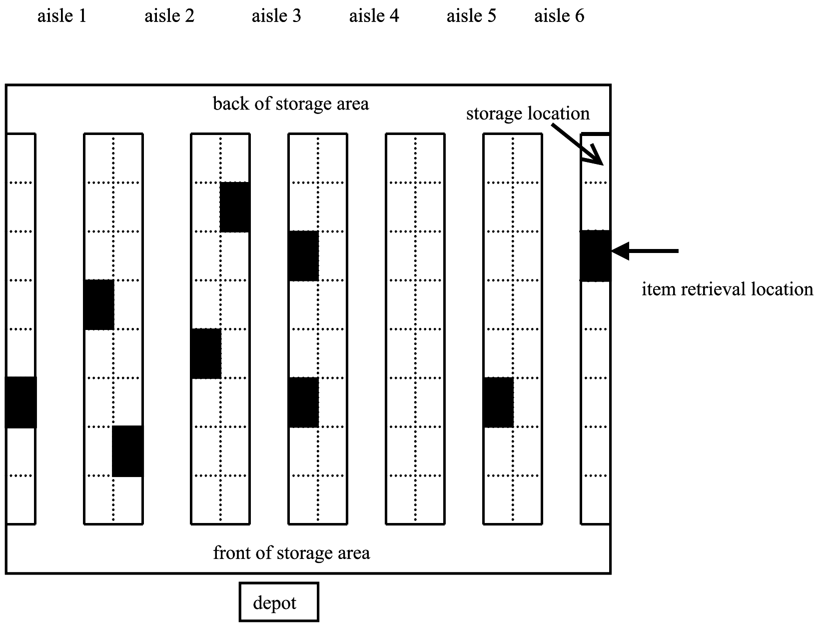

FIGURE 1.2. Example of a layout of a storage area consisting of one block of aisles. Black squares indicate locations at which item retrievals should be performed.

trucks, such that they can be delivered to customers. The products are transported between these areas in the warehouse by internal transport systems.

Cross-docking centres

By introducing cross-docking the two most expensive processes in warehouses, storage and orderpicking of goods, can be eliminated. *Cross-docking* is defined as follows: transporting goods from the receiving area to the shipping area with as little time as possible inbetween, and without storage.

Cross-docking exists in different forms and at different places in the supply chain, namely in manufacturing centres and distribution centres. In *current cross-docking* finished goods flow directly from the receiving area into the truck. In *future cross-docking*, finished products are temporarily buffered and thereafter they are loaded into the truck (see also Andel [3]). At distribution centres parcels of goods arrive by trucks and they are transshipped further by trucks in the same composition or in smaller parts (see Figure 1.3).

Not all products are suited for delivery to customers without being held in inventory. According to Richardson [221], cross-docking can be used for products with short delivery times, products with high demands and products with highly predictable demands. Examples of these products are food and mail. Furthermore, the possibility of applying cross-docking also depends on the delivery speed of the suppliers. To respond fast to customer orders without having products in inventory, products should be delivered fast by the supplier. To implement cross-docking successfully, attention could be paid at least to the following factors (see also Moore and Roy [193] and Schaffer [232]). Firstly, the relationship with suppliers and all partners in the supply chain has to be reliable. Products should be delivered at the right time and in the right quantity. Otherwise, trucks will travel too late or half-empty to customers. Secondly, information should be available through the entire supply chain. Arrival times, departure times and destinations have to be available in time such that the planning of trucks can be done in advance. Finally, to ensure that products arrive and depart on time, trucks should arrive on time. Transport should be reliable to prevent delays in delivery to customers. Except for the external factors mentioned, cross-docking can only be successful if the design of the cross-docking centre and the operating polices for all processes are executed efficiently. Potential advantages of cross-docking are, for example, reduction of inventories and associated costs, shorter lead times, improvement of service to customers, improvement of relations with suppliers and the possibility to take real-time decisions. Schaffer [233] discusses in detail how cross-docking can improve efficiency.

When an incoming truck arrives at a cross-docking centre, it is assigned to an open door or it is sent to a queue. Workers unload the truck and deliver the incoming products to the stack door designated to receive these specific loads. The unloading continues until the truck is empty. The truck is driven away and a new truck arrives to be unloaded. At the doors designated for leaving trucks, outgoing products are stored temporarily or loaded directly into the truck. Temporary storage occurs if the centre is too busy or if the specific truck is not there. Figure 1.3 gives an illustration of a possible layout of a cross-docking centre.

Container terminals

Containers are large boxes used to transport goods from one destination to another. With containers a bulk unit can be created out of the individual pieces of freight. As a result, *containerisation* can be defined, according to *The Containerization Institute*, as the utilisation, grouping or consolidating of multiple units into a larger container for more efficient movement. Compared to conventional bulk, the use of containers has several advantages, namely less product packaging, less damaging and higher productivity (Agerschou *et al.* [1]). The dimensions of containers have been

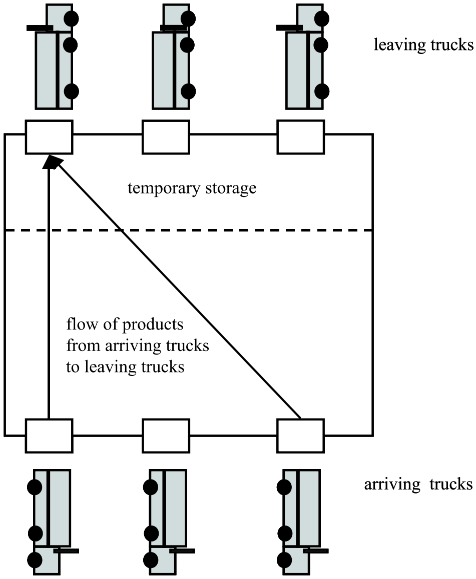

FIGURE 1.3. Example of a layout of a cross-docking centre.

standarised. The term *TEU* (twenty-feet-equivalent-unit) is used to refer to one container with a length of twenty feet. A container of 40 feet is expressed by 2 TEU. Several transportation systems can be used to transport containers from one destination to another. Transport over sea is carried out by ships. On the other hand, trucks or trains can be used to transport containers over land. To transship containers from one mode of transportation to another, ports and terminals can be used. For example, at a container terminal, a container can be taken off a train and placed on a ship.

Containers were used for the first time in the mid-fifties. Through the years, the proportion of cargo handled with containers has steadily increased. As a result of the enormous growth, the capacity of ships has been extended from 400 TEU to 4000 TEU and more. An extensive overview of the history of containers is given in Rath [217]. Furthermore, the importance of ports and terminals has grown. With the introduction of larger ships, small terminals have changed into large terminals. To ensure a fast transshipment process at large terminals information technology and automated control technology can be used. A detailed description of the use of these technologies in container terminals can be found in Johansen [116]. To use these kinds of technologies large investments have to be made and ongoing database management is required. Wan *et al.* [287] show that the application of information technology in the port of Singapore results in more efficiency and a higher performance. In Leeper [169] it is concluded that, in order to achieve an improvement of productivity and reduction in investment costs, an advanced automated control technology is a necessary condition.

The process of unloading and loading a ship at a container terminal is illustrated in Figure 1.4 and may be described as follows: when a ship arrives at the port, the containers have to be taken off the ship. This is done by manned Quay Cranes (QCs), which take the containers from the ship's hold and the deck. Next, the QCs put the containers on vehicles, like automated guided vehicles (AGVs). After receiving a container, the AGV moves to the stack. This stack consists of a number of lanes where containers can be stored for a certain period. These lanes are served by, for example, automatically controlled Automated Stacking Cranes (ASCs). When an AGV arrives at a lane, the ASC takes the container off the AGV and stores it in the stack. After a certain period the containers are retrieved from the stack by the ASCs and transported by the AGVs to transportation modes such as barges, deep-sea ships, trucks or trains. This process is also be executed in reverse order, to load containers on a ship.

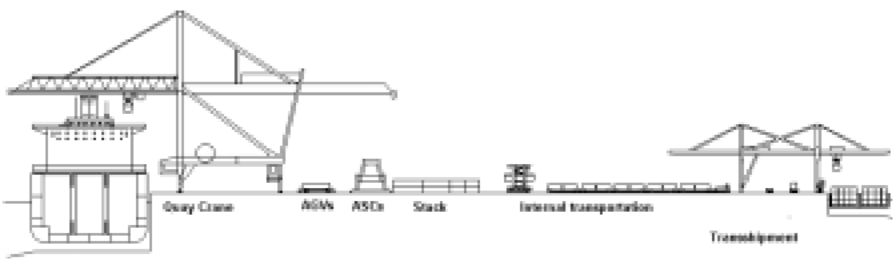

FIGURE 1.4. Logistics activities at a container terminal during the unloading and loading of a ship.

Manufacturing centres

Manufacturing centres are the parts of the supply chain at which products are assembled. According to Askin and Standridge [8] five interrelated functions can be distinguished in the manufacturing process, namely product design, process planning, production operations, material flow and production planning and control.

In *designing* the product much attention is paid to the wishes of customers, such that profitable products are created. In a specified order of operations the product is produced. The *process planning* indicates which instructions should be followed to manufacture the product from its raw materials. The *product operations* to manufacture the product are executed at assembly lines and machines. *Materials flow* from assembly line to assembly line through the material handling centre. To obtain an efficient manufacturing process attention can be paid to the layout of the manufacturing centre and the material handling systems used in the centre. To translate demands of customers, production capacity, inventory levels and planned production levels to the number of products to be produced various production plans can be used. (see Waller [286]).

In a *capacity requirements plan* prognoses of future needs of finished products are translated into the number of materials, labour hours and machine hours required to achieve the number of products. Costs and capacities of the firm are taken into account. In a *master production schedule* an indication for a number of weeks/months is given of what end terms, like number of finished products directly shipped to the customer and number of products kept in inventory, need to be completed. In a *material requirements plan* it is indicated which materials and in which quantity should be ordered or produced internally taking into account production times, delivery times and inventory levels.

1.3.2 Internal material handling equipment

Material handling equipment is used to perform logistics activities in material handling centres. From Section 1.3.1 it is clear that similar logistics processes occur in most material handling centres. These processes are, for example, the storage and retrieval of products from and into the storage area and the transportation of products through the centre. We will discuss some of the equipment used to perform these kinds of logistics activities. A more detailed description can be found in Tompkins *et al.* [265] and Frazelle [83]. Due to technological changes equipment is improving continuously.

The process of storage and retrieval of pallets or products from storage can be performed by various types of equipments. Some of these types of equipments are manned while others are automated. Types of equipments exist with which the order picker drives/walks to the products and collects them. Secondly, types of equipments exist which transport the products to the order picker. *Pedestrian stackers* and *counterbalance lift trucks* belong, for example, to the first type of equipment. A pedestrian stacker can lift and stack a pallet. The order picker walks behind the pedestrian stacker. A counterbalance lift truck is able to lift a pallet. The order picker drives with the vehicle to the right location and the pallet is lifted with the fork, which can go up and down. To use counterbalance lift trucks at warehouses, wide aisles are required, which allow the counterbalance lift truck to turn. An advanced system for handling pallets is an *automated storage and retrieval system (AS/RS)*. An *AS/RS* consist of racks handled by cranes travelling on rails and a control system. Cranes can move horizontally and vertically simultaneously. The aisles in which cranes are operating are very narrow. As a result, much storage space can be saved. However, the costs of the AS/RS are high. A more elaborate description of automated storage and retrieval systems is given in Chapter 5.

Instead of picking pallets, cases can also be stored and retrieved. Operators can use an *orderpicking truck* to go up and down to pick/store a case from/in a storage location. *Picking carts* can be used to pick individual items.

Carrousels belong to the category of equipment which transports the product to the order picker. A carrousel consists of horizontally/vertically rotating selves which can be controlled manually or automatically. The speed of orderpicking depends among other things on the rotation speed of the carrousel.

Due to the specific properties of containers, like size and weight, the most commonly used systems cannot be used to transport and store containers. *Straddle carriers*, *automated stack cranes* and *yard cranes* might be used for the storage and retrieval of containers in and from the stack. A straddle carrier can travel over a single row of containers. Stack and yard cranes move over a number of rows. *Quay cranes* perform the unloading and the loading of containers from and in the ship. Due to practical problem, like the exact positioning of containers, this system cannot be automatised. In Chapter 7 the types of equipments used in a container terminal are discussed more elaborately.

The products often have to be transported from one area to another area in material handling centres. Some of the already mentioned types of equipments, like counterbalance lift trucks can be used to take care of the internal transport of products. *Conveyors* can also be used. The products are transferred to specific locations over a fixed path. The products move on rollers, chains and so on. *Automated guided vehicles (AGVs)* transport products over fixed paths. For more information on AGVs see Chapter 2. *Automated guided vehicles*, *yard trucks* and *straddle carriers* can be used to transport single containers at a container terminal. Contrary to the two other systems, a straddle carrier is capable of lifting a container from the ground by itself instead of receiving it from a crane. To transport multiple containers simultaneously, *multi trailer systems* can be used. The vehicles used at a container terminal are discussed in more detail in Chapter 7.

1.3.3 Planning and control problems

Several planning and control decisions have to be made to ensure that the various logistics activities in material handling centres are performed efficiently. We will discuss some of these planning and control problems in this section.

One of the planning problems to be solved is the design of the centre and the choice of layout of various areas in material handling centres. In Schwind [236] several factors influencing the design of cross-docking centres are discussed. By designing a cross-docking centre, information on which materials and their quantity will be handled should be available. Furthermore, it should be known in advance how long materials will stay at the cross-docking centre. With answers to these questions, it could be decided how products are stored, for example, on the ground or in racks, and how large the storage area should be. The geometry of the centre determines the number of doors, location of the doors and the travel distances inbetween. In designing the layout of the centre doors need to be specified as doors for arriving and doors for leaving trucks, such that travel distances are minimised. Furthermore, destinations should be assigned to doors for leaving trucks. In practice, layouts are determined based on intuiton. Bartholdi and Gue [15] also discuss the problem of determining a layout of a cross-docking centre by taking into account travel time costs, waiting times and floor space congestion.

In Roodbergen [224], methods are given to determine the layout of an order picking area at warehouses. The objective is to develop a layout such that the average distance travelled by an order picker is minimised. In determining the layout of the storage area in a container terminal (i.e. stack), decision have to be made concerning, for example, the height of stacking of containers and the length of rows of containers. A review of the literature, concerning the stack layout and other planning problems in a container terminal, is given in Chapter 7.

Also, the number of vehicles required to transport all jobs between the various areas in a material handling centre within time should be determined. In Chapter 3 and 4 methods are given, which can be applied to determine the minimum number of vehicles required to transport loads at known release times or before specified due times. In Chapter 8 it is shown in which way the theoretical models from Chapters 3 and 4 can be applied to determine vehicle requirements at a container terminal.

Processes in material handling centres are controlled by operating policies. Rules for decisions such as where are incoming loads stored, in which sequence are storage and retrieval requests handled and which vehicle transports which load have to be determined. The performance of operating policies in material handling centres depends on its own efficiency, the layout of the centre, the performance of equipment, order data and so on. In Van den Berg [268] an overview of operating policies at warehouses is given.

Numerous ways of assigning products to storage locations exist. Products can be stored randomly over the storage area or at the closest open location or at a fixed location. Furthermore, products can be distributed over the storage area according to their turnover. In class-based storage the above-mentioned methods are combined. Products are divided into a number of classes depending on their demand frequency. Each class has a dedicated area at the warehouse in which the products are randomly stored.

To reduce lead times of processes in material handling centres, for example, order picking time can decrease. Routing policies can contribute to this. The problem is to determine a sequence of storage locations at which items need to be picked, by minimising total travel time. This problem is a special version of the travelling salesman problem. For warehouses with one or two blocks shortest routes can be determined (see, for example, Ratliff and Rosenthal [218]). For other layouts, heuristics should be applied. In Chapter 5 some of these heuristics and algorithms and related literature are discussed in more detail.

In the case that unit load storage and retrieval machines are used, it might be efficient to combine a storage and retrieval request in a route. Policies for sequencing unit load storage and retrieval requests are studied in detail in Chapter 5. In Chapter 6 a method is developed to schedule for a storage and retrieval machine a fixed number of storage and retrieval requests in a warehouse. The storage area consists in this case of one block with multiple aisles. In this block storage requests are available for storage at the front and back end of the warehouse. Retrieval requests need to be delivered to one of these ends. In Chapter 9 the methods of Chapter 6 are applied on the sequencing problem of storing and retrieving containers from and in the stack for automated stack cranes and straddle carriers.

As explained, internal transport occurs in material handling centres to transport goods from one area to another. Job should be assigned vehicles in an efficient way. This problem is closely related to the vehicle routing problem. Van der Meer [271] considers the control of guided vehicles in vehicle based systems. Different vehicle dispatching rules are tested in different environments. Furthermore, it should be decided which route is chosen to ensure efficient transport. Chapter 2 of this thesis gives an overview of operating policies and related literature for automated guided vehicle systems.

The assignment of trucks, to be unloaded, to doors is an operational problem at cross-docking centres. In practice, trucks are usually assigned to doors according to the first-come-first-served principle. Consequently, trucks are not assigned to doors by taking into account the destination of the freight inside the truck. In Gue [96] a look ahead scheduling method is presented to assign arriving trucks to doors. Freight is allocated to doors in an optimal way based on their locations and the locations of nearby doors designated for leaving trucks. It is shown that reductions between 15 and 20% in costs can be obtained by implementing this kind of scheduling.

1.4 Outline of the thesis

The performance of a supply chain is influenced by several trends, including lead time reduction, reverse flows and e-commerce. To deal with these trends and to ensure that the supply chain remains or becomes efficient and effective, adaptions in logistics activities might be required. Logistics activities occur between and in the various nodes of a supply chain. Material handling centres, like warehouses and terminals, function as nodes in a supply chain network.

To perform various logistics activities in material handling centres, like storing and internal transport of products, material handling equipment can be used. Planning and control systems are related to the equipment. In this thesis two different types of material handling equipments, namely automated guided vehicles and storage and retrieval systems, are examined.

Improvements in logistics activities in material handling centres and improvements in the performance of material handling systems might result in improvements in the overall performance of the supply chain. The objective of this thesis is, therefore, to develop new planning and control concepts for material handling systems.

In part I of this thesis, automated guided vehicles are studied. Firstly, in Chapter 2 a review of literature on the design and on related problems, of automated guided vehicle systems is given. Chapters 3 and 4 address the problem of the determination of the number of vehicles required to transport all loads in time. In Chapter 3, the deterministic case in which all loads have to be transported exactly at the moment they are available for transport is studied. A model and minimum flow algorithm are presented to solve the problem in polynomial time. This chapter is partly based on [282]. In Chapter 4, the deterministic case in which all loads have to be transported within their time-windows is examined. Two ways to model this situation are presented. Firstly, an integer linear programming model is formulated. Secondly, a formulation based on a set partitioning approach is given. This chapter is partly based on our research in [280].

In part II of this thesis, storage and retrieval systems are treated. Firstly, in Chapter 5, literature on scheduling of storage and retrieval requests is discussed. Literature on sequencing storage and retrieval requests at automated storage and retrieval systems is examined. Furthermore, some literature on the routing of orderpickers in a warehouse is discussed. In Chapter 6 a scheduling policy for a unit load AS/RS working in one block with multiple aisles is developed. Storage and retrieval requests, with their origin and destination at one of the sides of the block, are available to be handled within a number of aisles. Using a dynamic programming algorithm, the optimal order in which requests should be handled within an aisle, the optimal way in which each aisle should be traversed and the optimal order in which aisles should be visited, is determined in polynomial time.

To give an impression how the models and algorithms developed in part I and part II can be used in practice, a material handling centre, namely a container terminal, is examined in part III of this thesis. In Chapter 7 an overview of all processes within a container terminal is presented. A review of literature on various decision problems is given. This chapter is partly based on [281]. The methods from part I are used in Chapter 8 to give an impression how many automated guided vehicles and how many automated lifting vehicles are required to transport all containers in time. Furthermore, it is studied which variables, like cycle times of quay cranes, impact the number of vehicles required. This chapter is partly based on [283] and [284]. In Chapter 9, the methods for sequencing storage and retrieval requests from part II are applied to an automated stacking crane and a straddle carrier. The route lengths of both systems and several variables impacting these route lengths are studied. In Chapter 10 conclusions and suggestions for further research are presented. Appendix C gives a list with abbreviations used in this thesis.

Part I

Automated Guided Vehicles

2

Automated guided vehicle systems

An Automated Guided Vehicle (AGV) is a driverless internal transport system used for horizontal internal movement of materials in material handling centres. AGVs were introduced in 1955 (Müller [194]). The use of AGVs has grown enormously since their introduction. The number of areas of application and variation in types has increased significantly. The choice for using AGVs instead of other material handling equipment is based on economical and technical considerations. Müller [194] gives an extended overview of types of AGVs, reasons for implementation, areas of application and experiences with AGVs in practice.

In material handling centres, a number of AGVs are operating to transport jobs from one location to another. These AGVs belong to the *Automated Guided Vehicle System* (AGV system). In an AGV system several parts can be distinguished, namely the *vehicles*, the *transportation network*, the *physical interface between the production/storage system* and the *control system*. The transportation network connects all stationary installations (e.g. machines) in the centre. At stations, *pick-up and delivery points* are installed that operate as interfaces between the production/storage system and the transportation system of the centre. At these points a load is transferred by, for example, a conveyor from the station to the AGV and vice versa. AGVs travel from one pick-up and delivery point to another on fixed or free paths. Guidepaths are determined by, for example, wires in the ground or markings on the floor. More recent technologies allow AGVs to operate without physical guidepaths. We refer to this kind of AGVs as free-ranging AGVs.

AGVs are capable of transporting one or more loads at the same time. The size of the *unit load* has to be decided on by the management. This indicates a number of items arranged in such a way that they can be transported as a single object. An example of a unit load is, for example, a container (see Chapter 7). Clearly, the larger the size of the unit load, the lower the transportation costs per individual item and the lower the number of vehicles required in the system. Methods to determine the size of the unit load are given in [68] and [192]. Secondly, it has to be determined if *one-load-carrying* or *multiple-load-carrying AGVs* will be used in the system. Ozden [203] indicates with a simulation study that by transporting two loads at the same time instead of one, reductions in the vehicle fleet size can be obtained. Van der Meer [271] shows that increasing the capacity of a vehicle results in a reduction of the average throughput time. However, in literature, one-load-carrying AGVs are mainly discussed.

The transportation process in which one-load-carrying AGVs are used, might consist of the following steps. The load demands for transportation at the pick-up and delivery point of its origin. The load is available for transport at a certain time (i.e. release time). Sometimes, also a due time is defined to ensure that the load is transported at the latest at this time-point. An available AGV is assigned to this load (see Section 2.7) to transport it. At the origin of the load, the load is placed on the AGV. Thereafter, a route from origin to destination has to be determined via which the AGV travels to transport the load. Furthermore, times of arrival and departure have to be determined (see Section 2.8). At the destination of the load, the AGV is unloaded and ready to receive a new assignment. Clearly, many decisions have to be made for each transport of a load. Therefore, a high level of *control* is required in an AGV system for efficient routing, scheduling and dispatching of vehicles and deadlock avoidance (see Section 2.6).

In designing an AGV system many decisions have to be executed. Some of these decision variables are discussed in the remainder of this chapter. Furthermore, methods to solve these kinds of decision problems are discussed by giving a literature review. For other reviews of literature on AGV systems we refer to Johnson and Brandeau [118] (stochastic modeling of AGV systems), King and Wilson [144] (design and scheduling), Ganesharajah *et al.* [86] (design and operational issues), Ganesharajah and Sriskandarajah [87] (scheduling) and Co and Tanchoco [46] (dispatching, routing and scheduling). Finally, in Section 2.10, it is discussed which problems are studied in this part of the thesis.

2.1 Design of an AGV system

In design problems many decision variables arise. The impact of decisions on mutual interactions and performance might be difficult to predict. It might be hard to decide on one thing without considering other decision variables. At least the following tactical and operational issues have to be addressed in designing an AGV system (see, for example, [182]):

- flowpath layout,
- traffic management: avoidance of collisions and deadlocks,
- number and location of pick-up and delivery points,
- vehicle requirements,
- vehicle dispatching,
- vehicle routing,
- vehicle scheduling,
- positioning of idle vehicles,
- battery management,
- failure management.

A flowpath layout compromises the fixed guided paths on which vehicles can travel to the various pick-up and delivery points of loads. Traffic management is required to avoid collisions and deadlock situations in which two or more vehicles are blocked completely. To ensure that loads are transported in time, sufficient vehicles should be available and the right vehicle should be dispatched to the right load. Furthermore, routes for vehicles to transport loads from their origins to their destinations have to be determined. Based on this routing information, scheduling decisions on, for example, the sequence in which jobs should be handled by a vehicle, can be made. After transporting a job, an idle vehicle has to wait on a new assignment at a certain position in the system. If AGVs use batteries, frequent battery changing might be required. The time required for replacing or charging batteries can impact the number of vehicles required. Ebben [62] develops control rules to take battery constraints into account in the system. In most literature the impact of equipment failures on the system is neglected. In the case that only few AGVs are used failures will have little effect on the occurrence of congestion in the system and as a result on the performance of the system. For systems with large number of AGVs, failures may occur more often. Ebben [62] developed control methods to deal with AGV failures. More information on these major issues of AGV systems is given in Koff [148].

To address some of these problems simultaneously, simulation models (see, for example, [143] or combinations of simulation and analytical models might be used (see, for example, [178] and [182]). With numerous assumptions the design problems can be simplified and formulated as an analytical model. For example, Johnson and Brandeau [117] suggest to formulate the problem of designing an AGV system as a binary integer programming model and to solve it with enumeration algorithms. Johnson and Brandeau [119] incorporate inventory decisions in the design of an AGV system. According to the authors a trade-off exists between the amount of inventory and the number of vehicles required to provide adequate service. It is shown that by integrating inventory and design issues significant reductions in overall production costs can be obtained.

Analytical models often address one or two of these design problems. For example, the models in Chapter 3 and 4 deal with the problem of the determination of vehicle requirements. Literature on the separate decision variables will be discussed in the remaining sections of this chapter. In analytical models layout problems and control problems, which are highly interrelated, are often separated. By doing this, attention should still be paid to the overall performance of the system. A well developed layout with a bad control rule might result in a decrease of the performance of the system.

To measure the performance of the system various criteria can be used. For example, Bozer *et al.* [27] propose an analytical model to estimate the expected waiting times of loads to be transported. Nakano and Ohno [196] give a decomposition algorithm to evaluate the performance of an AGV system measured in terms of the average utilisation of machines in the system. According to the literature discussed in this chapter some of the objectives of an AGV system are:

- maximise throughput of the system (i.e. number of loads handled per time unit),
- minimise time required to complete all jobs (i.e. makespan),
- minimise vehicle travel times (empty or/and loaded),
- evenly distribute workload over AGVs,
- minimise total costs of movement,

- minimise time job is handled after its due time (i.e. tardiness),
- minimise maximum or average throughput times of AGVs to travel to the destination of new jobs,
- minimise expected waiting times of loads.

For example, in Chapter 3 the minimum number of vehicles required is determined such that each job is transported at its release time. In the next sections, we will discuss literature concerning some of the mentioned tactical and operational problems. For the literature discussed, it is mentioned which objective(s) of the system has to be met.

2.2 Design of a flowpath layout

AGVs can travel along fixed guidepaths, which are indicated by, for example, wires in the ground. A flowpath layout connects machines, processing centres, stations and other fixed structures along aisles. This layout is usually represented by a directed network in which aisles intersections and pick up and delivery locations can be considered as nodes. The arcs represent the guidepath the AGVs can travel on. Directed arcs indicate the direction of travel of vehicles in the system. The layout of this flowpath influences directly the performance of the system. For example, it impacts the travel time to transport a load from its origin to its destination, the number of vehicles required (see Section 2.5) and the degree of congestion (see Section 2.3).

The layout of the flowpath can be designed in various ways. Firstly, the facility layout, the layout of the flowpath and the location of pick-up and delivery points can be determined simultaneously. Secondly, the design of the flowpath and the location of pick-up and delivery points can be determined by considering the layout of the facility as an input factor. Finally, the flowpath can be designed, considering the layout of the facility and the location of pick-up and delivery points as input factors.

Using the information on the layout of the facility and the number and location of aisles and pick-up and delivery points a fully connected network consisting of arcs and nodes can be created. Nodes represent corners of aisles, intersections between aisles and pick-up and delivery points. In this network each node is connected with any other node and the complete path between two nodes can be traversed by a load without changing of vehicles. Directed arcs indicate possible directions of travel through the aisles. The direction of travel along these arcs can be *unidirectional* and *bidirectional*. If vehicles are allowed to travel in only one direction arcs are unidirectional. On the other hand, if vehicles are allowed to travel in both directions, bidirectional flow on arcs might occur. By using bidirectional traffic flows, instead of unidirectional flows, reductions in travel distance can be obtained due to the possibility of making short cuts. On the other hand, the control of unidirectional flows is easier due to the fact that no opposite traffic of AGVs is allowed. To obtain advantages of both options, *multiple lane guidepaths* can be introduced. If enough space is available multiple paths with opposite traffic flows can be inserted in one aisle. Furthermore, in one flowpath layout, it can be decided to use a *mix* of unidirectional and bidirectional paths. Traffic control simplicity and benefits from shorter distances can be obtained in this way. Finally, during operation it can be decided to change direction of certain unidirectional flows to meet the demand of the system. The problem for all described layouts is to determine direction of travel on the flowpath by minimising vehicle travel times and by ensuring ability to reach all pick-up and delivery points in the network. Sinriech [244] presents an overview of literature on several approaches for the design of flow networks. In this section, firstly, literature will be discussed on the layout problem with unidirectional, bidirectional, multiple lane and mixed uni-/bidirectional flowpaths. Thereafter, attention will be paid to developments in flowpath layouts, like single loop layouts.

Gaskins and Tanchoco [89] were one of the first to discuss the AGV guidepath layout problem with *unidirectional arcs*. The problem is presented as a network and formulated as a zero-one integer linear programming model. The objective is to minimise the total loaded transportation distance of AGVs (i.e. transportation costs). The solution of this model indicates the optimal direction of travel of each arc. For practical problems, the number of variables and related computation times increase enormously. Therefore, Kaspi and Tanchoco [121] describe a model with extra constraints and give a computationally efficient procedure, namely a branch and bound approach. This model is extended by Kim and Tanchoco [139] by considering also the fixed costs for construction, control and space of the system. It is concluded that for large-sized problems a better performance is obtained by applying their approach instead of the model of Kaspi and Tanchoco [121]. Also based on the model of Kaspi and Tanchoco [121] is the work of Sinriech and Tanchoco [246]. They propose a branch and bound method which has to deal with a smaller set of nodes in the flowpath network. As a result, the branch and bound process has been sped up. Kouvelis *et al.* [153] also study the problem of the design of unidirectional flowpaths by minimising the total distance travelled by vehicles. They develope five different heuristics and a composite heuristic, which combines the most successful heuristics.

This composite heuristics provides solutions comparable to solutions obtained by applying a simulated annealing approach.

In above-mentioned papers only the impact of the flow of loaded vehicles is taken into account. In Sun and Tchernev [259] the problem for unidirectional flowpath design is generalised by also taking into account the impact of empty vehicle flow. By applying a branch and bound algorithm based on a depth-first search method, the optimal direction for each arc can be found. The unidirectional guidepath layout problem is also studied in [109], [139], [243] and [277].

A second approach to establish the design of the flowpath network is by determining simultaneously the direction of traffic flow in an unidirectional network and the location of pick-up and delivery points (see also Section 2.4). Goetz and Egbelu [94] modelled this problem as a linear integer programming model. The objective is to minimise the total distance travelled. The main focus of this approach is on systems with major flows of traffic and a large number of pick-up and delivery points. The problem of assigning a machine for each operation, determining the machine visiting sequence and designing the unidirectional guidepath simultaneously is studied in Seo and Egbelu [239]. The objective is to minimise the total process and transport times of all parts in the system. Global solutions are found by splitting the problem into two subproblems and applying a heuristic.

In a *bidirectional network*, traffic flow takes place in either direction in each aisle. However, vehicles are not allowed to travel in opposite directions at the same time. Therefore, buffer areas exist for temporarily parking of vehicles. Questions, that arise in this context, are: how many nodes will have buffering facility? Which kind of buffer areas will be used? At which places will the buffer areas be located? Egbelu and Tanchoco [70] discuss three different designs of buffer areas, namely *loop design*, *siding design* and *spur design*. In a *loop design*, two uni-directional loops are located at the end of each aisle. A uni-directional siding is located at the end of each aisle in a *siding design*. A siding serves AGVs travelling in the same direction. Finally, in *spur design*, dead end spurs, capable of being transferred in both directions, are located at the end of each aisle. Contrary, to the other options, vehicles will leave the spurs, according to the last in-first out principle. Egbelu and Tanchoco [70] also discuss a model which describes the flow and control of AGVs in a bidirectional network. With simulation, it is shown, that in a specified situation the use of bidirectional guidepaths in networks with few AGVs can lead to an increase in productivity. Kim and Tanchoco [125] also present simulation results to compare the performance of unidirectional and bidirectional layouts in a particular network. For this network, it shown that the bidirectional layout outperforms the unidirectional one in terms of the number of jobs completed per time unit.

In the *multiple lane guidepaths*, various flowpaths exist between nodes (pick-up and delivery points, intersections) of the network. Commonly, there are two or more unidirectional paths in the same aisle. The problem is to determine the number and direction of feasible flowpaths. Gaskins *et al.* [90] present this problem as a multi commodity flow problem and formulate it as a linear integer programming model. Both loaded and empty vehicle data can be used in the model. Weaknesses of this approach are computational difficulties and the fact that the interaction of vehicles is not taken into account in the model. Therefore, the solution from the models should be evaluated with simulation before applying them in practice.

The configuration of a *mixed uni-/bidirectional flowpath* is studied in Rajotia *et al.* [213]. An unidirectional flowpath layout is taken as input. A heuristic method is developed to configure some unidirectional paths to bidirectional ones. Candidates for this configuration are paths from centre i to j with the highest value of the product of the material flow from centre i to j and the reverse flow from j to i. The purpose of this configuration is to reduce travel distances. It is indicated that benefits can be obtained in throughput rates and size of the vehicle fleet. However, the rate of vehicle congestion increases and as a result traffic control becomes more important.

Other developments in flowpath design are *single loops*, *tandem configurations* and *segmented flow configurations*. In a *single loop layout* AGVs travel in a (unidirectional) loop. This loop is a fixed sequence of processing centres which need to be visited. Single loops are comparable to networks for equipment like conveyors. The objective is to minimise the total length of the loop under the condition that all stations are included in the loop. Advantages of the single loop layout are diverse. For example, all vehicles travel in the same loop and blocking of vehicles only occurs as a vehicle has to stop to pick-up or deliver a load. Furthermore, due to lack of alternative paths in the layout, control of AGVs is relatively easy. Disadvantages of single loop systems also exist, compared to above mentioned flowpaths layouts. For example, due to vehicle failure, the complete loop will be unusable. Furthermore, once a station is passed, an AGV has to travel the complete loop, before it reaches the station again. Finally, the throughput of the system will be lower (see [248] and [264]).

The problem of finding a valid single loop guidepath is comparable to the Travelling Salesman Problem, except for the facts that the number of places to visit is bounded and not fixed, the start point of the loop is not fixed and the solution is a set of arcs forming a closed loop (see also Tanchoco and Sinriech [264]). However, in

De Guzman *et al.* [52] it is proven that finding optimal solutions for single loop configurations is *NP*-complete, due to the constraint that all stations in the layout should be included in the loop. Kouvelis and Kim [154] already proved that assigning machines to locations in an existing unidirectional loop to minimise total costs is *NP*-complete. Tanchoco and Sinriech [264] discuss the design of single loop guidepaths and the location of pick-up and delivery points such that the flow of parts in the system is minimised. They propose to solve the problem by applying enumeration. In Sinriech and Tanchoco [249] suggestions are given to speed up this enumeration method. Asef-Vaziri *et al.* [7] formulate the shortest loop design problem as a integer linear programming model. With some simplifications the size of the problem can be reduced effectively. Instances with maximal 40 cells in a production area can be solved to optimality by standard solvers. More literature on this subject includes [146], [167], [240].

A *tandem configuration flowpath design* consists of non-overlapping single vehicle loops with load transfer stations in between. To transport a load from its origin to its destination more than one AGV might be required. At the end of each zone the load is transferred from one AGV to another. This layout is proposed by Bozer and Srinivasan [28, 29]. The layout of the flowpath can also be designed by using a *segmented flow approach*. Sinriech and Tanchoco [250] introduce the concept of segmented flows. Mutually independent zones are divided in non-overlapping single vehicle segments. Congestion of vehicles can be reduced with both kinds of layouts. Therefore, these kinds of layout are discussed in more detail in Section 2.3.

The methods described in Chapter 3 and 4 to determine the minimum vehicle fleet size can be applied in AGV systems with almost all flowpath layouts described in this section. Clearly, the use of the models is not required for the tandem-configuration and the segmented flow layout. In these kinds of layouts, the number of vehicles is already fixed to one per zone or segment.

2.3 Traffic management: avoidance of collisions and deadlocks

In controlling and designing AGV systems the problem of prevention of AGV collisions and deadlocks should be addressed. By attaching sensors on AGVs, physical collisions can be avoided. However, in the case that AGVs, moving in opposite directions, are forced to stop in front of each other, blocking of vehicles occurs and no further transport is possible. Without manual intervention a deadlock situation is caused. Deadlocks can also occur at buffer areas of pick-up and delivery points. If a load is available for transport at a pick-up and delivery point and a loaded AGV is the first in line before an empty AGV then the loaded AGV cannot be unloaded and the new load cannot be transported. As a result, the performance of the system will decrease. In designing the system it is tried to avoid the occurrence of deadlock situations during operation. During operation, deadlocks and collisions are either detected and resolved by rerouting vehicles to buffer areas or deadlocks and collisions are predicted and avoided by preplanning of routes (see Section 2.8). Detection and solving instead of avoidance of deadlocks results in a lower performance of the system. Therefore, methods are developed to avoid collisions and deadlocks.

Literature in this area can be divided into three categories. Firstly, the layout of guidepaths can be designed in such a way that collisions and deadlocks are avoided. Secondly, dividing the traffic area into several *non-overlapping control zones* can contribute to avoidance of deadlocks and collisions. Thirdly, routeing strategies can be developed to prevent collisions and deadlocks. Literature from the first category will be discussed in Section 2.8. The layout of the guidepaths has already been discussed in Section 2.2. Literature from the third category is discussed below.

In this case, the flowpath network consists of a number of control zones. Only one vehicle at the same time is allowed to travel through the control zone. Consequently, at most one vehicle occupies a zone and other vehicles willing to enter the zone are stopped. One or more vehicles can wait outside a zone in a buffer area to enter the zone if the previous vehicle has left it. Vehicles are allowed to travel from one zone into another one. As a result, no transfer stations between zones are required to move a load from one vehicle to another. For example, Malmborg [182] discusses this kind of zone strategy. Similar to this kind of strategy, Faraji and Batta [75] use a flowpath network consisting of cells. According to Kim *et al.* [126] and Lee and Lin [164] deadlocks still can occur with this zone strategy, namely by competing vehicles willing to enter a zone from different directions. Lee and Lin [164] propose an algorithm to avoid deadlocks in unidirectional control zone networks. Petri Nets are used to represent the current state and to generate future states of the system to analyse deadlocks. Petri Nets are graphical modelling tools, which are ideal for representing flexible manufacturing systems (see [298]). The algorithm, in which deadlock prediction and travelling decisions are included, should be executed each time an AGV tries to travel from one zone to another. Yeh and Yeh [294] also address deadlock problems of unidirectional control zone AGV systems and they propose an algorithm to deal with it. The current states of the system are represented in a directed graph. This graph can also be used to generate future states of the system. The algorithm is applied each time a vehicle travels to a new zone and looks ahead to all future zones that have to be travelled by the vehicle.

Several variations exists on this traditional zone strategy, namely *tandem guidepath configurations* and *segmented flowpaths*. In a *tandem configuration* non-overlapping single AGV loops covering all stations exist. Between adjacent loops, pick-up and delivery points are situated to transfer the load from one vehicle to another. Each station is included in exactly one loop and only one AGV is used per loop. More details on this kind of configuration are given in Bozer and Srinivasan [28]. Collisions and deadlocks will not occur due to the fact that only one AGV operates in each zone. Therefore, a less complicated control system is required in tandem configurations. Adding one or more new loops to the system is possible without disruption of existing loops. However, there are also several disadvantages to tandem configurations (see [28]). Loads may have to be handled by a number of vehicles before they arrive at their destination. Vehicle breakdowns result in unaccessible stations. A zone can become the bottleneck of the system if the workload of an AGV is much higher than the workload of AGVs in other zones. To handle these disadvantages the design of a tandem configuration layout could be examined critically. Decisions have to be taken on, for example, the number of zones and which stations are located in each zone. Bozer and Srinivasan [29] have developed a method to configure tandem AGV systems. The objective is to distribute among all AGVs in the system evenly the workload . The model is a variation of the set partitioning problem. With simulation studies it is indicated that in some situations, from the perspective of throughput, tandem systems can be very competitive to other flowpath layouts.

The *segmented flowpath layout* consists of one or more mutually independent zones. There does not exist a flow of materials from one zone to another. Each zone is separated into non-overlapping segments. Each segment is served by a single AGV, which can travel in both directions on the segment. Between the ends of two segments stations are located where loads can be transferred from one AGV to another AGV. The stations are capable of serving AGVs from both sides simultaneously. By using this approach loads can be transported from their origin to their destination by using the shortest path. Furthermore, no congestion or deadlocks will occur. More information on this kind of zone strategy is given in Sinriech and Tanchoco [250]. Methods to design this kind of layout are described in, for example, [14] and [251].

All described zone strategies have one thing in common, namely the fact that the area of the zone cannot be changed. As a result, a bottleneck zone cannot be relieved from some workload by other zones. To overcome this problem Ho [105] developed a strategy to allow the system to adjust itself to accommodate to workload changes of vehicles. The objective is to maintain the workload of each vehicle approximately the same during time. With this dynamic zone strategy zones are redesigned during operation to avoid significant differences in workload between the different zones. Real-time feedback on the traffic in the system is also used in the dynamic close loop system of Reveliotis [219]. At the end of each zone either a new zone is allocated to the AGV or the AGV is commanded to wait in its current zone. This decision is based on safety and performance considerations.

2.4 Location of pick-up and delivery points

In the design of the layout of the AGV system the locations of pick-up and delivery points have to be determined. The pick-up and delivery points connect the AGV network to, for example, machines, (un)load stations, inspection stations and places of storage. Furthermore, the pick-up and delivery point can be used as a transfer station from one material handling network to another. For example, goods are transferred from an AS/RS to an AGV. The pick-up and delivery points are present in all different flowpath layouts described in Section 2.4.

The choice of the location of these points is important. It influences the operational performance, for example, measured in terms of total distance travelled, waiting times of loads, of the system. The choice of the locations can be decided on during the design of the system (see, for example, [6], [94] and [187]. Furthermore, the location of the pick-up and delivery points may be decided on in addition to an existing system. To obtain a better performance, only a small number of pick-up and delivery points should be assigned to a location at this late stage (see Kiran and Tansel [145]).

Kiran and Tansel [145] developed a facility location model for a directed network (also with loops and parallel arcs) to choose locations for the pick-up and delivery points in an existing layout. The objective is to minimise the total costs of the movement of material in the system. Kim and Klein [128] also formulate the problem as a facility location problem to minimise the total distance of materials transported by AGVs in a flowpath network. Firstly, the problem is presented as a quadratic assignment problem, which is NP-complete. Therefore, two heuristics are developed to solve the problem.

In tandem configurations (see Section 2.3) pick-up and delivery points are used to transfer loads from an AGV in one zone to an AGV in another zone. In Huang [110] a design concept is proposed to find the optimal location of the pick-up and delivery point for each zone. For each zone just a single the pick-up and delivery point is assigned. As a result, a simple traffic control for the movement of materials between zones is obtained.

2.5 Vehicle requirements

The minimum number of vehicles required in the system has to be determined when the AGV system is designed. To ensure that all tasks are performed within time, sufficient vehicles have to be available. However, for economical reasons the number of vehicles should not be overestimated. Furthermore, too many vehicles in the system leads to more congestion.

To determine an optimal AGV fleet size, capable of meeting all requirements, many factors have to be taken into account. Several of these factors are:

- number of units to be transported,
- points in time at which units can be or need to be transported,
- capacity of the vehicle,
- speed of the vehicle,
- costs of the system,
- layout of the system and guidepath,
- traffic congestion,
- vehicle dispatching strategies,
- number and location of pick-up and delivery points.

The point in time at which a job could be transported can be indicated in various ways. Firstly, a single time point is given at which the transport should start. Earlier or later transport is not allowed. Secondly, for each job a time-window with a release time and due time is indicated in which the transport of the job should start. Transport before the release time and transport after the due time is not allowed. In most cases, the layout of the system and the guidepath has already been defined before the minimum number of vehicles is determined. The way in which vehicles travel through the system (one way, bidirectional and so on) influence the vehicle fleet size. More information on guidepath design and the number and locations of the pick-up and delivery points is given in Sections 2.2 and 2.4. The capacity of the vehicle is in most cases a single load. Vehicle dispatching strategies are discussed in more detail in Section 2.7. Most models in literature consider the performance of the system by determining the minimum number of vehicles. The costs of this performance are neglected in most cases.

A procedure to determine the minimum number of vehicles required can be initiated by identifying a complete vehicle journey for transportation tasks. At a certain location the vehicle is assigned to a load. The vehicle drives to the pick-up point of this specific load. After receiving the load, the vehicle drives to the delivery point of this load. At the delivery point the vehicle is unloaded. Subsequently, this delivery point is the origin of the next journey of the vehicle. Consequently, the journey consists of empty travel time (i.e. origin journey to pick-up point), full travel time (i.e. pick-up to delivery) and waiting time (i.e. time at origin until assignment). Using this data Müller [194] proposes a simple formula to obtain a rough calculation of the number of vehicles required by taking into account rough estimates of total travel times and frequency of transport requests. However, the total travel time depends on the randomness in the system and the amount of congestion. Due to randomness in the arrival pattern of loads to be transported, the empty travel time is difficult to determine. The full travel time is influenced by the amount of congestion in the system (see also Section 2.3). The speed of the vehicle will vary, depending on the number of vehicles travelling on the same AGV path, the layout of the system and whether the AGV is loaded or not.

As a result, many stochastic inferences are present in AGV systems. Therefore, stochastic models can be used to determine vehicle requirements. However, in the literature also a number of deterministic approaches can be found. To solve the problem of the determination of the minimum number of vehicles, deterministic assumptions are made on the randomness of the arrival pattern of jobs and speed of vehicles. In this way, methods can be developed to determine the minimum number of vehicles in an optimal way and in polynomial time. Besides these two mathematical ways, simulation studies can be performed to approach this problem. With a simulation model, the entire system can be observed. However, this is a time-consuming job. Literature from these three categories will be discussed below.

The deterministic case of the determination of the minimum number of vehicles operating on unidirectional flowpaths (see also Section 2.2) is studied in Maxwell and Muckstadt [187]. They use an integer programming formulation for seeking the optimum solution. Leung *et al.* [170] extended this model to situations where the capacity as well as the operating speeds of each vehicle type differ. The authors developed a mixed integer linear programming model with the objective to minimise the total vehicle transportation time. According to Ilić [113], the total number of vehicles in simple systems can be estimated by the number of roundtrips that each vehicle can make per hour and the total number of roundtrips. In Rajotia *et al.* [212] a mixed integer programming model is discussed. Considering load handling time, empty travel time, waiting and congestion time the objective of the model is to minimise empty trips of the vehicles. The results of the model are compared with the results of a simulation study. It is concluded that the vehicle fleet size is underestimated with the analytical methods. In Dell'Amico *et al.* [55] the problem in which a given set of jobs with fixed start instants have to be assigned to vehicles at different depots is studied. A heuristic is given which guarantees to find the minimum number of vehicles in polynomial time. Sinriech and Tanchoco [247] develop a model which combines the throughput performance of the system and the costs related to this performance. The number of AGVs required is determined by using a trade-off between the two objectives. Several options are suggested in a decision table such that the management can make the final decision on the required number of vehicles. In Chapter 3 of this thesis a method is described to determine the minimum number of vehicles required in the AGV system if each job has to be transported at its known release time. A minimum flow algorithm is presented to solve the problem in polynomial time. In Chapter 4 of this thesis an integer linear programming model is presented, which can be used to determine the number of vehicles required, if each job has to be transported within its time window. More deterministic models for the determination of vehicle requirements can be found in, for example, [65],[79],[179] and [180].

In stochastic cases the problem can be solved by using queueing networks. For example, Mantel and Landeweerd [185] use a hierarchical queueing network approach to determine the number of AGVs. Tanchoco *et al.* [263] also developed a queueing theory based model. It is an approximation model based on the steady state behaviour of the system. Compared to results obtained with simulation the model underestimates the number of vehicles required. Other papers discussing queueing models for determining vehicle requirements are, for example, [293].

The problem can also be studied with the use of simulation. Models of real systems are designed and experiments are performed with these models to gain an understanding of the behaviour of AGV systems. Gobal and Kasilingam [93] and Kasilingam and Gobal [120] present a simulation model that gives an estimation of the number of AGVs needed to meet material handling requirements. This estimate is based on the sum of the idle times of vehicles and machines and waiting times of the transported parts. An increase in the number of vehicles reduces the waiting times of part and the idle times of machines. However, it results in an increase in the idle times of the vehicles. Newton [199] shows how a continuous based simulation model can be used to calculate how many AGVs are needed.

Problems, that have an analogy to the determination of the number of AGVs required at an AGV system, are e.g. the tanker scheduling problem and the scheduling of workers to different tasks. One of the first articles to describe a deterministic model to solve the tanker scheduling problem is Dantzig and Fulkerson [49]. The problem of determining the minimum number of oil tankers required to meet a fixed transportation schedule is formulated as a linear programming problem and solved with the simplex algorithm. More recently, the problem is studied by Ahuja *et al.* [2] (see also Chapter 3). While these authors look to the finite horizon version of the problem, Orlin [202] considers the case where a finite number of tasks are executed periodically over an infinite horizon. The problem is solved efficiently as a finite network problem.

2.6 Control of AGVs

One of the main objectives of a control policy is to satisfy demands for transportation as fast as possible and without occurrence of conflicts among AGVs. Therefore, at least the following activities need to be performed by a controller of the system:

- dispatching of loads to AGVs,
- route selection,
- scheduling of AGVs,
- dispatching of AGVs to parking locations (i.e. locations where idle vehicles are positioned).

Firstly, a vehicle is dispatched to a new transportation demand. The selected vehicle is assigned a route and schedule for the transportation task, such that the transport can be executed without occurrence of deadlocks and collisions. If no new jobs are available, an AGV can be routed to a parking location to wait for a new transportation demand.

If all transportation demands with origin, destination, release time and transportation time are known in advance, *offline control* of the system is possible. Transportation requests need to be perfectly predictable and information in the system needs to be accurate. Then all decisions on dispatching, routing and scheduling can be made in advance in offline control systems. Otherwise, due to the stochastic nature of the transportation process, control systems capable of making real-time decisions are required. These so-called *online control systems* can be used *decentrally* and *centrally*. According to Müller [194] AGVs can be controlled decentrally on board of the vehicles or from decentralised control systems located at several places in the system. Decentralised control can in practice be used at single path layouts and tandem configurations.

If a single control system simultaneously controls all AGVs in the system, we refer to centralised control. The central control system needs to retrieve frequently actual information on the traffic situation at a large number of points in the AGV network. Furthermore, in the control system a database is available with information on, for example, pick-up and delivery point of loads and vehicle positions. The control problem can be associated with the flowpath networks discussed in Section 2.2. The complexity of the control problem varies per type of layout. The control of vehicles in a single loop without intersections is easier than the control of vehicles in large bidirectional networks (see also Section 2.2).

Dispatching, routing and scheduling decisions can be made simultaneously or separately. In Taghaboni and Tanchoco [260] an intelligent control system is developed to dispatch, route and schedule a fleet of free ranging AGVs in a non-conflicting way, based on real-time information. The vehicle chosen to perform a certain transportation task is the one which can satisfy demand in the shortest time. Furthermore, the route with the shortest travel time is selected. The vehicle is scheduled along this route in a way that no conflicts with other AGVs will occur. The control system can be used in bidirectional and unidirectional flowpaths.

Bilge and Ulusoy [18], Ulusoy and Bilge [266] and Ulusoy *et al.* [267] study the problem of the simultaneous scheduling of machines and AGVs. Starting and completion times of operations for jobs at machines and of the transport of these jobs between workcentres are determined together with the assignment of jobs to vehicles. Genetic algorithms and iterative heuristics are used to generate machine schedules with completion times which are transformed into time-windows for each transport. The objective is to minimise the makespan. Ro and Kim [222] developed online scheduling and control rules for AGVs and machines by simultaneously considering the performance measures makespan, mean flowtime, mean tardiness, maximum tardiness and system utilisation.

Most of the literature, however, study one or two of the problems at the same time. In the next sections dispatching, routing and scheduling and idle positioning decisions will be discussed in more detail.

2.7 Dispatching of AGVs

Dispatching refers to a rule used to select a vehicle to execute a transportation demand. This problem exists already as long as people or goods have to be transported from one destination to another by, for example, public transport systems, trucks, trains and airlines (see, for example, [22], [103] and [220]). In this section we observe literature on the dispatching problem in AGV systems.

The dispatching problem can be observed from different points of view. Firstly, a load is available for transport and needs to be assigned to an idle AGV. Secondly, a vehicle becomes idle and need to be assigned to a new task. Consequently, the dispatching problem is divided into two categories, namely *workcentre initiated dispatching* and *vehicle initiated dispatching* rules (see Egbelu and Tanchoco [69]). The problem is workcentre initiated if the vehicle has to be selected from a set of idle vehicles to transport a load. The problem is vehicle initiated if an idle AGV has to choose a load from a set of transportation requests.

In offline control systems all data on transportation requests are available at the start of the transportation process. As a result, vehicles can be assigned to loads in an optimal way by formulating the dispatching problem as an assignment problem. The set of transportation requests and the set of vehicles form a complete bipartite graph with a weight (for example, transportation times) assigned to each arcs. By applying the Hungarian method the problem can be solved efficiently.

A simple heuristic used in online control systems is the *first-come-first-served rule*, which dispatches a free AGV to the load that requested transport at the earliest time. Bartholdi and Platzman [16] present the *first-encountered-first-served rule*, which can be applied for decentralised online control for AGVs travelling in a single loop. The AGV with multiple loads continually travels in a single loop and picks up, if space is available on the vehicle, the first load it encounters. The load is unloaded at its destination. With simulation it is shown that this heuristic outperforms other heuristics, like the first-come-first-served rule, if it is applied in a single loop.

According to Egbelu and Tanchoco [69] the following heuristic rules can be applied in decentralised control systems for workcentre initiated dispatching:

- *random vehicle rule*: pick-up task is randomly assigned to any available vehicle regardless the location of the vehicle and the load,

- *nearest vehicle rule*: the vehicle at the shortest distance of the load is assigned to the load,

- *farthest vehicle rule*: the vehicle at the greatest distance of the load is assigned to the new transportation request,
- *longest idle vehicle rule*: the vehicle that has remained idle for the longest time among all idle vehicles is dispatched to the load,
- *least utilised vehicle rule*: the vehicle with the minimum mean utilisation is assigned to the new job.

These last two rules contribute to balance the workload among all AGVs in the system. Dispatching rules for vehicles to choose a load from transportation requests are also discussed in Egbelu and Tanchoco [69]. Some of these assignment rules for the vehicle initiated dispatching problem are:

- *random workcentre rule*: a workcentre with a transportation requests is randomly chosen and the vehicle is dispatched to the load at this workcentre,
- *shortest travel times/distance rule*: the vehicle is dispatched to the workcentre closest by. The objective of this rule is to minimise empty travel times of vehicles,
- *maximum outgoing queue size rule*: the vehicle is dispatched to the centre with the largest number of loads awaiting transport,
- *modified first-come-first-served rule*: vehicles are assigned to centres, in chronological order to their time of request for transport. A workcentre can have only one request at this list at a time.

With simulation Egbelu and Tanchoco [69] tested the performance of some of these rules. It is shown that rules based on distance measures had some drawbacks if layout conditions are not met.

Another modification of the *first-come-first-served rule* is given in Srinivasan *et al.* [256]. An empty AGV first inspects if loads are available for transport at the station the AGV just delivered its previous job. In the case that one or more loads are available for transport, the AGV starts transporting the first in line at this station. Otherwise, the AGV is dispatched to the oldest unassigned request in the system. With this rule it is tried to reduce unnecessary empty travel times of AGVs by, if possible, dispatching the AGV to a load at its destination. Furthermore, it is tried to distribute workload evenly over all AGVs. With empirical results it is shown that the performance of this rule is nearly as good as the performance of the shortest-travel-time-first rule. Hodgson *et al.* [106] model an AGV system as a Markov decision process. Dispatching policies resulting from the semi-markov decision model are described and tested for single load and dual load vehicles. Kim *et al.* [127] propose an AGV dispatching procedure based on the objective to balance workload in the system. A dispatching rule with self-adaptive capability is introduced in Kim and Hwang [142]. A knowledge-based system for selecting an AGV and selecting a workcentre from a set of workcentres requesting transport simultaneously is presented in Kodali [147]. Yim and Linn [295] test the performance of various heuristics with petri net models. Bozer and Yen [32] present dispatching rules which take advantage of information available in centrally controlled systems. Van der Meer [271] discusses how some of the previously described vehicle dispatching rules behave in such environments as distribution centres and container terminals. It is shown that vehicle initiated rules are outperformed by load initiated rules in these kinds of environments. More literature on dispatching decisions in container terminals is presented in Chapter 7 of this thesis.

2.8 Routing and scheduling of AGVs

If the dispatching decision is made a route and schedule should be planned for the AGV to transport the job from its origin to its destination over the AGV network. A route indicates the path which should be taken by the AGV when making a pick-up or delivery. The related schedule gives arrival and departure times of the AGV at each segment, pick-up and delivery point and intersection during the route to ensure collision free routing. The selection of a certain route and schedule influences the performance of the system. The longer it takes to transport a job, the less jobs can be handled within a certain time. Therefore, one of the objectives of the routing of AGVs is to minimise transportation times of loads. Algorithms have to be developed to solve the routing problem. Two categories of algorithms can be distinguished, namely *static* and *dynamic* algorithms. With static algorithms the route from node i to node j is determined in advance and is always used if a load has to be transported from i to j. Herefore, a simple assumption is to choose the route with the shortest distance from i to j. However, these static algorithms are not able to adopt to changes in the system and traffic conditions. In dynamic routing, the routing decision is made based on real-time information and as a result various routes between i and j can be chosen.

Static routing problems in AGV systems are related to *vehicle routing problems* (VRP) studied in transportation literature. In the vehicle routing problem a set of n clients with known demands need to be served by a fleet of m vehicles with limited capacity. The vehicles are all housed at one depot. The route of each vehicles starts and ends at this depot. m least costs (length) routes have to be planned such that each customer is served exactly once and that the total demand of the customers served by each vehicle does not exceed the capacity of each vehicle. The objective is to minimise the total distance of all m routes under previously mentioned conditions. This is a *NP*-hard problem to solve. The vehicle routing problem is studied extensively in literature. Bodin *et al.* [22], Fisher [77] and Laporte [161] provide an overview of literature in this area. A more recent paper observing this problem is from Kelly and Xu [122]. They propose a set partitioning based heuristic. In a systematic way fragments of routes are combined to obtain high-quality solutions. *Vehicle scheduling problems* can be seen as routing problems with additional constraints concerning times at which certain activities (e.g. delivery of a load) have to be executed. Vehicle activities have to be sequenced both in time and space. An overview of methods to solve vehicle scheduling problems is given in Bodin *et al.* [22]. Also, genetic algorithms can be used to solve vehicle scheduling problems (see, for example, [11]).

The *vehicle routing problem with time windows (*VRPTW) is a generalisation of the vehicle routing problem. For each customer a time window is defined. The time window $[s, t]$ restricts the service time of the customer to fall into the time interval from s to t. Such time-windows arise, for example, due to traffic restrictions or fixed time schedules of customers and their products. Finding a feasible solution to the vehicle routing problem with time windows is a *NP*-complete problem. Numerous studies on the vehicle routing problem with time windows are executed. Branch and bound methods ([151]), insertion heuristics ([252]), extensions of vehicle routing problem heuristics ([253]), langrangian relaxation ([78], [150]), constrained shortest path relaxation ([56], [149]) and set covering formulations ([33]) can be used to find solutions to the problem. Desrochers *et al.* [57] provide an overview of solution methods. The methods of Desrochers *et al.* [56] and Fisher *et al.* [78] are capable of solving the problem to optimality for a problem size with 100 customers. Kohl *et al.* [149] even solve a problem with 150 customers by using a strong valid inequality, the 2-path cut, to produce better lower bounds and by using an effective separation algorithm. Due to the difficulty of the problem solving larger problems in reasonable times is hard.

Dynamic vehicle routing problems are also studied in transportation literature. Multiple demands for service envolving in a real-time way have to be satisfied by vehicles. Psaraftis [210] indicates the differences between static and dynamic vehicle routing. Gendreau *et al.* [91] propose a parallel tabu search method for real-time vehicle routing and dispatching. According to Savelsbergh and Sol [230] the dynamic routing of independent vehicles can be solved by applying a branch and price algorithm. Gans and Van Ryzin [88] represent the problem as a classical set covering model. Performance is measured in terms of congestion.

A generalisation of the vehicle routing problem with time windows is the *pick-up and delivery problem with time windows* (PDPTW). Optimal routes have to be constructed such that transportation requests requiring pick-up and delivery are met. The VRPTW is a special case of the PDPTW in which all destinations are the same depot. Dumas *et al.* [61] present an algorithm using column generation to solve the problem. Overviews of literature in this area are given in Solomon and Desrosiers [254] and Savelsbergh and Sol [231].

A special version of the pick-up and delivery problem and a combination of a vehicle routing and a vehicle scheduling problem is the *dial-a-ride problem.* This kind of problem is concerned with dynamic routing of vehicles and real-time response to customers. Each customer should be served within a time-window and penalty functions are used to minimise waiting times of customers.

Analogies between these problems from transportation literature and routing and scheduling problems for AGVs in automated guided vehicle systems are clear. A number of loads at various locations have to be transported by vehicles at a certain start time or at a certain moment within a time window. However, the use of the described models from transportation literature is not always possible. These models do not take into account congestion in the system. Furthermore, most models are not developed to deal with real time response to dynamically changing transportation requests. Therefore, attention is payed in the literature to develop *non-conflicting routes* for AGVs. With a non-conflicting route, an AGV arrives as early as possible at the destination without conflicting with other AGVs. Krishnamurthy *et al.* [159] observe a static routing problem in which AGVs have to be routed on a bidirectional network in a conflict-free way such that the makespan is minimised. The problem is solved by applying column generation. Kim and Tanchoco [124] also discuss the problem of finding conflict free routes in a bidirectional network. The algorithm, they propose, is based on the shortest path methods of Dijkstra.

Dynamic route planning can be done in two ways: *complete route planning* and *incremental route planning* (see Taghaboni-Dutta and Tanchoco [261]). With complete route planning the entire route from origin to the final destination is determined at once. With incremental route planning, the route is planned segment for segment until the vehicle reaches its destination. The disadvantage of complete route planning is that a route may become invalid during operation due to unexpected events. However, with applying incremental route planning, the optimality of the complete route is disregarded. Taghaboni-Dutta and Tanchoco [261] give a dynamic approach to incremental route planning. The traffic control is estimated by modelling the guidepath as a queueing network. Oboth *et al.* [201] discuss the problem of dynamic conflict routing of AGVs in bidirectional networks. The solution methodologies from Krishnamurthy *et al.* [159] are implemented in a dynamic environment. By using petri nets, also, conflict free routes can be determined (see [298]). Seifert *et al.* [238] introduce a dynamic vehicle routing strategy based on hierarchical simulation. At each time a route decision has to be made, simulations are performed to evaluate a set of possible routes. The route with the smallest estimated travel time is chosen. Rajotia *et al.* [211] add time windows to nodes to represent enter and leave times from the vehicle which is going to occupy the node. Other vehicles are allowed to travel through the specific node at a time point not included in one of the time windows. Furthermore, the direction on an arc for an arriving vehicle is indicated with a time window. Dijkstra's algorithm is applied on this network with time windows to find the least congested and fastest routes for AGVs.

Except for finding conflict free routes attention should be paid to the presence of interruptions in the system. Interruptions might occur due to, for example, vehicle breakdowns, objects on AGV paths and manual intervention. As a result of interruptions, AGVs may be blocked and routes cannot be finished. Therefore, if an AGV encounters an interruption it has to be rerouted in such a way that no conflicts with other AGVs occur. Narasimhan *et al.* [197] use simulation to analyse rerouting of AGVs.

Literature on routing and scheduling of AGVs in AGV systems also includes [17], [19], [21], [23], [41], [58], [100], [172], [228], [245], [255], [289] and [297].

2.9 Positioning of idle vehicles

An AGV becomes idle if it has delivered a job at its destination and it is not immediately assigned to a new job. One of the decisions to make is where to locate idle AGVs such that they can react as efficiently as possible on a new assignment. The locations where an idle vehicle can park are indicated with various names such as parking locations, dwell points or depots. Once an assignment of a parking location is made to an AGV, the trip to this location usually may not be interrupted by a new assignment. As a result, the idle AGV can only travel to the pick-up point of the new job, if it has reached the parking location. A related research area is the problem of determining a dwellpoint for an idle AS/RS.

To reduce waiting times of loads for transport, AGVs need to respond as quickly as possible to a new assignment. Therefore, the location of idle vehicles should be chosen well. Some criteria used in selecting a parking location are (Egbelu [67]):

- minimisation of the maximum response time of the vehicle to travel empty from the parking location to the pick-up point of the load,
- minimisation of the average vehicle response time,
- distribution of idle of vehicles evenly over the network.

According to Egbelu [67] the following rules are mostly used for positioning idle vehicles:

- central zone positioning rule,
- circulatory loop positioning rule,
- point of release positioning rule.

With a *central zone positioning rule* central parking areas are designated for buffering idle vehicles. Empty vehicles are routed to these areas regardless their current destination. With a *circulatory loop positioning rule*, one or more loops of the flow-path network are used as loops for positioning idle vehicles. AGVs travel on these loops until an assignment to a new job is made. Finally, with the *point of release positioning rule*, AGVs remain at the point where they are unloaded and the new assignment is received. However, by applying this rule, it might occur that one AGV blocks other AGVs on the same path.

Egbelu [67] was one of the first in literature to study the problem of positioning an idle AGVs in a loop layout. By minimising the maximum response time, how to determine parking locations for m AGVs in both unidirectional and bidirectional loop layouts with n pick-up points can be studied. In the case that one AGV is used, the problem can be solved in polynomial time. For more than one AGV the problem for unidirectional layouts can be formulated as an integer non-linear programming model. Further, the problem for the bidirectional layout can be solved by applying heuristics.

Gademann and Van de Velde [84], however, prove that the problems of minimising maximum and average response time in unidirectional loop layouts can be solved in polynomial time for a fixed number of m AGVs. If it is assumed that AGVs travel at constant speed even the problem of determining parking locations in a bidirectional loop layout with m AGV can be solved in polynomial time.

In Kim [129] and Kim and Kim [135] a static positioning strategy is proposed. In such a strategy, the location of a parking area is not changed. This version of the problem with one vehicle can be solved in polynomial time, by modelling it as a discrete time stationary markov chain.

A dynamic positioning strategy for m vehicles in which a new positioning location is assigned to each vehicle that becomes idle, is introduced by Kim [129]. The author also considers unidirectional and bidirectional single loop layouts. The objective is to minimise mean response time. The case in which already $m - 1$ AGVs have been located can be solved in polynomial time.

Chang and Egbelu [38] observe the problem of determining parking locations under the assumption that the number of loads to be picked up dynamically changes over time. Location models are presented based on the objective to minimise expected response times. Simulation studies indicate that, by applying these location models, the response times can be improved. Lu and Gerchak [177] indicate that delays occur if the AGV has to travel all its way to the parking location and new jobs are already assigned to it. Therefore, they propose an analytical model to choose an optimal anticipatory destination of an idle vehicle based on the location where it became idle and parameters of the system.

At a certain time, idle vehicles at a parking area or loop have to perform a new job. Consequently, new loads that have to be transported, have to be assigned to an idle vehicle. A question that arises is, for example, which idle vehicle transports which job? Dispatching rules in this context are discussed in Section 2.7.

2.10 Outline of part I of this thesis

In designing an AGV system a number of decision variables, like layout of the guide-paths and number of vehicles required, arise. This chapter gives an overview of literature concerning these and other decision variables.

The determination of the number of vehicles required is one of the planning problems in an AGV system. The objective is to determine vehicle requirements such that a sufficient performance level is obtained. This performance can be measured by observing empty trips of vehicles, travel times of vehicles and delay times of products. Stochastic, deterministic and simulation models can be used to solve the problem of determining vehicle requirements.

From the literature discussed in Section 2.5 it is clear that hardly any algorithm exists, for the deterministic case in which vehicle requirements are determined in an optimal way. In Chapter 3 we consider the case in which jobs must be transported exactly at the moment they are available for transport. A minimum flow algorithm is developed which solves the problem of determining vehicle requirements to optimality in polynomial time.

Furthermore, less attention has been paid in literature to the case in which jobs must be transported within a time-window. At a certain release time, jobs are available for transport in a buffer. Due to physical constraints of the buffer and to avoid congestion jobs finally must be transported at their due time. In Chapter 4 we give two solution approaches to solve the problem of determining vehicle requirements under time-window constraints. Firstly, the problem is formulated as an integer linear programming model. Thereafter, the problem is formulated as a set partitioning problem.

3

Determination of the number of vehicles required

In Chapter 2 we discussed various decision variables affecting the design of an automated guided vehicle system. The performance of the system is, among other things, influenced by its design. The efficiency of the processes in the material handling centre is affected by the performance of the AGV system. If, for example, not enough AGVs are available within the centre, inefficient routes and schedules might be chosen and congestion can occur. As a result, delays can occur in the production and distribution processes within material handling centres.

Consequently, one of the decisions in the design process of an AGV system is to determine the minimum vehicle fleet size required to ensure that all jobs are transported in time. This problem can be considered as a planning problem In the case that for each operation it has to be determined how many vehicles from a given vehicle fleet should be used.

This planning problem has many aspects. In this chapter we consider a deterministic case of this problem. An algorithm is described to determine the minimum number of vehicles required to transport all jobs at their known release times. Firstly, Section 3.1 contains a more detailed description of the problem. Section 3.2 discusses the literature related to the problem of determining vehicle requirements. Section 3.3 proposes a model to determine the minimum vehicle requirements. A polynomial time algorithm to solve the problem is presented in Section 3.4. To illustrate the way the model and algorithm could be used, a simple example is discussed in Section 3.5. Finally, some concluding remarks are given in Section 3.6.

3.1 Problem description

Automated guided vehicles (AGVs) can be used within material handling centres for the internal transport of products. We consider the case in which unit loads have to be transported from their origins to their destinations. Most AGVs are not capable of lifting a load by themselves. Another type of material handling equipment, for example a storage and retrieval machine (SRM) is required to place loads on and take loads off an AGV. This transfer process occurs at pick-up and delivery points. As a result, these two types of equipments and the related processes are strictly connected.

We assume that the transportation process is executed in such a way that delays are avoided in the other processes in the material handling centre. As a result, vehicles must be present the moment the other type of equipment has handled a load and has it available for transport. No space is available where loads can wait for transport by an AGV. Thus, loads (i.e. jobs) are available for transport by an AGV at a certain release time at *pick-up and delivery points* (p&d points). A transportation demand is translated into an assignment for an AGV to transport this specific load. An AGV transports the load from one p&d point to another. Different routes can exist between the p&d points in the AGV system. This depends on the kind of flowpath used (see also Chapter 2). The transportation time of the load depends on the route chosen for the AGV to transport this specific load.

Examples of this situation in practice are, for example, the transportation process of loads within warehouses and container terminals. Within warehouses products can be retrieved from storage by a storage and retrieval machine (SRM). In the case that no conveyors are used, the load needs to be placed on the AGV by the SRM itself. To avoid delays at the SRM an AGV must be present if a load is available for transport. At automated container terminals containers are unloaded of a ship by a crane. The containers are placed on AGVs by the crane. The AGVs transport the containers to the storage area. To ensure that a ship spends as little time as possible in the terminal, as small as possible delays should occur at the cranes. Consequently, if a container is unloaded of the ship an AGV should be present to receive the container at once. No buffers are used in these practical situations.

The objective is to ensure that the transport operation is carried out quickly and efficiently in an environment without buffer areas. One of the problems that has to be solved is the determination of the number of vehicles required to transport all loads at their release time. Beforehand the operation the concept of this chapter can be applied to estimate the number of vehicles required to obtain an efficient transportation process. During the real operation the number of vehicles still can be adjusted to unexpected events.

We make the following assumptions:

1. There are N jobs to be transported. For each of the jobs the availability time at the pick-up point is known.

2. The order in which jobs are handled by the machines is known.

3. There are K pick-up and delivery points (p&d-points) of jobs. At the p&d-points there is no interference between arriving AGVs and leaving AGVs. Arriving and leaving AGVs are separated. Arriving AGVs with a load wait at the delivery point and empty AGVs wait at the pick-up point. Therefore, the order in which jobs are handled by the other type of equipment does not result in deadlocks. Thus, it never occurs that, for example, a job needs an empty AGV for transport and the first in line is a full AGV which needs the other occupied machine, to be unloaded.

4. No buffer areas are available at pick-up and delivery points. Jobs that are ready for transportation, therefore, have to be transported immediately.

5. The release time of a job is the point in time at which the job arrives at the p&d point of its origin and is ready for transportation. That is, we assume that there is no interference at the other type of equipment between arriving and leaving jobs. It is therefore not necessary to define the handling time of the job at the other type of equipment (i.e. the time required to process the job and to bring it from, for example, the storage area to the p&d-point). This time is incorporated in the release time of the job.

6. The arrival time of a job is its arrival time at its destination p&d-point. The other type of equipment then has to take the load of the AGV, deliver it to, for example the storage area, and return to the p&d point. The corresponding handling time of the job is known in advance.

7. The capacity of an AGV is one unit load.

3.2 Literature

As mentioned in Chapter 2 there are several possibilities for solving the problem of the determination of the minimum number of vehicles required within an AGV system. Problems, that have an analogy to the determination of the number of vehicles required are, for example, the tanker scheduling problem and the scheduling of workers to different tasks. One of the first articles to describe a deterministic model to solve the tanker scheduling problem is Dantzig and Fulkerson [49]. The problem of determining the minimum number of oil tankers required to meet a fixed transportation schedule is formulated as a linear programming problem and solved with the simplex algorithm. The same tanker scheduling problem is also discussed by Ahuja *et al.* [2]. However, they suggest a different approach, namely to solve the problem by constructing a network and using any maximum flow algorithm. They also give an overview of different maximum flow algorithms. In general, each maximum flow problem can be formulated as a minimum cost flow problem. Another way of solving the problem of the determination of the number of vehicles is, therefore, to formulate it as a minimum cost flow problem and to use any minimum cost flow algorithm. A different approach, with the same time complexity as the method suggested in Ahuja *et al.* [2] is to use bipartite networks. This approach is used in Phillips and Garsia-Diaz [209] to determine the minimum number of workers necessary to accomplish a fixed schedule of tasks. The maximum flow in this bipartite network indicates pairs of tasks that should be performed by the same individual. To obtain the complete list of jobs for each individual, the arcs in the maximum flow have to be searched. Another method to solve the problem of the determination of the minimum number of individuals to meet a fixed schedule of tasks is given in Ford and Fulkerson [81]. The tasks can be partially ordered as follows: job i precedes job j if the start time of i is earlier than the start time of job j and if the jobs can be executed by the same individual. An ordered chain therefore represents a possible assignment of jobs to one individual. A minimum chain decomposition consequently represents the minimum number of individuals required.

In Chapter 2 literature on the specific problem of determining vehicle requirements in AGV systems has been discussed.

3.3 Model

A model and algorithm are developed in this chapter to solve the planning problem of determining vehicle requirements in a bufferless environment. The algorithm is capable of handling large data sets. The results obtained before the start of the operation, indicate how many vehicles should be used to ensure an efficient transportation process. During operation, it has to be decided which AGV from the vehicle fleet transports which load along which route and at which point in time. The possibility of adjusting the number of vehicles to accommodate unexpected events remains open. The required number of AGVs can be estimated by means of a simplified model in which loads are available for transportation at known points in time. By varying these time instants, robust estimates can be made of the necessary number of vehicles.

We choose to follow the approach of Ahuja *et al.* [2]. They discussed a similar problem and suggested that it can be solved by constructing a network and using any maximum flow algorithm. However, they only modelled the network and did not explicitly solve the problem. In this section, we explicitly formulate the network model for this specific situation and in Section 3.4 we construct a solution algorithm. A transformation of the graph is executed on the basis of a feasible flow in the original network. A maximum flow is thereafter determined in the transformed network. The values originating from the feasible and the maximum flow are used to determine a minimum flow through the original graph. The value of this minimum flow corresponds to the minimum number of AGVs needed in the AGV system.

To model the problem as a network model, we firstly have to define some specific aspects of each job. For each job the following is known:

s_{i_p} release time, the point in time at which job i $(1 \leq i \leq N)$ is available at origin pick-up point p $(1 \leq p \leq K)$ for transportation by an AGV.

$w_{i_{pd}}$ travel time of a full AGV from origin pick-up point p $(1 \leq p \leq K)$ of job i $(1 \leq i \leq N)$ to the destination delivery point d $(1 \leq d \leq K)$ of job i.

v_{i_d} arrival time, the point in time at which job i $(1 \leq i \leq N)$ arrives at the destination delivery point d $(1 \leq d \leq K)$. This time depends on the travel time of the full AGV and the release instant of the job. Consequently

$$v_{i_d} = s_{i_p} + w_{i_{pd}} \tag{3.1}$$

$r_{i_d j_p}$ travel time of an empty AGV from the destination d $(1 \leq d \leq K)$ of job i to the origin (pick-up point) p $(1 \leq p \leq K)$ of job j, with $s_j > s_i$ ($r_{i_d j_p} = 0$ if the destination of job i equals the origin of job j).

The travel times are assumed to be deterministic. We derive these deterministic travel times in the following way: if we want to be sure that the AGV is on time at the pick-up point of a next job, we should take into account the worst case, namely the largest travel time required for the corresponding trajectory. To obtain this deterministic value we can take the upper bound of a stochastic distribution representing the travel time for the longest existing route between the origin and destination.

As mentioned above, it takes the other type of equipment (machine) a certain amount of time to unload the job of the AGV, to bring it to, for example the storage area and return to the p&d point. This time is defined as follows:

b_{i_d} handling time of job i $(1 \le i \le N)$ at delivery point d $(1 \le d \le K)$.

As described in the introduction the load is taken off the AGV by a free machine. An AGV can obviously only start a new job if it is empty, so any previous load must first be removed. If the machine is handling another arriving or leaving job, a full AGV has to wait for the machine before it can leave for a new job. In the case that the previous job j is an arriving job, the point in time (t_{i_d}, delivery time) at which job i can be taken off the AGV by the machine at delivery point d $(1 \le d \le K)$ depends on the handling time and delivery time of the previous arriving job at delivery point d. in the case that the previous job j is a leaving job, the delivery time equals the release time of job j. The delivery time t_{i_d} (including the pick up time) is, therefore, determined as follows :

$$t_{i_d} = \begin{cases} \max(v_{i_d}, t_{j_d} + b_{j_d}), \text{ if job } j \text{ is an arriving} \\ \qquad \text{job at delivery point } d. \\ s_j, \text{ if job } j \text{ is a leaving job at delivery point } d \end{cases} \tag{3.2}$$

Equation (3.2) can be understood as follows. Assume that job j is an arriving job at the machine and has arrived earlier than job i $(v_{j_d} \le v_{i_d})$. The order in which jobs are handled by the machine is the same as the order in which jobs arrive at the machine and is known beforehand. Job j will therefore always be handled before job i $(t_{i_d} \ge t_{j_d})$. If the point in time at which job j was delivered at the delivery point d plus the handling time of job j at the machine is later than the point in time at which an AGV arrives with load i at the delivery point d, then the AGV has to wait at the machine until the machine is free. In this case, the point in time at which load i is delivered therefore equals the finish time of job j at the machine. In the case that job j is a leaving job, the crane first has to place the job on another AGV at a known release time. Thereafter, job i can be handled.

Summarising, the release times s_{i_d} of all loads as well as the travel times $w_{i_{pd}}$ and $r_{i_d j_p}$ and the handling times b_{id} are known. Consequently, all values of v_{i_d} and t_{i_d}

become known by applying Equations (3.1) and (3.2). Note the difference between v_{i_d} and t_{i_d}. v_{i_d} represents the arrival of job i at the p&d point d. t_{i_d} represents the time at which job i is taken off the AGV.

The network for our problem is constructed as follows: each job is represented as a node, and there is a directed arc from node i to node j with capacity one if one AGV can execute both jobs. In this case we call a set of jobs (i, j) compatible. Consequently, there is no directed arc (j, i), because job j can be executed after job i but, by definition, job i cannot be executed after job j. If job j can be transported after job i the release time of job j is later than the release time of job i because jobs are not allowed to wait for transport. Consequently, job j is released after job i and cannot be transported first. Since jobs are not allowed to wait for pick-up, jobs (i, j) are compatible if:

$$t_{i_d} + r_{i_d j_p} \leq s_{j_p} \tag{3.3}$$

Using the above data we construct a directed network $G = (V, A)$ with:

$$\begin{aligned} V &= \{1, \ldots, N\}, \text{where each node represents a job,} \\ A &= \{(i, j) : i, j \in V, \text{jobs } i \text{ and } j \text{ are compatible}\} \\ &\quad \text{in which the capacity of each arc equals one.} \end{aligned}$$

We further introduce a source node s and sink node t and add all arcs (s, i) and (i, t) with capacity 1. This is because the maximum flow algorithm, that we will use in the next section, seeks a feasible solution that sends the maximum amount of flow from s to t. Such a solution, the zero-one decision variable y_{ij}, would have to satisfy the following conditions:

- $$y_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \in A \text{ is used in the maximum flow,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

- For source node s and sink node t:

$$\left\{ \begin{array}{lcl} \sum_j y_{sj} & = & \sum_j y_{jt} \\ \sum_j y_{js} & = & 0 \\ \sum_j y_{tj} & = & 0 \end{array} \right. \tag{3.5}$$

- For the other nodes:

$$\sum_j y_{ij} = \sum_j y_{ji} = 1\text{: incoming flow equals outgoing flow} \tag{3.6}$$

Note that in this network a flow from the source to the sink represents a sequence of jobs that can be executed by a single vehicle. An example of constructing such a directed graph is given in Section 3.5 (Figure 3.1).

The constructed network makes it possible to solve the problem of determining the minimum number of AGVs required at a material handling centre. In this network a directed path corresponds with a feasible sequence of jobs which can be executed by one AGV. The aim is to determine the minimum number of directed paths, which corresponds with the minimum number of AGVs, such that each node in the network is included in exactly one path.

To solve the problem, the network (graph G) is transformed into G_1 as follows:

- split every node i (except source s and sink t) into two nodes i' (origin) and i'' (destination) and add the arc (i', i''). As a result, we obtain $V_1 = \{s, 1', 1'', \ldots, N', N'', t\}$ and $A_1 = \{(i'', j') : i'', j' \in V_1$, jobs i and j are compatible$\} \cup \{(i', i'') : i', i'' \in V_1\} \cup \{(s, i') : s, i' \in V_1\} \cup (j'', t) : j'', t \in V_1\}$.
- the lower bound for the flow on each arc (i', i'') $\forall i \in V$ is set at 1

The purpose of these two steps is to replace the capacity restrictions on the nodes with a capacity restriction on the connecting arc. One unit of flow obviously has to pass through this arc, since every job has to be executed. Consequently, every node will be visited.

After this transformation each directed path from s to t corresponds to a sequence of jobs that can be executed by one AGV. Consequently, a flow of value v (v directed paths each with a flow of one) corresponds with schedules for v AGVs. Hence, the problem is reduced to finding a feasible flow with minimum value.

3.4 A minimum flow algorithm to solve the model

To solve the problem of finding a feasible flow with minimum value in graph G_1, we introduce the following algorithm. Thereafter, the idea behind the algorithm is explained.

Step 1

a) Determine a feasible flow x through G_1. Without computation we can obtain a feasible flow in G_1 as follows:

b) Each node is included in a different path from s to t. This corresponds with the following flow x_{ij}:

$$\begin{aligned} x_{si'} &= 1 \quad \forall (s, i') \in A_1 \\ x_{i''t} &= 1 \quad \forall (i'', t) \in A_1 \\ x_{i''j'} &= 0 \quad \forall (i'', j') \in A_1 \\ x_{i'i''} &= 1 \quad \forall (i', i'') \in A_1 \end{aligned} \tag{3.7}$$

Note that this flow is actually a maximum flow in G_1.
Step 2
Assign the triple (l_{ij}, x_{ij}, u_{ij}) to every arc.

l_{ij} equals the lower bound of the capacity ($l_{ij} = 0 \, \forall (i,j) \in A_1 \backslash \{(i', i'')\}$ and $l_{ij} = 1$ otherwise)

u_{ij} equals the upper bound of the capacity ($u_{ij} = 1 \; \forall (i,j) \in A_1$)

x_{ij} corresponds with the value of the flow as given by Equation (3.7).

Step 3
Construct the graph G_2 as follows:
$A_2 = A_1 + \{(j,i) : (i,j) \in A_1\}$
$V_2 = V_1$.
A_2 consists of all arcs in A_1 plus a backward directed arc for each arc $(i,j) \in A_1$.

(a) The upper bound of the capacity of the forward arcs (i,j) in G_2 is:

$$\Delta_{ij} = x_{ij} - l_{ij} \tag{3.8}$$

b) The upper bound of the capacity of the backward arcs (j,i) in G_2 is:

$$\Delta_{ji} = u_{ij} - x_{ij} \tag{3.9}$$

Step 4

Determine a maximum flow x' through network G_2 by using any maximum flow algorithm (for an overview see Ahuja *et al.* [2]).

Step 5
Define the flow x^* as follows:

$$x^*_{ij} = x_{ij} - x^{'}_{ij} + x^{'}_{ji} \qquad \forall (i,j) \in A_1 \tag{3.10}$$

Step 6
The flow x^* from step 5 is the flow with minimum value in graph G_1.

The idea behind the algorithm is that the feasible flow from step 1 can be reduced to a minimum flow. By using the backward arcs, paths can be constructed which consist of several nodes. The maximum flow in graph G_2 is used to determine how the feasible flow in graph G_1 can be reduced to a minimum flow. Every backward arc between two jobs contained in the maximum flow in G_2 represents two jobs being transported by the same vehicle. Clearly, every combination of two jobs in the maximum flow in G_2 results in the saving of one vehicle. The value of the maximum flow consequently indicates the total saving of vehicles compared to the value of the feasible flow from step 1. In other words, this maximum flow in G_2 indicates which forward arcs have to be removed from the feasible flow in $G_1$and which backward arcs have to be added to the feasible flow in G_1. The value of the maximum flow in graph G_2 indicates how much the value of the feasible flow in graph G_1 must be reduced to obtain a minimum flow in graph G_1.

THEOREM 3.1.
The result of the algorithm *Minimum Flow* is a feasible flow of minimum value in graph G.
PROOF
see Appendix A.

THEOREM 3.2.
The minimum flow algorithm solves the problem of determining the minimum number of AGVs in $O(n^{5/2})$ time, where n is the number of jobs.
PROOF
From Ahuja *et al.* [2] it follows that the unit capacity maximum flow algorithm solves a maximum flow problem on unit capacity simple networks in $O(n^{1/2}m)$ time, where n is the number of nodes and m the number of arcs. In this kind of networks,

every node (except source and sink) has at most one incoming arc or at most one outgoing arc. The number of arcs in graph G_2 is $O(n^2)$. Hence, the complexity is $O(n^{1/2} \cdot n^2) = O(n^{5/2})$. ■

3.5 Illustrative example

Within a material handling centre jobs have to be transported by AGVs from one pick-up and delivery point to another. At these p&d-points other types of equipments are unloading and loading loads on AGVs. In this illustrative example, we assume that within this AGV system 3 pick-up and delivery points are present. Table 3.1 gives the travel times of empty and full AGVs from one pick-up and delivery point to another. We assume that the travel times are the same for both directions.

	$1 \leftrightarrow 2$	$1 \leftrightarrow 3$	$2 \leftrightarrow 3$
full AGV	4	5	3
empty AGV	2	3	1

TABLE 3.1. Travel times for full and empty AGVs between pick-up and delivery points.

We assume that no buffer areas are available at the pick-up and delivery points and as a result the waiting times of loads at pick-up and delivery points are zero. In this simplified example we consider a set of three loads. The characteristics of these loads are given in Table 3.2. The arrival times are determined by using the data from Table 3.1 and the aforementioned assumptions.

load i	origin p	destination d	release s_{i_p}	arrival v_{i_d}	handling b_{i_d}
1	1	3	1	6	4
2	2	3	5	8	2
3	2	1	11	15	1

TABLE 3.2. Load characteristics.

From the data in Table 3.2 it can be seen that the AGV with load 2 has to wait at the p&d point 3 until the crane has finished handling load 1 and has returned to the p&d point. From Equation (3.2) it follows that the values of t_{i_d} $(1 \leq i \leq 3)$ are as given in Table 3.3:

load i	t_{i_d}
1	6
2	10
3	15

TABLE 3.3. Delivery times.

As described, we solve this problem by constructing a network. Loads i and j can be transported by the same AGV if the start time of load j is later than or equal to the delivery time of load i plus the travel time from the destination of i to the origin of j. From Tables 3.2 and 3.3 it is obvious that the jobs $(1,3)$ and $(2,3)$ are compatible. By adding a source s and sink t and the arcs (s,i) and (i,t) $\forall i \in V$ we can construct graph G with $V = \{s,1,2,3,t\}$ and $A = \{(s,1);\ (s,2);\ (s,3);\ (1,3);\ (2,3);\ (1,t);\ (2,t);\ (3,t)\}$. This graph is presented in Figure 3.1.

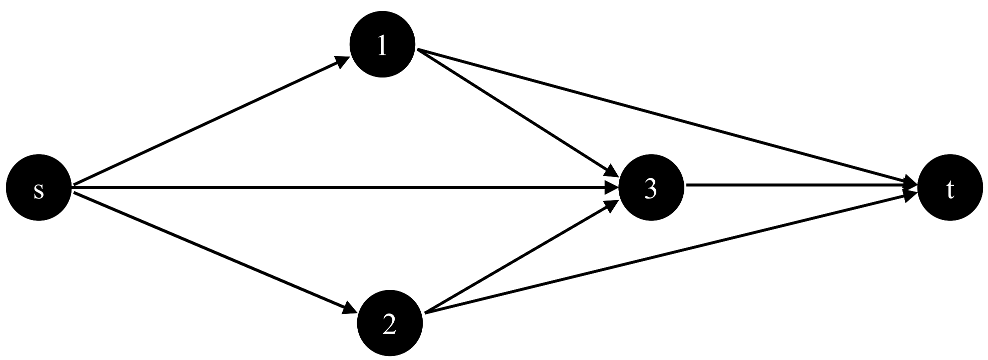

FIGURE 3.1. Graph G.

First, we perform the transformation as described before. To solve the problem of finding the minimum number of AGVs we use the minimum flow algorithm from the previous section.

From step 1 it follows that the maximum number of AGVs equals the number of jobs. Hence, the value of the maximum flow equals 3. This flow is shown in Figure 3.2.

Next, graph G_2 is constructed (see Figure 3.3). The upper bound of the capacity of each arc follows from Equations (3.8) and (3.9) and is shown in Figure 3.3.

Any maximum flow algorithm is used to determine the maximum flow through G_2. The maximum flow equals 1 and the following arcs are included in this maximum flow:

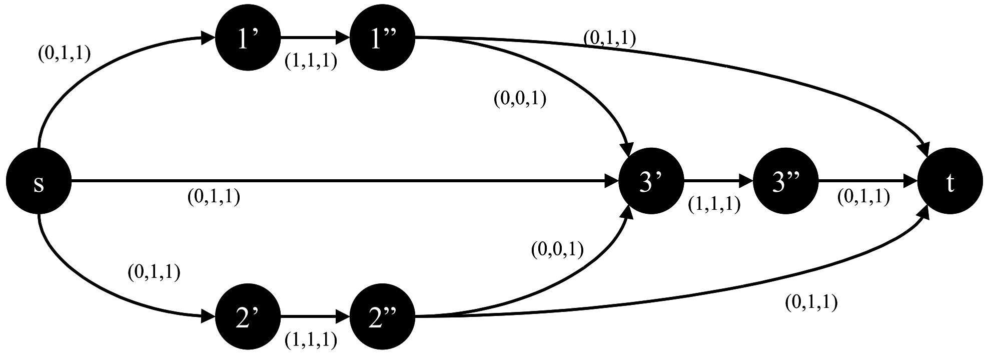

FIGURE 3.2. Graph G_1 with triple (l_{ij}, x_{ij}, u_{ij}).

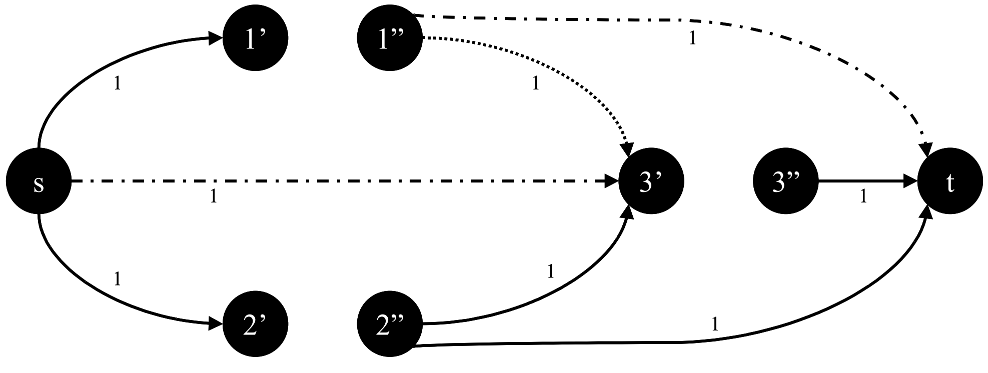

FIGURE 3.3. Graph G_2 with forward arcs (-.-.) and backward arcs (...) that belong to the maximum flow.

$(s, 3'); (3', 1''); (1'', t)$

The arcs that belong to this maximum flow are shown in Figure 3.3.

Using this maximum flow, the values of x^*_{ij} (see step 5 of the algorithm) are determined:

$$
\begin{array}{ll}
x^*_{s1'} = 1 & x^*_{2''t} = 1 \\
x^*_{s2'} = 1 & x^*_{3''t} = 1 \\
x^*_{s3'} = 0 & x^*_{1''3'} = 1 \\
x^*_{1''t} = 0 & x^*_{2''3'} = 0
\end{array}
$$

Consequently, the minimum value of a flow in graph G_1 is 2. All loads can therefore be transported by two AGV. The flow with minimum value is shown in Figure 3.4.

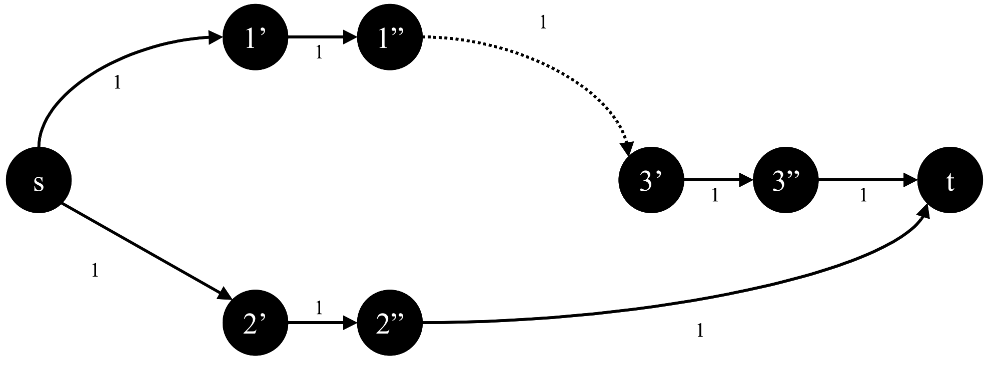

FIGURE 3.4. Minimum flow in graph G_1.

3.6 Concluding remarks

Automated Guided Vehicles can be used for internal transport of loads within material handling centres. In designing an automated guided vehicle system attention has to be paid to numerous factors. One of the problems that has to be solved in designing an AGV system is the determination of the number of AGVs required to transport all loads within time. In this chapter we developed a minimum flow algorithm to determine the minimum number of AGVs required at a material handling centre without buffer areas. The AGVs are used to transport loads from one pick-up and delivery point to another at known release times. By assigning as many loads as possible to an AGV the fewer AGVs are required. Loads have to be transported at their release times. Therefore, two loads can be transported by the same AGV only if the delivery time of the first job at its destination plus the travel time to the origin of the second job is smaller than or equal to the release time of this second job. All combinations of two loads, which can be transported by the same AGV, can be represented in a graph. A transformation is performed based on a feasible flow in this graph. We determine a maximum flow in this transformed graph. The minimum flow, which corresponds to the minimum number of AGVs required, equals the value of the feasible flow in the original graph minus the value of the maximum flow in the transformed graph. It proven that this is a strongly polynomial time algorithm.

With the concept developed in this chapter, we can determine the minimum number of AGVs required to transport all loads with known release times within an environment without buffer areas to optimality. In Chapter 8 of this thesis we will show how this algorithm can be applied to determine the number of AGVs required within a container terminal.

In Chapter 4, the assumption of a bufferless environment is skipped. We consider a material handling centre in which the transportation process is decoupled from the other processes in the centre. Loads are available for transport in buffer areas with a fixed capacity. The transport can start the moment the load is placed into the buffer. To avoid congestion and delays in the other processes in the centre, all loads have a deadline before which they need to be transported. Consequently, the problem is then to determine the minimum number of vehicles such that each load is transported within each time-window.

4

Vehicle requirements under time-window constraints

In this chapter we study an extension of the problem in Chapter 3. We consider the process of transporting loads by vehicles in an arbitrary material handling centre. A distinction can be made between two different types of material handling equipments, namely non-lifting and lifting vehicles.

Non-lifting vehicles need the help of another type of equipment to receive the load. For example, loads can become available for transport at conveyors. At the end of a conveyor the load is transferred to a vehicle. To ensure that no congestion occurs at the conveyor a finite number of loads can wait at the end of the conveyor for transport. If this number of loads is reached, the first load in line must be transported directly by a vehicle to create space for a new load. Consequently, to ensure that no congestion occurs a vehicle should pick up the load any time between the moment the load arrives at the end of the conveyor and the moment the load needs to be transported at the latest.

Lifting vehicles are capable of lifting a load by themselves. An example of such a vehicle is a forklift truck. No other material handling system is required to put the load onto the vehicle. Loads are waiting for transport in a buffer area, for example on the ground, with a fixed capacity. If the buffer is full the earliest arrived load must have been removed before a new load arrives. Consequently, for each job a deadline for transport (i.e. due time) can be defined. The lifting vehicle must pick up the load before this due time and transport it to its destination.

For both cases a similar problem arises, namely the problem of determining the minimum number of vehicles required to transport all jobs within their time-window. Each time-window is defined by a release time and a due time. The release time equals the moment the load arrives in the buffer. The due time equals the moment in time the job should have been removed from the buffer to avoid congestion.

In the literature, little attention has been paid to the problem of determining the minimum vehicle fleet size under time-window constraints. In this chapter we use a deterministic model in which loads need to be transported within predefined time-windows. New planning concepts are developed in this chapter to solve the problem to optimality.

In Section 4.1 we first describe the problem in more detail. An integer linear programming model to solve the problem is given in Section 4.2. Another solution approach is given in Section 4.3. The problem is formulated as a set partitioning problem. We compare both methods in Section 4.4. To illustrate the way in which both solution methods can be used, a simple example is discussed in Section 4.5. Finally, some concluding remarks are presented in Section 4.6.

4.1 Problem description

In this chapter we consider an arbitrary material handling centre in which the transportation process is decoupled by buffer areas of other processes, like the storage process.

Unit loads are transported by vehicles from one pick-up and delivery point to another one. Other types of equipments, such as storage and retrieval machines, place loads into buffers, that are located at each pick-up and delivery point. In these buffers jobs are waiting for transport. A buffer is assumed to have a fixed capacity. Similar to the case in Chapter 3 it is known for all jobs in advance at which release time they will be placed in the buffer by a machine. In this buffer the jobs can wait for transport, but when the buffer is full and a new job should be placed in the buffer, at least one of them has to be transported first. To create space in the buffer and to avoid congestion, we assume that the earliest arrived job has to be transported first. Consequently, there is, next to a first time point of transport (i.e. moment the job is placed in the buffer), a moment at which the job has to be transported at the latest. This deadline equals the release time of the new job that should be placed in the full buffer. As a result, each job that has to be transported from one pick-up and delivery point to another one has a *time-window* (release time and due time) in which the job should be transported. The time-window of each job can be determined in advance, because the release times of each job are known in advance. In the model the order in which jobs are transported can be chosen freely as long as each job is transported at the latest at its due time. In this way the situation that a storage and retrieval machine needs to wait to place a load in the buffer will not occur. The problem is now to determine the minimum number of vehicles required to transport all jobs within their time-windows.

In practice, these buffers can be located, for example, on the ground or at the end of a conveyor. In the case that loads are placed on the ground, vehicles, capable of lifting loads by themselves are used to transport loads. If lifting vehicles are used, it is not necessary for, for example, the system used for the storage process (i.e. storage and retrieval system) to place the job immediately on the vehicle. Therefore, the vehicles function more independently of other material handling equipments than in the case where non lifting vehicles are used (see Chapter 3). If loads wait for transport at the end of a conveyor, non lifting vehicles, like AGVs, can be used. The load is transferred to the vehicle at the end of the conveyor. If the buffer is full, the first in line needs to be transported by the vehicle.

The concepts developed in this chapter can be used beforehand the start of the real operation. By making assumptions on, for example, release times of loads vehicle requirements can be estimated. At the start of the operation the estimated number of vehicles can be used. If unexpected events occur, the number of vehicles still can be adjusted.

We make the following assumptions:

1. We have N jobs that need to be transported by vehicles. The availability time of each job is known in advance.
2. There are K pick-up and delivery points (p&d points) for jobs. At the p&d points there are separate buffers for arriving and leaving loads. Thus, there is no interference between arriving jobs and leaving jobs.
3. The capacity of a vehicle is one unit load.
4. The release time (i.e. the time at which the job is placed in the buffer and is available for transport by the vehicle) is known in advance.
5. The due time (i.e. the time at which the job ultimately has to be transported by the vehicle) is known in advance.
6. Due to the buffer areas a vehicle does not have to wait for another free system to complete or begin its job.

7. The position where a job is available for transport and the destination are known in advance.

8. Each buffer area has a fixed capacity.

9. For arriving loads sufficient space is always available in the buffer of the delivery point.

To solve the problem of determining vehicle requirements under time-window constraints we develop new planning concepts. Firstly, we model this problem as an integer linear programming model in Section 4.2. Thereafter, in Section 4.3 we explain in which way the problem can be formulated as a set partitioning problem.

4.2 Model

In this section, a model is developed to solve the problem of determining the minimum number of vehicles in a material handling centre in which the transportation process is uncoupled of other processes by buffer areas.

For each job i we define the following parameters:

a_{i_p} release time, the point in time at which job i $(1 \leq i \leq N)$ is available for transport at the buffer area of origin pick-up point p $(1 \leq p \leq K)$ by a lifting vehicle.

b_{i_p} due time, the point in time at which job i $(1 \leq i \leq N)$ ultimately has to be transported from origin pick-up point p $(1 \leq p \leq K)$ by a lifting vehicle.

$w_{i_{pd}}$ travel time of a full lifting vehicle from the origin pick-up point p $(1 \leq p \leq K)$ of job i $(1 \leq i \leq N)$ to the destination delivery point d $(1 \leq d \leq K)$ of job i. The time to lift and put down a load is incorporated in this travel time.

t_{i_d} arrival time, the point in time at which job i arrives at its destination delivery point d $(1 \leq d \leq K)$. This point in time depends on the travel time of the vehicle from the origin of i to the destination of i and the chosen start time of job i. As a result, an earliest and latest arrival time of each job exist, namely:

t_{if_d} earliest (first) arrival time at destination delivery point d $(1 \leq d \leq K)$ $t_{if_d} = a_{i_p} + w_{i_{pd}}$.

t_{il_d} latest arrival time at destination delivery point d $(1 \leq d \leq K)$ $t_{il_d} = b_{i_p} + w_{i_{pd}}$.

$r_{i_d j_p}$ travel time of an empty vehicle, from the destination delivery point d ($1 \leq d \leq K$) of job i to the origin pick-up point p ($1 \leq p \leq K$) of job j ($r_{i_d j_p} = 0$ if destination of i equals the origin of j).

Furthermore, we define the following decision variable for each job i ($1 \leq i \leq N$):

s_{i_p} start time, the point in time at which a vehicle lifts job i at origin pick-up point p ($1 \leq p \leq K$) and starts transporting the job to its destination: $a_{i_p} \leq s_{i_p} \leq b_{i_p}$

As a result, the arrival time t_{i_d} equals: $t_{i_d} = s_{i_p} + w_{i_{pd}}$.

To use as few vehicles as possible in the material handling centre, it is required to transport several jobs by one vehicle. If two jobs can be transported by the same vehicle, we call them compatible. Job i and j are compatible if:

$$\begin{cases} t_{i_d} + r_{i_d j_p} \leq s_{j_p} \\ a_{j_p} \leq s_{j_p} \leq b_{j_p} \end{cases}$$

Jobs are compatible, if the arrival time of job i plus the travel time to the origin of job j is smaller or equal to the chosen start time of job j. This start time should be chosen within the given time-window for job j.

Consequently, to solve the problem of determining the minimum number of vehicles required to transport all jobs within their time-window, two decisions have to be made, namely:

- Which jobs are transported by the same vehicle?
- At which start time is job i ($\forall\ i$) transported?

To model the problem each time-window $[a_i, b_i]$ can be discretised as shown in Figure 4.1.

By discretising the time-window we have obtained m_i copies of possible start times. Each job i ($1 \leq i \leq N$) can be executed at time points g_{ik} ($1 \leq k \leq m_i$). Because of the fact that for each job an other time-window (with its own size) is defined, m_i varies per job.

With these data we can construct a directed graph $G = (V, A)$ with

$$V = \{g_{11}, \ldots, g_{1m_1}, \ldots, g_{N1}, \ldots, g_{Nm_N}, \text{ where each node represents a time point at which a job can be transported}\}$$

$$A = \{(c, d) : c, d \in V, \text{ job with start time } c \text{ is compatible with another job with start time } d\}$$

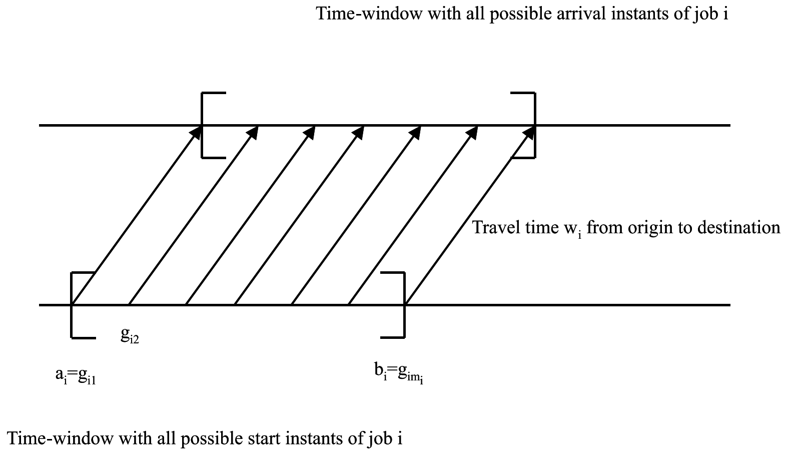

FIGURE 4.1. Discretised time window $[a_i, b_i]$.

Furthermore, we introduce a source node s and a sink node t and add all the arcs (s, c) and (c, t) $\forall c \in V$.

Graph G is acyclic and all nodes can be ordered. A directed path in this network corresponds to a feasible sequence of jobs with their start times, which can be executed by one vehicle.

In Figure 4.2 an example of such a graph is given. Three jobs have to be transported. For each job two possible start times are given. A path in this graph corresponds to a sequence of jobs that can be transported by the same vehicle. By determining the minimum number of directed paths we can obtain the minimum number of vehicles to transport all jobs. However, one extra condition is necessary. Namely, for each job only one start time can be chosen. It is not possible to transport the same job twice. For example, the following directed paths are not a valid solution for the problem:

$$\begin{aligned} s &\rightarrow g_{11} \rightarrow g_{22} \rightarrow t \\ s &\rightarrow g_{21} \rightarrow g_{32} \rightarrow t \end{aligned}$$

If both directed paths are chosen, job 2 is included in two paths en therefore need to be executed twice. As a result, to solve the problem we need to determine the minimum number of directed paths, such that each job is included in exactly one path and that for each job only one start time (modelled with condition (4.4) is chosen.

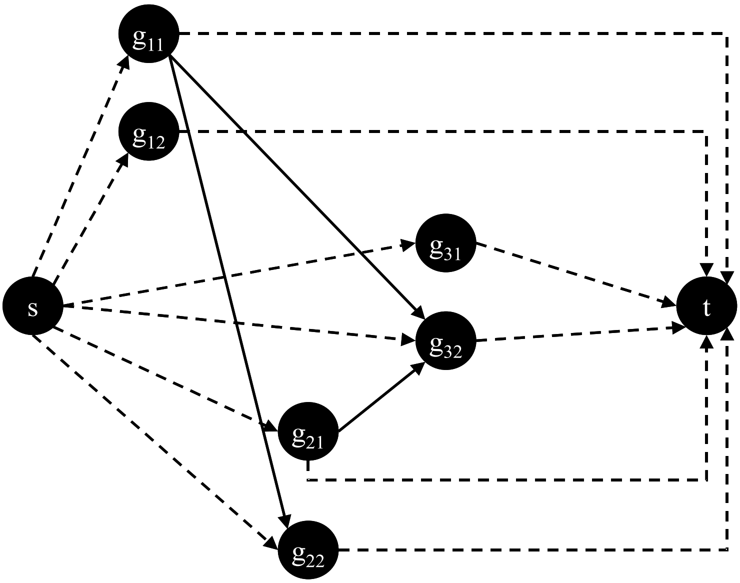

FIGURE 4.2. Example of graph G.

We can formulate the problem as follows as an *Integer Linear Programming Model*:

$$Min\ v$$

$$\sum_{\{c:(s,c)\in A\}} x_{sc} - \sum_{\{c:(c,s)\in A\}} x_{cs} = v \tag{4.1}$$

$$\sum_{\{c:(t,c)\in A\}} x_{tc} - \sum_{\{c:(c,t)\in A\}} x_{ct} = -v \tag{4.2}$$

$$\sum_{\{c:(d,c)\in A\}} x_{dc} - \sum_{\{c:(c,d)\in A\}} x_{cd} = 0\ \forall d \in V \backslash \{s,t\} \tag{4.3}$$

$$\sum_{k=1}^{k=m_i} \sum_{c} x_{cg_{ik}} = 1\ \forall i, 1 \leq i \leq N \tag{4.4}$$

$$0 \leq x_{ij} \leq 1 \qquad \forall (i,j) \in A \tag{4.5}$$

$$x_{ij} \text{ integer} \qquad \forall (i,j) \in A \tag{4.6}$$

The explanation of the equations is as follows:

(4.1) the value of all outcoming flow minus the value of all incoming flow in the source should be equal to v (= minimum number of directed paths), (4.2) the value of all outcoming flow minus the value of all incoming flow in the sink should be equal to $-v$,

(4.3) for all the other nodes the incoming flow should equal the outcoming flow,

(4.4) each job has to be transported only once. Consequently, only one start time can be chosen,

(4.5) the upper bound of the flow equals one and the lower bound equals zero,

(4.6) all flow should be integer.

The minimum number of vehicles to transport all jobs within their time-window can be obtained by solving this *Integer Linear Programming Model*.

4.3 Set partitioning approach

As explained in Section 4.2 the minimum number of lifting vehicles required to transport all jobs in time can be determined by solving the presented Integer Programming model. In this section, we introduce another way to model the problem. We use the same network as defined in the previous section.

A job can be transported only once. Therefore, exactly one start time for each job is chosen and this startpoint is included in exactly one directed path starting in the source and ending in the sink.

Clearly, the objective is to determine the minimum number of directed paths such that only one start time of each job is chosen and that each job is included in exactly one path. We formulate this problem as a *set partitioning* problem. This approach can be used for problems in which each object need to be covered exactly once (see, for example, Glover *et al.* [92]). We use the network formulation from Section 4.2. In this network, we assume n nodes and m arcs are included. Firstly, all possible directed paths from s to t need to be determined.

Therefore, we develop a *search method* based on the *depth first search method* (see, for example, Ahuja *et al.* [2]). For each node the *adjacency list* $A(i)$ of arcs is known.

$$A(i) = \{(i, j) \in A : j \in V\}$$

The search method starts in node s. The first arc from $A(s)$ is added to the directed path starting in s. This arc is called *visited*. In the next node i on this directed path, the first arc from $A(i)$ is added. The algorithm continues in the node reached by this arc. This is repeated until the sink node t is reached. The first path from s to t has been found. The method returns to the last node j before t in the path. The second arc in $A(j)$ is scanned and added to the path. Again this path is followed until t is reached. The second path has been found. If no *unvisited* arcs were left in node j the search method will go back one more node in the path. If we leave a node j all arcs in $A(j)$ become unvisited again. Finally, the search method will return in s. Then, the second arc from $A(s)$ will be scanned and the above mentioned procedure is repeated. This process is repeated until the last arc from $A(s)$ has been visited and all related paths on this route have been found.

We will explain this search method with the example in Figure 4.2.

$A(s) = \{(s, g_{11}); (s, g_{12}); (s, g_{21}); (s, g_{22}); (s, g_{31}); (s, g_{32})\}$,
$A(g_{11}) = \{(g_{11}, g_{22}); (g_{11}, g_{32}); (g_{11}, t)\}$,
$A(g_{12}) = \{(g_{12}, t)\}$,
$A(g_{21}) = \{(g_{21}, g_{32}); (g_{21}, t)\}$,
$A(g_{22}) = \{(g_{22}, t)\}$,
$A(g_{31}) = \{(g_{31}, t)\}$,
$A(g_{32}) = \{(g_{32}, t)\}$.

Figure 4.3 illustrates the search method. It is clear that 9 directed paths from s to t exist in the example in Figure 4.2.

THEOREM 4.1.
All directed paths from s to t can be found by applying the above mentioned search method in $O(nm)$ time.

PROOF
Each node is visited at most n times in this search method. The arcs in $A(i)$ are scanned at most n times. Therefore, the search algorithm examined a total of $\sum\limits_{i \in N} n\,|A(i)| = n \cdot m$ and therefore, terminates in $O(nm)$ time. ■

In Section 4.5 this search algorithm is applied to an example.

Assume p directed paths have been found. In the set partitioning approach we distinguish two parameters, namely the p directed paths and the N jobs. We create an incidence matrix a_{ij} in the following way: each row i $(1 \leq i \leq N)$ represents one of the jobs and each column j $(1 \leq j \leq p)$ represents a directed path. If a start time of the job is included in the directed path, the job is included in the directed path. Therefore, if $a_{ij} = 1$ then job i is included in path j. If $a_{ij} = 0$ then job i is not include in path j. The objective is to minimise the number of paths covering all jobs exactly once.

Identical columns can be deleted from the matrix such that one column of this type remains. This can be done, because of the fact that we are interested in the minimum

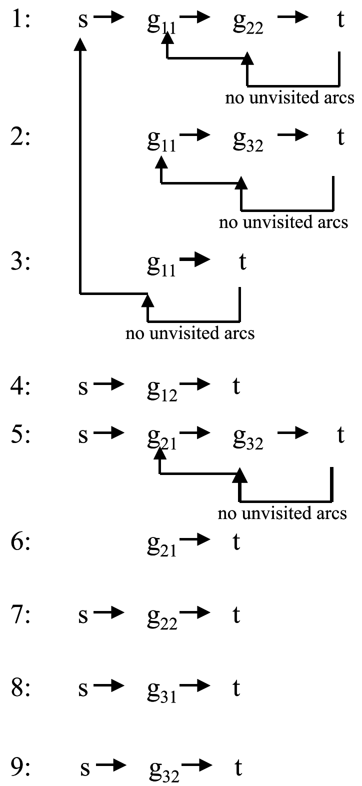

FIGURE 4.3. Example of the search method.

number of vehicles and not directly in the related routes. Furthermore, from the group of identical columns at most one column will be chosen in the solution because of the fact that otherwise jobs are covered by more than one directed path.

We formulate the problem in the following way (x_j has value 1 if path j is used.):

$$\min \sum_{j=1}^{N} x_j \tag{4.7}$$

$$\sum_{j=1}^{N} a_{ij} x_j = 1 \qquad \forall i$$

$$x_j \in \{0, 1\} \qquad \forall j$$

4.4 Comparison of the methods

In this chapter two methods are given for determining vehicle requirements under time-window constraints. Firstly, the problem is formulated as an integer linear programming model and secondly as a set partitioning problem. Before we can use one of these approaches we have to formulate a directed network. In this network each job is represented by multiple nodes, where each node represents a point in time at which the job can be transported. Furthermore, a source and a sink node are included in the network. Two jobs with a certain start time are compatible if they can be transported by the same vehicle. In that case the two corresponding nodes are connected by a directed arc. To solve the problem we have to determine the minimum number of directed paths from source to sink such that one start time of each job is chosen and that each job is included in exactly one path.

The objective of the integer linear programming model is to minimise the number of directed paths. The conditions are set such that it is ensured that each job is included in exactly one path. Furthermore, extra conditions are added to ensure that only one start time of each job is chosen. If the number of jobs grow, the corresponding number of nodes in the network grow. As a result, the number of conditions in the integer linear programming model increase. Consequently, the computation times to determine vehicle requirements increase.

To solve the problem as a set partitioning problem an extra step is required, namely the determination of all possible directed paths between source and sink. With the search method introduced, all possible directed paths are found in polynomial time. Directed paths including exactly the same jobs can be deleted such that only one directed path of this type remains. With a set partitioning integer programming model the minimum number of directed paths covering all jobs exactly once can be determined. Less conditions are needed by deleting similar directed paths.

Summarising, for both approaches first a directed network has to be constructed. After constructing the directed network one can apply an integer linear programming model directly. Alternatively, first a search method is applied, followed by a set partitioning integer programming model. Due to the method of construction, there are fewer conditions in the set partitioning integer programming model than in the regular integer linear programming model. Fewer conditions might result in lower computation times to obtain the solution in the set partitioning problem. However, the total computation time also depends on the time required to apply the search method. For future research it could be interesting to perform a large number of tests on examples from practice to study in which way the total computation times of both methods differ.

4.5 Illustrative example

We consider the same situation as in Section 3.5. Jobs have to be transported between three pick up and delivery points. Lifting vehicles are used to transport jobs from one pick up and delivery point to another one. In Table 4.1. travel times for full and empty lifting vehicles between p&d points are given. We assume that the travel times are the same for both directions.

	$1 \leftrightarrow 2$	$1 \leftrightarrow 3$	$2 \leftrightarrow 3$
full vehicle	4	5	3
empty vehicle	2	3	1

TABLE 4.1. Travel times for full and empty vehicles between pick-up and delivery points.

At the pick up and delivery points buffer areas with a capacity of one for leaving loads are located. Clearly, a load can wait in the buffer area until a new load needs to be placed in it. Therefore, each load has a release time and a due time. It is assumed that for arriving loads always sufficient space is available in the buffer. In this simplified example we consider a set of three loads. The characteristics of these loads are given in Table 4.2. The due time of load 2 equals the release time of load 3. The due times of the load 1 and 3 depend on the release times of new jobs at the related p&d points. These new jobs are not considered in this example. Therefore, we assume that the due time of load 1 equals 3 and that the due time of load 3 equals 13. The earliest t_{if} and latest arrival time t_{il} are determined by using the data from Table 4.1.

load i	origin p	destination d	release a_{i_p}	due b_{i_p}	earliest t_{if_d}	latest t_{il_d}
1	1	3	1	3	6	8
2	2	3	5	11	8	14
3	2	1	11	13	15	17

TABLE 4.2. Load characteristics.

Firstly, we discretise for each job i the time-window $[a_i, b_i]$. In Table 4.3. all possible start points for each job are presented.

load i	possible start times
1	1,2,3
2	5,6,7,8,9,10,11
3	11,12,13

TABLE 4.3. Possible start times for each job i.

As a result job 1 can be executed at time points g_{11}, g_{12} and g_{13}. For job 2 the time points $g_{21}, g_{22}, g_{23}, g_{24}, g_{25}, g_{26}, g_{27}$ exist. Job 3 can start at g_{31}, g_{32} and g_{33}. For each job one of these start times need to be chosen such that the number of vehicles required is minimised.

Loads can be transported by the same vehicle if the arrival time of load i plus the travel time to the origin of load j is smaller than or equal to the chosen start time of job j.

A network is constructed by representing all possible startpoints by a node, adding a source s and sink t. From Tables 4.2 and 4.3 we derive $A(i)\ \forall i \in V$. All these arcs in $A(i)\ \forall i \in V$ are added between the nodes in the network.

$$
\begin{aligned}
A(s) &= \{(s,g_{11});(s,g_{12});(s,g_{13});(s,g_{21});(s,g_{22});(s,g_{23});(s,g_{24});\\
&\quad (s,g_{25});(s,g_{26});(s,g_{27});(s,g_{31});(s,g_{32});(s,g_{33})\\
A(g_{11}) &= \{(g_{11},t);(g_{11},g_{23});(g_{11},g_{24});(g_{11},g_{25});(g_{11},g_{26});\\
&\quad (g_{11},g_{27});(g_{11},g_{31});(g_{11},g_{32});(g_{11},g_{33})\}\\
A(g_{12}) &= \{(g_{12},t);(g_{12},g_{24});(g_{12},g_{25});(g_{12},g_{26});\\
&\quad (g_{12},g_{27});(g_{12},g_{31});(g_{12},g_{32});(g_{12},g_{33})\}\\
A(g_{13}) &= \{(g_{13},t);(g_{13},g_{25});(g_{13},g_{26});(g_{13},g_{27});\\
&\quad (g_{13},g_{31});(g_{13},g_{32});(g_{13},g_{33})\}\\
A(g_{21}) &= \{(g_{21},t);(g_{21},g_{31});(g_{21},g_{32});(g_{21},g_{33})\}\\
A(g_{22}) &= \{(g_{22},t);(g_{22},g_{31});(g_{22},g_{32});(g_{22},g_{33})\}\\
A(g_{23}) &= \{(g_{23},t);(g_{23},g_{31});(g_{23},g_{32});(g_{23},g_{33})\}\\
A(g_{24}) &= \{(g_{24},t);(g_{24},g_{32});(g_{24},g_{33})\}\\
A(g_{25}) &= \{(g_{25},t);(g_{25},g_{33})\}\\
A(g_{26}) &= \{(g_{26},t)\}\\
A(g_{27}) &= \{(g_{27},t)\}\\
A(g_{31}) &= \{(g_{31},t)\}
\end{aligned}
$$

$$
\begin{aligned}
A(g_{32}) &= \{(g_{32}, t)\} \\
A(g_{33}) &= \{(g_{33}, t)\} \\
A(t) &= \emptyset
\end{aligned}
$$

The following integer linear programming model is formulated:

$$
\begin{aligned}
&Min\ v \\
v &= x_{sg_{11}} + x_{sg_{12}} + x_{sg_{13}} + x_{sg_{21}} + x_{sg_{22}} + x_{sg_{23}} + x_{sg_{24}} \\
&\quad + x_{sg_{25}} + x_{sg_{26}} + x_{sg_{27}} + x_{sg_{31}} + x_{sg_{32}} + x_{sg_{33}} \\
-v &= x_{g_{11}t} + x_{g_{12}t} + x_{g_{13}t} + x_{g_{21}t} + x_{g_{22}t} + x_{g_{23}t} + x_{g_{24}t} \\
&\quad + x_{g_{25}t} + x_{g_{26}t} + x_{g_{27}t} + x_{g_{31}t} + x_{g_{32}t} + x_{g_{33}t} \\
0 &= x_{g_{11}g_{23}} + x_{g_{11}g_{24}} + x_{g_{11}g_{25}} + x_{g_{11}g_{26}} + x_{g_{11}g_{27}} + x_{g_{11}g_{31}} \\
&\quad + x_{g_{11}g_{32}} + x_{g_{11}g_{33}} + x_{g_{11}t} - x_{sg_{11}} \\
0 &= x_{g_{12}g_{24}} + x_{g_{12}g_{25}} + x_{g_{12}g_{26}} + x_{g_{12}g_{27}} + x_{g_{12}g_{31}} + x_{g_{12}g_{32}} \\
&\quad + x_{g_{12}g_{33}} + x_{g_{12}t} - x_{sg_{12}} \\
0 &= x_{g_{13}g_{25}} + x_{g_{13}g_{26}} + x_{g_{13}g_{27}} + x_{g_{13}g_{31}} + x_{g_{13}g_{32}} + x_{g_{13}g_{33}} \\
&\quad + x_{g_{13}t} - x_{sg_{13}} \\
0 &= x_{g_{21}g_{31}} + x_{g_{21}g_{32}} + x_{g_{21}g_{33}} + x_{g_{21}t} - x_{sg_{21}} \\
0 &= x_{g_{22}g_{31}} + x_{g_{22}g_{32}} + x_{g_{22}g_{33}} + x_{g_{22}t} - x_{sg_{22}} \\
0 &= x_{g_{23}g_{31}} + x_{g_{23}g_{32}} + x_{g_{23}g_{33}} + x_{g_{23}t} - x_{sg_{23}} - x_{g_{11}g_{23}} \\
0 &= x_{g_{24}g_{32}} + x_{g_{24}g_{33}} + x_{g_{24}t} - x_{sg_{24}} - x_{g_{11}g_{24}} - x_{g_{12}g_{24}} \\
0 &= x_{g_{25}g_{33}} + x_{g_{25}t} - x_{sg_{25}} - x_{g_{11}g_{25}} - x_{g_{12}g_{25}} - x_{g_{13}g_{25}} \\
0 &= x_{g_{26}t} - x_{g_{11}g_{26}} - x_{g_{12}g_{26}} - x_{g_{13}g_{26}} \\
0 &= x_{g_{27}t} - x_{g_{11}g_{27}} - x_{g_{12}g_{27}} - x_{g_{13}g_{27}} \\
0 &= x_{g_{31}t} - x_{g_{11}g_{31}} - x_{g_{12}g_{31}} - x_{g_{13}g_{31}} - x_{g_{21}g_{31}} - x_{g_{22}g_{31}} \\
&\quad - x_{g_{23}g_{31}} \\
0 &= x_{g_{32}t} - x_{g_{11}g_{32}} - x_{g_{12}g_{32}} - x_{g_{13}g_{32}} - x_{g_{21}g_{32}} - x_{g_{22}g_{32}} \\
&\quad - x_{g_{23}g_{32}} - x_{g_{24}g_{32}} \\
0 &= x_{g_{33}t} - x_{g_{11}g_{33}} - x_{g_{12}g_{33}} - x_{g_{13}g_{33}} - x_{g_{21}g_{33}} - x_{g_{22}g_{33}} \\
&\quad - x_{g_{23}g_{33}} - x_{g_{24}g_{33}} - x_{g_{25}g_{33}} \\
1 &= x_{sg_{11}} + x_{sg_{12}} + x_{sg_{13}}
\end{aligned}
$$

$$
\begin{aligned}
1 &= x_{g_{11}g_{23}} + x_{g_{11}g_{24}} + x_{g_{11}g_{25}} + x_{g_{11}g_{26}} + x_{g_{11}g_{27}} + x_{g_{12}g_{24}} \\
&\quad + x_{g_{12}g_{25}} + x_{g_{12}g_{26}} + x_{g_{12}g_{27}} + x_{g_{13}g_{25}} + x_{g_{13}g_{26}} + x_{g_{13}g_{27}} \\
&\quad + x_{sg_{21}} + x_{sg_{22}} + x_{sg_{23}} + x_{sg_{24}} + x_{sg_{25}} + x_{sg_{26}} + x_{sg_{27}} \\
1 &= x_{g_{11}g_{31}} + x_{g_{11}g_{32}} + x_{g_{11}g_{33}} + x_{g_{12}g_{31}} + x_{g_{12}g_{32}} + x_{g_{12}g_{33}} \\
&\quad + x_{g_{13}g_{31}} + x_{g_{13}g_{32}} + x_{g_{13}g_{33}} + x_{g_{21}g_{31}} + x_{g_{21}g_{32}} + x_{g_{21}g_{33}} \\
&\quad + x_{g_{22}g_{31}} + x_{g_{22}g_{32}} + x_{g_{22}g_{33}} + x_{g_{23}g_{31}} + x_{g_{23}g_{32}} + x_{g_{23}g_{33}} \\
&\quad + x_{g_{24}g_{32}} + x_{g_{24}g_{33}} + x_{g_{25}g_{33}} + x_{sg_{31}} + x_{sg_{32}} + x_{sg_{33}} \\
0 &\leq x_{ij} \leq 1 \ \forall (i,j) \in A \\
x_{ij} & \quad \text{integer } \forall (i,j) \in A
\end{aligned}
$$

By solving this integer linear programming model the following x_{ij} have a value unequal to zero.

$$
\begin{aligned}
x_{sg_{11}} &= 1 \\
x_{g_{11}g_{23}} &= 1 \\
x_{g_{23}g_{31}} &= 1 \\
x_{g_{31}t} &= 1
\end{aligned}
$$

As a result, these three jobs can be transported within their time-window with one lifting vehicle.

As described in Section 4.3 this problem can also be presented as a set partitioning problem. Firstly, we have to search for all directed paths from the source s to the sink t. Therefore, we use the search method explained in Section 4.3. The resulting directed paths are given in appendix B.

In this network 56 directed paths from s to t exist. For each directed path we know which jobs are included in the path. With these data we can construct an incidence matrix a_{ij}, where i $(1 \leq i \leq 3)$ represents a job and j $(1 \leq j \leq 56)$ represents a directed path.

Columns 1-14:

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1
\end{pmatrix}
$$

Columns 15-28:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Columns 29-42:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Columns 43-56:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

By deleting identical columns the following matrix remains:

$$a_{ij} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

By using this incidence matrix a_{ij} in Equation (4.7) the minimum number of vehicles required to transport each job within its time-window can be determined. Clearly, by using path 3 all jobs are included in one path and the minimum number of vehicles equals 1.

4.6 Concluding remarks

Vehicles are used to transport loads from buffers at their origin to their destination. At a certain time (release time) a load is placed into a buffer area by another type of material handling equipment. This buffer has a fixed capacity. If the buffer is full the oldest load must have been picked up by a vehicle before a new load arrives. Consequently, except for a release time each job also has a due time. Somewhere, within this time-window the job must be transported. By discretising this time-window for each job a number of possible start points is obtained. The objective is to minimise the number of lifting vehicles such that each load is transported at a start time within its time-window. In this chapter, we have developed two methods to model this problem.

Firstly, the problem is formulated as an integer linear programming model. In this model, the objective is to minimise the number of directed paths from the source to the sink covering each job. A second condition is formulated, such that each job is transported at only one start time. A solution of the integer linear programming model indicates the minimum number of vehicles such that each job is transported and such that for each job only one start time is chosen.

Secondly, we formulate the problem as a set partitioning problem. This approach can be used for problems in which it is required to cover each object only once. In this case, it is required to cover each job with only one start time by only one directed path. In the first step of the solution approach we determine all directed paths in the network. A search method is developed which gives the number of directed paths and the exact contents of each path in polynomial time. With these directed paths and jobs an incidence matrix a_{ij} is presented. If a directed path j visits a start time of a certain job i the value of a_{ij} equals 1. Otherwise, the value of a_{ij} equals zero. Using this incidence matrix the minimum number of directed paths covering all jobs can be determined.

Both methods are applied on a simple example to show in which way the solution approaches can be applied. In Chapter 8 of this thesis the integer linear programming formulation is used to solve the problem of the determination of the minimum number of lifting vehicles at a container terminal.

Part II

Storage and retrieval machines

5

Some literature on sequencing storage and retrieval requests

Products can be temporarily held in inventory in material handling centres. Arriving products are stored in a storage area. To complete a customer order, products are retrieved from the storage area. Complete pallets, cases or just individual items are stored and retrieved. To perform these logistics activities material handling equipment might be used. To store and retrieve unit loads, for example pallets, (automated) storage and retrieval machines can be used in warehouses. A unit load, the unit to be moved, consists of material in, on or grouped together by something, like a pallet or container (see also Tompkins *et al.* [265]). Unit load machines usually perform a storage and retrieval request in one route starting and ending at a pick up and delivery point. Namely, after receiving a unit load at the pick up and delivery point, the machine stores the load at the right location. Thereafter, the machine travels empty to a retrieval request and transports it to the pick up and delivery point.

Orderpickers travel/walk with equipment, like pedestrian stackers or counterbalance trucks, through the storage area to collect items. Retrieval and storage requests are usually not combined. At a depot, the orderpicker receives a list with items to be picked. Usually, multiple items are retrieved in one route through the storage area starting and ending at the depot.

For both cases, a similar control problem can be distinguished. Namely, the sequencing of retrieval (and storage) requests such that the travel distance of the machine or the orderpicker is minimised. In this chapter, we discuss literature concerning control concepts for both sequencing problems.

Firstly, we consider the case of unit load requests. Most literature concerns automated storage and retrieval systems. Section 5.1 examines some specific characteristics of automated storage and retrieval systems. In Section 5.2 we discuss literature on control concepts for the sequencing of unit load storage and retrieval requests. Secondly, we study the problem of determining shortest routes for orderpickers to collect a given number of item retrievals. This sequencing problem is discussed in Section 5.3. Section 5.3 also explains in detail a dynamic programming algorithm to determine optimal routes for orderpickers to retrieve items from storage. This algorithm is extended in Chapter 6 of this thesis. Furthermore, other literature is mentioned on algorithms and heuristics to schedule item retrievals. Concluding remarks and open research questions are described in Section 5.4. Finally, in Section 5.5 it is presented in which way three research questions can be combined into a new research question, namely the scheduling of storage and retrieval requests of a unit load storage and retrieval machine working in multiple aisles. A new control concept addressing this problem is developed in Chapter 6.

5.1 Automated storage and retrieval systems

Automated storage and retrieval systems (*AS/RS*) are widely used in material handling centres. The main components of an AS/RS are racks and cranes. Unit loads are stored in the racks. The racks are placed alongside an aisle. In the aisle a fully automated crane takes care of the storage and retrieval of unit loads. This crane travels on rails through the aisle. An AS/RS consists of one or more parallel aisles. Often, each aisle is served by another crane. However, it might also occur that one crane serves multiple aisles. At the end(s) of an aisle the crane can travel from one aisle to another.

The crane can move horizontally and vertically at the same time. The travel time of the crane in an aisle is equal to the maximum of the horizontal and vertical travel time (*Chebysev distance metric*). Unit loads available for storage arrive at the input station of the AS/RS. These unit loads are delivered to this station by other material handling systems, like AGVs, conveyors or forklift trucks. Unit loads wait for their turn to be stored at a conveyor. In practice unit loads are usually stored according to a *first come first served* base. The assignment of unit loads to a certain storage location can be made in various ways. For example, products can be stored randomly over the racks, or they are stored at a dedicated location.

Unit loads that are retrieved from storage will be transported by a crane to an output station. Thereafter, another material handling system routes the unit load(s) to its (their) destination. The input and output station can be placed at the same level at the same end of the aisle. However, they can also be positioned at separated positions. In designing the system the number and exact location(s) of input and output stations are some of the decisions that need to be made.

Commonly, the capacity of the crane is one load. Therefore, an AS/RS operates usually in two ways, namely in a *single command cycle* or in a *dual command cycle*. In a *single command cycle* the crane performs either a single storage or a single retrieval request. The *storage cycle time* is, therefore, equal to the sum of the time to pick up a load at the input station, the time to travel to the storage location, the time to place the load in the rack and the time to return to the input station. The *retrieval cycle time* can be defined in the same way. An AS/RS performs both a storage and a retrieval request in a *dual command*. In this case the *cycle time* is defined as the sum of the time to pick up the load, the time to travel to the storage location and store the load, the empty travel time (*interleaving time*) from the storage location to the retrieval location and the time to pick the unit load and transport it to the output station. It is clear that the total time to perform all storage and retrieval requests reduces if dual commands are performed.

A number of variants on AS/RSs exist. We will discuss some of them briefly. To overcome the problem of the unit load capacity, *multiload AS/RSs* (*dual shuttle AS/RSs*) exist. Multiple load positions on the crane enable the system to perform a storage and a retrieval at the same location. The cycle time is reduced by eliminating the interleaving time. In warehouses, where small items are stored, *miniload AS/RSs* are used. Bins, containing multiple identical items, are stored in the racks. At the end of an aisle an orderpicker picks the requested amount of items from a bin. The crane transports the box to the orderpicker and stores the previous box in the rack. Clearly, the current retrieval request is the storage request in a next cycle of the crane. Also, in a miniload AS/RS travel times can be reduced by performing dual commands.

The use of AS/RSs in material handling centres has several advantages compared to non-automated systems, like savings in labor costs, savings in floorspace, increased reliability and reduced error rates. Disadvantages are increased maintenance costs, less flexibility and higher investments in control systems. To profit from these advantages and to overcome these disadvantages, various planning and control decisions need to be made. In designing the system questions like, how high and deep are the racks?, how many aisles are required and how many input/output stations should be incorporated in the system? should be answered. The objective is to design an AS/RS in such a way that, among other things, costs are minimised and operational and physical constraints are met. Literature on the design of an AS/RS includes [183], [226] and [276]. The design of a multi-aisle AS/RS system with a single storage/retrieval machine is discussed in Hwang and Ko [111]. Bozer and White [31] discuss the design of a miniload AS/RS.

Combined with design questions attention could simultaneously be paid to control problems of the system. An overview of literature on design and control policies for AS/RSs is given in the following papers [117], [227], [229] and [268]. In finding efficient control policies for an AS/RS expressions for travel times of the crane can be used. Analytical expression for travel times are derived in, for example, [30], [112], [204] and [291]. Control policies for operational decisions, like, where is an incoming load stored?, where is an idle crane positioned?, in which way are storage and retrieval request sequenced?, have to be determined. Various rules for storage assignment are discussed in, for example, [95], [97] and [104]. Literature on idle positioning of cranes includes [66] and [206]. Control policies for sequencing of unit load storage and retrievals will be discussed in more detail in the next section.

5.2 Sequencing of storage and retrieval requests

In this section we discuss literature concerning the control problem of sequencing unit load storage and retrieval requests for an automated storage and retrieval system. Most literature discusses unit load AS/RSs with one crane per aisle and one input/output station. One of the objectives in sequencing storage and retrieval requests is to minimise total travel times (distances) of the crane. We, especially want to minimise empty travel times of the crane. The loaded times, to store or retrieve loads, are namely fixed.

Storage requests are usually not time-critical. The exact point in time at which they are stored is not of much importance for the performance of the system. Storage requests are usually stored according to the first-come-first-served principle. In practice, storages routed to an AS/RS on a conveyor are stored in the order they arrive. In sequencing retrievals due times of retrievals should be met. There is no reason for retrievals to be sequenced in a first-come-first-served order. By sequencing the retrievals in a smart way, improvements in the throughput of the AS/RS might be obtained. The list of retrievals is continuously changing in time . The retrievals are handled and deleted from the list and new retrievals are added. Han *et al.* [101] suggest two ways to deal with this dynamic problem. Firstly, select a block of the most urgent storage and retrieval requests, sequence them and when they are completed select the next block etc. This is called *block sequencing*. Secondly, we can sequence the whole list of requests each time a new request is added and use due times or priorities. We refer to this kind of sequencing as *dynamic sequencing*. The performance of both approaches differs per situation. For example, Eben-Chaime [63] concludes that in a specified non-deterministic environment, the block sequencing strategy might be inappropriate. However, with dynamic sequencing the performance might be uncontrollable due to many changes in the schedule and long times required to develop new schedules. A block sequencing approach is much more transparent and simpler with respect to implementation.

In the sequencing problem two different kinds of cycles can be performed by the AS/RS, namely *single and dual command cycles* (see Section 5.1). The possibility of performing dual command cycles depends on the availability of storage and retrieval requests. If both types of requests are available dual command cycles give advantages with respect to travel times. Some warehouses, however, have patterns in arriving and leaving loads. For example, trucks with incoming loads arrive in the morning and trucks transporting outgoing loads arrive in the evening. In this case cranes might perform single command cycles. If arriving and leaving trucks overlap in time, dual command cycles can be performed. One of the first papers studying single command cycles is by Hausman *et al.* [104]. Graves *et al.* [95] also observe dual command cycles. They conclude that travel times can be reduced by approximately 30% if dual command cycles are performed.

Han *et al.* [101] indicate that in general the problem of optimally sequencing a given list of requests is *NP-hard*. One of the reasons for this complexity is the fact that the set of locations available for storages depends on where previous loads are stored and which products have already been retrieved. However, some special cases can be solved in polynomial time. Lee and Schaefer [165] study the sequencing problem in which storages can be stored at multiple empty storage locations. The locations of these empty slots dynamically change over time by retrieving and storing unit loads. It is assumed that the number of storages equals the number of retrievals. Firstly, the problem is formulated as an assignment problem in which a storage request together with a certain storage location is assigned to a retrieval request to obtain a dual command cycle. The solution of the assignment problem is not feasible if it contains a tour. This happens if a location is used for storage while the retrieval at that location has not been performed. It is proven that if the assignment solution does not have a tour an optimal sequence is obtained. Murthy's ranking algorithm (Murthy [195]) is applied to search the assignment solutions in increasing order of total travel times until a solution without tours is found. This is an optimal solution to the problem. However, for large problems, long computation times are required. In that case heuristics should be used to find a feasible solution. We will discuss some of these heuristics later on in this section.

The special case of block sequencing with dedicated storage, in which a set of storage and retrieval requests is given beforehand and no new requests come in during operation, can be solved to optimality in polynomial time (Van den Berg and Gademann [269]). In this case, for each storage an empty dedicated location is available. Lee and Schaefer [166] solve this problem by formulating it as an assignment problem. The problem can be formulated in this way only if the number of storages is smaller than or equal to the number of retrievals. Van den Berg and Gademann [269] solve the problem without this assumption. Furthermore, their results apply to systems with arbitrary and different positions of the input and output station. Travel times of loaded trips (storage or retrieval trips) are fixed. However, the sequence of loaded travels determines the travel times of empty trips. Therefore, the objective is to minimise the total empty travel time. A network (transportation problem) is formulated in which the left side of the network contains nodes representing the departure positions of empty travels and the right side of the network contains nodes representing the arrival positions of empty travels. Edges are added between nodes of the left and right side of the network if an empty trip can be performed from an endpoint of a loaded trip to the startpoint of another loaded trip. Costs associated with edges in between correspond to the travel time for each empty trip. It is shown that the optimal solution of the transportation problem corresponds to an optimal sequence of storage and retrieval requests. With some extensions this approach can be applied for dynamic sequencing of requests.

For the dynamic sequencing problem various heuristics exist to solve the problem of sequencing requests such that total travel distances are minimised. We will describe some of them briefly. Retrievals can be scheduled on a first-come-first-served basis. Alternatively, the request with the shortest completion time is served first in the *shortest completion time* rule. Linn and Wysk [173] study the performance of these control rules in combination with several rules for storage assignment. They conclude that control rules only impact the performance of the system if the traffic intensity is above a certain level. Han *et al.* [101] describe the *nearest neighbour heuristic*. Pairs of storages and retrievals are chosen in such a way that the distance between the storage and retrieval location is minimal. It is shown that the nearest neighbour heuristic can provide a lower average cycle time than the first come first served rule. Also, the *shortest leg* heuristic is presented. With this rule, the storage and retrieval requests are combined in such a way that no extra distance needs to be travelled to perform a storage. If possible, a free storage location located on the route from the input station to the retrieval location is chosen. As a result, the storage locations close to the input station are filled up first. Only locations far from the input station remain open. It is shown that in a specified situation, therefore, the nearest neighbour heuristic has a better performance than the shortest leg heuristic over the long run.

Eynan and Rosenblatt [74] study the performance of the nearest neighbour heuristic in a class-based storage environment. It is shown that quite significant savings can be obtained in interleaving times by combining this storage assignment rule and control rule. Ascheuer *et al.* [5] study the dynamic sequencing problem. They model the problem as an on-line asymmetric travelling salesman problem. To test the on-line behaviour of different heuristics, like random and greedy, computational experiments were performed. It is suggested that the crane should start work on a schedule resulting from a heuristic. In the mean time, the optimal sequence is solved and replaces the first sequence. Mahajan *et al.* [181] developed a nearest neighbour heuristic for the sequencing problem in a mini-load AS/RS. Contrary to AS/RS current retrieval requests become future storage requests, since loads are returned to their fixed locations after items have been picked from them. Therefore, only a queue of retrieval requests exists, which result in a less complicated problem. Also, the picker at the end of the aisle needs to be incorporated in the model. Retrieval requests are rearranged such that successive requests are located close to each other. Storages and retrievals which are close to each other are paired by the heuristic.

A number of simulation studies have been performed to examine scheduling rules under different conditions. Schwarz *et al.* [235] investigate the performance and predictions of previously developed deterministic models in a stochastic environment. It is found that the results of the models hold in this environment. However, the predictions in improvements are generally larger than the actual improvements. Randhawa and Shroff [216] use simulation to test the performance of scheduling rules in six different layouts. The performance measures used are system throughput and waiting times of requests.

Except for simulation studies, studies with intelligent systems can be executed to select and evaluate control rules for AS/RS. Examples of these types of systems are neural networks, expert systems, artificial intelligence and the Taguchi method. These intelligent systems can be applied to situations with high uncertainty and insufficient information. Furthermore, these systems are capable of learning and adapting to changes in the environment. Wang and Yih [288] use a neural network as control system. The network is able to deal with changes in the configuration of the system and performance measures. As output control strategies for storage assignment, retrieval location selection, queue selection and job sequencing are given. Linn and Wysk [174, 175] developed an expert system, based on decision rules, for control of the AS/RS. Various control policies, like first in first out and first come first served, are incorporated for functions, like determining storage and retrieval locations and job sequencing. The expert system is able to respond flexibly to fluctuations in demand and to maintain performance. Seidmann [237] suggests artificial intelligence for operational control of the AS/RS. The decision variables are which storage and which retrieval are performed. The system is able to adapt itself to changes, like the number of requests and the demand and arrival rates. Finally, Lim *et al.* [171] use the Taguchi method for determining the optimal configuration of operating policies including storage assignment policy and sequencing control. The principle of the Taguchi method is to determine settings of design factors such that the effects of noise factors affecting the performance are minimised.

An extension of the described problem is the problem in which storage and retrieval requests with release and due times need to be scheduled. Lee and Kim [168] propose four heuristics and a mixed integer linear programming model for solving the problem of minimising the sum of earliness and tardiness penalties in the case that all requests have one common due time. Elsayed and Lee [71] and Elsayed *et al.* [72] discuss methods to schedule retrievals of orders with due times. Orders can be picked individually or in batches. The objective is to minimise the total tardiness per group of orders. Linn and Xie [176] simulate an assembly system in which scheduling rules are used that give the highest priority to storages and retrievals with the least remaining time to their due times.

Most literature discusses AS/RS with one input/output station at one of the ends of an aisle. Randhawa *et al.* [215] study the effects of having an input/output station located at both ends of each aisle. With simulation various scheduling rules, like closest open location policy and nearest neighbor policy, are evaluated for systems with one input/output station at each aisle and two input/output stations at each aisle. It is shown that, compared to systems with one input/output station, several advantages can be obtained in systems with two input/output stations. Namely, reductions in expected crane round trip times and in throughput times.

In Section 5.1 we already discussed various variants of AS/RS. For efficient use of these systems also well performing control rules are required. Keserla and Peters [123] address the problem of sequencing storage and retrieval requests in a dual-shuttle AS/RS. A storage can be placed at the location of the retrieval. Furthermore, due to the capacity of the system quadruple command cycles can be performed by the system. In such a cycle three locations are visited. The authors developed the *minimum perimeter* heuristic. With this heuristic three locations are combined in one route if the perimeter of their triangle is minimal.

5.3 Sequencing of multiple item retrievals in a single route

Instead of retrieving unit loads, various products, forming an order, might be retrieved in one tour. These products can be retrieved manually by orderpickers instead of using automated storage and retrieval machines. Commonly, storage and retrieval requests are not combined in one route. We consider the situation in which products are stored in racks, which are positioned in parallel rows. Orderpickers travel through the aisles between the racks (see also Figure 1.2). The storage area can be divided into a number of *blocks*. *Cross aisles*, used to change aisles are positioned at the front and back of the warehouse and between blocks (see also Chapter 1). At a central point in the warehouse (i.e. *depot*) the *orderpicker* retrieves a list of items, with locations, to be picked. The orderpicker walks or drives through the warehouse to pick the items of the specific order. At the end of his/her route, the orderpicker returns to the depot to deliver the items. The objective is to minimise the length of the route of the orderpicker to collect a certain order.

Various control concepts exist to schedule m item retrievals in n aisles in such a way that the route length is minimised. This order picking routeing problem is a special case of the notorious *travelling salesman problem*. However, Ratliff and Rosenthal [218] showed that the commonly used layout in warehouses is suited to solve the routeing problem in polynomial time. In Chapter 6 the method of Ratliff and Rosenthal [218] is extended to obtain a new control concept. Therefore, this method is discussed in detail in this section.

The authors considered a rectangular warehouse with one block. The layout can be presented by an undirected network as follows:

node v_0 represents the depot
node v_i $(1 \leq i \leq m)$ represents the location of item i in the order
node a_j $(1 \leq j \leq n)$ represents the top end of aisle j
node b_j $(1 \leq j \leq n)$ represents the bottom end of aisle j

The adjacent locations in the warehouse are connected by an unlimited number of parallel undirected arcs. The length of these arcs is equal to the distance (travel time) between those locations. In Figure 5.1 the example in Figure 1.2 is represented as an undirected network. The unlimited number of parallel arcs are presented by two parallel arcs.

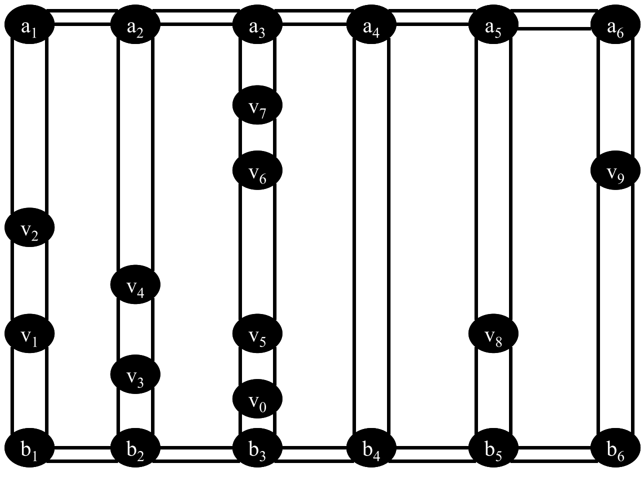

FIGURE 5.1. Network representation of the layout in Figure 1.2.

With this undirected network representation two of the main differences with the travelling salesmen problem are indicated. Firstly, no directed connection (undirected arc) exists between each pair of nodes. Secondly, a node might be visited more than once, because of the unlimited number of arcs available between each pair of nodes. A cycle in the graph is as a feasible orderpicking route if each of the vertices is visited at least once. In Ratliff and Rosenthal [218] a *dynamic programming algorithm* is presented to find an order picking route with minimum length starting and ending at the depot. The authors prove that in a shortest route no more than two undirected arcs between any pair of nodes are used. Each aisle j from left to right is considered in sequence. At each stage of the algorithm the subgraph L_j consisting of vertices a_j and b_j together with everything in the graph to the left of a_j and b_j is considered. To construct an orderpicking route based on subgraphs, three properties of each subgraph should to be known. These properties are represented by equivalence classes. An equivalence class is denoted by the following three components: degree parity of a_j, degree parity of b_j and number of components in the subgraph. The *degree parity* of a node indicates if the value of the degree (i.e. the number of undirected arcs incident with a node) of the node is zero, uneven or even. All possible subgraphs can be divided into one of the seven possible equivalence classes. Each subgraph L_j in an equivalence class has the same possibilities to be completed to an orderpicking route by adding arcs and nodes to the right of aisle j. Therefore, only the shortest of all subgraphs that form an equivalence class, has to be considered for the algorithm. We give a short explanation to the various steps of the algorithm.

Examining subgraph L_j, all possible paths for picking the items in aisle j are considered. These paths are added to the subgraph and new subgraphs are obtained. For each equivalence class the shortest subgraph is chosen and the algorithm continues with this subgraph per class. If the algorithm has finished searching aisle j it is determined in which way the crossover between aisle j and aisle $j+1$ is made. The least distance feasible configuration is selected for each equivalence class and added to the subgraph L_j which results in subgraph L_{j+1}. This procedure continuous until the last aisle n has been considered. The minimum value among the equivalence classes with even degree parity or zero degree and one component is selected as the minimum length orderpicking route through all aisles to collect the specific order. Summarising, starting from the leftmost aisle 1, each aisle and crossover is examined to construct sequentially a minimum length route per equivalence class. The computation time of the algorithm is linear in the number of aisles.

Several extensions of the algorithm Ratliff and Rosenthal [218] exist. De Koster and Van der Poort [54] extend the algorithm by allowing the optimal route to start and end at different locations. Roodbergen and De Koster [225] developed a dynamic programming algorithm capable of finding shortest routes in warehouses with two blocks. Extending this algorithm to more blocks results in higher computation times.

From a practical viewpoint, heuristics are developed to solve the problem of routeing order pickers. Hall [99] discusses a number of heuristics for warehouse with one block. Petersen [208] investigates the impact of several factors, like shape of the warehouse, on the performance of various routeing heuristics.

5.4 Concluding remarks

In material handling centres products are stored for a certain period. They need to be picked from storage to fulfil customer orders or to be used in the production process. Unit loads or individual items are stored and retrieved. Unit load storage and retrieval requests can be executed by unit load automated storage and retrieval systems in automated warehouses. To obtain a higher throughput a storage and a retrieval request can be combined in one trip of the crane. In the literature various methods are described to schedule storage and retrieval requests such that the total distance travelled by the AS/RS is minimised. The situation considered in the majority of the literature concerns a unit load AS/RS working in one aisle with one input/output station. For some specific cases methods exist for optimally sequencing requests. However, the general dynamic sequencing problem is NP-hard to solve and, therefore, heuristics are developed to find feasible schedules. Hardly any attention has been paid to methods for the scheduling of storage and retrieval requests in the case that each aisle has two input/output stations both capable of handling arriving and leaving loads. For example, at a container terminal containers are transferred from cranes to vehicles at both sides of the storage area. Consequently, at both sides storage and retrieval requests need to be performed by the crane. Besides this, in literature less attention has been paid to AS/RSs working in multiple aisles. Therefore and from a practical viewpoint, it might be interesting to develop new control concepts for this case.

In this chapter we also discussed literature on scheduling of item retrievals. Routes have to be determined for orderpickers to travel through multiple aisles to collect all items of a specific order. For warehouses with one or two blocks optimal routes can be found. Otherwise, heuristics should be applied.

In practice, it is not quite realistic to combine item retrieval and item storage in one orderpicking route. Commonly, items are grouped and stored at a unit load. They are retrieved separately or in small batches by an orderpicker. From a theoretical point of view, it might be interesting to see if the method used for determining optimal routes for picking multiple retrievals in multiple aisles can be extended for executing both storage and retrieval requests in multiple aisles.

5.5 Outline of part II of this thesis

In this chapter we described some literature on the scheduling of storage and retrieval requests at warehouses. Firstly, we discussed literature on scheduling of requests at automated storage and retrieval systems. Secondly, we paid attention to the scheduling of item retrievals. In Section 5.4 it was concluded that some research questions for both subjects are still interesting to be studied. In Chapter 6 we develop an algorithm to schedule storage and retrieval requests for a unit load storage and retrieval machine (SRM) working in multiple aisles. With this method we solve several of the questions mentioned in Section 5.4. We observe aisles with input/output stations at both ends of the aisle. An algorithm is developed to obtain the least distance required for a unit load SRM to perform all storage and retrieval requests in one aisle with two input and output stations. Furthermore, we want to sequence the storage and retrieval requests in multiple aisles. Therefore, we extend the dynamic programming algorithm of Ratliff and Rosenthal [218]. A feasible solution of the new algorithm indicates a route for the SRM through multiple aisles to perform storage and retrieval requests in all aisles.

6

Scheduling policy for a unit load storage and retrieval machine in multiple aisles

Storage and retrieval machines (SRM) can be used to store and retrieve unit loads in warehouses. An SRM can operate in one or more aisles. In this chapter, we consider a unit load SRM working in multiple aisles. The machine can change aisles at both ends of an aisle. Furthermore, at each end of the aisle an input/output station is located, which is used to transfer unit loads from the machine to other types of equipments and vice versa. According to Frazelle [83] one of the main reasons for a machine working in multiple aisles is when the storage requirement is higher relative to the throughput requirement. It is indicated that in such a case the throughput requirement does not justify the purchase of an SRM for each aisle. Furthermore, Frazelle [83] indicates that throughput requirements or facility design constraints allow the existence of multiple input/output stations located at various positions (for example at the end) of an aisle. These multiple input/output stations can be used to separate flows of unit loads in the system and to provide additional throughput capacity.

Several practical applications exist in which a single storage and retrieval machine works in multiple aisles with in each aisle multiple input/output stations. At a container terminal containers arrive and leave over land and over sea. These two types of flow can be separated. Each type of flow can arrive at a specified side of the storage area. Consequently, at both sides of the storage area storage and retrieval requests are performed by an SRM. This SRM usually works within multiple aisles. At warehouses (counterbalance) trucks can be used to store and retrieve unit loads in and from multiple aisle in the storage area. Inbound flows arrive from different destinations and outbound flows leave to different destinations. To separate these types of flows at warehouses trucks deliver and receive unit loads at both ends of an aisle. Rail-based SRMs can also perform requests in multiple aisles. Transfer cars can be used to transport SRMs between aisles.

One of the control problems for a unit load SRM working in multiple aisles is the scheduling of storage and retrieval requests such that travel times of the SRM are minimised. The objective is to determine the sequence in which requests in an aisle need to be handled, the direction of travel in each aisle and the sequence in which all aisles need to be visited, such that a route with minimum travel distance is found.

In this chapter we develop a polynomial-time algorithm that solves this sequencing problem to optimality. In this algorithm two new methods are developed. Firstly, the dynamic programming algorithm of Ratliff and Rosenthal [218] for item retrievals is extended such that it can be used in a warehouse with unit load storage and retrieval. As in Ratliff and Rosenthal [218] a picker with infinite capacity was assumed. Secondly, a new polynomial algorithm is presented for solving the scheduling problem of storage and retrieval requests in one aisle with input/output stations at both ends of the aisle. Both algorithms are combined in one algorithm. With this combined algorithm, the optimal sequence of requests in one aisle, the optimal start- and endpoints for the SRM in each aisle (i.e. direction of travel in an aisle) and the optimal order of visiting all aisles is determined. As a result, a route starting and ending at a fixed location (i.e. depot), handling all requests in all aisles is found. Furthermore, it is shown that this route can be found in polynomial time.

In Section 6.1 an exact description of the problem and some assumptions are given. The problem can be modeled as a network. This modelling approach is explained in Section 6.2. The dynamic programming algorithm is presented in Section 6.3. An example is discussed in Section 6.4 to illustrate the algorithm. Section 6.5 gives some concluding remarks.

6.1 Problem description

In this chapter we consider an SRM working in n parallel aisles. Aisles are numbered from left to right. Each aisle has two input/output stations I/O_1 and I/O_2 located at both ends of the aisle. At these I/Os unit loads are transferred from the SRM to other material handling equipment and vice versa. The SRM can change aisles at both ends of the aisle. In each aisle unit loads need to be stored and retrieved by a unit load SRM. At both sides of an aisle unit loads are stored. Figure 6.1 illustrates this layout.

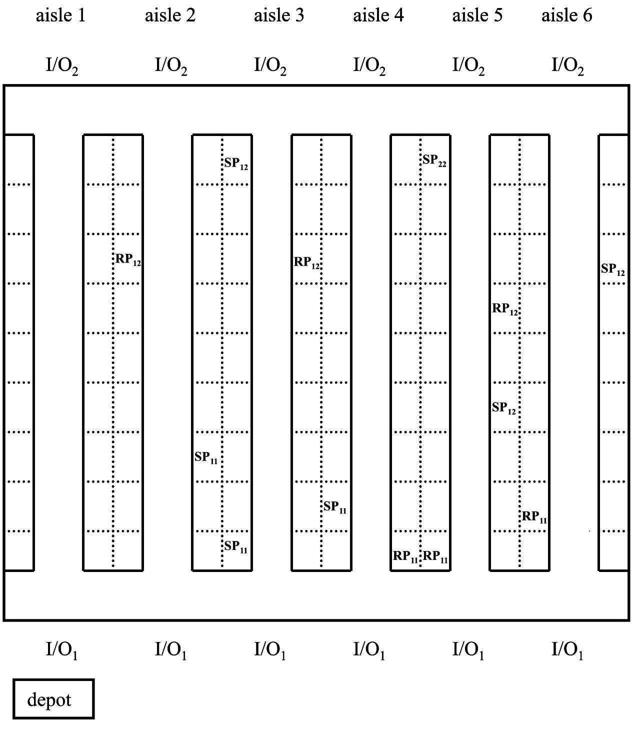

FIGURE 6.1. An example of a layout of a warehouse in which a storage and retrieval machine operates in 6 aisles. Each filled square indicates a location of a request.

We assume that unit loads need to be stored at prespecified locations. We refer to these locations as *storage locations*. Hence, the storage location of each storage request is known in advance. Storage requests in an aisle may be handled in any order. A trip to handle a storage request starts at the I/O at which the request is available and ends at the storage location of the request. Retrieval requests are available at their *retrieval locations*. From this retrieval location the retrieval request is transported to an I/O. Retrieval requests in an aisle may also be handled in any order by the SRM. In total m_s unit loads have to be stored and m_r unit loads have to be retrieved in n aisles. The total task of the SRM thus includes m unit loads, where $m = m_s + m_r$.

In this chapter we apply the block sequencing approach (see also Section 5.2). As a result, a list of storage and retrieval requests for all aisles is known and no new requests are added to this list. These requests need to be sequenced such that the SRM obtains a route with an exact order in which requests in all aisles need to be handled. We assume that the time to pick up or deposit a unit load is independent of the sequence of the requests. Therefore, the objective is to minimise travel times of the SRM. The SRM starts and ends its route at a depot. This depot might be located at any end of an aisle. If it is required that the SRM ends at an arbitrary location, the method of De Koster and Van der Poort [54] can be extended in the same way as we extend in this chapter the method of Ratliff and Rosenthal [218]. Clearly, the problem is to find a shortest route starting and ending in the depot and handling all storage and retrieval requests.

We use the following notation for each aisle j $(1 \leq j \leq n)$:

p_j number of storages in aisle j $(\sum_j p_j = m_s)$

q_j number of retrievals in aisle j $(\sum_j q_j = m_r)$

The storage and retrieval requests can be split in two groups, namely requests with their origin or destination in I/O_1 and requests with their origin or destination in I/O_2.

p_{j1} number of storages in aisle j with origin at I/O_1

p_{j2} number of storages in aisle j with origin at I/O_2, where $p_{j1} + p_{j2} = p_j$

q_{j1} number of retrievals in aisle j with destination at I/O_1

q_{j2} number of retrievals in aisle j with destination at I/O_2, where $q_{j1} + q_{j2} = q_j$

For each request in each aisle j the storage/retrieval location is known.

SP_{e1} storage position corresponding to request originating at I/O_1 for $e = 1, \ldots, p_{j1}$

SP_{f2} storage position corresponding to request originating at I/O_2 for $f = 1, \ldots, p_{j2}$

RP_{g1} retrieval position corresponding to request with destination at I/O_1 for $g = 1, \ldots, q_{j1}$

RP_{h2} retrieval position corresponding to request with destination at I/O_2 for $h = 1, \ldots, q_{j2}$

$t(A, B)$ empty traveltime from the destination of job A to the origin of job B

We make the following assumptions:

1. The travel time of the SRM satisfies the triangle inequality: $T(a,b) \leq T(a,c) + T(c,b)$ $\forall$ points a, b and c.

2. At the start of the route of the SRM the storage location of each storage request is known and initially empty.

3. The origin I/O is known for each storage request.

4. At the start of the route of the SRM all retrieval requests are available.

5. The destination I/O is known for each retrieval request.

6. All requests in an aisle need to be handled before a new aisle can be entered by the SRM. In practice this assumption is quite reasonable due to the relatively large time it takes the SRM to change aisles. In the case that this assumption is not made, then the equivalence classes in the dynamic programming algorithm will depend on the number of requests instead of the number of aisles. In Section 6.3.1 a more detailed discussion is presented on the number of equivalence classes.

7. The SRM starts and ends at the depot.

8. The capacity of the SRM equals one unit load.

With these data a directed network can be constructed.

6.2 Network representation

The n aisles are numbered from left to right. The unit loads arrive and leave at the top or bottom input/output station of the aisle in which they are handled. The route of the SRM starts and ends at the *depot*. The depot is located at one of the ends of one of the aisles. A *directed multiple aisle storage and retrieval tour* indicates in which sequence the requests have to be performed in each aisle, in which direction each aisle has to be travelled through and in which sequence the aisles have to be visited. We define a network to represent this situation. In defining the network we use a different notation from the one used in Section 6.1.

Each location that needs to be visited to perform a storage or retrieval request (SP_{e1}, SP_{f2}, RP_{g1} or RP_{h2}) is represented by a node in the following way:

$$\begin{aligned} \text{nodes } v_{si}\ (i &= 1,\ldots,m_s) \text{ correspond to the locations of storage requests} \\ \text{nodes } v_{ri}\ (i &= 1,\ldots,m_r) \text{ correspond to the locations of retrieval requests} \end{aligned}$$

The input/output stations at both ends of each aisle are also be represented by nodes, namely

$$\begin{aligned} \text{node } a_j\ (j &= 1,\ldots,n) \text{ corresponds to the top end of aisle } j \\ \text{node } b_j\ (j &= 1,\ldots,n) \text{ corresponds to the bottom end of aisle } j \end{aligned}$$

The depot can also be considered as a node in the network. Node v_0 corresponds to the location of the depot.

As assumed, before entering a new aisle all requests in the aisle have to be handled. In contrast to the item retrieval problem of Ratliff and Rosenthal [218] locations are usually not visited adjacently in an aisle. In other words, due to the unit load capacity of the SRM passing a location does not necessarily mean retrieving or storing a unit load at this location. For example, if the crane is on its way to a storage location with a storage request, than several retrieval locations can be passed without performing a retrieval. As a result, an SRM can pass a certain location multiple times without visiting it. Therefore, we model each storage and retrieval location by another node, which can be used to pass the location without performing the request.

$$\begin{aligned} \text{nodes } c_i\ (i &= 1,\ldots,m_s) \text{ correspond to the locations of storage requests} \\ \text{nodes } d_i\ (i &= 1,\ldots,m_r) \text{ correspond to the locations of retrieval requests} \end{aligned}$$

Clearly, the distance between c_i and v_{si} and between d_i and v_{ri} $\forall i$ equals zero.

The travel time of handling all requests in an aisle depends on the order in which requests are handled. The optimal order in which requests are handled depends on the start point and end point of the SRM in an aisle. The travel time for an SRM starting at the bottom end of an aisle, handling all requests in an optimal order and leaving the aisle at the top end of the aisle consequently can differ of the travel time for an SRM starting at the top end of the aisle, handling all requests in an optimal order and leaving at the bottom end of the aisle. Consequently, the related distances for these two situations can be unequal to each other. Therefore, we use, contrary to the item retrieval problem, multiple *directed* arcs between nodes. As a result, in a route of minimal length it might occur that the SRM passes a certain aisle without entering it and enters it the moment it arrives at the other end of the aisle.

Each node c_i $(i = 1, \ldots, m_s)$ and d_i $(i = 1, \ldots, m_r)$ might be passed multiple times and therefore has an unlimited number of incoming and outgoing directed arcs. The length of each arc indicates the distance (i.e. travel time) between the two connected nodes. Between c_i and v_{si} and d_i and v_{ri} $\forall i$ two directed arcs exist. Each request is handled exactly once. Consequently, the nodes v_{si} and v_{ri} $\forall i$ are visited exactly once. Adjacent top ends and adjacent bottom ends of the aisles are connected by an unlimited number of two-way arcs to ensure that the crossover between two aisles can be made multiple times. Due to the unlimited number of arcs between nodes, we can assume that each arc in a route through the aisles handling all requests is visited exactly once. The warehouse with storage and retrieval locations can be represented by a directed graph G consisting of described nodes and directed arcs. Figure 6.2 gives the directed graph of the warehouse in Figure 6.1. For reasons of simplicity the unlimited number of incoming and outcoming arcs are each represented by one single directed arc. The depot v_0 is located at zero distance of node b_4. Also, each node c_i and d_i $\forall i$ are located at zero distance of respectively v_{si} and v_{ri}.

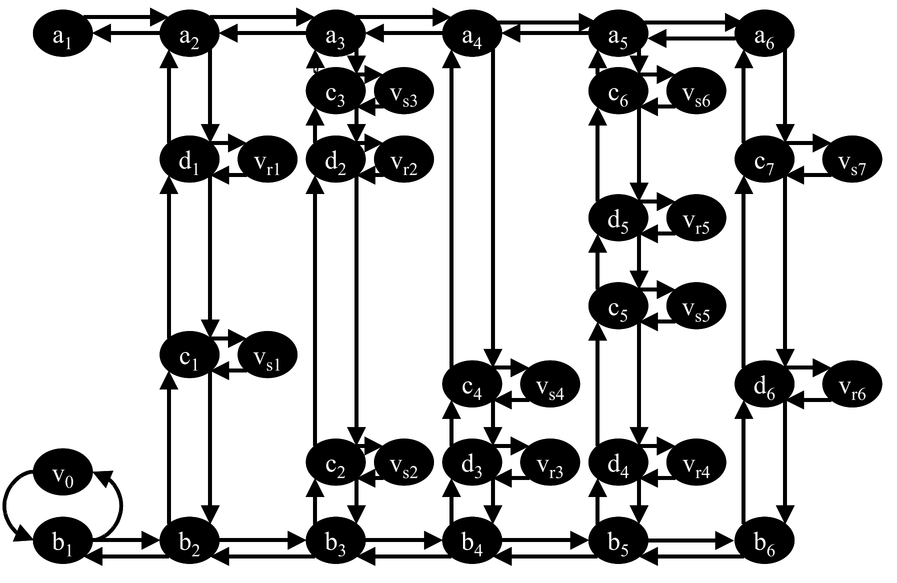

FIGURE 6.2. Network representation of the warehouses from Figure 6.1. Each directed arc represents an unlimited number of directed arcs. Distances between v_0 and b_4, between c_i and v_{si} and between d_i and v_{ri} equal zero.

A directed multiple aisle storage and retrieval tour, starting and ending at the depot, is a directed cycle in network G including all v_{si} and v_{ri} exactly once and all arcs at most once. Any directed multiple aisle storage and retrieval tour can be considered as a directed tour subgraph $T \subset G$ in the way presented in Theorem 6.1. This theorem is a special case of a theorem on *directed Eulerian graphs*. These graphs are strongly connected directed graphs in which every node has equal out- and indegrees (see, for example, Lawler *et al.* [163]). A *directed Eulerian tour* is a directed cycle in the graph which passes through each arc exactly once. The *degree* of a node in a directed graph equals the sum of the number of incoming arcs (i.e. *indegree*) and number of outgoing arcs (i.e. *outdegree*) in the node. Condition (a) indicates that each location with a unit load to be stored or retrieved need to be visited in the tour T. At least one incoming or outgoing arc should visit each unit load location. All unit loads need to be stored and retrieved in one tour. Therefore, the directed subgraph T should consist of one component, which includes all nodes with positive degree. This condition is incorporated in part (b) of Theorem 6.1. The *balance* of a node indicates the value of the number of incoming arcs minus the value of the number of outgoing arcs. Condition (c) requires that the number of incoming arcs equals the number of outgoing arcs.:

THEOREM 6.1.
A directed subgraph $T \subset G$ is a directed tour subgraph if and only if:

(a) all nodes v_0, v_{si} $(i = 1, \ldots, m_s)$ and v_{ri} $(i = 1, \ldots, m_r)$ have positive degree in T,

(b) excluding nodes with zero degree (i.e. nodes a_i, b_i which are not passed), T is a strongly connected directed graph,

(c) each node in T has equal out- and indegrees (balance equals 0).

The length of a directed tour subgraph T equals the sum of the length of all arcs in T. By applying the *tour construction* procedure from Section 6.3.4 a directed multiple aisle storage and retrieval tour can efficiently be derived from any directed tour subgraph T. Consequently, by applying the procedure on a shortest directed tour subgraph T an optimal directed multiple aisle storage and retrieval tour is found. Therefore, we will first discuss an algorithm to construct shortest directed tour subgraphs. This construction procedure is built up according to the steps of the algorithm in Ratliff and Rosenthal [218]. We use to a large extent their notation. Contrary to the network representation in this chapter, Ratliff and Rosenthal [218] apply their algorithm to an undirected network. Some of their theorems and proofs need to be adjusted to be applicable to directed networks. The first step in our construction procedure is the formal definition of partial directed tour subgraphs. It is shown which necessary and sufficient conditions are required to have a partial directed tour subgraph. These conditions and related proofs differ to a large extent from the ones in Ratliff and Rosenthal [218] because of the directed networks. The theorem of equivalence of partial directed tour subgraphs is similar to the proof of Ratliff and Rosenthal [218] in this context. The concept of equivalence is required to construct a dynamic programming algorithm. Equivalence classes, transitions between the states of the algorithm and the corresponding route lengths are defined to describe formally the dynamic programming algorithm. The equivalence classes, transitions and route lengths of Ratliff and Rosenthal [218] hardly can be used in this context because our equivalence classes depend on the number of aisles. In the next section, we show how to determine equivalence classes, transitions and route lengths in our case.

6.3 Constructing shortest directed tour subgraphs

We define L_j^- as the directed subgraph of G consisting of the nodes a_j and b_j together with all arcs and nodes to the left of a_j and b_j. Let M_j be the directed subgraph of G consisting of nodes a_j and b_j together with all nodes and arcs in G between a_j and b_j. L_j^+ is defined as follows: $L_j^+ = L_j^- \cup M_j$. We use L_j to indicate that a result holds if $L_j = L_j^-$ or $L_j = L_j^+$.

For any directed subgraph $L_j \subset G$ a subgraph $T_j \subset L_j$ is a L_j *partial directed tour subgraph* if there exists a *completion* C_j of T_j ($C_j \subset G - L_j$) such that $T_j \cup C_j$ is a directed tour subgraph of G.

Necessary and sufficient conditions for $T_j \subset L_j$ to be a L_j *partial directed tour subgraph* are defined in Theorem 6.2.

THEOREM 6.2.
Necessary and sufficient conditions for $T_j \subset L_j$ to be a L_j *partial directed tour subgraph* are:

(a) for all nodes $v_{si} \in L_j$ and for all nodes $v_{ri} \in L_j$ the degree of v_{si} and v_{ri} is positive in T_j,

(b) every node in T_j, except possibly for a_j and b_j, has equal out- and indegrees (i.e. balance equals zero),

(c) excluding nodes with zero degree T_j has either no connected component, a single connected component containing at least one of the nodes a_j and b_j or two connected components with a_j in one component and b_j in the other.

PROOF

Note: The sum of the balances of all vertices in T_j is zero.

Hence, if condition (b) is satisfied, if one of the nodes a_j or b_j has a balance unequal to zero then both have a balance unequal to zero and $balance\ a_j = -(balance\ b_j)$ to ensure that the sum of all balances equals zero. At most n aisles can be visited by different arcs leaving from a_j, because each aisle is traversed at most once. Therefore, the number of incoming or outgoing arcs from a node a_j or b_j is always smaller than or equal to the number of aisles n. Later on in this chapter, we will narrow this bound for the number of incoming or outgoing arcs. However, for ease of simplicity we use here this bound.

To prove sufficiency, we first consider the case where $balance\ a_j = -balance\ b_j \neq 0$ in T_j. We use induction to the value of the balance to prove that $T_j \cup C_j$ is a directed tour subgraph.

Firstly, suppose $balance\ a_j = 1$. Consequently, $balance\ b_j = -1$. It will be clear that the below mentioned proof also holds if $balance\ b_j = 1$ and $balance\ a_j = -1$. Let C_j be the subgraph of G consisting of all nodes in $G - L_j$, incoming arcs and outcoming arcs in each node in M_k such that each one is visited in a directed path starting and ending in a_j or starting and ending in b_j, for $k = j, j+1, \ldots, n-1$ (for $L_j = L_j^+$, $k = j+1, \ldots, n-1$). Furthermore, C_j consists of incoming and outgoing arcs in each node in M_n such that each one is visited in a directed path starting in a_j and ending in b_j, and one outgoing arc from a_k to a_{k+1} and one incoming arc from $b_{k+1}$to b_k, $k = j, j+1, \ldots, n-1$. $T_j \cup C_j$ then satisfies the conditions of Theorem 6.1. This completion is illustrated in the rightmost part of Figure 6.3 by arcs indicated with ·······>.

Next, assume that $T_j \cup C_j$ for some C_j is a directed tour subgraph if $balance\ a_j = 2, \ldots, n-1$ and consider the case that $balance\ a_j = n$ and $balance\ b_j = -n$. D_j is a subgraph of G consisting of nodes a_j and b_j and all nodes between a_j and b_j, incoming and outgoing arcs in each node in M_j such that each one is visited in a directed path starting in a_j and ending in b_j and $(n-1)$ outgoing arcs from a_j to a_{j+1} and $(n-1)$ incoming arcs from b_{j+1} to b_j. $T_{j+1} = T_j \cup D_j$. Consequently, $balance\ a_{j+1} = n-1$ and $balance\ b_{j+1} = -(n-1)$. The proof holds for the case in which the balance equals $(n-1)$. Therefore, there is a completion C_{j+1} of T_{j+1}. $D_j \cup C_{j+1}$ is a completion of T_j. As a result, $T_j \cup D_j \cup C_{j+1}$ is a directed tour subgraph if *balance*

$a_j = n$ and *balance* $b_j = -n$. Consequently, $T_j \cup C_j$ is a directed tour subgraph if *balance* $a_j = -$*balance* $b_j \neq 0$. Figure 6.3 illustrates this part of the proof for $n = 3$.

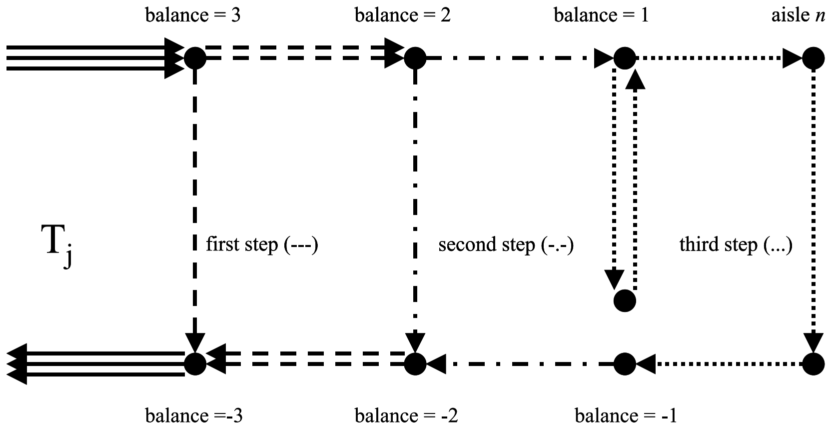

FIGURE 6.3. Illustration of the first part of the proof of Theorem 2.

Next, consider the case where both a_j and b_j have a balance equal to zero in T_j. If both nodes a_j and b_j have a degree of zero in T_j than there are no v_i in L_j. If the *balance* $a_j = 0$ and *degree* $a_j > 0$, than one incoming arc from the left arrives at a_j and one outgoing arc to the left begins at a_j. We first consider the case in which the degree of b_j equals zero. Let D_j be the subgraph containing a_j and b_j and all nodes between a_j and b_j, incoming and outgoing arcs in each node in M_j such that each node is visited in a directed path starting in b_j and ending in a_j, and an outgoing arc from a_j to a_{j+1} and an incoming arc from b_{j+1} to b_j. *balance* $a_{j+1} = 1$ and *balance* $b_{j+1} = -1$. Let C_{j+1} be the same directed subgraph of G described above for the case in which *balance* $a_j = 1$ and *balance* $b_j = -1$. $D_j \cup C_{j+1}$ is a completion of T_j. As a result, $T_j \cup D_j \cup C_{j+1}$ satisfies the conditions of Theorem 6.1. The same proof holds for the case in which the degree of a_j is zero and one incoming arc from the left arrives at b_j and one outgoing arc to the left begins at b_j. Secondly, we consider the case in which the degrees of a_j and b_j are larger than zero and the balance of both nodes equals zero. Then both nodes have one incoming arc from the left and one outgoing arc to the left. D_j and C_{j+1} can be constructed in the same way as in the first case. Consequently, $T_j \cup C_j$ is a directed tour subgraph if *balance* $a_j =$ *balance* $b_j = 0$. An illustration of this part of the proof is given in Figure 6.4.

The proof for necessity is the same as the proof for necessity in Theorem 2 of Ratliff and Rosenthal [218]. Namely, let, for some directed tour subgraph T, T_j be

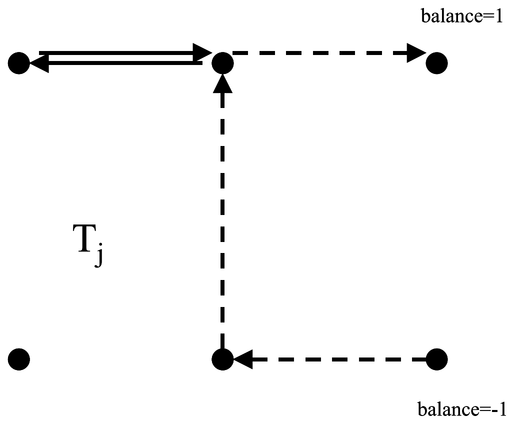

FIGURE 6.4. Illustration of the second part of the proof of Theorem 2.

$T_j = T - T \cap (G - L_j)$. Note that, except for possibly a_j and b_j, no arcs in $T \cap (G - L_j)$ are incident to nodes in T_j. Therefore, since conditions (a) and (b) must hold in T according to Theorem 6.1 , they also hold in T_j. Excluding nodes with zero degree, T is strongly connected. Hence, T_j cannot have a connected component which does not contain either a_j or b_j. ■

Two L_j partial directed tour subgraphs T_j^1 and T_j^2 are equivalent if for any completion C_j such that $T_j^1 \cup C_j$ is a directed tour subgraph, T_j^2 is also a directed tour subgraph. In Theorem 6.3 the sufficient conditions are formulated.

THEOREM 6.3.
Two L_j partial directed tour subgraphs T_j^1 and T_j^2 are equivalent if:

(a) the balance of a_j and b_j is the same for both

(b) excluding nodes with zero degree, both T_j^1 and T_j^2 have no connected components, both have a single connected component containing at least one of a_j and b_j or both have two connected components with a_j in one component and b_j in the other.

PROOF
The proof is similar to the proof of Theorem 3 of Ratliff and Rosenthal [218].

The concept of equivalence is used in a dynamic programming algorithm to construct a minimum directed tour subgraph. All possible subgraphs can be divided in several equivalence classes. In such an equivalence class all subgraphs have the same characteristics. Therefore, we only consider a shortest subgraph in each equivalence class. In the algorithm each aisle j ($j = 1, \ldots, n$) is considered sequentially. In the first step, we consider all subgraphs L_1^+. By adding arcs between aisle 1 and aisle 2 we obtain all possible subgraphs L_2^-. To get L_2^+ partial directed tour subgraphs nodes and directed arcs between a_2 and b_2 are added. This procedure continues until the last aisle n has been considered. A more formal description of the dynamic programming algorithm follows by defining the *equivalence classes*, the *transitions* between states of the algorithm and the *corresponding route lengths* in more detail.

6.3.1 *Equivalence classes*

An equivalence class, containing equivalent L_j partial directed tour subgraphs, is denoted by the following five characteristics: (degree parity of a_j, degree parity of b_j, balance of a_j, balance of b_j, connectivity), where *degree* $a_j =$ *indegree* $a_j +$ *outdegree* a_j and *balance* $a_j =$ *indegree* $a_j -$ *outdegree* a_j. The degree parity is denoted by U in the case that the degree is uneven, by E, in the case that the degree is even and by 0 in the case that the degree is zero. The use of degree parities is almost superfluous. The degree parity follows directly from the value of the balance. If the balance of a node has an uneven value, then the degree of this node is uneven. The degree of a node is even if the value of the balance of this node is even. In the case that the balance equals zero, no remarks can be made on the value of the degree parity. In that case the value of the degree parity is even or zero. Therefore, features 1 and 2 of the equivalence classes are only used in the case that *balance* $a_j =$ *balance* $b_j = 0$. The connectivity indicates the number of components in the graph. The connectivity equals 0, 1 or 2 (see also Theorem 6.2).

The number of equivalence classes depends on the number of aisles. To show this relation, we use Theorem 6.4.

THEOREM 6.4.
The number of incoming and outcoming arcs in nodes a_j and b_j is $\lfloor n/2 \rfloor$ at maximum.

PROOF
The maximal number of incoming and outgoing arcs in a_j and b_j is determined by the number of aisles. By allowing loops in the directed multiple aisle storage and retrieval tour, nodes a_j and b_j could be passed more than once. Figure 6.5 gives an example of a tour passing a_2 twice. From a_1 node a_2 is passed without entering the related aisle. The second time the machine arrives from b_2 in a_2 and the route passes a_2 again. Thus, a_2 is included in one loop including aisle 2 and aisle 4. In the same example, node a_3 is passed three times. Once in the loop circling around node a_2.

Furthermore, a_3 is included in a loop including aisles 3 and 5 and in a loop including aisles 3 and 6.

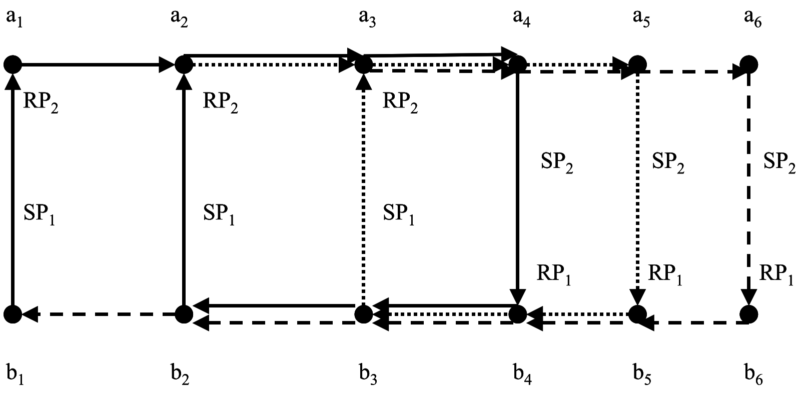

FIGURE 6.5. Example of a tour containing loops.

The maximum number of arcs arriving and leaving a node depends on the maximum number of loops circling around a node. Due to the assumption that, in the case that an aisle is entered, all requests in this aisle should be handled before entering a new aisle, at most one loop might exist per 2 aisles. As a result, the maximum number of loops circling around a_j equals:

$$\begin{cases} \min(n-j, j-1) - 1, & \text{if } n \text{ is odd.} \\ \min(n-j, j-1), & \text{if } n \text{ is even.} \end{cases} \tag{6.1}$$

Following from Equation (6.1) the maximum number of loops in the tour is obtained in the middle aisle. Consequently, in that aisle the maximum number of arcs arriving and leaving equals $\lfloor n/2 \rfloor$. For ease of computation, we assume that each node a_j and b_j might have a maximum of $\lfloor n/2 \rfloor$ incoming and outgoing arcs. ■

In Figure 6.5 a_3 has a maximal number of incoming and outgoing arcs, namely 3. Such a route will be obtained in the case that, for example, in each aisle a storage request and a retrieval request need to be handled. We assume that a storage request is available at the end where the route starts in aisle j and need to be stored at position SP_i. $i = 1$ in the case that the request is available at the bottom end b_j and $i = 2$ in the case that the request is available at the top end a_j. A retrieval is available at the latest location before the other end of the aisle (RP_i) and is designated for that end. $i = 1$ in the case that the request is designated for the bottom end b_j and $i = 2$ in the case that the request is designated for the top end a_j. These locations are illustrated in Figure 6.5. In that case the SRM travels in one trip from one end of the aisle to the other with minimal length combining both requests.

As a result of Theorem 6.4, the balance of a_j and b_j can have the following values: $-\lfloor n/2 \rfloor, (-\lfloor n/2 \rfloor + 1), \ldots, -1, 0, 1, \ldots, \lfloor n/2 \rfloor$. Consequently, there exist $(n+1)$ possible values for the balance of a_j and b_j. However, in the proof of Theorem 6.2. it is noted that in the case that the balance of a_j or the balance of b_j is unequal to zero that then *balance* $a_j = -$*balance* b_j. Using this and the previous theorems it can be proven that the following equivalence classes should be used in the algorithm:

$$\begin{cases} (0,0,0) \\ (0,0,1) \\ (E,E,0,0,1) \\ (E,E,0,0,2) \\ (E,0,0,0,1) \\ (0,E,0,0,1) \\ (\text{balance } a_j, -\text{balance } b_j, 1) \\ \text{for balance } a_j = -\lfloor n/2 \rfloor, (-\lfloor n/2 \rfloor + 1), \ldots, -1 \\ \text{and for balance } a_j = 1, \ldots, \lfloor n/2 \rfloor \end{cases} \tag{6.2}$$

The equivalence class $(0,0,0)$ is possible only if none of the aisles in L_j contains a unit load to be stored or a unit load to be retrieved. $(0,0,1)$ is possible only if none of the aisles in $G - L_j$ contains a unit load to be stored or a unit load to be retrieved. Note that not all classes (the ones with high balances) exist for all nodes, due to the assumption made in the proof of Theorem 6.4. With the equivalence classes from (6.2) we can derive a general analytical expression for the number of equivalence classes. The number of equivalence classes equals $6 + 2 \cdot \lfloor n/2 \rfloor$. The number of equivalence classes depends on the number of aisles because we consider directed networks and networks in which aisles with requests may be passed a number of times before the requests are really handled in the aisle. Therefore the number of equivalence classes differs from the number of equivalence classes used by Ratliff and Rosenthal [218].

From Theorem 6.1. it follows that after considering aisle n in the algorithm, a minimum length directed multiple aisle storage and retrieval tour is the shortest of the partial directed tour subgraphs in the equivalence classes $(0,0,1)$, $(E,E,0,0,1)$, $(E,0,0,0,1)$ and $(0,E,0,0,1)$.

6.3.2 Transitions

In the algorithm two transitions are made, namely in aisles and between aisles. To get L_j^+ partial directed tour subgraphs nodes and arcs between a_j and b_j are added to L_j^-. A minimum length L_j^+ partial directed tour subgraph for equivalence class is constructed by selecting for each equivalence class the least distance configuration. A method to determine the distance of a directed path in an aisle is presented in Section 6.3.3. To make the crossover from aisle j to aisle $j+1$ arcs have to be added between (a_j, a_{j+1}) and/or (b_j, b_{j+1}). In this way L_{j+1}^- partial directed tour subgraphs are obtained. The lengths of these transitions are equal to the sum of the length of the arcs added in the transition. We will discuss these transitions in more detail.

Transition from L_j^- to L_j^+

According to assumption 7 from Section 6.1, all storage and retrieval requests in one aisle are performed before entering a new aisle. As a result, the five ways presented in Figure 6.6 can be used to traverse aisle j.

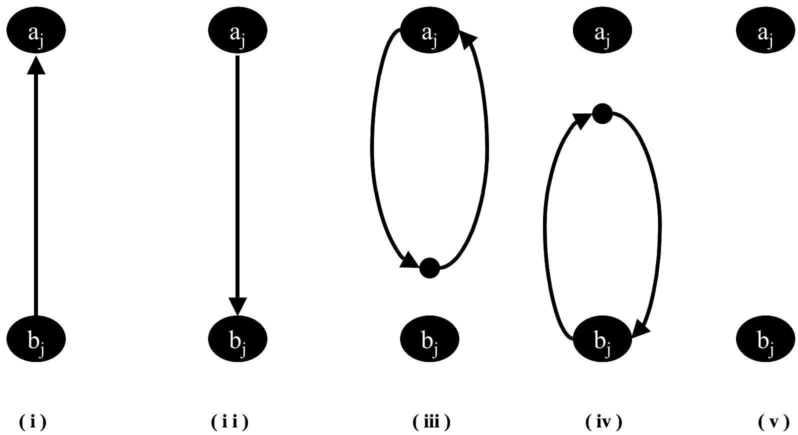

FIGURE 6.6. Five ways to traverse aisle j.

In possibilities (i) and (ii) the directed path through the aisle starts at one side of the aisle and ends in the other one meanwhile executing all requests in the aisle. This means that the nodes a_j and b_j may be visited multiple times in such a directed path. Transitions (iii) and (iv) start and end at the same side of the aisle. However, the other aisle endnode may still be visited while executing requests in the aisle. Transition (v) is only possible if there are no requests in aisle j. The order in which the requests in an aisle are handled is determined with the method described in Section 6.3.3. Ratliff and Rosenthal [218] also consider a transition in which an aisle is visited twice, namely from above and from below. Due to assumption 6 this transition is not allowed in our situation.

Table 6.1. indicates the equivalence classes for L_j^+ if we add the minimum length configurations from Figure 6.6 to the L_j^- minimum length partial directed tour subgraphs for each equivalence class, whereby k = *balance* a_j for *balance* $a_j = -\lfloor n/2 \rfloor, \ldots, -2$ and *balance* $a_j = 2, \ldots \lfloor n/2 \rfloor$. A dash $(-)$ indicates that no feasible solution is obtained by performing this transition.

	(i)	(ii)	(iii)	(iv)	(v)
(0,0,0)	(1,-1,1)	(-1,1,1)	(E,0,0,0,1)	(0,E,0,0,1)	(0,0,0)
(0,0,1)	-	-	-	-	(0,0,1)
(E,E,0,0,1)	-	-	-	-	(E,E,0,0,1)
(E,E,0,0,2)	(1,-1,1)	(-1,1,1)	(E,E,0,0,2)	(E,E,0,0,2)	(E,E,0,0,2)
(E,0,0,0,1)	(1,-1,1)	(-1,1,1)	(E,0,0,0,1)	(E,E,0,0,2)	(E,0,0,0,1)
(0,E,0,0,1)	(1,-1,1)	(-1,1,1)	(E,E,0,0,2)	(0,E,0,0,1)	(0,E,0,0,1)
(-k,k,1)	(-k+1,k-1,1)	(-k-1,k+1,1)	(-k,k,1)	(-k,k,1)	(-k,k,1)
(k,-k,1)	(k+1,-k-1,1)	(k-1,-k+1,1)	(k,-k,1)	(k,-k,1)	(k,-k,1)
(-1,1,1)	(E,E,0,0,1)	(-2,2,1)	(-1,1,1)	(-1,1,1)	(-1,1,1)
(1,-1,1)	(2,-2,1)	(E,E,0,0,1)	(1,-1,1)	(1,-1,1)	(1,-1,1)

TABLE 6.1. Equivalence classes resulting from performing transition L_j^- to L_j^+.

Transition L_j^+ to L_{j+1}^-

By adding the transitions given in Figure 6.7 the crossover between aisle j and aisle $j+1$ can be made. Clearly, other transitions will never lead to an optimal solution.

As explained in Theorem 6.4. the maximum number of incoming and outgoing arcs equals $\lfloor n/2 \rfloor$. This results in $2 \cdot \lfloor n/2 \rfloor$ transitions given in (1a) and further in Figure 6.7. Furthermore, the transitions in (0), (0a), (0b) and (0c) might exist (see also the proof of Theorem 6.2). Consequently, the total number of transitions, therefore, equals $4 + 2 \cdot \lfloor n/2 \rfloor$.

Table 6.2. indicates the equivalence classes for L_{j+1}^- if we add the minimum length configurations from Figure 6.7 to the L_j^+ minimum length partial directed tour subgraphs for each equivalence class whereby k = *balance* a_j for, *balance* $a_j = -\lfloor n/2 \rfloor, \ldots, -1$ and *balance* $a_j = 1, \ldots \lfloor n/2 \rfloor$. A dash $(-)$ indicates that no feasible solution is obtained by performing this transition. For values between $(0c)$ and ka no feasible solution exist.

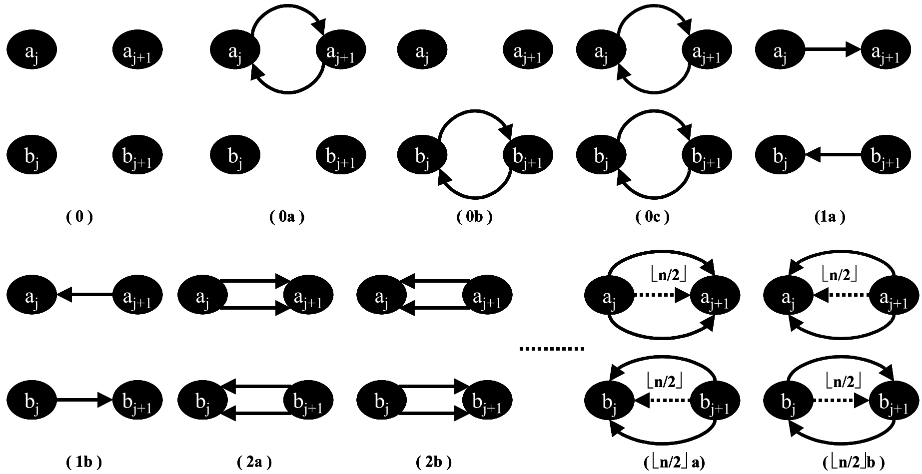

FIGURE 6.7. $4 + 2 * \lfloor n/2 \rfloor$ possible transitions to make the crossover from aisle j to aisle $j + 1$.

	(0)	(0a)	(0b)	(0c)	ka	kb
(0,0,0)	(0,0,0)	-	-	-	-	-
(0,0,1)	(0,0,1)	-	-	-	-	-
(E,E,0,0,1)	(0,0,1)	(E,0,0,0,1)	(0,E,0,0,1)	-	-	-
(E,E,0,0,2)	-	-	-	(E,E,0,0,2)	-	-
(E,0,0,0,1)	(0,0,1)	(E,0,0,0,1)	-	-	-	-
(0,E,0,0,1)	(0,0,1)	-	(0,E,0,0,1)	-	-	-
(-k,k,1)	-	-	-	-	-	(-k,k,1)
(k,-k,1)	-	-	-	-	(k,-k,1)	-

TABLE 6.2. Equivalence classes L^+_{j+1} resulting from performing transitions from L^+_j to L^-_{j+1}.

6.3.3 Determining routes for storages and retrievals in an aisle

In each aisle j $(1 \leq j \leq n)$ a number of storage and retrieval requests need to be performed by the storage and retrieval machine (SRM). An input and output station (I/O) is located at the bottom and top end of each aisle. At this I/O unit loads are transferred from the SRM to an other material handling system (for example, a conveyor) and vice versa. In Figure 6.1 the layout of aisle j can be seen. The storage location of each storage request is known in advance. Furthermore, for each storage request the origin I/O and for each retrieval request the destination I/O is known. The storage and retrieval requests in an aisle may be executed in each sequence as long as all requests are handled before entering a new aisle.

In Figure 6.6 five ways are illustrated to traverse aisle j. In this section, we describe a method to determine the sequence of storage and retrieval requests such that the tour begins and ends at the specified location(s). The objective is to minimise the travel time (or distance) of the SRM in each aisle. Travel times from the input stations to the storage locations and travel times from the retrieval locations to the output stations are fixed and known in advance and independent of the route. However, between each pair of requests the crane has to travel empty to a new location to pick up another request. As a result, the objective is to minimise empty travel times. By performing dual command cycles (see also Section 5.1) travel times might be reduced. Therefore, we try to combine a storage and a retrieval request in a path from an I/O to an I/O. In such a path it is not required to start and end at the same I/O.

Besides the notation already defined in Section 6.1 we define the following:

SP start position of SRM

EP end position of SRM

The locations of SP and EP depend on which transition from Figure 6.6 is chosen.

We consider the case in which at least one of the different types of requests (storages/retrievals) uses both input/output stations. This case is a special case of the Travelling Salesman Problem. Existing methods for aisles with requests available and designated for one input/output station, like Van den Berg and Gademann [269] cannot be used for this case, because solutions might have disjoint subtours. In this section we present a new method to determine an optimal sequence of storage and retrieval requests in polynomial time if both input/output stations are used for at least one of the types of requests.

We consider each request as a job with an origin and a destination that need to be sequenced in a non-prespecified order. This sequence gives a tour for the SRM starting and ending at a specified location and performing all requests. The locations in an aisle are numbered. In this section, we rotate each aisle by 90^0, such that we can consider an aisle as a line from left to right. The start position is numbered with zero and sequentially the locations are numbered to the other end of the aisle, which is an I/O. The number of the end position equals zero if it is equal to the start position. If the end position of the SRM (EP) is at the other end of the aisle, the number of the end position equals the highest number among the locations. I/O_1 is positioned at the same side of the aisle as the start position. I/O_2 is positioned at the end of the aisle at the position with the highest number.

Firstly, we construct a network. For each job i we create a node w_i. Define R_i as the rightmost location and L_i as the leftmost location of job i. For storage requests starting at I/O_1 and for retrieval requests ending at I/O_1 the leftmost location L_i equals zero and the rightmost location R_i equals the position of the storage/retrieval location. For storage requests starting at I/O_2 and retrieval requests ending at I/O_2 the leftmost location L_i equals the position of the storage/retrieval location and the rightmost location R_i equals the position of the location of I/O_2. The direction of travel between the leftmost and rightmost location of a job depends of the kind of job. Storage requests travel from I/O to their storage location and retrieval requests travel from their retrieval location to I/O. In Figure 6.8 R_i and L_i are indicated for all different types of requests.

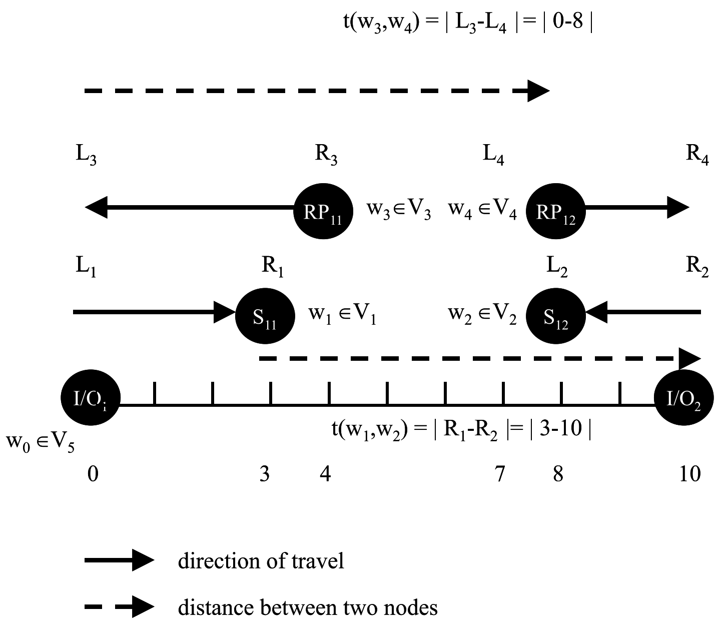

FIGURE 6.8. Leftmost and rightmost locations of all types of jobs. Example of distance calculations according to equation 6.3.

The SRM starts and ends at specified locations. The start location is at position $SP = 0$ and the end position at EP. Create an extra node w_0 for the startposition and an extra node $w_0^{'}$ for the endposition of the crane. Directed arcs connect nodes if the requests can be executed after each other. Furthermore, directed arcs exist from the startpoint w_0 to the requests and from the requests to the endpoint $w_0^{'}$. No directed arcs exist from node i to node i. The empty travel distances of each arc (i, j) depend on the destination of job i and the origin of job j. The distance of w_0 (start position) to an other node i equals the location of the origin of the job. The distance of any job i to $w_0^{'}$ (end position) equals the distance from the destination of job i to EP. These distances are not symmetric. We define $t(f, g)$ as the distance from the destination of request f to the origin of request g. Analytical expressions for these distances $t(f, g)$ are given in Equation 6.3. We construct the following network $G = (V, A)$:

$$V = V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_5$$

$$\begin{aligned}
V_1 &= \{w_k \ (1 \le k \le p_{j1})\} \\
V_2 &= \{w_{p_{j1}+l} \ (1 \le l \le p_{j2})\} \\
V_3 &= \{w_{p_{j1}+p_{j2}+m} \ (1 \le m \le q_{j1})\} \\
V_4 &= \{w_{p_{j1}+p_{j2}+q_{j1}+n} \ (1 \le n \le q_{j2})\} \\
V_5 &= \{w_0, w_0^{'}\}
\end{aligned}$$

Summarising, V_1 represents all storage requests with origin at I/O_1, V_2 represents all storage requests with origin at I/O_2, V_3 represents all retrieval requests with destination at I/O_1, V_4 represents all retrieval requests with destination at I/O_2 and V_5 represents the startposition and endposition of the SRM.

This node representation is indicated in Figure 6.8.

$$\begin{aligned}
A = \{(f, g) : f, g \in V \setminus \{w_0, w_0^{'}\} \quad & \cup \{(w_0, g) : \quad g \in V \setminus \{w_0, w_0^{'}\} \\
& \cup \{\{f, w_0^{'}) : \quad f \in V \setminus \{w_0, w_0^{'}\}\}
\end{aligned}$$

The distances related to the arcs equal:

$$\begin{aligned}
t(f, g) &= |R_f - 0| \quad f \in V_1, g \in V_1 \\
t(f, g) &= |R_f - R_g| \quad f \in V_1, g \in V_2 \\
t(f, g) &= |R_f - R_g| \quad f \in V_1, g \in V_3 \\
t(f, g) &= |R_f - L_g| \quad f \in V_1, g \in V_4
\end{aligned} \tag{6.3}$$

$$
\begin{aligned}
t(f,g) &= |L_f - 0| \quad f \in V_2, g \in V_1 \\
t(f,g) &= |L_f - R_g| \quad f \in V_2, g \in V_2 \\
t(f,g) &= |L_f - R_g| \quad f \in V_2, g \in V_3 \\
t(f,g) &= |L_f - L_g| \quad f \in V_2, g \in V_4 \\
t(f,g) &= 0 \quad f \in V_3, g \in V_1 \\
t(f,g) &= |0 - R_g| \quad f \in V_3, g \in V_2 \\
t(f,g) &= |0 - R_g| \quad f \in V_3, g \in V_3 \\
t(f,g) &= |0 - L_g| \quad f \in V_3, g \in V_4 \\
t(f,g) &= |R_f - 0| \quad f \in V_4, g \in V_1 \\
t(f,g) &= 0 \quad f \in V_4, g \in V_2 \\
t(f,g) &= |R_f - R_g| \quad f \in V_4, g \in V_3 \\
t(f,g) &= |R_f - L_g| \quad f \in V_4, g \in V_4 \\
t(w_0,f) &= L_f \quad f \in V_1, V_4 \\
t(w_0,f) &= R_f \quad f \in V_2, V_3 \\
t(f,w_0^{'}) &= |R_f - EP| \quad f \in V_1, V_4 \\
t(f,w_0^{'}) &= |L_f - EP| \quad f \in V_2, V_3
\end{aligned}
$$

A tour in such a network from w_0 to $w_0^{'}$ visiting all nodes in V can be considered as a connected directed graph in which all nodes in V have indegree one and outdegree one, node w_0 has outdegree one and node $w_0^{'}$ has indegree one. This network can be reformulated as an assignment problem. In a bipartite network we present all nodes from $V_1 \cup V_2 \cup V_3 \cup V_4$ at both sides of the network. Between all these nodes directed arcs from the left to the right side exist. Because of the fact that nodes w_0 and $w_0^{'}$ are always contained in the same cycle we may represent it by the same node, namely w_0. This statement is proven at the end of this section. Node w_0 is presented at the left and right side of the network. The w_0 at the left side is connected with a directed arc to all nodes from $V_1 \cup V_2 \cup V_3 \cup V_4$ at the right side. All nodes from $V_1 \cup V_2 \cup V_3 \cup V_4$ at the left side are connected with a directed arc to w_0 at the right side of the network. An assignment in such a network indicates which requests need to be handled after each other starting in w_0 and ending in w_0. The solution of the assignment problem indicates not necessarily connected directed cycles handling all requests. A solution to the assignment problem is only feasible if the startpoint and endpoint w_0 are contained in the same cycle. Such a solution has the property that the startpoint has an outdegree of one and the endpoint has an indegree of one. If we would allow the startpoint and the endpoint to be part of different cycles then one of the other nodes in the cycle with the startpoint must have an outdegree of zero, since the cycle would end there. This is a contradiction with the fact that all nodes except for the startpoint and endpoint have an indegree and outdegree of one.

The example in Figure 6.8 with startpoint and endpoint of the crane in I/O_1 is translated to an assignment problem in Figure 6.9

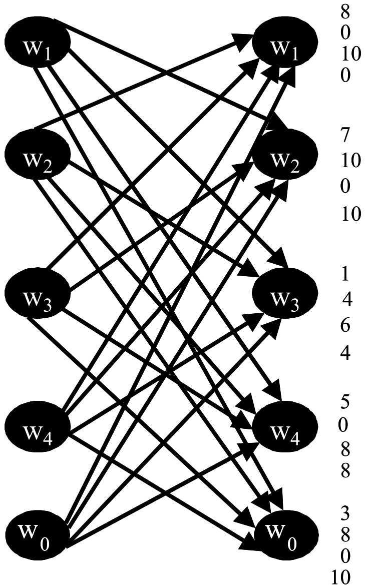

FIGURE 6.9. Example of an assignment problem.

Comparable to many algorithms for special case of the TSP (see Kollias *et al.* [152]), we first construct an optimal assignment. An shortest assignment can be found in polynomial time (see, for example, Papadimitriou and Steiglitz [205]). An optimal assignment may contain several disconnected cycles. If it does not contain cycles an optimal tour has been found. An optimal assignment in Figure 6.9 is: $\{w_0, w_1, w_3, w_0\} \cup \{w_2, w_4, w_2\}$ with costs of 1. Clearly, this assignment contains two disconnected cycles.

Disconnected cycles need to be 'patched' together to find a shortest tour. A detailed description of the general theory and proofs of subtours patching are given in Lawler *et al.* [163]. The *basic cycle* contains w_0 and other nodes of V. The other nodes of V are contained in *non basic* cycles. Non basic cycles are repeatedly merged among themselves and with the basic cycle, until one cycle remains. This remaining cycle is an optimal tour. Firstly, mergings are performed that add nothing to the total distance. Cycles can be merged without extra costs if the interval [leftmost L_i, rightmost R_j] (i.e. *span*) of one of the cycles is intersecting with the span of an other cycle. This procedure is repeated until cycles with disjoint spans remain.

There is only one possibility for having disjoint cycles. This occurs if one cycle contains all requests related to I/O_1 and the other one contains all requests related to I/O_2 and both cycles are not intersecting and the SRM starts and ends at the same I/O. In all other cases, the cycles will be non-disjoint because of the fact that if requests are not separated by I/O, we know that the crane can arrive after a maximum of 2 requests in both I/O locations. Furthermore, we know that the startpoint and endpoint are contained in the same cycle. If the startpoint and endpoint are at different locations the SRM visits both I/O in one cycle. Other, non-disjoint cycles can be patched together without extra costs because both I/O locations are visited in the basic cycle.

Finally, these two cycles with disjoint spans are merged. Each cycle is considered as a node. The distance between those two nodes equals the smallest distance between two points of the spans of these cycles. In other words, the minimum distance between those two disjoint cycles equals the smallest gap between the two farthest located storage/retrieval locations of each cycle. Clearly, the length of a shortest route containing all requests is greater than or equal to the length of both cycles plus twice this smallest gap. Therefore, if we combine both cycles at the corners of the smallest gap a shortest route is obtained. As a result, in exactly one storage/retrieval location of both cycles two incoming and two outgoing arcs occur. This is used, to overcome the problem that the direction of travel of a storage job is reverse to the direction of travel of a retrieval job. By allowing two incoming and outgoing arcs in the requests at the corners of the gap, it can be decided during the route at which moment the job is really executed. We know for sure that the job is really executed, because of the fact that both incoming and outgoing arcs will be visited. We will describe this in more detail for the example in Figure 6.9: the retrieval job w_3 in the basis cycle $\{w_0, w_1, w_3, w_0\}$ is combined with the storage job w_2 in the other cycle $\{w_2, w_4, w_2\}$ (see Figure 6.10). The route starts at the startpoint, executes the storage request w_1 from the basis cycle and arrives empty at the corner retrieval location w_3. Before performing the retrieval request the distance $(= 4)$ between the two cycles is travelled and the crane arrives empty at the storage location w_2. It passes the storage location and travels empty to the retrieval location of request w_4 and executes this request. The last job of this cycle is to transport the storage job from I/O_2 to the storage location at the corner of the cycle. The job is stored and again the crane travels empty the distance of 4 between both cycles and arrives at the retrieval location w_3. The load at this location is retrieved and the crane continues to handle this request from the basis cycle. Finally, the cranes arrives at the endpoint of the route. Clearly, all requests are handled in an optimal sequence with an empty travel distance of 9.

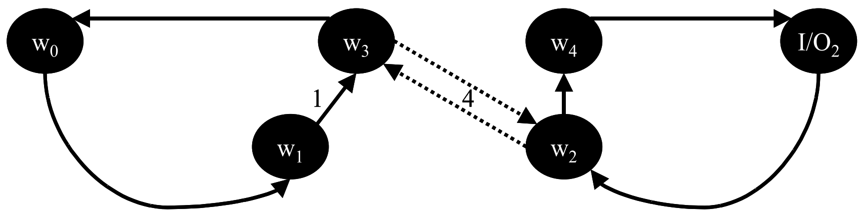

FIGURE 6.10. Example of combining two disjoint subtours.

The route is optimal since either an optimal assignment without cycles is found or an optimal assignment with disjoint subtours is connected in an optimal way. By applying the *Hungarian Method*, an optimal assignment can be found in $O(n^3)$ time, where $2n$ nodes are used in the complete bipartite network. A description of the Hungarian Method is given in, for example, Papadimitriou and Steiglitz [205]. To define the span of a cycle, all nodes have to be considered to find the rightmost and leftmost location of all nodes in the cycle. At most n nodes need to be considered. Thereafter, at most n cycles need to be merged. Thus, the sequencing of storage and retrieval requests in an aisle with two I/O can be solved to optimality in $O(n^2) + O(n^3) = O(n^3)$ time. In Section 6.4 this method is applied on an example.

In the case that all storage requests are available at one I/O station and all retrieval requests are available at one I/O station the method of Van den Berg and Gademann [269] can be used. This method is discussed in Section 5.2.

6.3.4 Procedure for constructing an orderpicking tour

With the dynamic programming algorithm described in the previous section, we have obtained a minimum length directed tour subgraph T. Ratliff and Rosenthal [218] describe an algorithm to construct an optimal directed multiple aisle storage and retrieval tour from T. This algorithm can also be used for directed tours. The procedure constructs a directed cycle in T starting and ending at v_0. In Theorem 4 of Ratliff and Rosenthal [218] it is proven that each arc is included in this cycle. The proof of their Theorem 4 is similar for directed and undirected tour subgraphs.

6.3.5 Complexity of the algorithm

In this section we will show that a directed multiple aisle storage and retrieval tour can be found in polynomial time.

THEOREM 6.5.
The problem of sequencing storage and retrieval requests of an SRM working in multiple aisles can be solved to optimality in $O(n^3 + n^2m^3)$ time, where n equals the number of aisles and m the number of requests.

PROOF
To bound the running time of the complete algorithm, we first bound the running time of each of its operations. The running time of an operation can be bounded by the total number of steps associated with this operation. The bound of the running time of the algorithm equals the product of the bound on the number of steps per operation and the bound on the number of operations (see also Ahuja *et al.* [2]).

In this algorithm an operation consists of performing transitions in aisle j (Figure 6.6) combined with transitions for the crossover from aisle j to $j+1$ (Figure 6.7). The total number of aisles equals n and therefore the total number of operations is bounded by n.

The worst case of the procedure to sequence requests in a single aisle occurs if all requests m are located in this single aisle. As a result, $O(m^3)$ is the upper bound of the time to perform this part of the operation. All $6+2\cdot\lfloor n/2 \rfloor$ equivalence classes need to be searched for the smallest value. In each equivalence class a constant number of values needs to be searched. Therefore, this operation has a time complexity of $O(n)$. To perform the operation in an aisle the upper bound of time equals $O(m^3 + n)$.

There exist $4 + 2 \cdot \lfloor n/2 \rfloor$ possible ways to make the crossover from aisle j to $j+1$. Therefore, the upper bound of this part of the operation equals n.

In the complete operation all transitions in an aisle are combined with all transitions between aisles. Again, n equivalence classes need to be searched for a minimum value among a constant number of values. Therefore, the upper bound of an operation equals $O(n(m^3 + n + 1))$. As a result, the complexity of the algorithm is $O(n^2(m^3 + n + 1)) = O(n^3 + n^2m^3)$. ■

6.4 Illustrative example

In this section, we apply the above described method to the example in Figure 6.1. Summarising, we consider a warehouse with 6 aisles in which storage and retrieval requests are available for transport by an SRM. The depot is located at the bottom end of aisle 1. At both ends of an aisle an input/output station is located at which requests are transferred from the SRM to another material handling system and vice versa. For each request the aisle, origin and destination are known. The objective is to schedule all requests such that the total empty travel distance is minimised. The exact position of each storage/retrieval location is indicated in the network in Figure 6.11. The network is constructed according to the method described in Section 6.2.

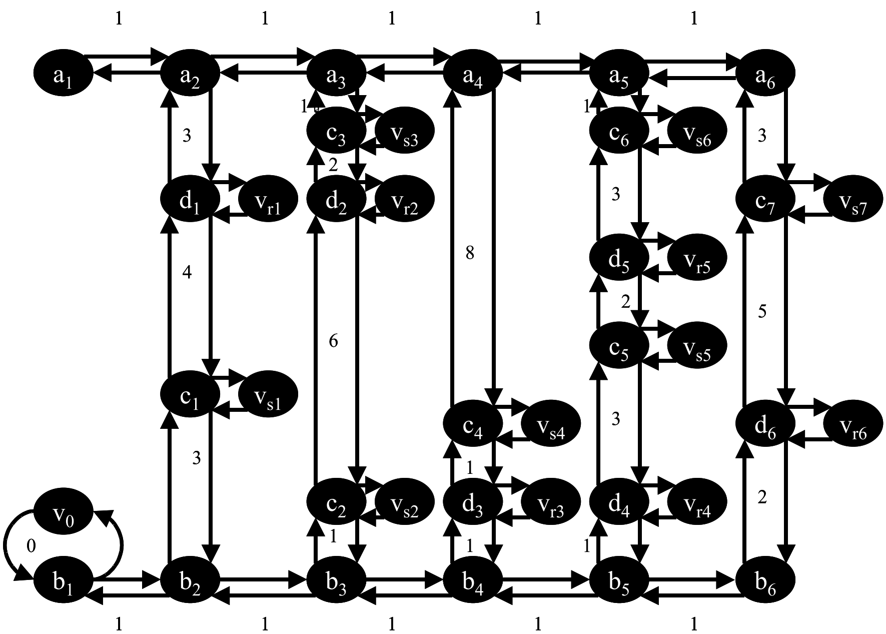

FIGURE 6.11. Example of a warehouse with a number of storage and retrieval requests in 6 aisles.

Firstly, we determine the empty travel distance (i.e. route length) per transition in each aisle to transport all requests starting and ending at the specified positions. The methods from Section 6.3.3 are applied to obtain the various lengths. In Table 6.3. optimal route lengths for each transition in each aisle are shown.

transition	aisle 1	aisle 2	aisle 3	aisle 4	aisle 5	aisle 6
i)	10	4	7	9	14*	15
ii)	10	16	13	11	6	5
iii)	20	14	15	17	10	10
iv)	0	14	15*	1	10	15
v)	-	-	-	-	-	-

TABLE 6.3. Route length (i.e. empty travel distance) to visit all nodes in aisle i using transition i), ii), iii), iv) and v). The values with* are explained below.

We will explain the way in which these route lengths are obtained by considering aisle 3 with transition iv), aisle 4 with transition iv) and aisle 5 with transition i). In transition iv) the start- and endpoint equals b_j. The requests in aisle 3 can be represented, as given in Figure 6.12, on a straight line. The origins and destinations of each job are numbered. Furthermore, the start- and endpoint are indicated.

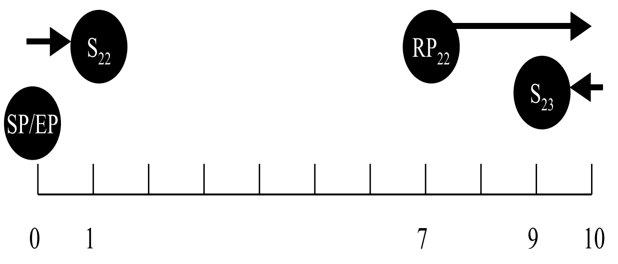

FIGURE 6.12. Straight line with all requests in aisle 3.

We translate this to an assignment problem (see Figure 6.13), in the way described in Section 6.3.3.

After solving the assignment problem, an optimal solution to this assignment problem equals 3, namely connect w_0 with w_1 and w_1 with w_0. Furthermore, node w_3 is connected with w_2 and w_2 with w_3. As a result, two cycles are obtained, namely (w_0, w_1, w_0) and (w_3, w_2, w_3). The span of (w_0, w_1, w_0) equals $(0, 1)$. The span of (w_3, w_2, w_3) equals $(7, 10)$. Consequently, we have two disjoint cycles. The minimum distance between those two spans equals 6. An optimal route for this problem has a length equal to at least the sum of the lengths of the cycles + twice the length of the gap. As a result, an optimal route can be constructed with length 15 in the following way. We combine the two disjoint cycles to the cycle $(w_0, w_1, w_3, w_2, w_0)$ with length equal to 15.

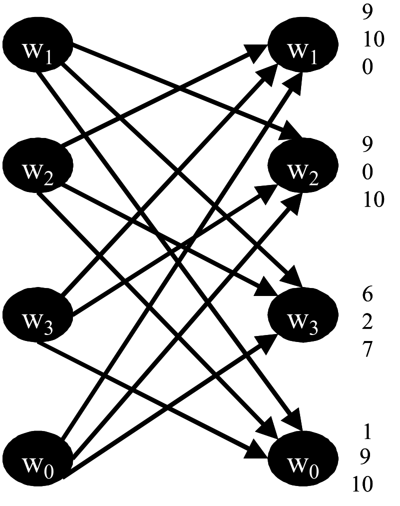

FIGURE 6.13. Assignment problem related to aisle 3 with start- and endpoint in b_3. The empty travel times related to the arcs from top to bottom of each node are indicated at the right of the network.

If we solve the assignment problem for aisle 5 with transition 1 we obtain the following two cycles: (w_0, w_3, w_0) and (w_1, w_4, w_2, w_1). The related costs equal 14. The span of $(w_0, w_3, w_0) = (0, 10)$ and the span of $(w_1, w_4, w_2, w_1) = (4, 10)$. These cycles are not disjoint, and therefore the cycles can be patched together without extra costs. An optimal route is $(w_0, w_3, w_1, w_4, w_2, w_0)$ with costs 14.

According to Theorem 6.4. the maximum number of incoming and outcoming arcs in nodes a_j and b_j equals $\lfloor 6/2 \rfloor = 3$. Therefore, we have the following equivalence classes (see Equation 6.2): $(0,0,0)$, $(0,0,1)$, $(E,E,0,0,1)$, $(E,E,0,0,2)$, $(E,0,0,0,1)$, $(0,E,0,0,1)$, $(-3,3,1)$, $(-2,2,1)$, $(-1,1,1)$, $(1,-1,1)$, $(2,-2,1)$, $(3,-3,1)$. Tables 6.1 and 6.2. are reformulated for this special example. The results are presented in Tables 6.4. and 6.5.

Equivalance class	Transitions				
	(i)	(ii)	(iii)	(iv)	(v)
(0,0,0)	(1,-1,1)	(-1,1,1)	(E,0,0,0,1)	(0,E,0,0,1)	(0,0,0)
(0,0,1)	-	-	-	-	(0,0,1)
(E,E,0,0,1)	-	-	-	-	(E,E,0,0,1)
(E,E,0,0,2)	(1,-1,1)	(-1,1,1)	(E,E,0,0,2)	(E,E,0,0,2)	(E,E,0,0,2)
(E,0,0,0,1)	(1,-1,1)	(-1,1,1)	(E,0,0,0,1)	(E,E,0,0,2)	(E,0,0,0,1)
(0,E,0,0,1)	(1,-1,1)	(-1,1,1)	(E,E,0,0,2)	(0,E,0,0,1)	(0,E,0,0,1)
(-1,1,1)	(E,E,0,0,1)	(-2,2,1)	(-1,1,1)	(-1,1,1)	(-1,1,1)
(-2,2,1)	(-1,1,1)	(-3,3,1)	(-2,2,1)	(-2,2,1)	(-2,2,1)
(-3,3,1)	(-2,2,1)	-	(-3,3,1)	(-3,3,1)	(-3,3,1)
(1,-1,1)	(2,-2,1)	(E,E,0,0,1)	(1,-1,1)	(1,-1,1)	(1,-1,1)
(2,-2,1)	(3,-3,1)	(1,-1,1)	(2,-2,1)	(2,-2,1)	(2,-2,1)
(3,-3,1)	-	(2,-2,1)	(3,-3,1)	(3,-3,1)	(3,-3,1)

TABLE 6.4. Equivalence classes resulting from performing transition L_j^- to L_j^+ for a warehouse with 6 aisles.

Equivalence class	Transitions			
	(0)	(0a)	(0b)	(0c)
(0,0,0)	(0,0,0)	-	-	-
(0,0,1)	(0,0,1)	-	-	-
(E,E,0,0,1)	(0,0,1)	(E,0,0,0,1)	(0,E,0,0,1)	-
(E,E,0,0,2)	-	-	-	(E,E,0,0,2)
(E,0,0,0,1)	(0,0,1)	(E,0,0,0,1)	-	-
(0,E,0,0,1)	(0,0,1)	-	(0,E,0,0,1)	-

Equivalence class	Transitions					
	1a	1b	2a	2b	3a	3b
(-1,1,1)	-	(-1,1,1)	-	-	-	-
(-2,2,1)	-	-	-	(-2,2,1)	-	-
(-3,3,1)	-	-	-	-	-	(-3,3,1)
(1,-1,1)	(1,-1,1)	-	-	-	-	-
(2,-2,1)	-	-	(2,-2,1)	-	-	-
(3,-3,1)	-	-	-	-	(3,-3,1)	-

TABLE 6.5. Equivalence classes resulting from performing transition L_j^+ to $L_{j+1}^-$for a warehouse with 6 aisles.

In Table 6.6. we present for this example all information regarding the minimum length partial directed subtours. The numbers in each cell indicate respectively, the minimum length, the equivalence class of the predecessor and the chosen arc configuration. We have numbered the equivalence classes according to column one of the Table 6.6. The values for L_1^+ are obtained by adding the mentioned arc configuration to a null graph. By adding transitions from Table 6.5. we obtain L_2^-. The minimum length among all values corresponding to a specific equivalence class is chosen for each equivalence class. By adding transitions from Table 6.4. we obtain L_2^+. Again we choose the minimum value for each equivalence class. For example, the equivalence class $(E, E, 0, 0, 1)$ can be obtained in two different ways. First, by performing transition (i) to $(-1, 1, 1)$, which results in a length of 16. Secondly, by performing transition (ii) to $(1, -1, 1)$ we also obtain equivalence class $(E, E, 0, 0, 1)$. The length in this case equals 28. Clearly, the minimum length equals 16. The process is continued in this way for each aisle. All results are presented in Table 6.6.

Equivalence class	aisle 1		aisle 2		aisle 3	
	L_1^-	L_1^+	L_2^-	L_2^+	L_3^-	L_3^+
1 (0,0,0)	-	-	-	-	-	-
2 (0,0,1)	-	-	0/6/0	-	16/6/0	-
3 (E,E,0,0,1)	-	-	-	16/7/i	-	21/10/ii
4 (E,E,0,0,2)	-	-	-	16/6/iii	20/4/0c	33/6/iii
5 (E,0,0,0,1)	-	20/-/iii	22/5/0a	36/5/iii	18/3/0a	33/5/iii
6 (0,E,0,0,1)	-	0/-/iv	2/6/0b	16/6/iii	18/6/0b	33/6/iv
7 (-1,1,1)	-	10/-/ii	12/7/1b	18/7/iii	20/7/1b	27/8/i
8 (-2,2,1)	-	-	-	28/7/ii	32/8/2b	33/7/ii
9 (-3,3,1)	-	-	-	-	-	45/8/ii
10 (1,-1,1)	-	10/-/i	12/10/1a	6/6/i	8/10/1a	23/10/iii
11 (2,-2,1)	-	-	-	16/10/i	20/11/2a	15/10/i
12 (3,-3,1)	-	-	-	-	-	27/11/i

Equivalence class	aisle 4		aisle 5		aisle 6	
	L_4^-	L_4^+	L_5^-	L_5^+	L_6^-	L_6^+
1 (0,0,0)	-	-	-	-	-	-
2 (0,0,1)	21/3/0	-	24/5/0	-	34/3/0	-
3 (E,E,0,0,1)	-	38/7/i	-	34/10/ii	-	<u>37/10/ii</u>
4 (E,E,0,0,2)	37/4/0c	24/5/iv	28/4/0c	36/5/iv	40/4/0c	48/6/iii
5 (E,0,0,0,1)	23/3/0a	24/5/iv	26/5/0a	36/5/iv	38/5/0a	48/5/iii
6 (0,E,0,0,1)	23/3/0b	24/6/iv	26/6/0b	36/6/iii	38/6/0b	53/6/iv
7 (-1,1,1)	29/7/1b	30/7/iv	32/7/1b	32/6/ii	34/7/1b	43/5/ii
8 (-2,2,1)	37/8/2b	38/8/iv	42/8/2b	38/7/ii	42/8/2b	39/7/ii
9 (-3,3,1)	51/9/3b	48/8/ii	54/9/3b	48/8/ii	54/9/3b	47/8/ii
10 (1,-1,1)	25/11/1a	26/10/iv	28/10/1a	<u>30/11/ii</u>	<u>32/10/1a</u>	42/10/iii
11 (2,-2,1)	<u>19/11/2a</u>	<u>20/11/iv</u>	<u>24/11/2a</u>	34/11/iii	38/11/2a	47/10/i
12 (3,-3,1)	33/12/3a	34/12/ii	40/12/3a	38/11/i	44/12/3a	53/11/i

TABLE 6.6. Solution to the example problem.

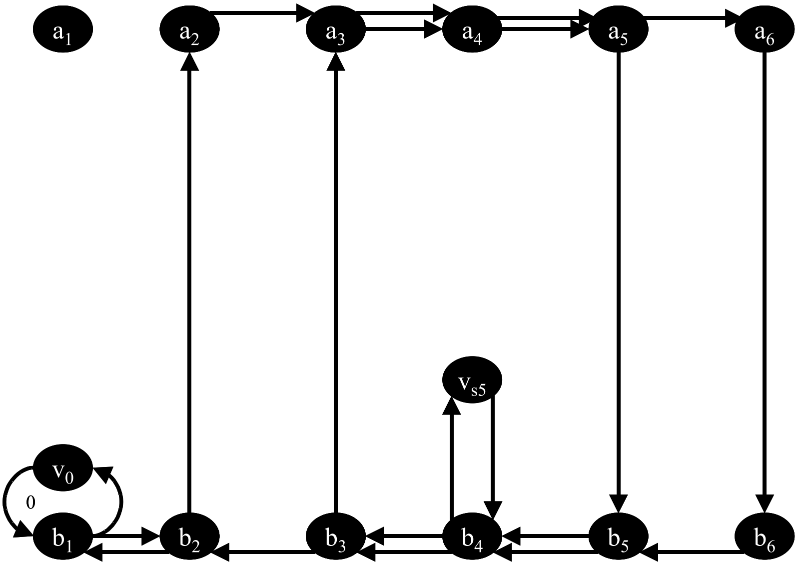

FIGURE 6.14. Shortest directed multiple aisle storage and retrieval tour for the example problem.

The solution to the problem is indicated by the underscored values. Starting in aisle 6, we travel backwards through the Table to obtain, the arc configurations which should be used. A shortest *directed multiple aisle storage and retrieval tour* related to the data from Table 6.6. is given in Figure 6.14. The length of the empty travel equals 37. The total travel time, including the fixed loaded travel times, equals 68.

6.5 Concluding remarks

In this chapter we have studied the problem of sequencing storage and retrieval requests of an unit load SRM working in multiple aisles. Each aisle has two input/output stations located at the end of the aisle. A method is developed to construct a shortest *directed multiple aisle storage and retrieval tour* containing all requests starting and ending at a depot. Therefore, two algorithms are developed. Firstly, a polynomial algorithm is created which can be used to schedule to optimality requests in one aisle with two input/output stations. The objective is to minimise empty travel times of the crane to transport all requests if the crane starts and ends at specified end(s) of the aisle.

Furthermore, a dynamic programming algorithm is developed to schedule storage and retrieval requests in multiple aisle. We extended the item retrieval algorithm of Ratliff and Rosenthal [218] such that it can be used in warehouses in which two different types of requests (storages and retrievals) have to be performed. The lengths of the route in an aisle is acquired by applying the algorithm for scheduling requests in an aisle with two input/output stations. The algorithm considers the aisle sequentially, starting at aisle 1 and ending at aisle n. It is shown that a *directed multiple aisle storage and retrieval tour* can be found in polynomial time.

In Chapter 9 this method will be applied to a straddle carrier working in a container terminal.

Part III

A practical application: a container terminal

7

Transshipment of containers at a container terminal: an literature overview

As described in Chapter 1 various material handling centres can be distinguished, namely centres in which products are produced, stored or transshipped. The planning and control concepts developed in the previous chapters can be applied to equipment used at an arbitrary material handling centre. To give an impression in which way this can be done, we have applied the methods to some of the material handling equipments used at a semi-automated container terminal. In Chapter 8 we consider a terminal in which automated guided vehicles or self lifting vehicles are used. The algorithms of Chapter 3 and 4 are applied to determine vehicle requirements at a terminal. In Chapter 9, the storage process is considered. The methods from Chapter 6 are applied to determine routes for performing storage and retrieval requests by automated stacking cranes and straddle carriers.

At container terminals, containers are transshipped from one mode of transportation to another. At a terminal different types of material handling equipments are used to transship containers from ships to barges, trucks and trains and vice versa. Over the past decades, ships have strongly increased in size, up to 8000 TEU (i.e. Twenty feet Equivalent Unit container). In order to use these big ships efficiently, the docking time at the port must be as short as possible. This means that large numbers of containers have to be loaded, unloaded and transshipped in a short time span, with the minimum use of expensive equipment.

This chapter gives a classification of the decision problems that arise at container terminals. For various decision problems, an overview of relevant literature is presented. Quantitative models from this literature, which try to solve the problems are discussed. Finally, some general conclusions are given.

The organisation of this chapter is as follows: Section 7.1 describes the processes at container terminals. The planning of a complete terminal is examined in Section 7.2. Furthermore, some general conclusions and subjects for further research are given in Section 7.3. Finally, the outline of this part of the thesis is presented.

7.1 Processes at container terminals

Container ships are nowadays unloaded and loaded at large container terminals. Figure 7.1 illustrates the unloading and loading process at a typical container terminal. This loading and unloading process can be divided into different subprocesses, described below. When a ship arrives at the port, quay cranes (QCs) take the import containers off the ship's hold or off the deck. Next, the containers are transferred from the QCs to vehicles that travel between the ship and the stack. This stack consists of a number of lanes, where containers can be stored for a certain period. Equipments, like cranes or straddle carriers (SCs), serve the lanes. A straddle carrier can both transport containers and store them in the stack. It is also possible to use dedicated vehicles to transport containers. If a vehicle arrives at the stack, it puts the load down or the stack crane takes the container off the vehicle and stores it in the stack. After a certain period the containers are retrieved from the stack by cranes and transported by vehicles to transportation modes like barges, deep sea ships, trucks or trains. To load export containers onto a ship, these processes are also executed in reverse order.

Most of the terminals make use of manned equipments, like straddle carriers, cranes and multi-trailer-systems. However, a few terminals, like some terminals in Rotterdam, are semi-automated. At such terminals Automated Guided Vehicles (AGVs) are used for the transport of containers. Furthermore, the stacking process can also be done automatically by Automated Stacking Cranes (ASCs).

This section describes the different subprocesses and their corresponding types of material handling equipments in more detail. Furthermore, we present different decision problems of the various types of material handling equipments. We also discuss ways of solving these problems.

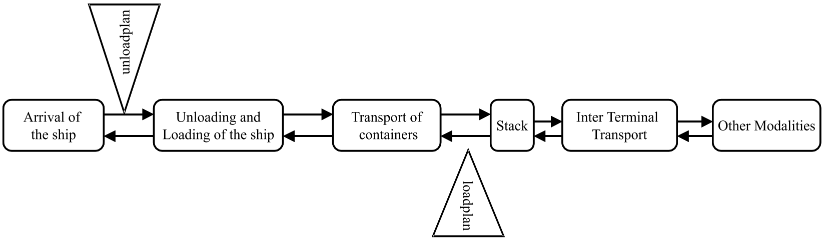

FIGURE 7.1. Processes at a container terminal.

7.1.1 Arrival of the ship

When a ship arrives at the port, it has to moor at the quay. For this purpose, a number of berths (i.e. place to moor) are available. The number of berths that should be available at the quay is one of the decisions that has to be made at the strategic level. In Edmond and Maggs [64] queueing models are evaluated, which can be used in making this decision. They conclude that some of these queueing models can be used when the model and parameters are chosen carefully and the results are evaluated precisely.

At the operational level the allocation of a berth to the ship has to be decided on. Imai *et al.* [114] studies how to allocate berths to ships while optimising the berth utilisation. On one hand optimal berth allocation can be obtained by minimising the sum of port staying times. This leads to ships mooring at the quay according to the first come first served principle. On the other hand berths can be allocated, without consideration of ship's arrival order, by allocating ships at a berth closest by the area in the stack in which most containers for this specific ship are located. As a result, the resulting terminal utilisation will be maximal, but ship owners will be dissatisfied by the long waiting times of the ships. Consequently, a trade-off exists between the total staying time in the port and the dissatisfaction of ship owners caused by the order in which ships are berthed. The berth allocation problem could be considered as a machine scheduling problem. However, the introduction of a multi objective approach is according to Imai *et al.* [114] really new in machine scheduling problems. A two objective nonlinear integer program is formulated to identify the set of berth allocations which minimises the dual objectives of overall staying time (i.e. waiting time plus berthing time) and dissatisfaction on order of berthing.

After defining the two objective nonlinear integer program, the problem is reduced to a single objective problem. The resulting single objective problem is similar to a classical assignment problem. The objective consists of two parts, namely the sum of the waiting times plus the sum of dissatisfaction. From numerical experiments, it can be concluded that the trade-off increases if the size of the port increases. The dynamic version of the berth allocation problem is studied in *Imai et al.* [115]. The difference with the static problem is that ships arrive while work is in progress. The dynamic problem can be represented by a three dimensional assignment problem, which cannot be solved in polynomial time. Therefore, a langrangian relaxation based heuristic is developed to solve the problem. Nishimura *et al.* [200] study the same problem. However, solutions for dynamic berth assignment are found by applying genetic algorithms.

7.1.2 Unloading and loading of the ship

As described, decisions have to be made for questions that arise at three different levels. At the strategic level questions arise concerning, for example, which type of material handling equipments will be used for the unloading and the loading of containers from the ship. Automated and manned terminals both use quay cranes. QCs are manned because automation of this process encounters practical problems, like exact positioning of containers. Figure 7.2 illustrates a QC. The QCs have trolleys that can move along the crane arm to transport the container from the ship to the transport vehicle and vice versa. A spreader, a pick up device attached to the trolley, picks the containers. The QCs move on rails to the different holds to take/put containers off/on the deck and holds. It can occur that at the same moment one QC is unloading containers while another QC is loading containers.

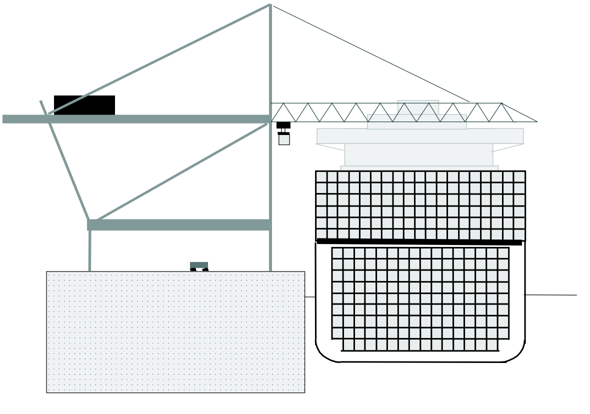

FIGURE 7.2. Quay Crane.

At the tactical level the number of QCs have to be determined that work simultaneously on one ship. One of the objectives is to minimise the staying time of ships at the terminal.

The most general case of the crane scheduling problem, is the case in which ships arrive at different times in the port and queue for berthing space if the berths are full. The objective in this case is to serve all the ships while minimising the total delay of the ships. Daganzo [48] describes ships by the number of holds they have. Only one crane can work on a hold at a time. Daganzo [48] discusses the static crane allocation problem in which a collection of ships is available at a berth to be handled at the start of the planning horizon and no other ships will arrive during this planning horizon. A number of identical quay cranes have to be allocated to the holds to minimise the total delay (i.e. costs) of the ship. The problem can be formulated as a mixed integer program. The solution indicates the average number of cranes used on each hold at every instance. As a result, an implementable crane allocation scheme has been found. Due to computational reasons, this exact solution method is only usable for a small number of ships. When a mathematical programming solution is not effective a heuristic procedure based on some scheduling principles, derived from optimal solutions, can be applied to solve the studied problem.

Daganzo [48] also studies the dynamic case of the crane allocation problem. Within a finite horizon, ships arrive at instants within the horizon and cannot be handled before those instants instead of having ships ready at the start of the horizon. It is required to repeat the static allocation procedure for ships at the berth after each ship's arrival. Only loads remaining in the ships have to be considered. As a result, an arriving ship can get pre-emptive priority over a ship that is already being served.

For all described methods, it is assumed that cranes can operate on all ships present at the port. A formulation and solution method for the general case in which berth length is limited, is not provided in this paper. However, the results in the paper may lead to analytical expressions to predict crane productivity and ship delay.

Peterkofsky and Daganzo [207] also treat the minimisation of the total delay of ships. The goal of this paper is to give an exact solution method for a class of problems considered in the paper of Daganzo [48]. This is interesting for practical use and theoretical use to test the performance of the heuristic methods. The problem is decomposed into two stages, namely finding the best departure schedule for the ships and finding a crane allocation scheme. A branch and bound method is given to solve the static case of the crane scheduling problem. This method is based on the property that the optimum is restricted to only certain kinds of departure schedules. It is proved that the search for the optimum can be restricted to boundary points.

Boundary points are feasible schedules that lie on the boundary between the solution space's infeasible and feasible regions. Ten problems, based on real world problems are generated to test the method. The performance declines quickly when the problem size grows. According to the authors the described model can be extended to take into account ships with different known arrival times. Furthermore, the model can be applied in other situations, like machine scheduling problems.

The results for the static case from the heuristic method based on principles in Daganzo [48] and from the branch and bound approach in Peterkofsky and Daganzo [207] are compared in the paper of Daganzo [48]. For small problems with four ships, it can be concluded that the heuristic approach of Daganzo [48] is comparable to the optimisation procedure of Peterkofsky and Daganzo [207]. In larger cases, the use of the heuristic approach is preferred over the branch and bound approach due to rapidly increasing computation times. However, the performance of the heuristic is not tested for large problems.

The number of import containers that has to be unloaded at the terminal is in practice usually only known shortly before the arrival of the ship. At the operational level an unloading and loading plan have to be made. The unloading plan indicates which containers should be unloaded and in which hold they are situated in the ship. Successively, these containers are unloaded. In a hold the crane driver is almost free to determine the order in which the containers are unloaded. The unloading time of a container depends on its place in the ship.

In contrast with the unloading process, there is hardly flexibility in the loading process. A good distribution of containers over the ship is necessary. At the operational level a stowage planning is made. A stowage plan indicates for each container the exact place in the ship. Containers with the same destination, category, weight, size, contents and so on, belong to the same category. Sometimes, only for each category the positions in the ship are given. Locations of containers belonging to the same category can be exchanged between containers of this category. In making the stowage planning attention should be paid to the order in which containers need to be unloaded. Unnecessary moves should be avoided by placing containers designated for a terminal visited later during the journey on top of containers designated for the earlier visited terminals.

According to Shields [242], the containers, that will be stowed, have to satisfy a variety of constraints, which arise as a result of physical limitations of the ship and containers and the sequence in which ports are visited by the ship. Shields [242] presents a system which can assist in this planning process. The stowage problem is solved with the Monte Carlo method. Many different possible ship loadings are generated and the most efficient one is given such that rehandlings are avoided to a large extent, physical limitations are met and unloading and loading time is minimised. This system has been used worldwide since 1981. A most efficient plan is displayed with the precise loading order of export containers.

According to Wilson and Roach [292], the container stowage problem is one the size of which depends upon the capacity of the ship and the supply and demand of containers at each port. Finding an optimal solution is not realistic in reasonable times, because of the fact that the stowage plan has to be made across a number of ports and the lack of information at the different ports. Therefore, the authors propose to decouple the process into two subprocesses, namely a tactical and a operational process. The first process assigns generalised containers to a block in the ship. Secondly, specific containers are assigned to specific locations in the blocks determined in the first phase. The block stowage problem can be solved by applying a branch and bound algorithm. The problem in the tactical phase can be solved by applying tabu search. In this way, good but not always optimal solutions will be found in reasonable computation times.

Avriel *et al.* [9] present a binary linear programming formulation to solve the problem of stowage planning such that the number of unnecessary moves is minimised. It is shown that the stowage planning problem is NP-hard. Therefore, also a heuristic is developed for solving the problem. Avriel *et al.* [10] show a relation between the stowage planning problem and the coloring of circle graphs problem.

7.1.3 Transport of containers from ship to stack and vice versa

As described containers have to be transported from the ship to the stack and vice versa. In designing a terminal, one of the decisions at the strategic level concerns the type of material handling equipment, that takes care of the transport of containers. For the transport of a container at a manned terminal, vehicles like forklift trucks, yard trucks or straddle carriers can be used. Straddle carriers, (see Figure 7.3), and forklift trucks can pick up containers from the ground.

A crane is needed to put the container on the yard truck. According to Baker [12] the use of straddle carriers instead of non lifting trucks can mean improved Quay Crane productivity. For the transport of multiple containers, multi-trailer systems can be used. The system in Figure 7.4, uses a truck that pulls five trailers, each capable of carrying 2 TEU.

At an automated container terminal Automated Guided Vehicles (AGVs) are used for the internal transport. AGVs (see Figure 7.5) are robotic vehicles which travel

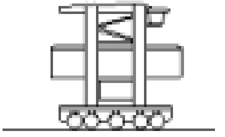

FIGURE 7.3. Straddle Carrier.

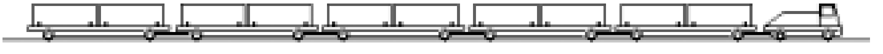

FIGURE 7.4. Multi-trailer system.

along a predefined path. The road system consists of electrical wires in the ground, or a grid of transponders, that control accurately the position of the AGV. More detailed information on AGVs is given in Chapter 2.

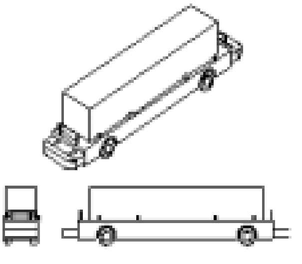

FIGURE 7.5. AGV from different points of view.

Currently, an AGV can carry either one 20 feet, 40 feet or 45 feet container. In the future, Europe Combined Terminals (ECT), in the port of Rotterdam, will use AGVs capable of carrying one 40 feet, one 45 feet or two 20 feet containers. To do so, the capacity of the AGV will increase from 40 tonnes to 60 tonnes (see [4]). AGVs are only practical in ports with high labour costs because of the high initial capital costs.

In ports with low labour costs, the system of manned vehicles is preferable. At this moment, research is done with respect to a new type of automated vehicle, namely an Automated Lifting Vehicle (ALV). An ALV can lift and transport one container without using a crane.

After the decision which system will be used has been made, one of the problems at the tactical level that has to be solved is the determination of the necessary number of transport vehicles. In Steenken [257] an optimisation system is developed to determine the number of straddle carriers and their route. Because of the fact that the system had to be implemented into a radio data transmission system, the system had to fulfil the conditions of a real-time application. The problem is solved as a linear assignment problem. In Chapter 2 of this thesis a model and an algorithm are presented to determine the necessary number of AGVs at an automated container terminal. To solve the problem a network formulation is given and a minimum flow, strongly polynomial time algorithm is developed.

At the operational level it should be decided which vehicle transports which container and which route is chosen. A complete review of the routing and scheduling of vehicles in general is given in Bodin *et al.* [22]. Steenken [257] and Steenken *et al.* [258] describe the more specific problem of the routing of straddle carriers at the container terminal. The objective is to minimise empty-travel distances by combining unloading and loading jobs. Routing and scheduling systems are tested and integrated into a radio data transmission system of a real terminal. Steenken [257] obtains savings of 13% in empty drives compared with the previously existing situation at the terminal, by solving the problem as a linear assignment problem. Steenken *et al.* [258] solve the problem by formulating it as a network problem with minimum costs. Savings of 20-35% in empty-travel distances can be obtained in quite acceptable computation times.

In Kim and Bae [132] mixed integer linear programming formulations and a heuristic method are given for dispatching containers to AGVs such that the delay of the ship and the total travel time of the AGVs is minimised.

In Chen *et al.* [43] an effective dispatching rule is given that assigns AGVs to containers. Other decisions, like determination of a storage location for the import container in the stack, routing of AGVs and traffic control are considered as input. They have developed a greedy algorithm to solve this problem. In the case of a single ship with a single crane, the greedy algorithm is optimal. k AGVs are assigned to the first k containers available. Thereafter, the next container is assigned to the first available AGV. In the case of a single ship with multiple cranes, the greedy algorithm assigns an available AGV to the first available ship crane. Examples can be constructed that demonstrate that the greedy algorithm does not necessarily find the optimal solution in this case. However, with a simulation study it is shown that solutions of the greedy algorithm are close to optimal. Furthermore, the impact of this rule on other decisions, like throughput times of AGVs, crane idle times and number of cranes, is examined. Bish *et al.* [20] observe an extension of this problem, namely the problem of dispatching vehicles to containers in combination with the location problem of containers. In other words, in this vehicle-scheduling-location problem each container has to be assigned to a location in the stack (see also Section 7.1.4) and vehicles have to be dispatched to containers such that the total time to unload all containers from the ship is minimised. It is proven that this is a *NP hard* problem. Therefore, in Bish *et al.* [20] a heuristic method is proposed to solve this problem. Firstly, an assignment problem is formulated in which containers are assigned to locations by minimising the total distance travelled by vehicles from quay cranes to locations in the stack. This objective is subject to the fact that each container must be assigned to a location and secondly to the fact that each location cannot be assigned to more than one container. The heuristic method consists of two steps. Firstly, the assignment problem is solved and locations are assigned to the containers based on this solution. Secondly, the greedy algorithm from Chen *et al.* [43] is applied to the containers and their locations. It is shown that to use this effective heuristic in practice other issues, like avoidance of congestion, identifying routes need to be incorporated into the analysis and algorithms.

In Van der Meer [271] the control of guided vehicles in vehicle based internal transport systems, like container terminals, is studied. Results are presented that show how different vehicle dispatching rules behave in different environments. In Evers and Koppers [73] the traffic control of large numbers of AGVs is studied. A formal tool to describe traffic infrastructure and its control is developed by using four types of entities: node, track, area and semaphore (i.e. a non-negative integer variable which can be interpreted as free capacity). The tool is evaluated with simulation. It can be concluded that the technique is a powerful tool for modelling transportation infrastructure and its control and that the performance and the capacity of the area increases.

7.1.4 Stacking of containers

Two ways of storing containers can be distinguished: storing on a chassis and stacking on the ground. With a chassis system each container is individually accessible. With stacking on the ground containers can be piled up, which means that not every container is directly accessible. As a consequence of limited storage space, nowadays stacking on the ground is most common.

The stack (see Figure 7.6) is the place where import and export containers can be stored for a certain period. The stack is divided into multiple blocks/lanes, each consisting of a number of rows. The height of stacking varies per terminal between two and eight containers high. At the end of each lane a transfer point might be situated. At this point the crane takes/places the container off/on the vehicle that transports the container. Empty containers are usually stored separately. The distribution of empty containers to ports is a related problem. It is for example studied in Cheung and Chen [44], Crainic *et al.* [47] and Shen and Khoong [241].

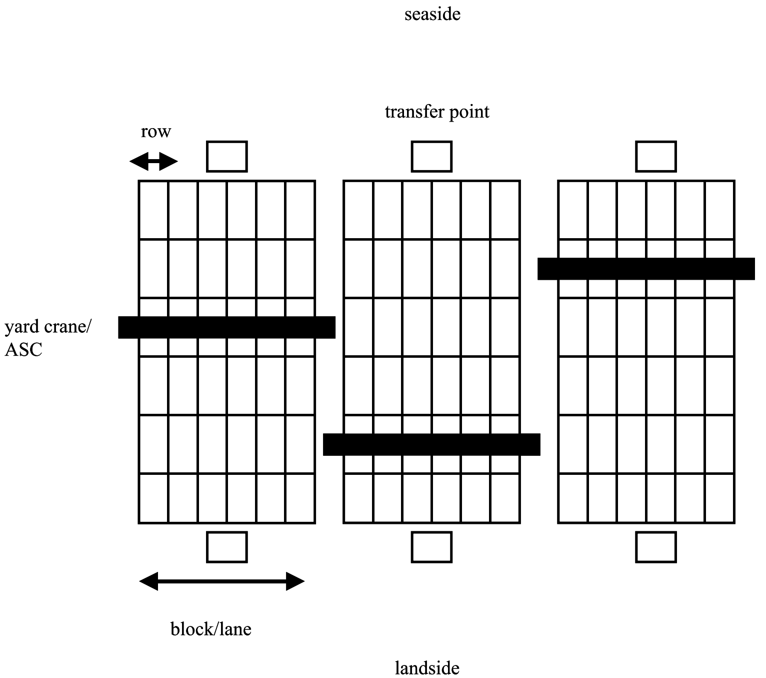

FIGURE 7.6. Schematic top view of the stack.

A decision at the strategic level that has to be made, is choosing the type of material handling equipment that will take care of the storage and retrieval of containers in and from the stack. Systems like forklift trucks, reach stackers, yard cranes and straddle carriers can be chosen. Yard cranes (see Figure 7.7) move on rubber tires or on rails over the containers. They can provide high density storage and can be automated. These automated cranes are called Automated Stacking Cranes (ASCs). ASCs, see

Figure 7.7, move on rails and are controlled by the central operating system. The ASC takes/places the container with a spreader from/on the AGV. At the port of Rotterdam, the containers can be stacked six wide and two or three levels high per ASC.

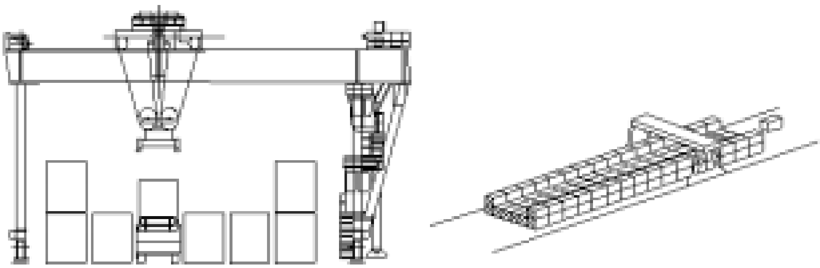

FIGURE 7.7. Yard Crane/Automated Stacking Crane.

Most of the described terminal operations have their origin and destination at the stack, for example the transport of containers from the stack to the ship and vice versa. The process of storing and retrieving containers should be executed such that the remaining operations in the terminal can be carried out effectively. The efficiency of stacking depends among other things on the stack height and strategies for storage and retrieval planning of import and export containers.

Consequences of higher stacking are a higher number of reshuffles/rehandles. To reach a specific container it can be necessary to rehandle containers that are placed on top of the demanded container. To minimise delay by removing containers, reshuffling of the stack can be done in advance. On the other hand, the higher the stacking the less ground space is needed for the same number of containers. Obviously, one of the problems at the strategic level is to determine a good stack layout. Questions concerning storage planning are, for example, where is an incoming container stored, in which order are containers stored and which crane handles which container. For retrieval planning it has to be decided in which order containers are retrieved and which crane handles the request. Firstly, we will discuss literature on the stack configuration. Secondly, we treat literature concerning storage and retrieval planning.

In Chen [42] it is concluded that higher stacking needs the improvement of all the other relevant conditions at the same time to reduce its possible impact. Otherwise, large numbers of unproductive container movements are needed. Chung *et al.* [45] develop and test strategies that can reduce the unproductive movements of the stack crane during the loading process and as a result reduce the total container loading time. They propose the idea of using a buffer area, located halfway between the stack and ship, where a number of empty chassis are available to store export containers temporarily. A simulation model is developed to investigate the effects of this buffer area on the port's operation. The conclusion is drawn that intermediate storage is effective. On average, 4% reduction in the total loading time can be obtained by using a buffer space. According to the authors, this method can be implemented in every container port.

De Castilho and Daganzo [51], conclude that for a good configuration of the stack, methods are needed to estimate the number of moves to retrieve a container as a function of stack height and operation strategy. As a result it is possible to trade off extra handling effort for higher stacking against space requirements. Moreover, the best operating strategy can be selected for the chosen configuration. Furthermore, the problem of optimal space allocation related to optimal storage pricing policies is discussed in Holguín-Veras and Jara-Díaz [108].

In Chen *et al.* [40] the storage space allocation problem is examined. A time-space network is developed to assist in assigning containers to storage locations in advance. A time-space network can represent entities moving in time and space. Thereafter, a mathematical programming model can be developed. The objective of this model is to minimise total costs of operation. A test case and a real world case are solved with a branch and bound algorithm.

In Kim and Kim [134] the storage space allocation problem is also treated, with decision variables stack height and allocation space. The problem is discussed for import containers with a dynamic space requirement. The objective is to minimise the number of reshuffles under the condition that the space requirements are met. Firstly, the case in which import containers arrive with a constant arrival rate is observed. The stack height and the amount of space are decision variables. It is derived, that the optimal height of the stack equals the total number of import containers during the length of the planning horizon divided by the total number of available locations in the stack.

Secondly, it is assumed that the arrival rate of import containers follows a cyclic pattern with the period of one week. Thirdly, the case in which the arrival rate of containers varies in an irregular way on a rolling horizon is observed. Both problems can be solved by formulating a linear program model. The solution can be obtained by solving the dual problem and related subproblems by applying the subgradient optimisation technique. Summarising, the problem is solved for different cases by determining firstly a formula representing the relationship between the stack height and number of rehandles and secondly by determining methodologies based on Lagrangian relaxation.

In Kim [130] methods are given for the evaluation of the rehandling of containers, when import containers are picked up in a random way. Methodologies are presented to estimate the expected number of rehandles for the next container to be picked up and the total number of rehandles to pick up all containers. As expected the total number of rehandles increases when the height of the stack increases. The paper only observes the case in which containers for different ships are separated. According to Cao and Uebe [35] the repositioning of containers is closely related to the p-median transportation problem, namely the transportation problem of containers from rows to be emptied to p rows not to be emptied. Kim and Bae [131] study the problem of the repositioning of containers such that containers are stacked in the same order of loading. The loading plan of a ship can be translated to a target layout of the stack. The objective is to replace containers such that the current layout of the stack is converted into the target layout such that the total travel time of the cranes is minimised. Mathematical programming techniques, like dynamic programming, are used to solve the problem.

In Taleb-Ibrahimi *et al.* [262] results are obtained for long-term and operational planning. They give a description of handling and storage strategies for export containers and quantify their performance according to the amount of space and number of handling moves. At the strategic level the minimum amount of storage space needed is determined. At the operational level the problem of minimising and predicting the amount of work is discussed. Models are given that reflect the relationship between available handling effort, storage space and traffic demand. In Kim *et al.* [138] the problem of determining storage locations for export containers with a certain weight is considered. It is required to minimise the expected number of rehandles for the loading of containers on the ship. These rehandles occur for example if lighter containers are stacked on top of heavier containers, which, as assumed in this paper, are needed first in the ship. A dynamic programming model is formulated to solve this problem. For making real time decisions a decision tree is given. The performance of this decision tree is evaluated by comparing its solutions to the solutions of the dynamic programming model. Maximally 5.5% of the decisions made with the decision tree is wrong.

At the tactical level the number of transfer cranes has to be determined necessary to ensure an efficient storage and retrieval process. In Kim and Kim [133] it is discussed how the optimal number of straddle carriers can be determined for import containers. According to the authors, there exists a trade off between the storage density, the accessibility, investment and service to outside trucks. A model is developed to solve analytically this trade off. The sum of all costs is minimised with respect to the number of straddle carriers and amount of space.

If straddle carriers take care of the storage and retrieval of containers from the stack, it has to be decided at the operational level how to route straddle carriers through the stack. In Kim and Kim [137, 140, 141] optimal routes of a single crane during loading operations of a ship are determined. Containers are divided over a number of categories. The locations where containers of a certain category are positioned are known. The container handling time (i.e. the total travel time of the crane) has to be minimised by optimally determining the stack lane sequence and the number of containers of each category to be picked up at each stack lane. The loading schedule has to satisfy the work schedule of the quay cranes which is assumed to be input. A tour of a crane consists of connected subtours. A subtour is a sequence of stack lanes which are visited by a crane to pick up all containers which will be loaded together at a ship's hold. A tour of a crane can be expressed as a route on a network. In the constructed network the problem is to find a path from the source to the sink and to determine the number of containers to be picked up at each node during the tour (the sum equals the number of containers in the work schedule of the crane) while minimising the travel time. This problem can be formulated as a mixed integer program. The total distance travelled in a lane is constant regardless the loading sequence of containers. Therefore, only the movements between lanes are considered in the total travel distance. With the special structures of this formulation an efficient solution method is developed. This solution algorithm consists of two procedures, namely a procedure to determine basic feasible solutions to the problem of determining the number of containers to be picked up at each lane. Secondly, a dynamic programming procedure is given to determine the route of the crane. Twenty-four problems are tested and it is concluded that the runtime of the algorithm depends highly on the number of combinations of the basic feasible solutions. From the computations it can be concluded that practical problems of a moderate size can be solved by using this algorithm. Kim and Kim [136] study the same problem. In this paper the problem is solved by using a beamsearch algorithm. This is a heuristic method for solving large combinatorial problems by efficiently exploring search trees. Each node in this tree represents a partial path from the first partial tour to the current one. Compared to branch and bound methods a beamsearch algorithm rejects unpromising nodes in an aggressive careless manner. The performance of the algorithm is tested by solving 360 sample problems. For small-sized problems the average value of solutions from the heuristic is 114.3% greater than the optimal solution. In both papers, the routing algorithm is only developed for a single crane. To have real practical use, the model should be extended to address multiple cranes. Furthermore, the model does not address both unloading and loading operations simultaneously. Also, operations at the landside are neglected at the model.

Namely, another typical problem for a container terminal is that containers have to be stored and retrieved at two sides of the stack (see Figure 7.1 and Figure 7.6), namely seaside (to/from the ship) and landside (to/from other modalities). This can be done by the same yard crane/ASC. Some of the decisions that have to be made to ensure an efficient process are: which side has the highest priority (commonly seaside) and how long can containers wait before they are stored or retrieved.

The problem to decide which ASC carries out which job, can be examined in two ways. If every container is treated as an individual (QC asks for a specific container from the stack), then it is clear which ASC should carry out the job. However, one can also distinguish container categories in a stack. This holds especially for empty containers. Containers with, for example, the same destination, the same weight, contents and size belong to the same category. The problem of the packaging of containers is studied in Bortfeldt and Gehring [25], Chen *et al.* [39], Davies and Bischoff [50] and Scheithauer [234]. If the QC asks for a container from a certain category a choice can be made between different containers in different stack lanes taking into account the planned workload of the ASC. The problem when the job should be carried out should be examined at the same moment.

Furthermore, at the operational level a schedule of the order in which containers are retrieved has to be determined. Kozan and Preston [158] use genetic algorithms as a technique to schedule the retrieval of containers from the stack. The objective is to minimise the time ships spend at the berth for the unloading and loading process. Therefore, they want to minimise the sum of setup times (i.e. the time necessary to retrieve containers from the stack) and travel times (i.e. the time necessary to transport containers from the stack to the ship). The authors suggest that research should be done into the use of other heuristics, like neural networks or tabu search, to see if they are more efficient than genetic algorithms.

7.1.5 Inter Terminal Transport and other modes of transportation

Containers have to be transported from the stack to other modes of transportation, like barges, rail and road. It is expected that, with the growth of terminals in the future, this inter terminal transport becomes more and more important. According to Van Horssen [275], new concepts and technologies have to be developed to handle the large numbers of inter terminal container transports expected in the future. Furthermore, research has to be done to the various transport systems by which containers can be transported between the terminals.

Multi-trailer systems (see Figure 7.4) and automated guided vehicles (see Figure 7.5) can carry out this inter terminal transport. In certain terminals it is possible that containers are put directly on, for example, trains without using transport vehicles.

Kurstjens *et al.* [160] studies the multi-trailer system. A method is presented that can be used for the planning of the inter terminal transport. This method is based on a technique which tries to minimise the number of empty trips. To obtain the minimum number of trucks needed an integer linear problem model is developed. For a particular case, it is concluded that the utilisation of the multi trailer systems can be reduced. But on the other hand the number of transport vehicles can hardly be reduced. Duinkerken *et al.* [60] developed an object oriented simulation model to investigate the performance of AGVs, Automated Lifting Vehicles (ALVs) and MTSs used for the inter terminal transport of containers. It is concluded that the number of ALVs required is equal to half the number of AGVs required.

One way of transporting containers to other destinations is by rail. Kozan [155] develops an analytically based computer simulation model to describe the container progress at a rail container terminal. Furthermore, the major factors influencing the throughput time of containers, which is a function of cranes, stackers and transfer systems, are discussed. The simulation model combined with heuristic rules describes the progress of containers in the system. Firstly, a cyclic heuristic rule is used to assign handling equipment to trains. This rule selects the first available resource from a group of resources beginning with the successor of the previous resource seized of the group. As a result, workloads are balanced and utilisation of handling equipment and throughput are higher. Secondly, a new heuristic rule is developed to dispatch trains to tracks. When a train enters the system there may or may not be a queue for the tracks. If there are no free tracks, the train will join the queue. Otherwise, the system sends trains first to track 1 and then to track 2 or 3 if they become available for track 1 and if they minimise total throughput time. In the case that more than one track is used, the train with the fewest number of containers will be unloaded first. A simulation model is developed by using data from a terminal in Australia. Due to cyclic train schedules a weekly simulation period was used. It is concluded, by applying the Wilcoxon Rank Test between the simulation output and the observed data for the total througput times of containers, that the simulation program imitates the rail terminal effectively. The rail terminal is also a starting point of Bostel and Dejax [26]. They observe the allocation of containers on trains. Different models and solution methods are given and tested on realistic data. It can be concluded that the number of container moves and the use and quantity of equipment can be decreased. Newman and Yano [198] address the problem of simultaneously determining train scheduling and container routing. Direct and indirect trains are scheduled and containers are allocated to these trains. Operating costs should be minimised while meeting on-time delivery requirements.

Another way of transporting containers to other destinations is on the road by trucks. In Ballis and Abacoumkin [13] a simulation model is developed that can be used in the design and evaluation of terminal facilities at the landside. Five heuristics are incorporated in the model to investigate the performance of the system. The comparison between different studies indicates that a shorter truck service time is feasible but that this leads to an increase of traffic conflicts in the internal transport network.

7.2 Complete container terminals

In Section 7.1 only problems for individual types of material handling equipments are discussed. It is obvious that in order to obtain an efficient terminal, it is also necessary to address all problems as a whole. The methods and algorithms obtained by optimising the single processes can be used as a base to optimising all processes simultaneously. To evaluate control concepts, layouts and material handling equipments, simulation can be used. For better understanding of all processes and decision making, various experiments with the model should be carried out. By using simulation, contrary to most analytical models, it is possible to study problems that arise simultaneously at several levels and furthermore to investigate results that are obtained by examining different material systems simultaneously.. On the other hand it is a time consuming job to develop and validate the model.

In Gambardella *et al.* [85] it is shown how operations research techniques can be used to generate resource allocation plans. These plans can be used by terminal managers to determine the best management strategies. Ramani [214] develops an interactive planning model to analyse container port operations and to support its logistics planning. It is assumed that all unloading operations are completed before loading operations are started. In the simulation model of Yun and Choi [296] an object-oriented approach is used. The performance of a simple model, in which many design parameters affecting the performance are changed, is observed. Other simulation models for container terminals are developed in Merkuryev *et al.* [191].

Instead of the time consuming simulation models, analytical models can be used. Contrary to simulation models, it is in general necessary to simplify the problem in such a form that it can be solved. In Van Hee and Wijbrands [274] a decision support system for the capacity planning of container terminals is developed. Several mathematical models, each describing parts of the complete process, are incorporated in this system. The system can support decisions at the strategic and tactical level. It is not meant for day to day planning. This decision support system is partly based on the system, for a breakbulk terminal, developed by Van Hee *et al.* [273].

In Kozan [156] analytical and simulation planning models for a complete terminal are compared. A batch-arrival multi-server queuing model is developed and compared with a simulation model. The results of this comparison indicate that, at a 95% level of significance, there exists little difference between the models. However, before implementing the analytical model, long term data collection is necessary.

In Kozan [157] the problem is examined of the minimisation of handling and travelling times of import and export containers from the time the ship arrives at the port until the time they are leaving the terminal and vice versa. The complete trajectory that containers go through from the ship to road or rail terminals via storage areas is catched into a network model. The objective in this model is to minimise total throughput time, which is the sum of handling and travelling times of containers. It is shown that the expected number of moves per container is the average of the maximum stack height and the minimum stack height. It is explained that this model can be used as decision tool in the context of investment appraisals of multimodal container terminals. Before implementing the model long-term data collection is required.

Meersmans [188] gives mathematical models for the integrated scheduling of handling equipment at container terminals. Exact and heuristic algorithms are presented to solve these models. To test the performance of these algorithms computational studies are performed. Meersmans and Wagelmans [190] propose a model that deals with the integrated scheduling of all types of material handling equipments at an automated container terminal. The objective is to optimise the overall performance of the container terminal. by minimising the makespan of the schedule. A branch and bound algorithm is proposed which produces optimal or near optimal schedules. For large problems a beam search heuristic is presented. It is shown that with the heuristic solutions close to optimal solutions can be obtained in reasonable computation times.

7.3 Concluding remarks

In this chapter we have successively described all subprocesses at a manned or automated container terminal and also the planning of the complete terminal. All separate types of material handling equipments and their decision problems at manned container terminals and automated container terminals are discussed. As a result, we have obtained a classification of decision problems at a container terminal. For each decision problem an overview of literature is given. In most cases analytical or simulation models are used to solve the problem.

From examining the literature, it is apparent that, it is in general considered to be necessary to simplify the problem, due to its complexity, before it can be solved by analytical models. Analytical models that are used most often are queueing models, network models and assignment problems. Branch and bound approaches are often used to solve the problems. On the other hand, simulation models can be used. In general it is a time-consuming job to develop and validate this kind of models.

In this chapter it is shown that already numerous research has been done to solve decision problems at container terminals. However, a number of questions are still open for research. Firstly, the priority planning of containers at the stack can be studied. In which way should we deal with the fact that the containers originated and designated for the seaside have a higher priority than landside containers. Furthermore, it is necessary to extend models for simple cases to more realistic situations. For example, the case of routing a single straddle carrier during loading operations should be extended to the case of routing a number of straddle carriers during loading and unloading operations at both sides of the stack.

7.4 Outline of part III of this thesis

In the literature little attention has been paid to the problem of the determination of the number of vehicles required to transport all containers in time. In Chapter 8 we study a container terminal in which automated guided vehicles or self lifting vehicles can be used. In the case that AGVs are used, containers must be transported the moment they are available for transport to ensure that no delays occur at the quay cranes. Assuming that the release time of each container is known, the minimum flow algorithm of Chapter 3 can be applied to determine the minimum number of AGVs at a container terminal. In the case that lifting vehicles are used, the transportation process and storage process can be decoupled by buffer areas with a fixed capacity.

For each container a time-window can be defined in which the transport should start. We explain in which way the integer linear programming of Chapter 4 can be applied to solve the problem of determining vehicle requirements under time-window constraints. With a simulation study we can obtain a feeling about the difference in number of AGVs and number of lifting vehicles required at a terminal. Furthermore, the impact of several factors on the number of vehicles is studied.

In Chapter 9 we consider the storage and retrieval process of containers. From the literature mentioned in this chapter it can be concluded that in literature only the problem of routing a straddle carrier during loading operations is studied. We discuss the problem of routing a straddle carrier during loading and unloading operations at both sides of the stack. It is explained in which way the dynamic programming algorithm of Chapter 6 can be applied to determine shortest routes for a straddle carrier performing storage and retrieval requests at both sides of the stack in multiple rows of containers. To compare route lengths between several types of equipments, we also study the determination of route lengths for automated stacking cranes. The algorithm of determining an optimal order of handling storage and retrieval requests in one aisle with two input/output stations of Chapter 6 can be applied for the routeing problem of automated stacking cranes. A simulation study is performed to gain insight in the way route lenghts differ for both types of equipments. Furthermore, a sensitivity analysis on several factors impacting the route length is executed.

8

Minimum vehicle fleet size at a container terminal

One of the main activities at a container terminal is transporting containers from the ship to the stack (i.e. storage area) during the unloading of the ship (see also Chapter 7). To ensure that this transportation process is carried out fast and efficiently, one of the problems that has to be solved in the planning process is the determination of the number of vehicles required to transport all containers in time from the ship to the storage area. This is a complex problem, both with stochastic and deterministic aspects. To gain more insight into the problem we consider two idealised cases with full information that do not pretend to be actual modellings of real terminals. However, they allow a thorough analysis of this planning problem in a container terminal. The objective of this chapter is to compare the following two idealised cases.

Firstly, we consider a deterministic case in which containers are transported immediately by non lifting automated guided vehicles without the use of buffers. In other words, the vehicle must be present if the quay crane takes the container off the ship. The quay crane places a container on the vehicle. Thereafter, the vehicle transports the container to the stack. At the stack a stacking crane takes the container off the AGV and stores it in the stack. As a result, the transportation process is tightly connected to the other processes. To avoid delays in the unloading process, immediate transport of containers is required. The problem is to determine the minimum number of vehicles required to transport each container at its release time.

Secondly, we consider a container terminal in which buffer areas with limited capacity and lifting vehicles are used. Consequently, the (un)-loading process and transport are decoupled. Quay cranes take the containers off the ship and place them in buffers. Lifting vehicles lift the containers off the ground and transport them to the stack. At the stack also buffer areas are located. The vehicle puts the container in the buffer. After a certain period of time the stacking crane stores the container in the stack. Due to the decoupling of the processes in the terminal, immediate transport of containers is not required. However, due to the limited capacity of the buffers each container has a final time of transport. The problem is to determine the minimum lifting vehicle fleet size during the unloading process under time-window constraints. After determining the minimum number of non lifting and lifting vehicles, we can compare both cases.

In Section 8.1 a more elaborate description of both problems is presented. The algorithms from Chapters 3 and 4 can be applied to determine the minimum number of vehicles in these two cases. In Section 8.2.1 it is discussed in which way the method from Chapter 3 can be applied to the problem of determining the minimum number of non lifting vehicles. The number of lifting vehicles can be determined by applying the method from Chapter 4. The two cases are compared by a simulation study to observe whether the use of buffer areas can give an advantage over the immediate transport of containers. In Section 8.3 results of this simulation study are presented. Finally, concluding remarks are given in Section 8.4.

8.1 Problem Description

We consider the unloading process at an automated container terminal. We only consider the unloading process, since the way in which the number of vehicles is determined, is the same for both loading and unloading. To transport containers, automated guided vehicles or automated lifting vehicles can be used. For the storage and retrieval of containers in the stack, terminals can use automated stacking cranes. Containers, unloaded off the ship by manned quay cranes (QCs), are transported from the ship to the stack. There are two possibilities for the transfer of containers from the quay crane to the transport vehicle. Firstly, the QC places the container directly on the automated guided vehicle (AGV). Secondly, the QC places the container in a buffer area and at a certain moment an automated lifting vehicle (ALV) lifts the container and transports it to the stack. By making a choice for non lifting or lifting vehicles a choice between buffering and non-buffering needs to be made. We will discuss both situations in more detail in the next subsections.

8.1.1 *Terminal without buffers*

In the case that non lifting vehicles are used, the QC directly places a container on a vehicle. After receiving the container, the vehicle transports the container from the ship to the storage area (i.e. stack). Automated guided vehicles drive on predefined tracks (see also Chapters 2 and 7). At the terminal, we consider vehicles travelling in a loop between stack and ship. Furthermore, each crane has its own AGV-track. This layout is represented in Figure 8.1. When an AGV arrives at the stack, the stack crane takes the container off the vehicle. Thereafter, the stack crane stores the container in the stack.

Since the waiting times of cranes are much more expensive than the waiting times of the vehicles, containers should be transported immediately in the case that no buffer areas are used. Consequently, a vehicle should be ready to transport the container once the crane has taken the container out off the ship or stack. To obtain a vehicle control concept that ensures that each container is transported in time, one of the problems that has to be solved, is the determination of the minimum number of vehicles required to transport all containers. This problem is examined in general in Chapter 3. The number of vehicles obtained serve as input for the operation. Which vehicle transports which container along which route should be decided during operation. Therefore, the possibility to adjust the number of vehicles, due to unexpected events, like breakdowns, is still open.

We assume the following:

1. N containers need to be transported from the ship to the stack.
2. The capacity of an automated guided vehicle is one container.
3. Jobs at QCs and stacking cranes are known in advance and are considered as input. By assuming that there is no interference at the QCs and no interference at the stacking cranes, we aggregate all QCs to one QC and all stacking cranes to one stacking crane. As a result, unloading jobs can be considered to be released at one QC and be transported to one stacking crane. The number of automated guided vehicles required for all quay cranes equals the product of the number of cranes and the number of automated guided vehicles required at one crane.
4. The release time of each container can be determined with an empirical distribution. To use the methods of Chapters 3 and 4 the release time of each container is determined in advance.
5. The travel times between ship and stack are deterministic and known in advance

6. The position where a container is available for transport and the destination are known in advance.

7. At the start of the simulation the AGVs are directly available at the QC.

8. No delays occur at the stacking crane. Cranes are directly available to lift a container from an AGV.

By assuming to have full information on some aspects a thorough analysis of both cases is possible. These assumptions are used in both idealised cases and therefore do no impact the comparison between both cases.

8.1.2 Terminal with buffers

QCs take containers off the ship and put them in buffers. Automated lifting vehicles can lift a container from the ground. Furthermore, a lifting vehicle carrying one container can drive over another container placed on the ground. After a certain period of time the ALV transports the container from the buffer to the stack. When an ALV arrives at the stack, the ALV puts the container in the buffer area. Thereafter, the stacking crane takes the container from the buffer and stores it in the stack. Automated lifting vehicles also travel on predefined tracks. In this situation, we also consider a terminal in which vehicles travel in a loop from the ship to the stack. Figure 8.1 illustrates this layout.

Buffer positions can be created under the crane on an ALV-track. In a terminal in which four cranes per ship are used and in which lifting vehicles are used capable of travelling with one container over another container, we assume that the maximum capacity of a buffer area under a crane equals four containers each positioned at an AGV track. Containers are not positioned next to each other in the buffer to ensure that the crane does not have to make extra moves, with extra travel times, to the left or right. If a vehicle arrives at the stack it either puts the container in the buffer area, or the stack crane takes the container off the vehicle. This layout is presented in Figure 8.1.

The advantage of buffer areas is the decoupling of the (un)-loading process at the Quay Crane and the transport process of the containers. The QC and stacking crane can continue with their work when no vehicles are available. On the other hand, the use of buffer areas can imply the use of self-lifting vehicles. The costs of such vehicles are considerably higher than the costs of non lifting vehicles. Consequently, the disadvantages and advantages should be weighed against each other by choosing a type of equipment.

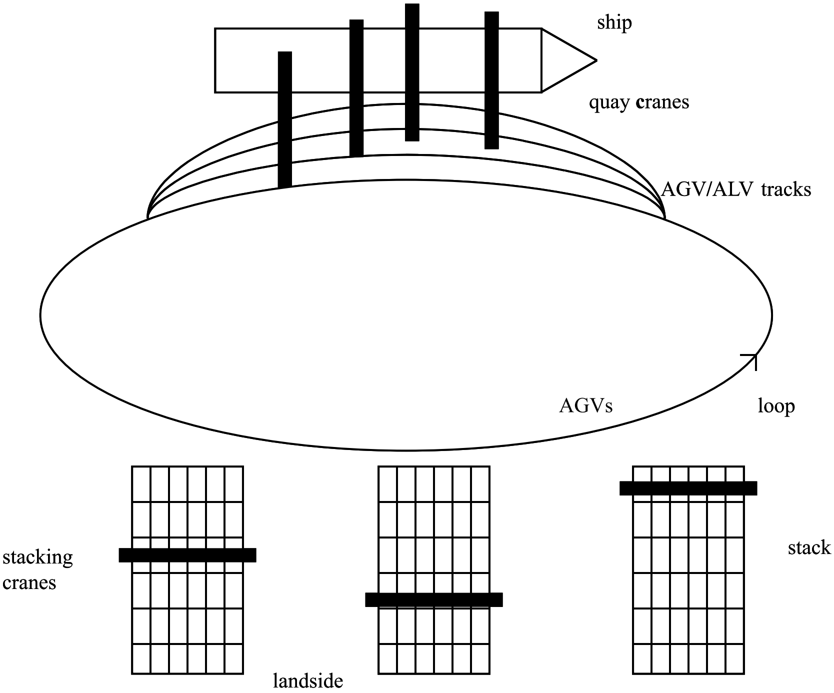

FIGURE 8.1. AGV/ALV tracks at a container terminal.

In the case that buffers are used, immediate transport of the containers is not necessary. Due to the fixed capacity of buffers and the fact that a new container needs to be positioned in a full buffer on the location of the oldest container in the buffer, each container has a final time of transport. Consequently, for each container a release time and a due time for pick up can be derived (see also Section 8.2.2). The problem is now to determine the minimum number of vehicles required to transport all containers within their time-windows. In Chapter 4 this problem is studied for vehicles in general. Besides the assumptions made in the previous section we assume the following:

1. A lifting vehicle carrying one container can drive over another container placed on the ground.

2. Buffer areas have a fixed capacity.

In the next section, we will explain in which way the methods from Chapters 3 and 4 can be applied to these specific situations. Thereafter, both cases are compared with simulation.

8.2 Usage of algorithms

8.2.1 Number of automated guided vehicles

To ensure that cranes do not have to wait for vehicles and that every container is transported in time, sufficient vehicles should be available to transport all containers. In Chapter 3 a deterministic model has been developed, which can be applied to determine the minimum number of vehicles required to transport all containers in time.

To solve the problem, a directed network is formulated in which all nodes represent a container. Two nodes are connected if the two containers can be transported by the same vehicle. This is only possible if the vehicle has finished transporting the first container and has travelled to the origin of the second container before or exactly at the start time of the second container. In this network, a directed path corresponds to a feasible sequence of containers, which can be transported by one vehicle. To obtain the minimum number of vehicles, it is necessary to determine the minimum number of directed paths such that each node is included in exactly one path. By determining a feasible flow in the original network and a maximum flow in the transformed network (backward arcs are added) the minimum flow in the network can be found. This minimum flow represents the minimum number of AGVs required such that each container is transported at its start time.

8.2.2 Number of automated lifting vehicles

If the vehicles that are used for the transport of containers can lift containers by themselves, then it is not necessary for the crane to place the container immediately on the vehicle. Instead, the crane places the container in a buffer, where it waits for transport.

The buffer is assumed to have a fixed capacity. For all containers it is known in advance at which time point they will be placed in the buffer by the crane. In this buffer the containers can wait for transport, but when the buffer is full and a new container should be placed in the buffer, at least one of them has to be transported first. To create space in the buffer, the earliest arrived container has to be transported first. Consequently, there is, next to a first point in time of transport (i.e. moment the container is placed in the buffer), a point in time at which the container ultimately has to be transported. This final point in time equals the time at which a new container should be placed in the full buffer. As a result, every container that has to be transported from the ship to the stack (i.e. unloading container) has a time-window (release time and due time) in which the container should be transported. This time-window can be discretised such that a finite number of start times for the container result. For each container, it needs to be decided at which of these start times the container is transported. By using this data and applying the method from Chapter 4, the minimum number of vehicles required such that each container is transported in its time-window can be determined.

8.3 Numerical comparison

8.3.1 Description of the simulation study

In this section both cases are compared by a simulation study. We consider the unloading process at an automated container terminal. We use realistic data to gain insight into the problem of determining vehicle requirements for both described cases. It is not pretended to model a real terminal, but to model a terminal which allows a thorough analysis.

The manned QC take containers off the ship. This takes a certain cycle time. Based on data used in a study at a large container terminal in the Netherlands, the cycle time (in seconds) has an empirical distribution which is represented in Table 8.1 (see Celen *et al.* [37]). Possible large disturbances, such as breakdowns of equipment, are not taken into account. With these data we can generate the points in time at which the containers are available for transport as follows: randomly select a class of cycle times according to the empirical distribution. In this class each cycle time has an equal probability to be chosen. That is, in a class, cycle times are assumed to follow a continuous uniform distribution. The release time of each container is determined in advance and therefore each release time is known when performing the experiments.

As explained in Section 8.1 we observe the release of containers at one QC. All containers have to be transported from one QC to one stack crane. The automated guided vehicles and automated lifting vehicles travel in a loop from the QC to the stack and vice versa (see Figure 8.1). The speed of a full vehicle is 4.0 m/s. The speed of an empty vehicle is 5.5 m/s. The distance between the QC and stack is 600 meters. As a result, the travel time of a full vehicle from the QC to the stack equals 150 seconds. The travel time of an empty vehicle from the stack to the QC equals 109 seconds. We assume that an unloading container is directly taken off the vehicle by the stack crane in the bufferless case or that the container can be placed immediately in the buffer.

Fraction	Unloading cycle time in seconds
0.05	30-40
0.15	40-50
0.20	50-60
0.19	60-70
0.19	70-80
0.10	80-90
0.08	90-120
0.03	120-150
0.01	150-180

TABLE 8.1. Empirical distribution of the unloading cycle times.

With the data from Table 8.1, we can derive a theoretical upper bound, lower bound and average for the number of automated guided vehicles necessary to transport all containers in time in the bufferless case. By assuming that all containers are unloaded off the ship by quay cranes in 30 seconds and by using the given travel times the theoretical upper bound for the minimum number of vehicles per crane equals nine. If every 180 seconds a container is unloaded from the ship the lower bound for the minimum number of vehicles per crane equals two. The average cycle time of the mentioned distribution equals 68.7 seconds. If the unloading cycle time of each container equals 68.7 seconds, a minimum of four vehicles is necessary.

In the simulation study, we consider the following instances: 50, 80 or 100 containers have to be unloaded. For every instance a number of replications is executed. As described, we can generate the points in time at which the containers are available for transport with the data from Table 8.1. At the previously mentioned terminal, a crane needs to work on average one hour to handle 50 containers. The planning horizon for 100 containers, therefore, covers two hours on average.

As described, we observe two cases, namely the use of buffers and the immediate transport of containers. Vehicles, with capacity one take care of the transport. In the case of immediate transport, the container has to be transported at the generated time point. In the case that a buffer area is used, we assume that the buffer has a fixed capacity. In Section 8.3.3 the effects of the buffer size are described. As a result, the first container placed in the buffer area finally has to be transported at the moment the buffer area is full and a new container is released. In the same way, for each container a final transport time (i.e. due time) can be generated. To ensure that the last unloading containers, which do not have to create space for other containers, are also transported, it is assumed that these containers finally have to be picked up for transport at their release time + 100 seconds. Consequently, every container has to be transported at some moment between its release time and its due time (i.e. within its time-window).

To solve the problem with the buffer area, the time-window has to be discretised (see Section 8.2.2). In the simulation, the time-window is discretised as follows: the release time equals the moment the container is placed in the buffer. This is also the first possible start instant of the job. Consequently, the second possible transport moment of the container equals the first possible transport moment plus the discretisation step. This is repeated until the new created possible transport moment is equal to or greater than the due time of the container. The last possible start instant is set to the due time of the job. Consequently, for each container a number of possible start instants is defined. In Section 8.3.3, it is explained which discretisation step is chosen for this simulation study.

With all these data, the problem of the determination of the number of vehicles required per used Quay Crane, can be solved using the methods described in Sections 8.2.1 and 8.2.2.

For a number of independent replications of each described instance, the two cases are solved and compared. To obtain a valid estimation of the number of vehicles per used Quay Crane, the necessary number of replications for each simulation experiment should be determined.

According to Law and Kelton [162], an approximation for the minimum number of replications i, such that the relative error is smaller than γ $(0 < \gamma < 1)$ with a probability of $1 - \alpha$ equals:

$$i \geq S^2(i)[z_{1-\alpha/2}/\gamma'\overline{X}(i)]^2 \tag{8.1}$$

where $S^2(i)$ is the sample variance, $z_{1-\alpha/2}$ the $1 - \alpha/2$ percentile of the normal distribution, $\overline{X}(i)$ the sample mean and $\gamma' = \gamma/(1 + \gamma)$. The minimum number of replications is determined, by solving Equation (8.1) with data from a limited number of replications. To obtain a relative error γ smaller than 2% with a probability of 95%, a replication size of 100 is sufficient for all experiments in this chapter. Therefore, for every simulation experiment 100 replications are generated.

As a result, by executing for example a simulation experiment for 50 containers, 100 times a release time for each container is generated by using the data from Table 1. Herewith and with the other data, like travel times, buffer size (if used) mentioned in this section 100 models are generated according to the methods described in the Chapters 3 and 4. By applying the methods described in Section 8.2.1 and Section 8.2.2 and using the solver CPLEX 6.5 for every replication the minimum number of vehicles necessary to transport all containers is obtained. To obtain an estimation of the number of vehicles that should be used to transport the N unloading containers, the maximum value obtained from the 100 replications is chosen. Furthermore, the average of the 100 replications is observed.

8.3.2 *Terminal without buffer areas*

The results for the situation where all containers have to be transported immediately are presented in Table 8.2. We observe the following instances: 50, 80 and 100 containers. For each instance, it is indicated how many replications are found that need the given number of automated guided vehicles to transport the containers. The maximum and the average number of AGVs are also given.

number of containers	k= 5	k= 6	k= 7	average # of vehicles	maximum # of vehicles
50	45	55	0	5.55	6
80	21	79	0	5.62	6
100	15	84	1	5.86	7
110	14	84	2	5.88	7
130	15	83	2	5.87	7
150	11	87	2	5.91	7

TABLE 8.2. Number of replications found needing a given number of vehicles (k) and the average and maximum value if no buffer areas are used.

From the results in Table 8.2, it is clear that per Quay Crane six or seven AGVs should be used to transport all containers at their release time. It can be seen that by increasing the workload from one hour to three hours (50 to 150 containers), the average number of AGVs increases little. An explanation therefore could be that the crane needs some warm up period.

Considering, the average number of AGVs, a minimum of six vehicles per crane is necessary. However, at one replication it is found that a minimum of seven AGVs should be used. From a safe point of view, a minimum of 28 vehicles for a terminal with four Quay Cranes is advisable. Only in a small number of cases 7 vehicles are needed. By using 6 vehicles per crane and by sharing one extra vehicle between the four cranes during operation, it might be possible that 25 vehicles will be sufficient.

This minimum is comparable with the number of vehicles that is used for a ship that is unloaded at the previously mentioned terminal by using four cranes. At this terminal, 25-35 AGVs are used per ship with four cranes. Some extra vehicles are used to deal with expected events, like preventive maintenance and filling up of the vehicles and unexpected events, like breakdowns of vehicles.

8.3.3 Terminal with buffer areas

Choice of the size of the discretisation step

As described in Section 8.2.2 and Section 8.3.1 for each container a time-window exists in which the container should be transported if buffer areas are used. In practice it can be decided at each moment in this time-window if the automated lifting vehicle starts transporting the container. However, to use the method described in Section 8.2.2 it is necessary to discretise the time-window. By discretising the time-window a number of time points are generated at which the container can be transported. As explained in Section 8.3.1 the time between two possible start points equals the discretisation step. It is clear that the size of the discretisation step influences the number of possible time points that are defined. To approximate the continuous reality as many points in time as possible should be generated in the time-window. Consequently, the discretisation step should be chosen very small. However, by using many time points the computation time of instances of the model increases. By increasing the size of the discretisation step, the number of time points at which the container can be transported will decrease. By doing this, it might be possible that less jobs are compatible. As a consequence the result of the model can be a larger number of vehicles than is necessary. Therefore, the choice of the size of the discretisation step should be made carefully.

To determine an appropriate size of the discretisation step for the other experiments in this chapter, we have executed the following simulation experiment. 50 containers have to be transported by automated lifting vehicles. The cycle times of the containers are generated by using the distribution represented in Table 8.1. As a result, for every container the release time is known. The size of the buffer is in this experiment set to one. Consequently, the due time of every container equals the release time of the next container. The due time of the last unloading container equals its release time plus 100 seconds. To use the model of Section 8.2.2, the time-window of every container is discretised. To observe the effect of the discretisation step on the minimum number of vehicles necessary to transport all containers within their time-window various discretisation steps have been tested, namely 50, 40, 30, 20, and 10 seconds. In Figure 8.2 the results of the experiment are given.

The exact numbers as depicted in Figure 8.2 are also represented in Table 8.3.

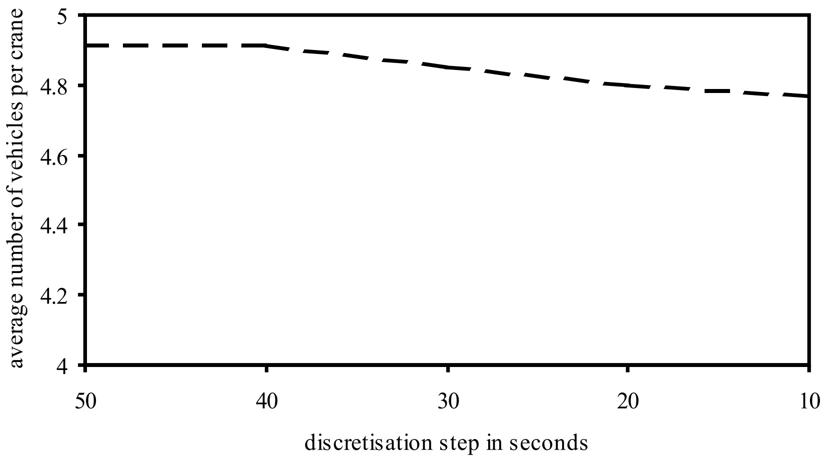

FIGURE 8.2. Average number of vehicles per crane if 50 containers have to be transported and the buffer size equals 1. The size of the discretisation step varies between 50 and 10 seconds.

size of the discretisation step	50	40	30	20	10
average value	4.91	4.91	4.85	4.8	4.77

TABLE 8.3. Average number of vehicles per crane if the discretisation step varies between 50 and 10 seconds.

The maximum value among the replications that has been found in the simulation experiment equals five vehicles per crane for all different discretisation steps. From Figure 8.2 it can be concluded that the average number of vehicles per crane decreases slightly (maximal 2.9%) if the size of the discretisation step decreases from 50 seconds to 10 seconds. As mentioned before, the smaller the discretisation step, the larger the number of possible time points at which the container can be transported. Therefore, it might be possible that with a smaller discretisation step more jobs can be transported by the same vehicle, which results in a smaller number of vehicles that is necessary to transport all containers within their time-window. However, it is clear that the reduction in the average number of vehicles that is obtained by reducing the size of the discretisation step, is relatively small. Furthermore, the maximum value found among the replications is the same for all sizes. As a result, we choose a discretisation step of 50 seconds for the other experiments in this chapter.

Choice of the buffer size

The moment at which the containers finally should be transported depends on the size of the buffer area. The larger the buffer, the later the moment the buffer is full. Consequently, the final transport moment is later than by using a smaller buffer. As a result, it might be possible to transport more containers by the same vehicle. This could result in the use of less vehicles. However, by using a larger buffer the computation time of instances of the model increases. In practice the capacity of the buffer area is also limited. As explained in Section 8.1, the maximum physical capacity of the buffer at the QC equals four. Considering all these aspects, the choice of the buffer size should be made carefully.

To determine an applicable size of the buffer for the other experiments in this chapter the following experiment is executed. 50 containers have to be transported by lifting vehicles. The cycle times of the containers are generated by using the distribution represented in Table 8.1. The discretisation step equals 50 seconds (see Section 8.3.3). To observe the effect of the buffer size on the minimum number of vehicles necessary to transport all jobs within their time-windows the buffer size varies between zero and four.

FIGURE 8.3. Average number of vehicles per crane if 50 containers have to be transported and the discretisation step equals 50 seconds. The size of the buffer varies between 0 and 4.

size of the buffer area	4	3	2	1	0
average value	4.40	4.51	4.69	4.91	5.55

TABLE 8.4. Average number of vehicles per crane if the size of the buffer area varies between 0 and 4.

From the results in Table 8.4 and Figure 8.3 it is clear that a reduction in the number of vehicles can be obtained by introducing a buffer area. The maximum value among the replications that has been found for the minimum number of vehicles equals five for all buffers with capacity greater than zero. Furthermore, it can be seen that the reduction in the average number of vehicles that can be obtained by increasing the size of the buffer from two to three or four is very small. Considering this and computation times of instances of the model by using larger buffer areas, we have chosen to use a buffer size of two for the following experiments.

Results

In the following simulation study it is examined how many vehicles are required by introducing a buffer area of size two (see Section 8.3.3). Again a discretisation step of 50 seconds is used. The results for this situation are presented in Table 8.3. We observe the following instances: 50, 80 and 100 containers. For every instance it is indicated how many replications are found with the given number of vehicles. The maximum and the average number of vehicles are also given.

number of containers	# replications with # veh. = 4	# replications with # veh. = 5	average # of vehicles	maximum # of vehicles
50	31	69	4.69	5
80	10	90	4.90	5
100	4	96	4.96	5

TABLE 8.5. Number of replications found for a given number of vehicles and the average and maximum value if a buffer of size two is used.

From Table 8.5 it is clear that five lifting vehicles per crane should be used to transport all containers within their time-window. Consequently, for a terminal in which ships are unloaded by using four cranes at least 20 lifting vehicles are necessary to transport all containers in time.

8.3.4 Comparison

As explained in Section 8.1, one of the decisions that container terminals have to make is the choice of equipment that is used. Therefore, it has to be considered among other things if lifting vehicles or non lifting vehicles are used. In the case lifting vehicles are used, buffer areas become a part of the layout of the terminal. Because of the fact that lifting vehicles are more expensive than non lifting vehicles, it is interesting to find out if a reduction in the minimum number of vehicles necessary to transport all containers in time can be obtained by introducing buffer areas in the terminal. Therefore, it is also in this simulation study interesting to compare both cases.

From Table 8.2 it is clear that 28 non lifting vehicles should be used in a container terminal without buffer areas. In a terminal with buffer areas, it can be concluded from Table 8.3 that 20 lifting vehicles are required. As a result, a reduction of eight vehicles (i.e. 29% of the total number of vehicles) can be obtained by using buffer areas in a terminal using four cranes. Therefore, from the results of comparing both idealised cases it can be concluded that the use of a buffer with capacity two is still interesting if lifting vehicles are at most 30% more expensive than non lifting vehicles.

8.3.5 Varying cycle times

It is clear that many factors, like number of cranes, costs, unloading times, influence the number of vehicles that will be used in the terminal. In this section the influence of the cycle times of the containers on the minimum number of vehicles will be tested. The time necessary for a quay crane to take a container out the ship depends for example on the capabilities of the crane driver, the specifications of the crane, the position of the container in the ship and so on. If a crane is capable of unloading containers very fast, the time between two successive containers is smaller. Because of the fact that the driving times of the vehicles remain the same as mentioned in Section 8.3.1, it might be expected in the bufferless case that less containers can be transported by the same vehicle. As a result, we expect an increase in the number of automated guided vehicles. If the unloading of a container takes more time, the time between two successive containers will increase. By the same driving times of the vehicles, it might be possible that in the bufferless case more jobs can be transported by the same vehicle. As a consequence, the necessary number of vehicles will decrease.

In the following study, we observe the effects of a crane that unloads containers faster and slower, than the data given in Table 8.1, on the number of vehicles that is necessary to transport all containers in time. Faster and slower in this context means that the intervals mentioned in Table 8.1 will decrease or increase with a given number of seconds. The fractions remain the same as in Table 8.1. If a crane unloads for example 5 seconds faster, the first interval with fraction 0.05 becomes $25 - 35$ seconds. Consequently, the average cycle time will be 5 seconds (7.3%) less than the average cycle time mentioned in Section 8.3.1.

Firstly, the effects of a faster and slower unloading process in a terminal without a buffer area are observed. The simulation study is executed for 50, 80 and 100 containers. We observe a crane that unloads containers 2, 5, 7 and 10 seconds slower and faster than the distribution in Table 8.1. The results are given in Figure 8.4. In this Figure the results for the original distribution (deviation of zero) are also given.

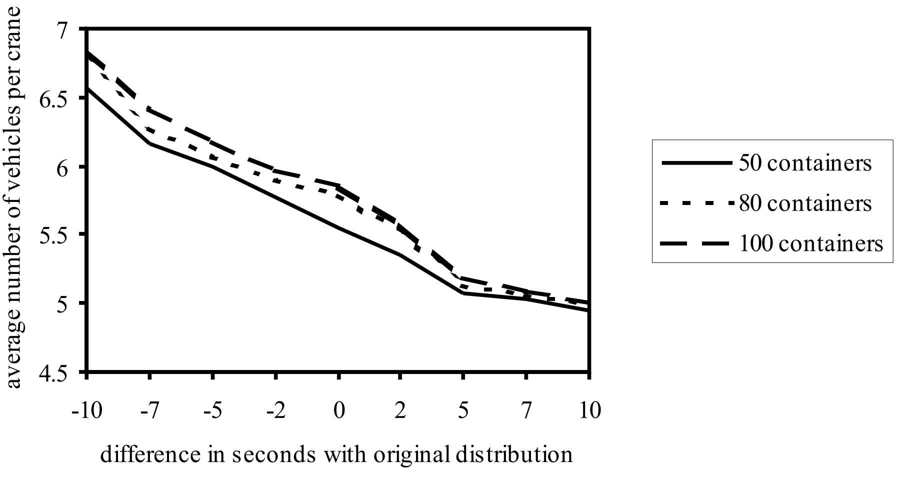

FIGURE 8.4. Average number of vehicles per crane if the cycle time deviation equals -10, -7, -5, -2, 0, 2, 5, 7 and 10 seconds of the distribution in table 1 in a terminal without buffer areas.

The exact numbers depicted in Figure 8.4 are also represented in Table 8.6.

difference in seconds	50 containers	80 containers	100 containers
-10	6.57	6.76	6.85
-7	6.16	6.28	6.42
-5	6	6.06	6.17
-2	5.77	5.9	5.97
0	5.55	5.79	5.86
2	5.35	5.53	5.58
5	5.07	5.13	5.18
7	5.03	5.06	5.08
10	4.94	4.99	5

TABLE 8.6. Average number of vehicles if the cycle time deviates -10, -7, -5, -2, 0, 2, 5, 7, 10 seconds of the original distribution.

From Figure 8.4 it can be concluded that the average number of vehicles decreases if the cycle times increase and that the average number of vehicles increases if the unloading cycle times decrease. In the case that the crane unloads 10 seconds faster (15%) the number of vehicles increases with 16.9%. If the cycle times increase with 10 seconds the number of vehicles decreases with 14.7%. These percentages are comparable. A decrease or increase of 15% in the cycle times results in respectively an increase or decrease in the number of vehicles of approximately 15%. If the cycle times of the containers are 10 seconds shorter than the distribution in Table 8.1 the maximum value among the replications equals eight vehicles per crane. On the other hand, if the cycle times of the containers are 10 seconds higher the maximum value among the replications equals five vehicles per crane.

The increase of the average and maximum number of vehicles is relatively higher. A difference of two vehicles is found by decreasing the cycle times with 10 seconds in contrast to a difference of one vehicle found by increasing the cycle times with 10 seconds. In this study the total travel time of a vehicle to transport the container and return to the crane remains 259 seconds (see Section 8.1). If jobs are unloaded faster the influence of a relative large travel time is larger than in the case jobs are unloaded 10 seconds slower.

By observing the maximum values among the replications it can be concluded that a crane can unload at most 5 seconds faster than the distribution given in Table 8.1 to keep the same number of vehicles as used in the original case (see Section 8.3.2).

Further, we observe a terminal with a buffer area of size two (see Section 8.3.3). The discretisation step used is 50 seconds (see Section 8.3.3). Again the study is executed for 50, 80 and 100 containers. We observe a crane that unloads containers 2, 5, 7 and 10 seconds faster and slower than the distribution in Table 8.1 In Figure 8.5 the results of the study are given. The results for the original distribution (deviation of zero) are also given.

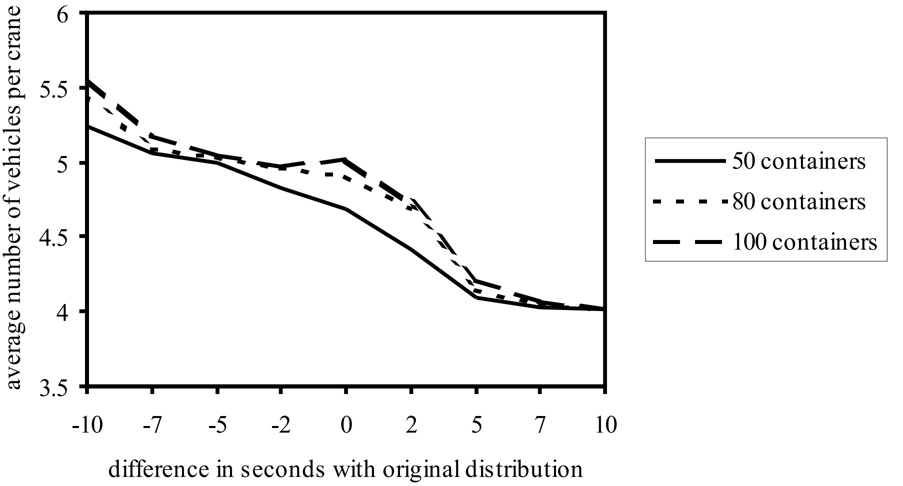

FIGURE 8.5. Average number of vehicles per crane if the cycle time deviate -10,-7, -5, -2, 0, 2, 5, 7 and 10 seconds of the distribution in table 1 in a terminal with a buffer area of size 2.

Table 8.7 represents the exact numbers depicted in Figure 8.5.

difference in seconds	50 containers	80 containers	100 containers
-10	5.24	5.45	5.56
-7	5.06	5.09	5.18
-5	5	5.03	5.04
-2	4.83	4.95	4.97
0	4.69	4.9	5.02
2	4.42	4.81	4.73
5	4.09	4.14	4.21
7	4.03	4.04	4.07
10	4.01	4.01	4.02

TABLE 8.7. Average number of lifting vehicles if the cycle time deviates -10, -7, -5, -2, 0, 2, 5, 7, 10 seconds of the original distribution.

From Figure 8.5 it can also be concluded that the average number of vehicles decreases with 19.9% if the cycle times of the containers increase with 15%. The average number of vehicles increases with 10.8% if the cycle times of the containers decrease with 15%. If the mean of cycle times increases, while the variance remains the same, then the percentage deviation from the mean will decrease. In other words, the uncertainty decreases and the buffers can easier compensate for these small changes in cycle times. Thus, enabling a percentage decrease in vehicles which is even larger than the increase in cycle times. The maximum value among the replications equals six vehicles per crane if the cycle times decrease with 10 seconds. If the cycle times increase with 10 seconds the maximum value among the replications equals five vehicles per crane. Furthermore, it can be concluded that a crane can unload only 2 seconds faster to keep the same number of vehicles as in the original case (see Section 8.3.3).

For 100 containers, the average number of vehicles equals 4.97 if the crane unloads 2 seconds faster than the original crane. In the original case the average number of vehicles equals 5.02. This small, unexpected, increase in the average number of vehicles is likely to be due to the effect of random numbers in the simulation experiment. Recall that the half width of the 95% confidence interval is 2%. A global decreasing trend can be noticed.

8.4 Concluding remarks

At a container terminal containers are transshipped from a ship to other modes of transportation and vice versa. One of the processes that takes place at a container terminal is the transport of containers from the ship to the storage and vice versa. This transport is carried out by self lifting or non lifting vehicles, like automated lifting vehicles or automated guided vehicles. One of the decisions that has to be made is the determination of the minimum number of vehicles required to transport all containers in time.

In this chapter two cases are examined. Firstly, a terminal is observed in which containers are transported by vehicles that cannot lift a container by itself. As a result, containers have to be transported immediately if they are available for transport and no buffer areas are used. In Section 8.2.1 it is explained in which way the polynomial time algorithm of Chapter 3 should be applied to this specific situation.

Secondly, vehicles that lift a container by itself are operating at the container terminal. Consequently, containers do not have to be transported immediately and buffer areas can be used. For every container a release time is given, which depends on the cycle times of the crane. The due time of the container is determined by the capacity of the buffer. In Section 8.2.2 the algorithm of Chapter 4 is explained to determine the necessary number of vehicles to transport all containers within their time-window.

At the strategic level the container terminal has to decide which type of vehicles will be used at the terminal. Lifting vehicles are more expensive than non lifting vehicles. Therefore, it is interesting to observe if a reduction in the number of vehicles can be obtained if lifting vehicles are used. In Section 8.3.4 it is concluded that a reduction of approximately 30% in the number of vehicles can be obtained in a terminal with a buffer area of capacity two. As a result, savings in the number of vehicles still can be obtained if the costs of lifting vehicles are at most 30% higher. Furthermore, the effects of the cycle times on the number of vehicles are studied. It can be concluded that slower or faster cranes have an impact on the necessary number of vehicles. Compared to the original distribution, cranes can unload at most 5 seconds faster to keep the same number of non lifting vehicles and at most 2 seconds faster to keep the same number of lifting vehicles.

9

Scheduling of container requests at stacking yards

One of the activities at a container terminal is the temporarily storage of containers. Containers are stored and retrieved in and from the stack by material handling systems (see also Chapter 7). One of the operational problems, that has to be solved, is the sequencing of storage and retrieval requests such that empty travel distances of the equipment are minimised. This is a complex control problem, both with stochastic and deterministic aspects. To gain more insight into the problem we discuss two cases in which a fixed number of storage and retrieval requests with known characteristics, like location in the stack, origin and destination, need to be sequenced without due times. With these deterministic cases a thorough analysis of this control problem in a container terminal is possible.

Various types of storage and retrieval equipments can be used to store and retrieve containers in and from the stack. Each type of equipment has it own characteristics. In this chapter we consider two different types of equipments. Firstly, we treat the case in which containers are stored and retrieved by straddle carriers. A straddle carrier operates in a block of containers. This block consists of a number of rows in which storage and retrieval requests need to be handled. The straddle carrier travels over a row of containers to perform a storage or retrieval. At both ends of each row pickup and delivery points are located at which the containers are transferred from and to another material handling system. At the top and bottom end of each row the straddle carrier can change rows.

Secondly, we consider the case in which automated stacking cranes store and retrieve containers in and from the stack. The automated stacking crane operates over a block of containers, which consists of a number of rows. Also in this case pickup and delivery points are located at both ends of each row. At each location in the row the trolley of the automated stacking crane (i.e. pick-up device) can change rows.

In both cases a number of storage and retrieval requests should be handled. The problem is to route the material handling equipment through the stack to handle all requests such that the empty travel distance is minimised.

In Section 9.1 a description of the problem is presented. The dynamic programming algorithm from Chapter 6 can be applied to solve the problem of sequencing the requests for a straddle carrier. The route for an automated stacking crane can be determined by applying the algorithm from Section 6.3.3. Section 9.2 explains in more detail in which way both methods should be applied to solve the problem for the described cases.

The two cases are compared by a simulation study. Various simulation experiments are executed in Section 9.3 to examine in which way average minimal empty travel distances differ for both types of equipments. Furthermore, the relation between empty travel distance and the total route length is studied. A sensitivity analysis is performed to study the impact of the number of containers, the number of rows and the number of locations in a row on the average minimum empty travel distance for both types of equipments. Finally, concluding remarks are given in Section 9.4.

9.1 Problem description

Various material handling systems can be used to store and retrieve containers in and from the stack. An elaborate discussion on the systems and stacking of containers is given in Chapter 7. In this chapter we consider two different types of material handling equipments, namely *automated stacking cranes* and *straddle carriers*. These types of equipments have different characteristics, which we will discuss later on.

To handle storage and retrieval requests the two types of equipments have to travel through the storage area. The total distance to be travelled consists of two components, namely full and empty travel distances. Both systems are capable of carrying one load at a time. To store or retrieve a container the system travels full from the origin to the destination of the container. This full travel distance is the same for both types of equipments, since they handle the same container. Furthermore, this full travel distance is known beforehand for each container since origin and destination of each container are known. To handle all requests a machine has to travel empty from one request to another. The total empty travel distance travelled to handle all requests can differ between those types of equipments. Instead of examining total travel distances we observe the varying empty travel distances only. The objective of this chapter is, therefore, to compare average empty travel distances for handling a fixed number of requests for both types of equipments by a simulation study. We consider the following situation. Straddle carriers and automated stacking cranes operate in a block of containers. Containers are stored for a certain period in such a block for further transport. To load the container on a ship or other modality the container is retrieved from the stack. The stack has two ends, namely one at the seaside and one at the landside. A block consists of a number of rows of containers with a pickup and delivery point at both ends of each row. These points are located at both sides such that containers required at the landside do not interfere with containers required at the seaside. At a pickup and delivery point the container is taken from or placed on the ground. These pickup and delivery points have infinite capacity. The transport of the container to and from this point is executed by another type of material handling equipment, for example a multi trailer system. Clearly, the storage and transportation processes are decoupled.

Both types of equipments have their own characteristics. A straddle carrier travels over a single row of containers to store or retrieve a container. At the bottom and top end of each row the straddle carrier can make the crossover to another row of containers (see Figure 9.1).

An automated stacking crane operates over a fixed number of rows. The automated stacking crane can change rows at any location in a row. Sometimes, depending on the current position in the stack and the next location, an ASC can move simultaneously horizontally and vertically. However, in the model we assume that horizontal and vertical movements are made sequentially (see Figure 9.2).

For both systems it is required to schedule storage and retrieval requests in such a way that empty travel distances are minimised. The total route length consists of empty travel distances and fixed full travel distances. Therefore, to minimise the total route length the order of requests should be chosen in such a way that the empty travel distance between requests is minimal. We use a block sequencing approach to determine the order in which requests should be handled. This means that a number of requests is sequenced and handled. Thereafter, a new number of requests is selected.

The following assumptions are made:

1. N storage and retrieval requests need to be scheduled.

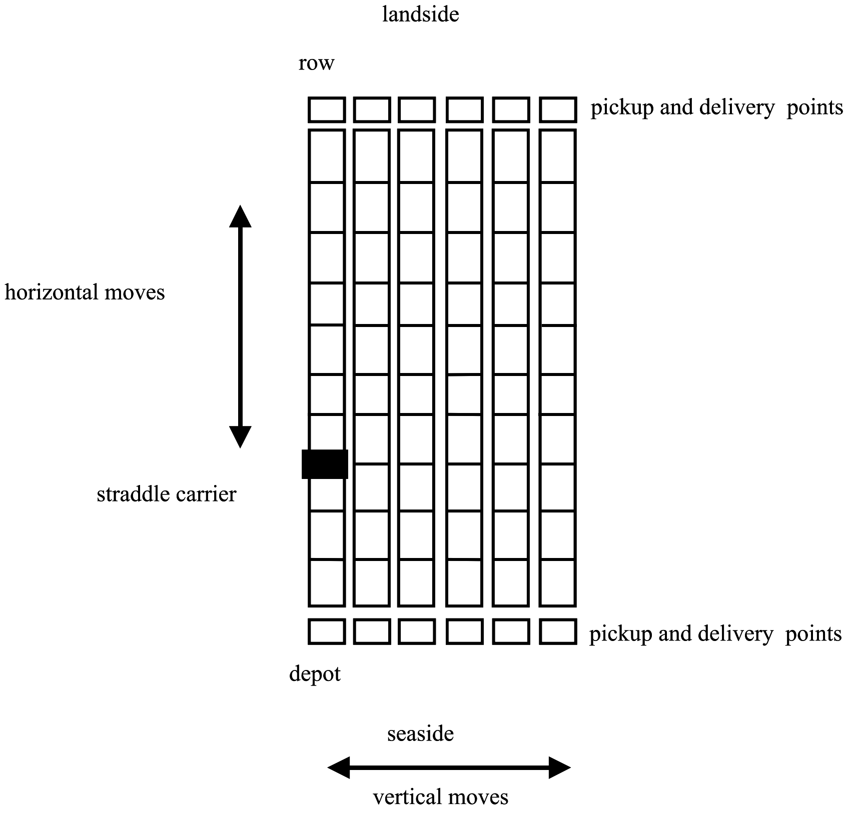

FIGURE 9.1. Layout of stack in which a straddle carrier operates.

2. Containers can be accessed directly. Therefore, to retrieve or store a container no reshuffles need to be made.

3. The time to pickup or release a container is independent of the order in which requests are handled. Therefore, only horizontal and vertical travel distances between requests are considered.

4. The travel times satisfy the triangle equality.

5. The storage position of each container is known.

6. All the storage positions for storage requests to be sequenced are empty initially.

7. The origin pickup and delivery point is known for each storage request.

8. The destination pickup and delivery point is known for each retrieval request.

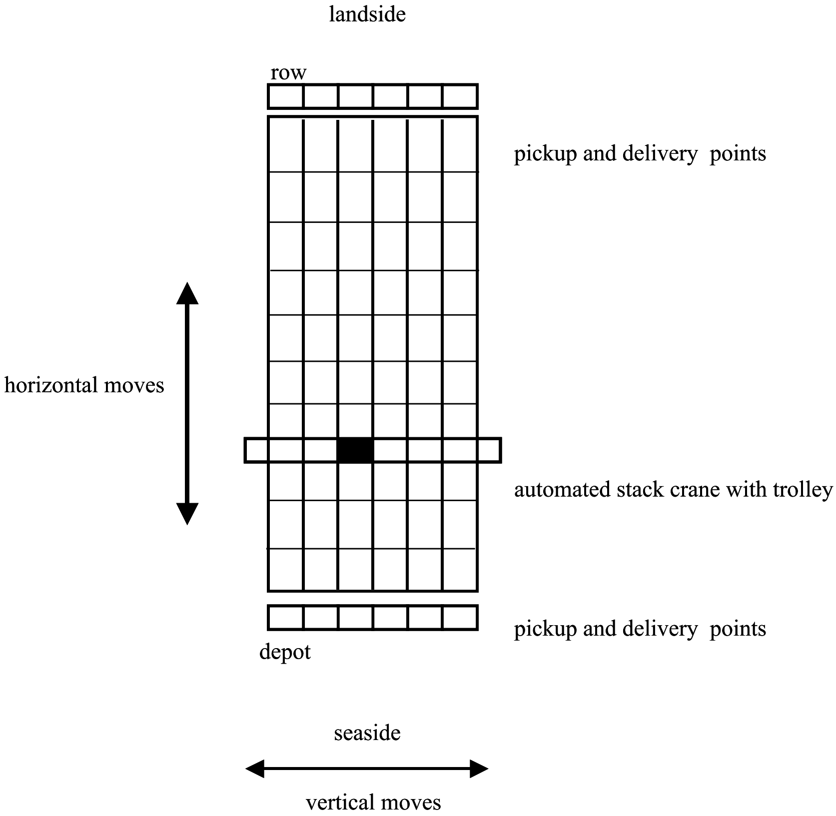

FIGURE 9.2. Layout of a stack in which an automated stacking crane operates.

9. All requests are available at the start of the route.
10. Requests can be scheduled in any order.
11. The route starts and ends at the depot.
12. The capacity of the material handling system is one container.
13. A straddle carrier needs to handle all requests in one row of containers before a new row can be entered.

To sequence storage and retrieval requests at straddle carriers and automated stacking cranes methods from Chapter 6 can be used. In the next section, we explain shortly in which way these methods are applied to these practical situations. In

Section 9.3 route lengths are determined in various situations for both systems by applying the methods from Chapter 6.

9.2 Usage of algorithms

In this section we explain how the dynamic programming algorithm from Chapter 6 can be used to sequence requests at a straddle carrier. Furthermore, the method from Section 6.3.3 is considered for sequencing requests at automated stacking cranes.

9.2.1 Straddle carriers

To construct a route for a straddle carrier starting and ending at the depot and handling all requests in all rows, the dynamic programming algorithm of Chapter 6 can be used. A straddle carrier travels over rows of containers to store/retrieve containers in/from these rows. This is similar to a storage and retrieval machine handling requests in multiple aisles. Storage requests are available at both ends of a row. Retrieval requests have their destination at one of the ends of the row (see Figure 9.1).

We construct a network by representing each location which need to be visited to perform a storage or retrieval request as a node. Furthermore, the pick up and delivery points of each row and the depot are also represented by nodes. Between two nodes multiple directed arcs exist. The objective of the dynamic programming algorithm is to construct a directed tour subgraph visiting all nodes at least once and all arcs at most once.

Equivalence classes are used to refer to a group of equivalent partial directed tour subgraphs (see also Section 6.3.1). Each partial directed tour subgraph can be completed to a directed tour subgraph. In an equivalence classes all partial directed tour subgraphs have the same characteristics. Therefore, we only consider a shortest subgraph in each equivalence class.

In the algorithm two types of transitions are made, namely in rows and between rows. For each equivalence class the least distance configuration is chosen. For each row the minimum length of all possible transitions needs to be calculated (see Section 6.3.2). In other words, for each row the order in which requests should be handled starting and ending at known locations such that the travel distance is minimal, is determined. The length of the transition between rows equals the sum of the length of the arcs added between two rows.

In the dynamic programming algorithm rows are considered sequentially starting with the leftmost row and ending with the rightmost row. In sequential steps of the algorithm directed arcs in a row and between rows are added for each equivalence class. The algorithm continues with the shortest subgraph in each equivalence class. This procedure continues until the last row has been considered and a minimal length directed tour has been found.

9.2.2 Automated stacking cranes

An automated stacking crane stores and retrieves containers in a block with a number of rows (see Figure 9.2). At the end and start of each row a pickup and delivery point is located. The trolley of the automated stacking crane can change easily from row to row at any location. The ASC operates in one area, in which horizontal and vertical movements can be made. As a result, we can consider the block of containers with pickup and delivery points as a single aisle in which requests need to be handled. From the destination of a container to another container horizontal or vertical distances might be travelled. If two containers are located in the same row only horizontal distances occur. Otherwise, also vertical distances need to be travelled to change rows. Horizontal and vertical movements are indicated in Figure 9.2.

The order in which requests need to be handled is determined such that empty travel distances are minimised. To solve this problem we apply the algorithm from Section 6.3.3. With this algorithm minimum length routes in an aisle with input/output stations at both ends and a known start and end point can be calculated. The depot is the start and end point for each route of an automated stacking crane.

A network is created in which each request and the depot are represented by a node. Directed arcs connect nodes if they can be handled after each other. The costs of each arc (i, j) equal the distance between the destination of i to the origin of j. Furthermore, directed arcs connect the depot with all requests and directed arcs connect all requests with the depot. This network can be reformulated as a complete bipartite network in which all nodes occur at both sides of the network. Firstly, an optimal assignment is constructed. If the assignment does not contain cycles an optimal route has been found. Otherwise, the cycles are patched together in an optimal way until a shortest tour has been found.

9.3 Experiments

We perform a simulation study to gain insight in differences in average empty travel distances in handling a fixed number of requests by automated stacking cranes and straddle carriers. In this study we consider travel distances. By incorporating speed of the types of equipments in the model travel times can be considered as a performance measure. In practice a straddle carrier can move faster through the stack than an automated stacking crane. In discussing the results of travel distances, we also refer to the speed and related travel times of the equipments.

We will discuss results from various simulation experiments. Two cases are studied, namely the scheduling of requests at straddle carriers and automated stacking cranes. For both cases routes are determined to fulfil all requests such that empty travel distances are minimised. Firstly, we describe the simulation study in more detail. Thereafter, both cases are studied separately. Finally, the results from both cases are compared.

9.3.1 Description of simulation study

The layout of the terminal in the case that straddle carriers are used, is presented in Figure 9.1. Figure 9.2 represents the layout of the terminal in the case that automated stacking cranes are used. Firstly, we consider the situation in which the material handling systems operate in a block of containers consisting of *six* rows of containers. At a large terminal in the Netherlands one row of containers has a length of 42 TEU (i.e. twenty equivalent unit). One TEU equals approximately 6 meters. We assume that all containers in a block have the same size. Therefore, 42 locations exist at which containers can be stored. As explained in Section 9.1 we only consider horizontal and vertical distances. The time to pick up or release a container is independent of the order in which containers are handled. Furthermore, all retrievals can be accessed directly. The distance unit is set to one gross container length (6 meters). Consequently, the length of a row equals 42 distance units. The pick-up and delivery points are located at both ends of these 42 locations. The distance between two containers i and j in the same row equals $|location\ i - location\ j|$. If automated stacking cranes are used the distance between containers in two rows is 0.4 distance units. Straddle carriers need more space to change rows. It is assumed that in this case the distance travelled by a straddle carrier to go from one row to another is 2.4 distance units. Except for the normal distance between two containers in different rows, the straddle needs space to switch rows. These crossover distance are estimated on the basis of container dimensions. These distances are illustrated in Figure 9.3.

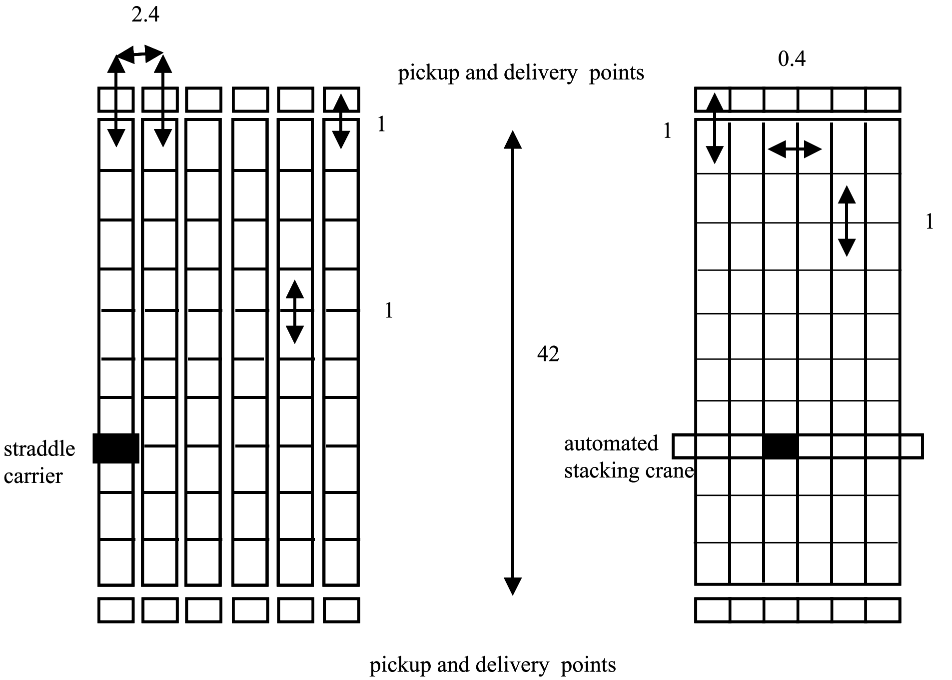

FIGURE 9.3. Illustration of distances in stack used in this simulation study.

Both types of equipments have unit load capacity. Therefore, a container need to be handled before another one can be stored or retrieved. At both ends of a row *pickup and delivery points* with infinite capacity are located. At these points containers are picked up by the system to be stored or containers are delivered to this point if they are retrieved from the stack. The travel distances to handle a request are fixed and known in advance. For each request origin and destination are known. The route of both systems starts and ends at the *depot* which is located at the front end of the leftmost row (see Figure 9.1 and Figure 9.2).

A straddle carrier handles on average 15 containers an hour. We assume that at most *10* containers are scheduled in advance. For each container we need to know the following characteristics:

1. row i ($1 \leq i \leq 6$) in/from which container need to be stored/retrieved,
2. location j ($1 \leq j \leq 42$) at which container is located or should be placed,
3. type of job (storage (1) or retrieval (2)),
4. origin (storage) or destination (retrieval) of container (I/O_1 (1) or I/O_2 (2)).

For each container these four numbers are randomly selected. Each number for each characteristic has an equal probability to be chosen. Firstly, we perform an experiment in which 10 containers need to be scheduled in six rows in each row 42 locations. For straddle carriers a route is determined by applying the method described in Section 9.2.1. Automated stacking cranes are routed according to the method from Section 9.2.2. The objective is to minimise *empty travel distances*. Therefore, the results in the next sections consist of empty travel distances.

To obtain a valid estimation of the average minimum empty travel distances, the number of replications required for each simulation experiment should be determined. The minimum number of replications is determined by solving Equation 8.1 with data from a limited number of replications. A replication size of 2500 is sufficient for all experiments such that a relative error γ smaller than 2% with a probability of 95% is obtained. As a result, the average route length for a certain instance is determined by executing 2500 times the same experiment with new input data.

Besides determining the average route length for the basic situation a *sensitivity analysis* is performed to both cases. Firstly, the effect of the number of requests to be handled on the average minimum empty travel distance is studied. Furthermore, the impact of changing the number of aisles and locations is observed. In the next sections, we discuss results for both cases.

9.3.2 Results

We first consider the case in which 10 containers need to be scheduled such that the route for the straddle carrier and automated stacking crane has a minimum empty travel distance. The 10 containers are stored or retrieved from six rows with 42 locations in each row. The following results for average minimum empty travel distance, average full travel distance and average minimum total travel distance are obtained, by executing 2500 replications:

	empty	st. dev.	full	st.dev	total	st.dev
SC	138.63	31.62	214.36	37.52	352.99	51.69
ASC	80.55	38.83	214.36	37.52	294.91	58.01

TABLE 9.1. Average minimum empty travel distance, average full distance, average minimum total distance and related standard deviation for a straddle carrier and automated stacking crane if 10 requests are scheduled.

The full travel distance is the same for automated stacking cranes and straddle carriers. Due to differences in empty travel distances the average total route length can vary for ASCs and SCs. From Table 9.1 it can be seen that the proportion of empty travel distance in the total route length is smaller for ASCs (27%) than for SCs (39%). The empty travel distance for a straddle carrier is larger than the empty travel distance for an automated stacking crane. This difference mainly occurs due to the fact that the length of crossovers between two adjacent aisles is six times larger for straddler carriers. Straddle carriers need to travel empty a distance of 2.4 distance units between two rows. Automated stacking cranes only need to travel empty 0.4 distance units between two rows. Another reason for this difference in route length might be the fact that a straddle carrier handles all requests in a row before entering another row. An ASC can combine requests in different rows and as a result it might be possible that combinations with less empty travel distances are obtained in this way.

If we also consider travel speed of both types of equipments, the performance of the straddle carrier is less worse than when we only observe travel distances. In a stack, a straddle carrier can travel 6 m/s versus an automated stacking crane 4 m/s. The speed of the trolly of an ASC for changing rows is only 0.8 m/s. A straddle carrier, therefore, moves much faster through the stack. If the main difference in routelenghts occurs due to the large crossovers, the performances of both types of equipments are comparable. In this comparison we do not take into account pick up and set down times.

To obtain insight in factors impacting empty travel distance, we performed an sensitivity analysis on the number of requests, the number of aisles and the number of locations in each aisle. The results will be presented in the following sections.

9.3.3 Varying the number of containers

Firstly, we vary the number of containers vary. In Figure 9.4 the results are presented for the average minimum empty travel distances if 5, 6, 7, 8, 9 and 10 containers are scheduled for a straddle carrier. Furthermore, the average minimum total travel distance, including the full travel distance, is presented. The solid line indicates the average minimum total route length and the dashed line indicates the average minimum empty travel distance.

It is clear that if more containers need to be handled the total route length of the straddle carrier increases. To handle a container, a fixed full route need to be travelled by the straddle carrier. The larger the amount of containers to be handled, the larger the distance the SC travels fully. The empty travel distance, which is part of the

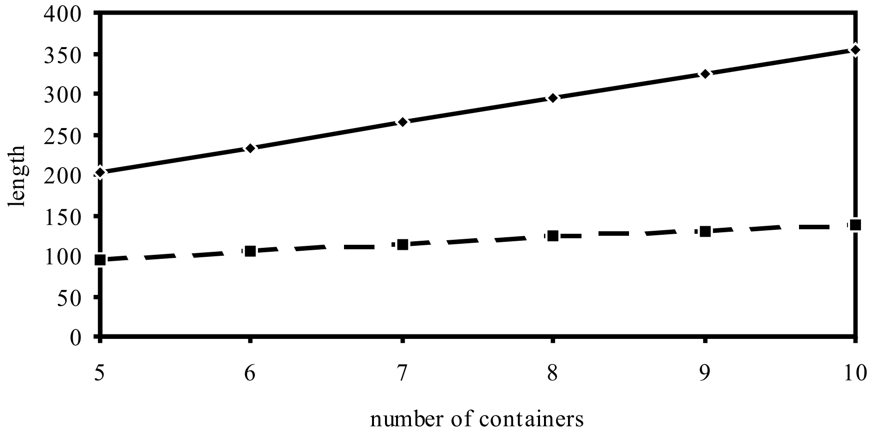

FIGURE 9.4. Average minimum empty travel distance (- - -) and average minimum total travel distance (_) if 5, 6, 7, 8, 9 or 10 requests need to be handled by a straddle carrier.

total route length will also increase. However, it can be noticed that the proportion of empty travel distance in the total route length decreases from 47% to 39%. In the case that a large number of containers are available in the stack for scheduling, the empty travel distances between containers will become relatively smaller. It might even be expected that in the case that the number of containers to be scheduled increases to infinity, the total empty travel distance between containers even will decrease to zero. In the interval studied, a linear relation exists between the number of requests to be handled and the average minimum empty travel distance.

Figure 9.5 represents the average minimum empty travel distance and average minimum total travel distance for an automated stacking crane if the number of containers increases from 5 to 10.

Also, in this case it can be concluded that the proportion of empty travel distance in the total route length decreases if the number of containers increases. In the case that 5 containers are handled, on average 36% of the total route length consists of empty travel distance. The empty travel distance is 27% of the total route length if 10 containers are scheduled. The average travel distance remains for all instances smaller than for a straddle carrier.

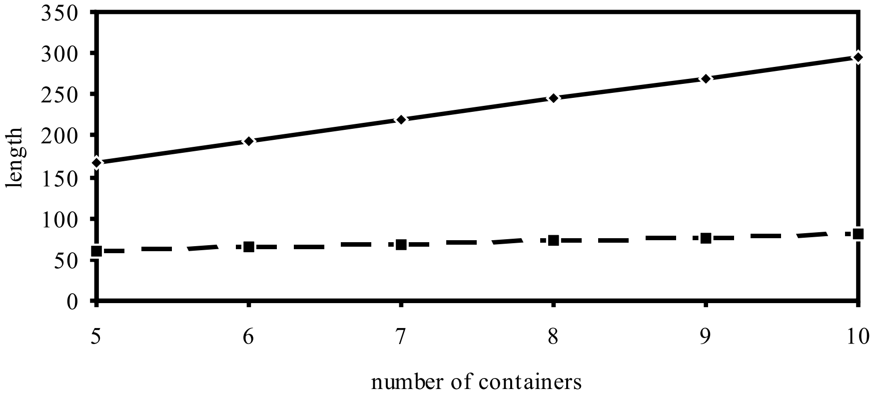

FIGURE 9.5. Average minimum empty travel distance (- - -) and average minimum total travel distance (_) if 5, 6, 7, 8, 9 or 10 requests need to be handled by an automated stacking crane.

9.3.4 Varying the number of rows

In this section, we study the impact of the number of rows on the average minimum empty travel distance. In Table 9.2 results are presented for the case in which 5 requests are divided over 3, 4, 5 and 6 rows. The length of each row remains the same.

	3	4	5	6
SC	78.72 (25.19)	84.84 (24.90)	90.60 (24.90)	95.65 (24.70)
ASC	57.16 (27.71)	58.00 (27.66)	58.80 (27.62)	59.21 (27.74)

TABLE 9.2. Average minimum empty travel distance and standard deviation (between brackets) of a straddle carrier and an automated stacking crane if 5 requests are divided over 3, 4, 5 or 6 rows.

It can concluded that the empty travel distance of a straddle carrier decreases with 17.7% if 5 containers are handled in three rows instead of six rows. The main reason for this reduction is that less crossovers between aisles need to be made by empty straddle carriers. The difference between the average for three rows and six rows consists mainly of six extra crossovers with length 2.4 that might be made. Due to the smaller length of crossovers (0.4), the impact of decreasing the number of rows is much smaller for automated stacking cranes. Only a decrease of 3.5% is found in the average empty travel distance if the number of rows decreases from 6 to 3.

9.3.5 Varying the number of locations in each row

In the basis situation each row consists of 42 locations. We studied the effect on average minimum empty travel distance if 5 containers need to be handled in 6 rows in the case that each row has 36, 38, 40, 42, 44, 46 or 48 locations. The average minimum empty travel distance for a straddle carrier (solid line) and an automated stacking crane (dashed line) are presented in Figure 9.6.

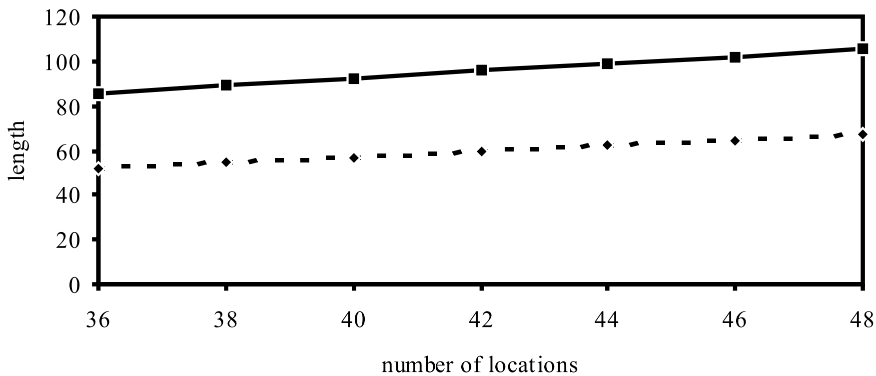

FIGURE 9.6. Average minimum empty travel distance for a straddle carrier (_) and an automated stacking crane (- -) if the number of locations in each aisle changes from 36 to 48.

From Figure 9.6 it can be concluded that the route length increases if the number of locations increase. An increase of 33% in the number of locations results in an increase of average minimum empty travel distance of 23% for a straddle carrier and of 30% of an automated stacking crane. The reason that the average minimum empty travel distance does not increase with the same proportion as the number of locations is that the distance between rows remains equal and does not increase with 33%. The increase in empty travel distance is smaller for a straddle carrier than for an automated stacking crane, because of the fact that the distance between rows is much larger for SCs than for ASCs.

9.4 Concluding Remarks

At a container terminal containers are transshipped from one modality to another. Before containers can be transshipped they are stored temporarily in a stack. Material handling systems take care of the storage and retrieval of containers in the stack. In this chapter we observed two different types of material handling systems, namely straddle carriers and automated stacking cranes.

Containers arrive and leave at pickup and delivery points. The straddle carrier or automated stacking crane stores and retrieves the containers in the stack. They operate in a block of containers which consists of a number of rows. The main difference between both types of equipments is that a straddle carrier changes rows at the end of a row. The automated stacking crane can change rows at any location in a row.

One of the control problems at a container terminal is the sequencing of requests such that empty travel distances of the equipment are minimised. A block sequencing approach is used in this chapter. An optimal order of handling storage and retrieval requests is determined such that the total route length of the straddle carrier or automated stacking crane handling all requests is minimal. In Section 9.2.1 it is explained in which way the polynomial time dynamic programming algorithm of Chapter 6 should be applied to solve the problem of the scheduling of requests at straddle carriers. The scheduling of requests for an automated stacking crane is similar to the problem of determining a route for an automated storage and retrieval system working in an aisle with two input/output stations. In Section 9.2.2 it is shown in which way the algorithm of Section 6.3.3 can be used for solving the sequencing problem of requests for an automated stacking crane.

To gain more insight in travel distances to handle a fixed number of requests of straddle carriers and automated stacking cranes in the stack we performed a simulation study. Furthermore, to study the impact of various decision variables on the route length a sensitivity analysis is performed. Various simulation experiments are executed and the results are described in Section 9.3. For both cases routes are calculated such that empty travel distances are minimised. Both cases are studied and their results are compared. The basis situation consists of a block consisting of 6 rows with each 42 locations and pickup and delivery points at both ends of each row. 10 storage and retrieval requests need to be sequenced. For each experiment 2500 replications are executed. The results consist of average minimum empty travel distance, average full travel distance and average minimum total travel distance for a straddle carrier and an automated stacking crane. The full travel distances are the same for both types of equipments. Due to the fact that a straddle carrier needs to travel a longer distance to make the crossover to another row, the average minimum empty travel distance for a straddle carrier is longer than the average minimum empty travel distance of an automated stacking crane. As a result, the total travel distance of a SC is larger than the total travel distance of an ASC. It might be expected that, due to differences in the speed of both types of equipments, the average travel times to handle all request are comparable.

Besides determining the average minimum empty travel distance for this basis situation a sensitivity analysis is performed. Firstly, the impact of the number of requests on the average minimum empty travel distance and related total travel time is studied. It can be concluded that the average minimum empty travel distance increases if the number of requests increases from 5 to 10. However, the proportion of empty travel distance in total travel time decreases. Clearly, the distance between requests decreases if the number of requests increases.

Secondly, the effect of decreasing the number of rows in a block is observed. It is shown that the average minimum empty travel distance decreases if the same number of requests is handled in less rows. The main reason therefore, is the fact that less crossovers need to be made by an empty machine. The effect is more significant for straddle carriers than for automated stacking cranes. This is due to the fact that the travel distance between two rows is much smaller for automated stacking cranes. Finally, the number of locations in each row is varied. The number of locations varies between 36 and 48. An increase of 33% in the number of locations results in a smaller increase of empty travel distances. This is due to the fact that the length of crossovers remains the same.

From the results for the straddle carrier it can be noticed that the average minimum empty travel distance decreases if the number of locations in a row decreases. In the case that less locations are used a straddle carrier is able to switch rows at an earlier location. Consequently, creating extra space to allow a straddle carrier to switch rows at more locations might result in a decrease in empty travel distances. The use of extra cross aisles between rows except for the aisle at the bottom and top end of a row might therefore be advisable. At some terminals, cross aisles are used in blocks in which straddle carriers perform storage and retrieval requests. An interesting subject for further research might be to identify how many cross aisles need to be used and at which locations these cross aisles need to be positioned. Using cross aisles for straddle carriers results, contrary to our simulation study, in different layouts for both types of equipments. A further research question is consequently to determine methods to optimise the layout of the stacking yard depending on the type of equipment used.

10
Conclusions and further research

Numerous technological innovations in communication, like the Internet, impact the process of ordering and delivering of products. To deal with these and other trends and to remain competitive, several adaptions in the supply chain and logistics branch might be required. Chapter 1 discusses in more detail trends, which impact the performance of the supply chain.

Logistics activities occur at several places in a supply chain, namely between and in various material handling centres of a supply chain, such as warehouses, transshipment terminals, cross-docking centres and manufacturing centres. In Chapter 1 some logistics activities in centres, like storing and internal transport of products are discussed. To perform the various logistics activities, material handling equipment are used.

Material handling systems consist of the equipment, personnel, information, materials and related planning and control systems. Planning and control concepts are used to control the systems in the centres. By developing new planning and control concepts the performance of material handling systems might increase. Such improvements in the performance can result in an increase in the performance of the material handling centre itself and as a result in the performance of the supply chain. In this thesis, new planning and control concepts for material handling systems are developed and tested in a practical application. The types of equipment studied are automated guided vehicles and storage and retrieval machines. The planning and control concepts developed in this thesis deal with the problem of the determination of the number of vehicles required in the system and the problem of the sequencing of storage and retrieval requests for a unit load storage and retrieval machine. Both concepts were tested with simulation on the material handling systems used at a container terminal.

10.1 Determination of the number of vehicles required

In material handling centres internal transport equipment is used to transport products from one location to another. In this thesis, we study automated guided vehicles (AGVs). AGVs are driverless internal transport systems, which transport products on fixed or free guidepaths. An AGV system has the following components: the vehicles, the transportation network, the interface between the production/storage system and the control system. Chapter 2 discusses AGVs and the related system in more detail.

In designing an AGV system a number of decision variables arise. Some of the issues to be addressed are the layout of the guidepaths, the number of vehicles required and scheduling of vehicles. Chapter 2 considers these and other decisions variables more elaborately. Furthermore, methods solving these kinds of decision problems are discussed by presenting a review of relevant literature.

One of the planning problems in an AGV system is the determination of the number of AGVs required in the system to obtain a sufficient performance level. From the literature discussed in Chapter 2 it is clear that several objectives can be used by determining vehicle requirements. Some of these objectives are minimising empty trips of vehicles, minimising travel times of vehicles and minimising delay times of products to be transported.

Stochastic, deterministic and simulation approaches can be used to solve this problem. In this thesis, we use a deterministic approach. By making deterministic assumptions on arrival patterns of jobs and travel times of vehicles, methods can be developed to determine the number of vehicles in an optimal way. In deterministic models release times of jobs are derived from stochastic distributions and are considered to be known. Furthermore, the destination and origin of each load are known. Related travel times are estimated based on the upper bound of the stochastic distribution. Non polynomial time methods like integer and mixed integer programming models are presented in Chapter 2. Furthermore, heuristics are discussed to determine feasible vehicle requirements. However, to determine the number of vehicles in an optimal way and in polynomial time a new method is developed in Chapter 3. Chapters 3 and 4 discuss new methods for determining vehicle requirements in an automated guided vehicle system.

In Chapter 3, we consider a material handling centre in which products are transported that need the help of an other type of equipment to be placed on automated guided vehicles. As a result, the transportation process is strongly connected to the other processes in the centre. The main objective of the transportation system is to prevent the occurrence of delays in the main processes in the centre. If a product is available for transport, it has to be transported immediately. By assuming that the availability instant of each job and related travel time are known, the problem can be formulated as a network problem. In the network each job is represented by a node. Two nodes i and j are connected by a directed arc (i, j) if the jobs are compatible. This is the case if the start instant of job j is greater than or equal to the moment job i is delivered at its destination plus the travel time from there to the origin of job j. By adding a source and a sink node directed paths from the source to the sink connecting several jobs can be found. The minimum number of directed paths, covering each node in exactly one path, corresponds to the minimum number of vehicles required to transport all jobs in time.

The algorithm developed to determine the minimum number of AGVs consists of several steps. Firstly, a transformation, based on a feasible flow in the network, is performed on the network. Secondly, we determine a maximum flow in this transformed graph. The value of the feasible flow in the original network minus the value of the maximum flow in the transformed graph equals the value of the minimum flow in the original graph. The value of this minimum flow corresponds to the minimum number of AGVs required. It is shown that the problem of determining the minimum number of vehicles required to transport n jobs in time is solved in $O(n^{5/2})$ time. Thus, a new planning concept is developed to determine in strongly polynomial time the minimum number of vehicles required to transport all jobs in time.

An extension of this problem is the problem in which each job needs to be transported within its time-window. Instead of immediate transport of jobs at their release times, some delays are allowed. In Chapter 4 we consider a material handling centre in which the transportation process is decoupled of the other processes by buffer areas. Loads are placed into such buffer areas to wait for transport. Each buffer area has a fixed capacity. The release time a job is placed in a buffer is assumed to be known for each job. If a buffer is a full and a new job needs to be placed into the buffer, the first job placed in the buffer needs to be transported. Consequently, each job has a deadline (i.e. due time) at which it should be transported. As a result, each job has a time-window [*release time*, *due time*] in which the transport should start.

Except for the question which jobs are transported by the same vehicle, a second question should be answered. Namely, at which start instant within the time-window the transport of a job is started. To solve these questions, we first discretise each time-window such that for each job a finite number of start instants is obtained. By representing each start instant of each job as a node, a network is formulated. A directed arc is added if a job, with a certain start instant, is compatible with another job at a certain start instant. The minimum flow algorithm of Chapter 3 cannot be applied to this network. Except for the condition that each job is included at most once in the directed paths an extra condition is required. Namely, for each job only one start instant can be chosen.

We formulate this problem as an integer linear programming model. The objective is to minimise the number of directed paths such that each job with at most one start instant is covered only once by the set of directed paths.

Furthermore, we use a set partitioning approach to solve the problem. Three steps are formulated. Firstly, with a new search method all possible directed paths from source to sink are determined in polynomial time. Secondly, an incidence matrix a_{ij} is constructed in which each row i represents one of the jobs and each column j one of the directed paths. If $a_{ij} = 1$ then job i is included with one of its start instants in the directed path j. Thirdly, similar columns can be deleted from the matrix until one column of this kind remains. This can be done, because of the fact that we are only interested in the number of vehicles and not especially in the related route. Using the incidence matrix created, a minimum number of directed paths covering all jobs only once can be determined.

10.2 Sequencing storage and retrieval requests

Storage and retrieval equipment is used to store and retrieve products in and from the storage area in material handling centres. Products can be stored on pallets, in cases or just as individual items. To complete an order of a customer, a pallet or just some products from a pallet are retrieved from storage. Storage and retrieval machine retrieve pallets. Individual items are collected by an orderpicker. An unit load (automated) storage and retrieval machine can be used to store and retrieve unit loads, for example pallets, in and from the storage area. In one trip of the machine usually a storage request is combined with a retrieval request. Compared to performing one request in each trip a higher throughput will be obtained.

Orderpickers travel/walk through the storage area to retrieve items from storage. In a route multiple items can be picked. Commonly, retrieval and storage requests are not combined in one route. In both cases a similar control problem arises. Namely, the sequencing of retrieval (and storage) requests such that the total distance traveled by the machine or orderpicker is minimised.

In Chapter 5 we present a literature review on methods for sequencing requests. Firstly, literature on the sequencing problem of storage and retrieval requests for an automated storage and retrieval system (AS/RS) is studied. The majority of the literature considers a unit load automated storage and retrieval machine working in one aisle with one input/output station. At such a station loads are transferred from the AS/RS to another type of equipment and vice versa. The general case of sequencing requests is *NP-hard* to solve. Therefore, heuristics have been developed to find feasible schedules. However, in some special cases, for example, if a fixed number of requests are scheduled with the use of dedicated storage, optimal methods exist.

Secondly, literature on scheduling of item retrievals is discussed. Routes are determined for orderpickers to collect items in multiple aisles. Optimal methods exist for storage areas consisting of one or two blocks. Heuristic methods are also treated.

In Chapter 5 it becomes clear that for both subjects some research questions still remain unanswered. Hardly any attention has been paid to the question of scheduling storage and retrieval requests in case each aisle has an input/output station at both ends of the aisle. This is the case in material handling centres in which flows are separated. An example of such a centre is a container terminal, in which containers arrive and leave at both sides of the stack, namely the seaside and the landside. Besides this, scheduling rules for AS/RSs working in multiple aisles are scarce. Both questions are addressed in Chapter 6 of this thesis. For item retrieval optimal methods exist. It might be interesting to see if these methods can be extended for executing both storage and retrieval requests in multiple aisles. These research questions are combined to one new research question which is solved in Chapter 6. A new control concept is developed to schedule storage and retrieval requests for a unit load storage and retrieval machine working in multiple aisles.

In Chapter 6, we consider aisles with an input/output station at both ends. Two new algorithms are developed. Firstly, a polynomial time algorithm is developed to schedule to optimality storage and retrieval requests in one aisle with two input/output stations. The objective is to minimise empty travel times between requests such that all requests are handled by a machine starting and ending at specified end(s) of the aisle. This problem is formulated as an assignment problem. A solution of this assignment problem without cycles indicates an optimal sequence of requests. An optimal assignment may contain several disconnected cycles, which need to be patched together. It is explained in which way disjoint cycles can be patched in an optimal way. As a result, we can obtain in an optimal way the length of each possible route starting and ending at one of the end(s) of the aisle handling all requests in an aisle. It is assumed that the machine only can change aisles if all requests are handled in the aisle. As a result, the lenghts, obtained from applying this method, can be used in the second control concept developed in Chapter 6.

A dynamic programming algorithm is developed to schedule storage and retrieval requests in multiple aisles for a unit load machine starting and ending its route at the depot. The problem is represented by a directed network. Each node corresponds to a location at which a storage or retrieval request need to be performed. Furthermore, the depot and the input/output stations at both ends of each aisle are also represented by a node. The objective is to find a directed tour in the directed network in which it is indicated in which order aisles need to be handled, at which end(s) of an aisle the machine should start and end and the sequence in which the requests in an aisle should be handled. This is done in such a way that all m requests are handled and the total empty travel distance is minimised. The dynamic programming algorithm presented in Chapter 6 sequentially considers all aisles starting at aisle one and ending at aisle n. It is shown that the problem can be solved in $O(n^3 + n^2m^3)$ time. Thus, a new control concept is developed to determine, in polynomial time, an optimal sequence of storage and retrieval requests for a unit load machine working in multiple aisles.

10.3 Container terminals

Container terminals can be considered as nodes in a supply chain network. Raw materials, semi-finished products or finished products can be transported in a container from suppliers to customers. Over land this transport is executed by trains or trucks. Ships transport containers over water. At a container terminal containers are transshipped from one mode of transportation to another.

If a ship arrives containers are taken off the ship by cranes. Thereafter, containers are transported by vehicles to new modes of transportation or to the storage area where they are stored temporarily. After a certain period of time the containers are retrieved from storage and transported to the next destination by trains, trucks or barges. This process is executed in reverse order for containers arriving over land. Chapter 7 discusses all process at a container terminal in more detail. At a container terminal various decision problems arise. A classification of these decision problems and related literature is also presented in Chapter 7.

Clearly, at a container terminal an internal transport process and a storage and retrieval process occur. One of the planning questions for the transport system is the determination of the number of vehicles required. For the storage and retrieval process one of the control problems concerns the sequencing of storage and retrieval requests. The planning and control concepts from this thesis can be applied to solve the mentioned problems in a container terminal. The choice for a certain model depends, among other things, on the type of equipment used for the transport or storage and retrieval process.

In Chapter 8 we consider a container terminal in which AGVs or self lifting vehicles can be used. In case non lifting vehicles are used, the transport process is strongly connected to processes in the terminal, in which delays must be avoided. As a result, if containers are available for transport, they need to be transported immediately. It is explained in which way the minimum flow algorithm of Chapter 3 can be used to determine the minimum number of non lifting vehicles to transport all containers in time.

Secondly, we consider the case in which vehicles are used which are capable of lifting a container by itself. Consequently, the storage and transport process are decoupled and buffer areas are used. Each container has a time-window which is defined by a release time and a due time which depends on the capacity of the buffer. The integer linear programming model of Chapter 4 can be used to solve the problem of the determination of the number of lifting vehicles required to transport all containers within their time-window.

To obtain a feeling about the difference in the number of non lifting vehicles and the number of lifting vehicles required, a simulation study is performed. In a terminal in which each crane has a buffer area with capacity two a reduction of approximately 30% in the number of vehicles can be obtained if lifting vehicles are used. Furthermore, the effects of cycle times (i.e. time to unload a container of the ship) on the number of vehicles are considered. It can be concluded that faster and smaller cranes impact the number of vehicles required. Compared to the original cycle times cranes can unload, for example, at most 5 seconds faster to keep the same number of non lifting vehicles and at most two seconds faster to keep the same number of lifting vehicles.

In Chapter 9, we consider the storage process in a container terminal. Material handling equipment takes care of the storage and retrieval of containers in the storage area (i.e. stack). Two different types of equipment are considered, namely automated stack cranes and straddle carriers. Both types of equipment operate in a stack consisting of a number of parallel rows of containers. At both ends of each row a pick up and delivery point is located at which containers are transferred from the transport system to the storage system and vice versa. The main difference between both types of equipment is the fact that automated stack cranes can change rows at any location in a row and the fact that a straddle carrier only can change rows at the ends of a row. One of the control problems at the stack is the sequencing of requests such that empty travel distances are minimised.

The sequencing problem for a straddle carrier is comparable to the problem of sequencing requests for a unit load machine working in multiple aisles. Chapter 9 indicates in which way the dynamic programming algorithm of Chapter 6 can be applied to solve this specific case.

The rows, in which an automated stack crane operates, can be considered as one aisle, because of the fact that an automated stack crane can make a crossover between two rows at any location. As a result, the sequencing problem of an automated stack crane can be compared with the problem of sequencing requests in one aisle with input/output stations at both ends of the aisle. The polynomial time algorithm of Chapter 6 can be used to solve this specific problem.

A simulation study is executed to gain insight in the way empty travel distances of a straddle carrier differ from empty travel distances of an automated stack crane. It is concluded that the average minimum empty travel distance is longer for straddle carriers than for automated stack cranes. The main reason for this is the longer distance a straddle carrier needs to travel between rows to make the crossover.

Furthermore, a sensitivity analysis is performed to study the impact of the number of requests on the average minimum empty travel distance and related total travel time. It is shown that the average minimum empty travel distance increases if the number of requests increases from 5 to 10. The proportion of empty travel distance in total travel time decreases, due to the fact that the distance travelled empty between requests decreases if the number of requests increases. Secondly, the impact of the number of rows in a block on average minimum empty travel times is studied. The average minimum empty travel distance decreases if the same number of requests is handled in less rows and consequently less crossovers need to be made empty. It is concluded that the effect is more significant for straddle carriers than for automated stacking cranes. The main reason for this is the fact that the empty travel distance between two rows is much smaller for automated stacking cranes. Finally, the impact of the number of locations in each row on the average minimum empty travel distance is examined. By increasing the number of locations from 36 to 48 (33%) a smaller increase of empty travel distances is found. This is explained by the fact that the length of crossovers remains the same.

10.4 Further research

This thesis provides some new planning and control concepts for material handling systems. These methods can be used to solve specific problems. Variations in the problems may result in adjustments in the methods. In practice stochastic influences might be noticed. Due to, for example, congestion, breaking down of vehicles and other types of delays travel times of vehicles between two locations might vary. It might be interesting to study in which way the minimum flow algorithm of Chapter 3 to determine vehicle requirements can be adjusted if stochastic travel times are incorporated in the system. Each different type of equipment has it own characteristics. Some vehicles are capable of transporting more than one load at a time. Questions that arise are: can the method in this case still be used? Which new decisions need to be made and how should these decisions be incorporated in the model?

Instead of adjusting the method it might be sometimes necessary to develop a completely new model to deal with the problem. Incorporating randomness in availability instants will lead to the development of a new concept. It might be interesting to develop methods which solve the stochastic vehicle requirements problem. Furthermore, restrictions from practice can be incorporated in the model. Examples of these restrictions are limits on the maximum number of vehicles allowed in a certain zone, the interference between arriving and leaving vehicles at pick-up and delivery points and the occurrence of deadlocks and congestion.

The control concept for sequencing a fixed number of storage and retrieval requests can be applied for a unit load machine working in multiple aisles. The destination input/output station of each retrieval is known in advance. The order in which requests are handled can be chosen freely. Except for the objective of minimising travel times other objectives can be used in practice. To complete orders of customers in time, final times of retrieval might be defined for retrieval requests. In which way should we deal in the method of Chapter 6 with final times for retrievals and simultaneously with minimising travel times of storage and retrieval machines? Is the method still useful if the scheduling process need to be executed dynamically? Which changes need to be incorporated or should a new control concept be developed? What will be the consequences if the choice of the destination input/output station of retrievals is not fixed but free to choose?

Instead of having machines with unit load capacity, machines can be used with a larger capacity. This will result in making extra decisions. Is the concept of Chapter 6 applicable or should new concepts be developed? In multiple aisles more than one machine can be used. Questions arise, like which machine handles which requests and in which order and in which way are congestion and collisions between machines avoided?

The above-mentioned unsolved questions also exist at container terminals. Furthermore, from the literature review in Chapter 7 it becomes clear that some specific decision problems at container terminals are still open for research. Attention can be paid to the priority planning problem at the stack. Usually, retrieval requests with destination ship have priority over retrieval requests designated for trucks and trains. Can this priority aspect be incorporated in the sequencing concept of this thesis or should a new concept be developed? From the simulation study in Chapter 9 it can be concluded that the layout of a stacking yard influences the travel distances of the various types of equipment. It might be expected that travel distances decrease in case layouts are optimised for a certain type of equipment. At some terminals cross aisles are used in blocks to allow a straddle carrier to change rows at several places. An interesting research question is to develop methods to determine the number of cross aisles needed and their locations in the rows. Furthermore, methods can be developed for optimising the layout of the stacking yard taking into account the type of equipment used.

Also, the majority of the research in material handling centres only addresses single types of material handling equipment. In our opinion more attention could be given to the combination of various types of equipment. For example, the simultaneous scheduling of jobs at storage and retrieval machines and automated guided vehicles. Joint optimisation of several material handling systems is certainly a topic for future research. In this context, one can think of combining both problems studied in this thesis. The problem then becomes to sequence storage and retrieval requests in such a way that the number of transport vehicles is minimised.

References

[1] Agerschou, H., Lundgren, H., Sørensen, T., Ernst, T., Korsgaard, J., Schmidt, L.R., Chi, W.K. (1983), *Planning and Design of Ports and Marine Terminals*, John Wiley & Sons, Chichester.

[2] Ahuja, R.K., Magnanti, T.L., Orlin, J.B. (1993), *Network Flows, Theory, Algorithms, and Applications*, Prentice Hall, New Jersey.

[3] Andel, T. (1994), Define cross-docking before you do it, *Transportation & Distribution* 35(11), 93-98.

[4] Anonymous (1999), More AGVs for ECT, *Cargo Systems,* June, 5.

[5] Ascheuer, N., Grötschel, M., Abdel-Hamid, A.A.A. (1999), Order picking in an automatic warehouse: solving online asymmetric TSPs, *Mathematical Methods of Operations Research* 49, 501-515.

[6] Asef-Vaziri, A., Dessouky, M., Sriskandarajah, C. (2001), A loop material flow system design for automated guided vehicles, *International Journal of Flexible Manufacturing Systems* 13(1), 33-48.

[7] Asef-Vaziri, A., Laporte, G., Sriskandarajah, C. (2000), The block layout shortest loop design problem, *IIE Transactions* 32, 727-734.

[8] Askin, R.G., Standridge, C.R. (1993), *Modeling and Analysis of Manufacturing Systems*, John Wiley & Sons, New York.

[9] Avriel, M., Penn, M., Shpirer, N., Witteboon, S. (1998), Stowage planning for container ships to reduce the number of shifts, *Annals of Operations Research* 76, 55-71.

[10] Avriel, M., Penn, M., Shpirer, N. (2000), Container ship stowage problem: complexity and connection to the coloring of circle graphs, *Discrete Applied Mathematics* 103, 271-279.

[11] Baita, F., Pesenti, R., Ukovich, W., Favaretto, D. (2000), A comparison of different solution approaches to the vehicle scheduling problem in a practical case, *Computers & Operations Research* 27, 1249-1269.

[12] Baker, C. (1998), High time for straddles, *Cargo Systems,* October, 23-26.

[13] Ballis, A., Abacoumkin, C. (1996), A container terminal simulation model with animation capabilities, *Journal of Advanced Transportation* 30(1), 37-57.

[14] Barad, M., Sinriech, D. (1998), A petri net model for the operational design and analysis of segmented flow topology (SFT) AGV systems, *International Journal of Production Research* 36(5), 1401-1425.

[15] Bartholdi, J.J., Gue, K.R. (2000), Reducing labor costs in an LTL crossdocking terminal, *Operations Research* 48(6), 823-832.

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

[17] Basu, S., Dutta, S.P., North, W.P.T. (1999), A process modelling framework for AGV control, *International Journal of Production Research* 37(16), 3671-3696.

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

[19] Bing, W.X. (1998), The application of analytic process of resource in an AGV scheduling, *Computers & Industrial Engineering* 35(1-2), 169-172.

[20] Bish, E.K., Leong, T.Y., Li, C.L., Ng, J.W.C., Simchi-Levi, D. (2001), Analysis of a new vehicle scheduling and location problem, *Naval Research Logistics* 48, 363-385.

[21] Blair, E.L., Charnsethikul, P., Vasques, A. (1987), Optimal routing of driverless vehicles in a flexible material handling system, *Material Flow* 4, 73-83.

[22] Bodin, L.D., Golden, B.L., Assad, A.A., Ball, M.O. (1983), Routing and scheduling of vehicles and crews: the state of the art, *Computers & Operations Research* 10(2), 63-211.

[23] Bookbinder, J.H., Krik, M.D. (1997), Lane selection in an AGV-based asynchronous parallel assembly line, *Computers & Industrial Engineering* 32(4), 927-938.

[24] Borsje, N., Keijer, R.J., Tibboel, M., Vis, I. en Roodbergen, K.J. (2001), Minder blikken op voorraad in distributieketen van Akzo Nobel en Impress, *Tijdschrift voor Inkoop en Logistiek* 17(7/8), 38-40.

[25] Bortfeldt, A., Gehring, H. (2001), A hybrid genetic algorithm for the container loading problem, *European Journal of Operational Research* 131, 143-161.

[26] Bostel, N., Dejax, P. (1998), Models and algorithms for container allocation problems on trains in a rapid transshipment shunting yard, *Transportation Science* 32(4), 370-379.

[27] Bozer, Y.A., Cho, M.S., Srinivasan, M.M. (1994), Expected waiting times in single-device trip-based material handling systems, *European Journal of Operational Research* 75, 200-216.

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

[29] Bozer, Y.A., 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.

[30] Bozer, Y.A., White, J.A. (1984), Travel-time models for automated storage/retrieval systems, *IIE Transactions* 16(4), 329-338.

[31] Bozer, Y.A., White, J.A. (1990), Design and performance models for end-of-aisle order picking systems, *Management Science* 36(7), 852-866.

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

[33] Bramel, J., Simchi-Levi, D. (1997), On the effectiveness of set covering formulations for the vehicle routing problem with time windows, *Operations Research* 45(2), 295-301.

[34] Brockmann, T. (1999), 21 warehousing trends in the 21st century, *IIE Solutions* 31(7), 36-40.

[35] Cao, B., Uebe, G. (1993), An algorithm for solving capacitated multicommodity p-median transportation problems, *Journal of the Operational Research Society* 44(3), 259-269.

[36] Carter, P.L., Carter, J.R., Monczka, R.M., Slaight, T.H., Swan, A.J. (2000), The future of purchasing and supply: A ten-year forecast, *The Journal of Supply Chain Management* 36(1), 14-26.

[37] Celen, H.P., Slegtenhorst, R.J.W., Van der Ham, R.Th., Nagel, A., De Vos Burchart, R., Van den Berg, J., Evers, J.J.M., Lindeijer, D.G., Dekker, R., Meersmans, P.J.M., De Koster, M.B.M., Van der Meer, R., Carlebur, A.F.C., Nooijen, F.J.A.M. (1997), FAMAS NewCon, concept *TRAIL Report*.

[38] Chang, S.H., 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.

[39] Chen, C.S., Lee, M.S., Shen, Q.S. (1995), An analytical model for the container loading problem, *European Journal of Operational Research* 80, 68-76.

[40] Chen, C.Y., Chao, S.L., Hsieh, T.W. (2000), A time-space network model for the space resource allocation problem in container marine transportation, *paper presented at the 17th international symposium on mathematical programming 2000, Atlanta, USA*.

[41] Chen, M. (1996), A mathematical programming model for AGVs planning and control in manufacturing systems, *Computers & Industrial Engineering* 30(4), 647-658.

[42] Chen, T. (1999), Yard operations in the container terminal - a study in the 'unproductive moves', *Maritime Policy & Management* 26(1), 27-38.

[43] Chen, Y., Leong, Y.T., Ng, J.W.C., Demir, E.K., Nelson, B.L., Simchi-Levi, D. (1998), Dispatching automated guided vehicles in a mega container terminal, *paper presented at INFORMS Montreal 1998, Canada*.

[44] Cheung, R.K., Chen, C.Y. (1998), A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem, *Transportation Science* 32(2), 142-162.

[45] Chung, Y.G., Randhawa, S.U., McDowell, E.D. (1988), A simulation analysis for a transtainer-based container handling facility, *Computers & Industrial Engineering* 14(2), 113-125.

[46] Co, C.G., Tanchoco, J.M.A. (1991), A review of research on AGVS vehicle management, *Engineering Costs and Production Economics* 21, 35-42.

[47] Crainic, T.G., Gendreau, M., Dejax, P. (1993), Dynamic and stochastic models for the allocation of empty containers, *Operations Research* 41, 102-126.

[48] Daganzo, C.F. (1989), The crane scheduling problem, *Transportation Research B* 23(3), 159-175.

[49] Dantzig, G.B., Fulkerson, D.R. (1954), Minimizing the number of tankers to meet a fixed schedule, *Naval Research Logistics Quaterly* 1, 217-222.

[50] Davies, A.P., Bischoff, E.E. (1999), Weight distribution considerations in container loading, *European Journal of Operational Research* 114, 509-527.

[51] De Castilho, B., Daganzo, C.F. (1993), Handling strategies for import containers at marine terminals, *Transportation Research B* 27(2), 151-166.

[52] De Guzman, M.C., Prabhu, N., Tanchoco, J.M.A. (1997), Complexity of the AGV shortest path and single-loop guide path layout problems, *International Journal of Production Research* 35(8), 2083-2092.

[53] De Jong, R., Jonker, L., Van der Voet, J., Vis, I. en Roodbergen, K.J. (2001), E-commerce in de designmeubelbranche?, *Tijdschrift voor Inkoop en Logistiek* 17(9), 54-56.

[54] De Koster, R., Van der Poort, E. (1998), Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions, *IIE Transactions* 30, 469-480.

[55] Dell'Amico, M., Fischetti, M., Toth, P. (1993), Heuristic algorithms for the multiple depot vehicle scheduling problem, *Management Science* 39(1), 115-125.

[56] Desrochers, M., Desrosiers, J., Solomon, M. (1992), A new optimization algorithm for the vehicle routing problem with time windows, *Operations Research* 40(2), 342-354.

[57] Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., Soumis, F. (1988), Vehicle routing with time windows: optimization and approximation, in: *Vehicle Routing: Methods and Studies, Studies in Management Science and Systems,* Golden, B.L., Assad, A.A. (eds.), 65-84.

[58] Dhouib, K., Kadi, D.A. (1994), Expert system for AGV managing in bidirectional networks: KADS methodology based approach, *International Journal of Production Economics* 33, 31-43.

[59] Dornier, P.P., Ernst, R., Fender, M., Kouvelis, P. (1998), *Global Operations and Logistics, Text and Cases*, John Wiley & Sons, New York.

[60] Duinkerken, M.B., Ottjes, J.A., Evers, J.J.M., Kurstjens, S.T.G.L., Dekker, R., Dellaert, N.P. (1996), Simulation studies on inter terminal transport at the Maasvlakte, *TRAIL Proceedings 2nd TRAIL Congress 1996.*

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

[62] Ebben, M. (2001), Logistic control in automated transportation networks, *Beta Ph.D. thesis series D-40*, Twente University Press, Enschede.

[63] Eben-Chaime, M. (1992), Operations sequencing in automated warehousing systems, *International Journal of Production Research* 30(10), 2401-2409.

[64] Edmond, E.D., Maggs, R.P. (1978), How useful are queue models in port investment decisions for container berths?, *Journal of the Operational Research Society* 29(8), 741-750.

[65] 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.

[66] Egbelu, P.J. (1991), Framework for dynamic positioning of storage/retrieval machines in an automated storage/retrieval system, *International Journal of Production Research* 29(1), 17-37.

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

[68] Egbelu, P.J. (1993), Concurrent specification of unit load sizes and automated guided vehicle fleet size in manufacturing system, *International Journal of Production Economics* 29, 49-64.

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

[70] Egbelu, P.J., 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.

[71] Elsayed, E.A., Lee, M.K. (1996), Order processing in automated storage/retrieval systems with due dates, *IIE Transactions* 28, 567-577.

[72] Elsayed, E.A., Lee, M.K., Kim, S., Scherer, E. (1993), Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals, *International Journal of Production Research* 31(3), 727-738.

[73] Evers, J.J.M., Koppers, S.A.J. (1996), Automated guided vehicle traffic control at a container terminal, *Transportation Research A* 30(1), 21-34.

[74] Eynan, A., Rosenblatt, M.J. (1993), An interleaving policy in automated storage/retrieval systems, *International Journal of Production Research* 31(1), 1-18.

[75] Faraji, M., Batta, R. (1994), Forming cells to eliminate vehicle interference and system locking in an AGVS, *International Journal of Production Research* 32(9), 2219-2241.

[76] Fisher, E.L., Farber, J.B., Kay, M.G. (1988), MATHES: an expert system for material handling equipment selection, *Engineering Costs and Production Economics* 14, 297-310.

[77] Fisher, M. (1995), Vehicle routing, in: *Network Routing*, Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (eds.), Elsevier, Amsterdam.

[78] Fisher, M.L., Jörnsten, K.O., Madsen, O.B.G. (1997), Vehicle routing with time windows: two optimization algorithms, *Operations Research* 45(3), 488-492.

[79] Fitzgerald, K.R. (1985), How to estimate the number of AGVs you need, *Modern Materials Handling,* October, 79.

[80] Fleischmann, M. (2001), Quantitative models for reverse logistics, *Lecture Notes in Economics and Mathematical Systems*, Volume 501, Springer Verlag, Berlin.

[81] Ford, L.R., Fulkerson, D.R. (1962), *Flows in Networks*, Princeton University Press, Princeton, New Jersey.

[82] Fraser, J., Fraser, N., McDonald, F. (2000), The strategic challenge of electronic commerce, *Supply Chain Management: an International Journal* 5(1), 7-14.

[83] Frazelle, E.H. (1996), *World-class Warehousing*, Logistics resources international, Inc., Atlanta, GA.

[84] Gademann, A.J.R.M., Van de Velde, S.L. (2000), Positioning automated guided vehicles in a loop layout, *European Journal of Operational Research* 127, 565-573.

[85] Gambardella, L.M., Rizzoli, A.E., Zaffalon, M. (1998), Simulation and planning of an intermodal container terminal, *Simulation* 71(2), 107-116.

[86] Ganesharajah, T., Hall, N.G., Sriskandarajah, C. (1998), Design and operational issues in AGV-served manufacturing systems, *Annals of Operations Research* 76, 109-154.

[87] Ganesharajah, T., Sriskandarajah, C. (1995), Survey of scheduling research in AGV-served manufacturing systems, *Advances in Instrumentation and Control, Proceedings of the International Conference and Exhibit* 50(1), 87-94.

[88] Gans, N., Van Ryzin, G. (1999), Dynamic vehicle dispatching: optimal heavy traffic performance and practical insights, *Operations Research* 47(5), 675-692.

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

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

[91] Gendreau, M., Guertin, F., Potvin, J.Y., Taillard, É. (1999), Parallel tabu search for real-time vehicle routing and dispatching, *Transportation Science* 33(4), 381-390.

[92] Glover, F., Klingman, D., Phillips, N.V. (1992), *Network Models in Optimization and their Applications in Practice*, John Wiley & Sons, New York.

[93] Gobal, S.L., Kasilingam, R.G. (1991), A simulation model for estimating vehicle requirements in automated guided vehicle systems, *Computers & Industrial Engineering* 21 (1-4), 623-627.

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

[95] Graves, S.C., Hausman, W.H., Schwarz, L.B. (1977), Storage-retrieval interleaving in automatic warehousing systems, *Management Science* 23(9), 935-945.

[96] Gue, K.R. (1999), The effects of trailer scheduling on the layout of freight terminals, *Transportation Science* 33(4), 419-428.

[97] Hackman, S.T., Rosenblatt, M.J. (1990), Allocating items to an automated storage and retrieval system, *IIE Transactions* 22(1), 7-14.

[98] Hagdorn-Van der Meijden, L., Van Nunen, J.A.E.E. (2000), Informatie- en communicatietechnologie en de rol van het distributiecentrum in vraaggestuurde netwerken, *Praktijkboek Magazijnen/Distributiecentra*, Kluwer, Deventer, 1.4-01-1.4-14.

[99] Hall, R.W. (1993), Distance approximations for routing manual pickers in a warehouse, *IIE Transactions* 25(4), 76-87.

[100] Han, M.H., McGinnis, L.F. (1989), Control of material handling transporter in automated manufacturing, *IIE Transactions* 21(2), 184-190.

[101] Han, M.H., McGinnis, L.F., Shieh, J.S., White, J.A. (1987), On sequencing retrievals in an automated storage/retrieval system, *IIE Transactions* 19(1), 56-66.

[102] Handfield, R.B., Nichols Jr., E.L. (1999), *Introduction to Supply Chain Management*, Prentice Hall, New Jersey.

[103] Hane, C.A., Barnhart, C., Johnson, E.L, Marsten, R.E., Nemhauser, G.L., Sigismondi, G. (1995), The fleet assignment problem: solving a large-scale integer program, *Mathematical Programming* 70, 211-232.

[104] Hausman, W.H., Schwarz, L.B., Graves, S.C. (1976), Optimal storage assignment in automatic warehousing systems, *Management Science* 22(6), 629-638.

[105] 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.

[106] Hodgson, T.J., King, R.E., Monteith, S.K., Schultz, S.R. (1987), Developing control rules for an AGVS using Markov decision processes, *Material Flow* 4, 85-96.

[107] Hoek, L., Hogewoning, M., Verplancke, T., Vis, I., Roodbergen, K.J. (2001), Het stroomlijnen van retourstromen binnen supermarktketens, *Tijdschrift voor Inkoop en Logistiek* 17(10), 54-56.

[108] Holguín-Veras, J., Jara-Díaz, S. (1999), Optimal pricing for priority service and space allocation in container ports, *Transportation Research B* 33, 81-106.

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

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

[111] Hwang, H., Ko, C.S. (1988), A study on multi-aisle system served by a single storage/retrieval machine, *International Journal of Production Research* 26(11), 1727-1737.

[112] Hwang, H., Lee, S.B. (1990), Travel-time models considering the operating characteristics of the storage and retrieval machine, *International Journal of Production Research* 28(10), 1779-1789.

[113] Ilić, O.R. (1994), Analysis of the number of automated guided vehicles required in flexible manufacturing systems, *International Journal of Advanced Manufacturing Technology* 9, 382-389.

[114] Imai, A., Nagaiwa, K., Tat, C.W. (1997), Efficient planning of berth allocation for container terminals in Asia, *Journal of Advanced Transportation* 31(1), 75-94.

[115] Imai, A., Nishimura, E., Papadimitriou, S. (2001), The dynamic berth allocation problem for a container port, *Transportation Research B* 35, 401-417.

[116] Johansen, R.S. (1999), Gate solutions, *paper presented at the Containerport and Terminal Performance Conference, February, Amsterdam.*

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

[118] Johnson, M.E., Brandeau, M.L. (1996), Stochastic modeling for automated material handling system design and control, *Transportation Science* 30(4), 330-350.

[119] Johnson, M.E., Brandeau, M.L. (1999), Design of an automated shop floor material handling system with inventory considerations, *Operations Research* 47(1), 65-80.

[120] Kasilingam, R.G., Gobal, S.L. (1996), Vehicle requirements model for automated guided vehicle systems, *International Journal of Advanced Manufacturing Technology* 12, 276-279.

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

[122] Kelly, J.P., Xu, J. (1999), A set-partitioning-based heuristic for the vehicle routing problem, *Journal on Computing* 11(2), 161-172.

[123] Keserla, A., Peters, B.A. (1994), Analysis of dual-shuttle automated storage/retrieval systems, *Journal of Manufacturing Systems* 13(6), 424-434.

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

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

[126] Kim, C.W., Tanchoco, J.M.A., Koo, P.H. (1997), Deadlock prevention in manufacturing systems with AGV systems: Banker's algorithm approach, *Journal of Manufacturing Science and Engineering* 119, 849-854.

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

[128] Kim, J., Klein, C.M. (1996), Location of departmental pickup and delivery points for an AGV system, *International Journal of Production Research* 34(2), 407-420.

[129] 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, 201-214.

[130] Kim, K.H. (1997), Evaluation of the number of rehandles in container yards, *Computers & Industrial Engineering* 32(4), 701-711.

[131] Kim, K.H., Bae, J.W. (1998), Re-marshaling export containers in port container terminals, *Computers & Industrial Engineering* 35(3-4), 655-658.

[132] Kim, K.H., Bae, J.W. (1999), A dispatching method for automated guided vehicles to minimize delays of containership operations, *International Journal of Management Science* 5(1), 1-25.

[133] Kim, K.H., Kim, H.B. (1998), The optimal determination of the space requirement and the number of transfer cranes for import containers, *Computers & Industrial Engineering* 35(3-4), 427-430.

[134] Kim, K.H., Kim, H.B. (1999), Segregating space allocation models for container inventories in port container terminals, *International Journal of Production Economics* 59, 415-423.

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

[136] Kim, K.H., Kim, K.Y. (1999), Routing straddle carriers for the loading operation of containers using a beam search algorithm, *Computers & Industrial Engineering* 36(1), 109-136.

[137] Kim, K.H., Kim, K.Y. (1999), An optimal routing algorithm for a transfer crane in port container terminals, *Transportation Science* 33(1), 17-33.

[138] Kim, K.H., Park, Y.M., Ryu, K.R. (2000), Deriving decision rules to locate export containers in container yards, *European Journal of Operational Research* 124, 89-101.

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

[140] Kim, K.Y., Kim, K.H. (1997), A routing algorithm for a single transfer crane to load export containers onto a containership, *Computers & Industrial Engineering* 33(3-4), 673-676.

[141] Kim, K.Y., Kim, K.H. (1999), A routing algorithm for a single straddle carrier to load export containers onto a containership, *International Journal of Production Economics* 59, 425-433.

[142] Kim, S.H., Hwang, H. (1999), An adaptive dispatching algorithm for automated guided vehicles based on an evolutionary process, *International Journal of Production Economics* 60-61, 465-472.

[143] King, R.E., Kim, K.S. (1995), AgvTalk: an object-oriented simulator for AGV systems, *Computers & Industrial Engineering* 28(3), 575-592.

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

[145] Kiran, A.S., Tansel, B.C. (1989), Optimal pickup point location on material handling networks, *International Journal of Production Research* 27(9), 1475-1486.

[146] Kiran, A.S., Unal, A.T., Karabati, S. (1992), A location problem on unicyclic networks: balanced case, *European Journal of Operational Research* 62, 194-202.

[147] Kodali, R. (1997), Knowledge-based system for selection of an AGV and a workcentre for transport of a part in on-line scheduling of FMS, *Production Planning & Control* 8(2), 114-122.

[148] Koff, G.A. (1987), Automatic guided vehicle systems: applications, controls and planning, *Material Flow* 4, 3-16.

[149] Kohl, N., Desrosiers, J., Madsen, O.B.G., Solomon, M.M., Soumis, F. (1999), 2-path cuts for the vehicle routing problem with time windows, *Transportation Science* 33(1), 101-116.

[150] Kohl, N., Madsen, O.B.G. (1997), An optimization algorithm for the vehicle routing problem with time windows based on Lagrangian relaxation, *Operations Research* 45(3), 395-406.

[151] Kolen, A.W.J., Rinnooy Kan, A.H.G., Trienekens, H.W.J.M. (1987), Vehicle routing with time windows, *Operations Research* 35(2), 266-273.

[152] Kollias, J.G., Manolopoulos, Y., Papadimitriou, C.H. (1990), The optimum execution order of queries in linear storage, *Information Processing Letters* 36, 141-145.

[153] Kouvelis, P., Gutierrez, G.J., Chiang, W.C. (1992), Heuristic unidirectional flowpath design approaches for automated guided vehicle systems, *International Journal of Production Research* 30(6), 1327-1351.

[154] Kouvelis, P., Kim, M.W. (1992), Unidirectional loop network layout problem in automated manufacturing systems, *Operations Research* 40(3), 533-550.

[155] Kozan, E. (1997), Increasing the operational efficiency of container terminals in Australia, *Journal of the Operational Research Society* 48(2), 151-161.

[156] Kozan, E. (1997), Comparison of analytical and simulation planning models of seaport container terminals, *Transportation Planning and Technology* 20, 235-248.

[157] Kozan, E. (2000), Optimising container transfers at multimodal terminals, *Mathematical and Computer Modelling* 31, 235-243.

[158] Kozan, E., Preston, P. (1999), Genetic algorithms to schedule container transfers at multimodal terminals, *International Transactions in Operational Research* 6, 311-329.

[159] Krishnamurthy, N.N., Batta, R., Karwan, M.H. (1993), Developing conflict-free routes for automated guided vehicles, *Operations Research* 41(6), 1077-1090.

[160] Kurstjens, S.T.G.L., Dekker, R., Dellaert, N.P., Duinkerken, M.B., Ottjes, J.A., Evers, J.J.M. (1996), Planning of inter terminal transport at the Maasvlakte, *TRAIL Proceedings 2nd TRAIL Congress 1996.*

[161] Laporte, G. (1992), The vehicle routing problem: an overview of exact and approximate algorithms, *European Journal of Operational Research* 59, 345-358.

[162] Law, A.M., Kelton, W.D. (1999), *Simulation Modeling & Analysis,* McGraw-Hill International Editions, New York.

[163] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B. (1985), *The Traveling Salesman Problem, a Guided Tour of Combinatorial Optimization,* John Wiley & Sons, Chichester.

[164] Lee, C.C., Lin, J.T. (1995), Deadlock prediction and avoidance based on Petri nets for zone-control automated guided vehicle systems, *International Journal of Production Research* 33(12), 3249-3265.

[165] Lee, H.F., Schaefer, S.K. (1996), Retrieval sequencing for unit-load automated storage and retrieval systems with multiple openings, *International Journal of Production Research* 34(10), 2943-2962.

[166] Lee, H.F., Schaefer, S.K. (1997), Sequencing methods for automated storage and retrieval systems with dedicated storage, *Computers & Industrial Engineering* 32(2), 351-362.

[167] Lee, J., Lee, M.K., Zhu, Z. (1996), WASA: a decision support system for workstations arrangement in single-vehicle closed-loop AGV systems, *Computers & Industrial Engineering* 30(1), 41-49.

[168] Lee, M.K., Kim, S.Y. (1995), Scheduling of storage/retrieval orders under a just-in-time environment, *International Journal of Production Research* 33(12), 3331-3348.

[169] Leeper, J.H. (1988), Integrated automated terminal operations, *Transportation Research Circular* 33, 23-28.

[170] Leung, L.C., Khator, S.K., Kimbler, D.L. (1987), Assignment of AGVs with different vehicle types, *Material Flow* 4, 65-72.

[171] Lim, J.M., Kim, K.S., Yum, B.J., Hwang, H. (1996), Determination of an optimal configuration of operating policies for direct-input-output manufacturing systems using the Taguchi method, *Computers & Industrial Engineering* 31(3/4), 555-560.

[172] Lin, J.T., Chang, C.C.K., Liu, W.C. (1994), A load-routeing problem in a tandem-configuration automated guided vehicle system, *International Journal of Production Research* 32(2), 411-427.

[173] Linn, R.J., Wysk, R.A. (1987), An analysis of control strategies for an automated storage/retrieval system, *INFOR* 25(1), 66-83.

[174] Linn, R.J., Wysk, R.A. (1990), An expert system framework for automated storage and retrieval system control, *Computers & Industrial Engineering* 18(1), 37-48.

[175] Linn, R.J., Wysk, R.A. (1990), An expert system based controller for an automated storage/retrieval system, *International Journal of Production Research* 28(4), 735-756.

[176] Linn, R.J., Xie, X. (1993), A simulation analysis of sequencing rules for ASRS in a pull-based assembly facility, *International Journal of Production Research* 31(10), 2355-2367.

[177] Lu, X., Gerchak, Y. (1998), Minimizing the expected response time of an idled server on a line, *IIE Transactions* 30, 401-408.

[178] Mahadevan, B., Narendran, T.T. (1990), Design of an automated guided vehicle-based material handling system for a flexible manufacturing system, *International Journal of Production Research* 28(9), 1611-1622.

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

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

[181] Mahajan, S., Rao, B.V., Peters, B.A. (1998), A retrieval sequencing heuristic for miniload end-of-aisle automated storage/retrieval systems, *International Journal of Production Research* 36(6), 1715-1731.

[182] Malmborg, C.J. (1990), A model for the design of zone control automated guided vehicle systems, *International Journal of Production Research* 28(10), 1741-1758.

[183] Malmborg, C.J. (2001), Rule of thumb heuristics for configuring storage racks in automated storage and retrieval systems design, *International Journal of Production Research* 39(3), 511-527.

[184] Manda, B.S., Palekar, U.S. (1997), Recent advances in the design and analysis of material handling systems, *Journal of Manufacturing Science and Engineering* 119, 841-848.

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

[186] Matson, J.O., White, J.A. (1982), Operational research and material handling, *European Journal of Operational Research* 11, 309-318.

[187] Maxwell, W.L., Muckstadt, J.A. (1982), Design of automatic guided vehicle systems, *IIE Transactions* 14(2), 114-124.

[188] Meersmans, P.J.M. (2002), Optimization of container handling systems, *Ph.D. Thesis,* Tinbergen Institute 271, Erasmus University Rotterdam.

[189] Meersmans, P.J.M., Vis, I.F.A., De Koster, M.B.M., Dekker, R. (1999), FAMAS-NewCon, Een model voor korte-termijn stacking. Modelbeschrijving, Report Econometric Institute EI-9942/A, Erasmus University Rotterdam.

[190] Meersmans, P.J.M., Wagelmans, A.P.M. (2001), Effective algorithms for integrated scheduling of handling equipment at automated container terminals, *ERIM Report Series Research in Management* ERS-2001-36-LIS, Erasmus University Rotterdam.

[191] Merkuryev, Y., Tolujew, J., Blümer, E., Novitsky, L., Ginters, E., Vitorova, E., Merkuryeva, G., Pronins, J. (1998), A modelling and simulation methodology for managing the Riga Harbour container terminal, *Simulation* 71(2), 84-95.

[192] Moon, S.W., Hwang, H. (1999), Determination of unit load sizes of AGV in multi-product multi-line assembly production systems, *International Journal of Production Research* 37(15), 3565-3581.

[193] Moore, T., Roy, C. (1998), Manage inventory in a real-time environment, *Transportation & Distribution* July, 68-73.

[194] Müller, T. (1983), *Automated Guided Vehicles*, IFS (Publications) Ltd, UK, Springer-Verlag, Berlin.

[195] Murthy, K.G. (1968), An algorithm for ranking all the assignments in order of increasing costs, *Operations Research* 16(3), 682-687.

[196] Nakano, M., Ohno, K. (1999), Decomposition algorithm for performance evaluation of AGV systems, *Production and Operations Management* 8(2), 193-205.

[197] Narasimhan, R., Batta, R., Karwan, M.H. (1999), Routing automated guided vehicles in the presence of interruptions, *International Journal of Production Research* 37(3), 653-681.

[198] Newman, A.M., Yano, C.A. (2000), Scheduling direct and indirect trains and containers in an intermodal setting, *Transportation Science* 34(3), 256-270.

[199] Newton, D. (1985), Simulation model calculates how many automated guided vehicles are needed, *Industrial Engineering* 17(2), 68-78.

[200] Nishimura, E., Imai, A., Papadimitriou, S. (2001), Berth allocation planning in the public berth system by genetic algorithms, *European Journal of Operational Research* 131, 282-292.

[201] Oboth, C., Batta, R., Karwan, M. (1999), Dynamic conflict-free routing of automated guided vehicles, *International Journal of Production Research* 37(9), 2003-2030.

[202] Orlin, J.B. (1982), Minimizing the number of vehicles to meet a fixed periodic schedule: an application of periodic posets, *Operations Research* 30(4), 760-776.

[203] Ozden, M. (1988), A simulation study of multiple-load-carrying automated guided vehicles in a flexible manufacturing system, *International Journal of Production Research* 26(8), 1353-1366.

[204] Pan, C.H., Wang, C.H. (1996), A framework for the dual command cycle travel time model in automated warehousing systems, *International Journal of Production Research* 34(8), 2099-2117.

[205] Papadimitriou, C.H., Steiglitz, K. (1982), *Combinatorial Optimization, Algorithms and Complexity*, Prentice-Hall, Inc., Englewood Cliffs.

[206] Park, B.C. (2001), An optimal dwell point policy for automated storage/retrieval systems with uniformly distributed, rectangular racks, *International Journal of Production Research* 39(7), 1469-1480.

[207] Peterkofsky, R.I., Daganzo, C.F. (1990), A branch and bound solution method for the crane scheduling problem, *Transportation Research B* 24(3), 159-172.

[208] Petersen, C.G. (1997), An evaluation of order picking routeing policies, *International Journal of Operations & Production Management* 17(11), 1098-1111.

[209] Phillips, D.T., Garsia-Diaz, A. (1981), *Fundamentals of Network Analysis*, Prentice Hall, Inc., Englewood Cliffs, New York.

[210] Psaraftis, H.N. (1988), Dynamic vehicle routing problems, in: *Vehicle Routing: Methods and Studies, Studies in Management Science and Systems*, Golden, B.L., Assad, A.A. (eds.), 223-248.

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

[212] Rajotia, S., Shanker, K., Batra, J.L. (1998), Determination of optimal AGV fleet size for an FMS, *International Journal of Production Research* 36(5), 1177-1198.

[213] Rajotia, S., Shanker, K., Batra, J.L. (1998), An heuristic for configuring a mixed uni/bidirectional flow path for an AGV system, *International Journal of Production Research* 36(7), 1779-1799.

[214] Ramani, K.V. (1996), An interactive simulation model for the logistics planning of container operations in seaports, *Simulation* 66(5), 291-300.

[215] Randhawa, S.U., McDowell, E.D., Wang, W.T. (1991), Evaluation of scheduling rules for single- and dual-dock automated storage/retrieval system, *Computers & Industrial Engineering* 20(4), 401-410.

[216] Randhawa, S.U., Shroff, R. (1995), Simulation-based design evaluation of unit load automated storage/retrieval systems, *Computers & Industrial Engineering* 28(1), 71-79.

[217] Rath, E. (1973), *Container Systems*, John Wiley & Sons, New York.

[218] Ratliff, H.D., Rosenthal, A.S. (1983), Orderpicking in a rectangular warehouse: a solvable case of the traveling salesman problem, *Operations Research* 31(3), 507-521.

[219] Reveliotis, S.A. (2000), Conflict resolution in AGV systems, *IIE Transactions* 32, 647-659.

[220] Rexing, B., Barnhart, C., Kniker, T., Jarrah, A., Krishnamurthy, N. (2000), Airline fleet assignment with time windows, *Transportation Science* 34(1), 1-20.

[221] Richardson, H.L. (1999), Cross docking: information flow saves space, *Transportation & Distribution* 40(11), 51-54.

[222] Ro, I.K., Kim, J.I. (1990), Multi-criteria operational control rules in flexible manufacturing systems (FMSs), *International Journal of Production Research* 28(1), 47-63.

[223] Rockafellar, R.T. (1984), *Network Flows and Monotropic Optimization*, *Pure & Applied Mathematics*, John Wiley & Sons, New York, 100-101.

[224] Roodbergen, K.J. (2001), Layout and routing methods for warehouses, *ERIM Ph.D. Series Research in Management* 4, Erasmus University Rotterdam.

[225] Roodbergen, K.J., De Koster, R. (2001), Routing order pickers in a warehouse with a middle aisle, *European Journal of Operational Research* 133, 32-43.

[226] Rosenblatt, M.J., Roll, Y., Zyser, V. (1993), A combined optimization and simulation approach for designing automated storage/retrieval systems, *IIE Transactions* 25(1), 40-50.

[227] Rouwenhorst, B., Reuter, B., Stockrahm, V., Van Houtum, G.J., Mantel, R.J., Zijm, W.H.M. (2000), Warehouse design and control: framework and literature review, *European Journal of Operational Research* 122, 515-533.

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

[229] Sarker, B.R., Babu, P.S. (1995), Travel time models in automated storage/retrieval systems: a critical review, *International Journal of Production Economics* 40, 173-184.

[230] Savelsbergh, M., Sol, M. (1998), Drive: dynamic routing of independent vehicles, *Operations Research* 46(4), 474-490.

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

[232] Schaffer, B. (1997), Implementing a succesful crossdocking operation, *IIE Solutions* 29(10), 34-36.

[233] Schaffer, B. (1998), Cross docking can increase efficiency, *Automatic I.D. News* 14(8), 34-37.

[234] Scheithauer, G. (1999), LP-based bounds for the container and multi-container loading problem, *International Transactions in Operational Research* 6, 199-213.

[235] Schwarz, L.B., Graves, S.C., Hausman, W.H. (1978), Scheduling policies for automatic warehousing systems: simulation results, *AIIE Transactions* 10(3), 260-270.

[236] Schwind, G.F. (1996), A systems approach to docks and crossdocking, *Material Handling Engineering* 51(2), 59-62.

[237] Seidmann, A. (1988), Intelligent control schemes for automated storage and retrieval systems, *International Journal of Production Research* 26(5), 931-952.

[238] Seifert, R.W., Kay, M.G., Wilson, J.R. (1998), Evaluation of AGV routeing strategies using hierarchical simulation, *International Journal of Production Research* 36(7), 1961-1976.

[239] Seo, Y., Egbelu, P.J. (1999), Integrated manufacturing planning for an AGV-based FMS, *International Journal of Production Economics* 60-61, 473-478.

[240] Sharp, G.P., Liu, F.H.F. (1990), An analytical method for configuring fixed-path, closed loop material handling systems, *International Journal of Production Research* 28(4), 757-783.

[241] Shen, W.S., Khoong, C.M. (1995), A DSS for empty container distribution planning, *Decision Support Systems* 15, 75-82.

[242] Shields, J.J. (1984), Container stowage: a computer-aided preplanning system, *Marine Technology* 21(4), 370-383.

[243] Shiizuka, H., Suzuki, M. (1994), Modeling of AGV networks in flexible manufacturing systems, *Computers & Industrial Engineering* 27(1-4), 81-86.

[244] Sinriech, D. (1995), Network design models for discrete material flow systems: a literature review, *International Journal of Advanced Manufacturing Technology* 10, 277-291.

[245] Sinriech, D., Palni, L. (1998), Scheduling pickup and deliveries in a multiple-load discrete carrier environment, *IIE Transactions* 30, 1035-1047.

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

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

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

[249] Sinriech, D., Tanchoco, J.M.A. (1993), Solution methods for the mathematical models of single-loop AGV systems, *International Journal of Production Research* 31(3), 705-725.

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

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

[252] Solomon, M.M. (1987), Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research* 35(2), 254-265.

[253] Solomon, M.M., Baker, E.K., Schaffer, J.R. (1988), Vehicle routing and scheduling problems with time window constraints: efficient implementations of solution improvement procedures, in: *Vehicle Routing: Methods and Studies, Studies in Managment Science and Systems,* Golden, B.L., Assad, A.A. (eds.), 85-104.

[254] Solomon, M.M., Desrosiers, J. (1988), Time window constrained routing and scheduling problems, *Transportation Science* 22(1), 1-13.

[255] Soylu, M., Özdemirel, N.E., Kayaligil, S. (2000), A self-organizing neural network approach for the single AGV routing problem, *European Journal of Operational Research* 121, 124-137.

[256] Srinivasan, M.M., Bozer, Y.A., Cho, M. (1994), Trip-based material handling systems: throughput capacity analysis, *IIE Transactions* 26(1), 70-89.

[257] Steenken, D. (1992), Fahrwegoptimierung am containerterminal unter echtzeitbedingungen, *OR Spektrum* 14, 161-168.

[258] Steenken, D., Henning A., Freigang, S., Voß, S. (1993), Routing of straddle carriers at a container terminal with the special aspect of internal moves, *OR Spektrum* 15, 167-172.

[259] Sun, X.C., Tchernev, N. (1996), Impact of empty vehicle flow on optimal flow path design for unidirectional AGV systems, *International Journal of Production Research* 34(10), 2827-2852.

[260] Taghaboni, F., Tanchoco, J.M.A. (1988), A LISP-based controller for free-ranging automated guided vehicle systems, *International Journal of Production Research* 26(2), 173-188.

[261] Taghaboni-Dutta, F., Tanchoco, J.M.A. (1995), Comparison of dynamic routeing techniques for automated guided vehicle system, *International Journal of Production Research* 33(10), 2653-2669.

[262] Taleb-Ibrahimi, M., De Castilho, B., Daganzo, C.F. (1993), Storage space vs handling work in container terminals, *Transportation Research B* 27, 13-32.

[263] Tanchoco, J.M.A., Egbelu, P.J., Taghaboni, F. (1987), Determination of the total number of vehicles in an AGV-based material transport system, *Material Flow* 4, 33-51.

[264] Tanchoco, J.M.A., Sinriech, D. (1992), OSL-optimal single-loop guide paths for AGVS, *International Journal of Production Research* 30(3), 665-681.

[265] Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A., Trevino, J. (1996), *Facilities Planning*, John Wiley & Sons, New York.

[266] Ulusoy, G., Bilge, Ü. (1993), Simultaneous scheduling of machines and automated guided vehicles, *International Journal of Production Research* 31(12), 2857-2873.

[267] Ulusoy, G., Sivrikaya-Serifoglu, F., Bilge, Ü. (1997), A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles, *Computers & Operations Research* 24(4), 335-351.

[268] Van den Berg, J.P. (1999), A literature survey on planning and control of warehousing systems, *IIE Transactions* 31, 751-762.

[269] Van den Berg, J.P., Gademann, A.J.R.M. (1999), Optimal routing in an automated storage/retrieval system with dedicated storage, *IIE Transactions* 31, 407-415.

[270] Van der Baan, C.A.S. (1997), Knelpunten bij uitbesteding: prijsvechten of strategisch partnership?, in: *Logistieke Knelpunten in het Nederlandse Bedrijfsleven*, De Koster, M.B.M., Roos, H.B., De Vaan, M.J.M. (eds.), 107-120.

[271] Van der Meer, J.R. (2000), Operational control of internal transport, *ERIM Ph.D. Series Research in Management* 1, Erasmus University Rotterdam.

[272] Van der Meer, R., Vis, I., De Koster, R., Meersmans, P., Dekker, R. (1999), Containerterminals in het Famas NewCon concept, in: *SIMLOG Simulatie van Logistieke Ketens Rond de Haven*, Verspui, L., Van Nunen, J. (eds.), Eburon, Delft, 89-104.

[273] Van Hee, K.M., Huitink, B., Leegwater, D.K. (1988), Portplan, decision support system for port terminals, *European Journal of Operational Research* 34, 249-261.

[274] Van Hee, K.M., Wijbrands, R.J. (1988), Decision support system for container terminal planning, *European Journal of Operational Research* 34, 262-272.

[275] Van Horssen, W. (1996), The trouble with growth, *Cargo Systems* February, 51-52.

[276] Van Oudheusden, D.L., Zhu, W. (1992), Storage layout of AS/RS racks based on recurrent orders, *European Journal of Operational Research* 58, 48-56.

[277] Venkataramanan, M.A., Wilson, K.A. (1991), A branch-and-bound algorithm for flow-path design of automated guided vehicle systems, *Naval Research Logistics* 38, 431-445.

[278] Vis, I., Roodbergen, K.J. (2001), Distributieketens onderzocht, *Tijdschrift voor Inkoop en Logistiek* 17(7/8), 36-37.

[279] Vis, I.F.A. (1997), Kasvoorraadbeheer bij aangesloten banken, Master Thesis, Leiden University.

[280] Vis, I.F.A. (2002), A set partitioning approach to determine the minimum number of vehicles required within a facility, working paper, Erasmus University Rotterdam.

[281] Vis, I.F.A., De Koster, R. (2001), Transshipment of containers at a container terminal: an overview, under second revision of *European Journal of Operational Research.*

[282] Vis, I.F.A., De Koster, R., Roodbergen, K.J., Peeters, L.W.P. (2001), Determination of the number of automated guided vehicles required at a semi-automated container terminal, *Journal of the Operational Research Society* 52, 409-417.

[283] Vis, I.F.A., De Koster, R., Savelsbergh, M.W.P. (2000), Estimation of the number of transport vehicles at a container terminal, in: *Progress in Material Handling Research: 2000,* Graves, R.J., McGinnis, L.F., Ogle, M.K., Peters, B.A., Ward, R.E., Wilhelm, M.R. (eds.), Material Handling Institute, Charlotte, North Carolina, 404-420.

[284] Vis, I.F.A., De Koster, R., Savelsbergh, M.W.P. (2001), Minimum vehicle fleet size at a container terminal, *ERIM Report Series Research in Management* ERS-2001-24-LIS, Erasmus University Rotterdam.

[285] Vis, I.F.A., Roodbergen, K.J. (to appear), Examining supply chains from practice, in: *New Developments in Distribution Logistics and Supply Chain Management*, Klose, A., Speranza, M.G., Van Wassenhove, L.G. (eds.), Springer-Verlag, Berlin.

[286] Waller, D.L. (1999), *Operations Management, a Supply Chain Approach*, International Thomson Business Press, London.

[287] Wan, T.B., Wah, E.L.C., Meng, L.C. (1992), The use of information technology by the port of Singapore authority, *World Development* 20(12), 1785-1795.

[288] Wang, J.Y., Yih, Y. (1997), Using neural networks to select a control strategy for automated storage and retrieval systems (AS/RS), *International Journal of Computer Integrated Manufacturing* 10(6), 487-495.

[289] Watanabe, M., Furukawa, M., Kakazu, Y. (2001), Intelligent AGV driving toward an autonomous decentralized manufacturing system, *Robotics and Computer Integrated Manufacturing* 17, 57-64.

[290] Welgama, P.S., Gibson, P.R. (1996), An integrated methodology for automating the determination of layout and materials handling system, *International Journal of Production Research* 34(8), 2247-2264.

[291] Wen, U.P., Chang, D.T., Chen, S.P. (2001), The impact of acceleration/deceleration on travel-time models in class-based automated S/R systems, *IIE Transactions* 33, 599-608.

[292] Wilson, I.D., Roach, P.A. (2000), Container stowage planning: a methodology for generating computerised solutions, *Journal of the Operational Research Society* 51, 1248-1255.

[293] Wysk, R.A., Egbelu, P.J., Zhou, C., Ghosh, B.K. (1987), Use of spread sheet analysis for evaluating AGV systems, *Material Flow* 4, 53-64.

[294] Yeh, M.S., Yeh, W.C. (1998), Deadlock prediction and avoidance for zone-control AGVS, *International Journal of Production Research* 36(10), 2879-2889.

[295] Yim, D.S., 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.

[296] Yun, W.Y., Choi, Y.S. (1999), A simulation model for container-terminal operation analysis using an object-oriented approach, *International Journal of Production Economics* 59, 221-230.

[297] Zaremba, M.B., Obuchowicz, A., Banaszak, Z.A., Jedrzejek, K.J. (1997), A max-algebra approach to the robust distributed control of repetitive AGV systems, *International Journal of Production Research* 35(10), 2667-2687.

[298] Zeng, L., Wang, H.P., Jin, S. (1991), Conflict detection of automated guided vehicles: a petri net approach, *International Journal of Production Research* 29(5), 865-879.

Appendix A

Proof of Chapter 3

THEOREM 3.1.
The result of the algorithm *Minimum Flow* is a feasible flow of minimum value in graph G.
PROOF
Original problem.
The original problem consists of finding a minimum flow in a graph $G = (V, A)$ with lower bounds l_{ij} and upper bounds u_{ij} on the arcs $(i, j) \in A$. This problem is formally formulated as follows

$$
\begin{array}{lllr}
\mathcal{P} = & Minimise & v & \\
& subject\ to & \displaystyle\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} v & i = s \\ 0 & i \not= s, t \\ -v & i = t \end{cases} & \forall i \in V \\
& & l_{ij} \leq x_{ij} \leq u_{ij} & \forall (i, j) \in A
\end{array}
$$

Initialisation (step 1).

Given is a feasible flow x in the graph $G_1 = (V_1, A_1)$ with value K, i.e. given x_{ij} $\forall (i, j) \in A_1$ satisfying

$V_1 = \{s, 1', 1'', .., N', N'', t\}$

$A_1 = \{(i'', j') : i'', j' \in V_1$, jobs i and j are compatible$\} \cup \{(i', i'') : i', i'' \in V_1\}$ $\cup\{(s, i') : s, i' \in V_1\} \cup (j'', t) : j'', t \in V_1\}$.

$$\sum_{j:(i,j)\in A_1} x_{ij} - \sum_{j:(j,i)\in A_1} x_{ji} = \begin{cases} K & i = s \\ 0 & i \neq s,t \\ -K & i = t \end{cases} \qquad \forall i \in V_1$$
$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in A_1$$

Note that such a flow can easily be constructed by computing a maximum flow in G_1.

Maximum flow in transformed graph $G_2 = (V_2, A_2)$.

Define (according to step 3) a graph $G_2 = (V_2, A_2)$ where the node set V_2 equals V_1 and the arc set A_2 consists of all arcs in A_1, and of a backward directed arc (j,i) for each arc $(i,j) \in A_1$. The capacities of the arcs in A_2 are defined by

$$\begin{aligned} \Delta_{ij} &= x_{ij} - l_{ij} \quad \forall (i,j) \in A_2 : (i,j) \in A_1 \\ \Delta_{ij} &= u_{ji} - x_{ji} \quad \forall (i,j) \in A_2 : (j,i) \in A_1 \end{aligned}$$

Solve the maximum flow problem stated below (step 4):

$$\begin{aligned} \mathcal{P}' = \quad & Maximise \quad v' \\ & subject\ to \quad \sum_{j:(i,j)\in A_2} x'_{ij} - \sum_{j:(j,i)\in A_2} x'_{ji} = \begin{cases} v' & i = s \\ 0 & i \neq s,t \\ -v' & i = t \end{cases} \qquad \forall i \in V_2 \\ & \qquad 0 \leq x'_{ij} \leq \Delta_{ij} \qquad \forall (i,j) \in A_2 \end{aligned}$$

A solution to the minimum flow problem (step 5).

An optimal solution x^* to $\mathcal{P}$ can be constructed by combining the feasible flow x with the maximum flow x' as follows:

$$x^*_{ij} := x_{ij} - x'_{ij} + x'_{ji} \ \ \forall (i,j) \in A_1.$$

Proof of feasibility and optimality of x^*.

Consider an arc $(i,j) \in A_1$, and the corresponding variables x'_{ij} and x'_{ji} from $\mathcal{P}'$. For these variables the following holds:

$$\begin{aligned} 0 \le x'_{ij} \le x_{ij} - l_{ij} \\ 0 \le x'_{ji} \le u_{ij} - x_{ij}. \end{aligned}$$

Adding these two inequalities gives

$$l_{ij} - x_{ij} \le -x'_{ij} + x'_{ji} \le u_{ij} - x_{ij}.$$

Adding x_{ij} to this inequality gives $x_{ij} - x'_{ij} + x'_{ji} = x^*_{ij}$ for the middle term, and it follows that x^* is feasible with respect to the flow bounds:

$$l_{ij} \le x^*_{ij} \le u_{ij} \quad \forall (i,j) \in A_1$$

Next, consider the flow conservation constraint in $\mathcal{P}'$

$$\begin{aligned} &\sum_{j:(i,j)\in A_2} x'_{ij} - \sum_{j:(j,i)\in A_2} x'_{ji} = \\ &\sum_{j:(i,j)\in A_1} x'_{ij} + \sum_{j:(j,i)\in A_1} x'_{ij} - \sum_{j:(j,i)\in A_1} x'_{ji} - \sum_{j:(i,j)\in A_1} x'_{ji} = \\ &\sum_{j:(i,j)\in A1} (x'_{ij} - x'_{ji}) - \sum_{j:(j,i)\in A_1} (x'_{ji} - x'_{ij}). \end{aligned}$$

The step from the first to the second equation follows by considering A_1 rather than A_2 as the index set over which to sum, and by correspondingly indexing the variables x'_{ij} for which $(i,j) \not\in A$ on the backward directed arc (j,i) that does exist in A_1.

Using the definition of x^*_{ij} we obtain

$$\begin{aligned} &\sum_{j:(i,j)\in A_1} (x'_{ij} - x'_{ji}) - \sum_{j:(j,i)\in A_1} (x'_{ji} - x'_{ij}) = \\ &\sum_{j:(i,j)\in A1} (x_{ij} - x^*_{ij}) - \sum_{j:(j,i)\in A_1} (x_{ji} - x^*_{ji}) = \\ &\left(\sum_{j:(i,j)\in A_1} x_{ij} - \sum_{j:(j,i)\in A_1} x_{ji} \right) - \left(\sum_{j:(i,j)\in A_1} x^*_{ij} - \sum_{j:(j,i)\in A_1} x^*_{ji} \right) \end{aligned}$$

From the initialisation we know that

$$\sum_{j:(i,j)\in A_1} x_{ij} - \sum_{j:(j,i)\in A_1} x_{ji} = \begin{cases} K & i = s \\ 0 & i \not= s, t \\ -K & i = t \end{cases}$$

With all the above, problem $\mathcal{P}'$ can be rewritten in terms of x^*_{ij} as

$$\begin{aligned} \mathcal{P}' = \quad & \mathit{Maximise} \quad v' \\ & \mathit{subject\ to} \quad \sum_{j:(i,j)\in A_1} x^*_{ij} - \sum_{j:(j,i)\in A_1} x^*_{ji} = \begin{cases} K - v' & i = s \\ 0 & i \neq s,t \\ v' - K & i = t \end{cases} \qquad \forall i \in V_1 \\ & \qquad\qquad l_{ij} \leq x^*_{ij} \leq u_{ij} \qquad \forall (i,j) \in A_1 \end{aligned}$$

Introducing $v^* = K - v'$, the objective becomes to maximise $v' = K - v^*$, or, equivalently, to minimise $v^* - K$, where K is a constant and can thus be left out of the objective. We finally write

$$\begin{aligned} \mathcal{P}' = \quad & \mathit{Minimise} \quad v^* \\ & \mathit{subject\ to} \quad \sum_{j:(i,j)\in A_1} x^*_{ij} - \sum_{j:(j,i)\in A_1} x^*_{ji} = \begin{cases} v^* & i = s \\ 0 & i \neq s,t \\ -v^* & i = t \end{cases} \qquad \forall i \in V_1 \\ & \qquad\qquad l_{ij} \leq x^*_{ij} \leq u_{ij} \qquad \forall (i,j) \in A_1 \end{aligned}$$

It follows that x^* is both feasible and optimal for $\mathcal{P}$. ■

Appendix B

Directed paths of network of example in Chapter 4

1:	s	g_{11}	t			16:	s	g_{12}	t		
2:	s	g_{11}	g_{23}	t		17:	s	g_{12}	g_{24}	t	
3:	s	g_{11}	g_{23}	g_{31}	t	18:	s	g_{12}	g_{24}	g_{32}	t
4:	s	g_{11}	g_{23}	g_{32}	t	19:	s	g_{12}	g_{24}	g_{33}	t
5:	s	g_{11}	g_{23}	g_{33}	t	20:	s	g_{12}	g_{25}	t	
6:	s	g_{11}	g_{24}	t		21:	s	g_{12}	g_{25}	g_{33}	t
7:	s	g_{11}	g_{24}	g_{32}	t	22:	s	g_{12}	g_{26}	t	
8:	s	g_{11}	g_{24}	g_{33}	t	23:	s	g_{12}	g_{27}	t	
9:	s	g_{11}	g_{25}	t		24:	s	g_{12}	g_{31}	t	
10:	s	g_{11}	g_{25}	g_{33}	t	25:	s	g_{12}	g_{32}	t	
11:	s	g_{11}	g_{26}	t		26:	s	g_{12}	g_{33}	t	
12:	s	g_{11}	g_{27}	t		27:	s	g_{13}	t		
13:	s	g_{11}	g_{31}	t		28:	s	g_{13}	g_{25}	t	
14:	s	g_{11}	g_{32}	t		29:	s	g_{13}	g_{25}	g_{33}	t
15:	s	g_{11}	g_{33}	t		30:	s	g_{13}	g_{26}	t	

31:	s	g_{13}	g_{27}	t	39:	s	g_{22}	t	
32:	s	g_{13}	g_{31}	t	40:	s	g_{22}	g_{31}	t
33:	s	g_{13}	g_{32}	t	41:	s	g_{22}	g_{32}	t
34:	s	g_{13}	g_{33}	t	42:	s	g_{22}	g_{33}	t
35:	s	g_{21}	t		43:	s	g_{23}	t	
36:	s	g_{21}	g_{31}	t	44:	s	g_{23}	g_{31}	t
37:	s	g_{21}	g_{32}	t	45:	s	g_{23}	g_{32}	t
38:	s	g_{21}	g_{33}	t	46:	s	g_{23}	g_{33}	t

47:	s	g_{24}	t		52:	s	g_{26}	t
48:	s	g_{24}	g_{32}	t	53:	s	g_{27}	t
49:	s	g_{24}	g_{33}	t	54:	s	g_{31}	t
50:	s	g_{25}	t		55:	s	g_{32}	t
51:	s	g_{25}	g_{33}	t	56:	s	g_{33}	t

Appendix C

List of abbreviations

AGV	Automated Guided Vehicle
ALV	Automated Lifting Vehicle
ASC	Automated Stacking Crane
AS/RS	Automated Storage and Retrieval System
EDI	Electronic Data Interchange
I/O	Input/Output Station
MTS	Multi Trailer System
p&d points	Pick-up and delivery points
QC	Quay Crane
SC	Straddle Carrier
SRM	Storage and Retrieval Machine
TEU	Twenty feet Equivalent Unit

Samenvatting

De technologische ontwikkelingen staan heden ten dage niet stil. Van mobiel bellen tot sms-en en van e-mailen tot wappen. Nieuwe communicatiemogelijkheden worden bedacht en ontwikkeld. Hierdoor verlopen allerlei processen in de maatschappij steeds sneller. Zo kan de consument via de electronische snelweg tegenwoordig producten rechtstreeks vanuit huis bestellen. Consumenten raken meer en meer gewend aan deze snelle en gemakkelijke manier van bestellen. Als gevolg daarvan verwachten zij een vergelijkbaar snelle en gemakkelijke manier van aflevering van producten. Om te kunnen reageren op continu veranderend klantengedrag kan het voor ondernemers van belang zijn de aandacht te vestigen op de eigen processen en op de interactie met de overige onderdelen van de distributieketen. Een distributieketen omvat alle stromen en transformaties van goederen, van grondstoffen tot eindproducten, alsmede de gerelateerde informatiestromen. Retourstromen stromen terug van de consument naar de leveranciers of producenten. Deze stromen ontstaan ondermeer door de van overheidswege opgelegde verplichting tot het recyclen van producten. Alsmede door garantiebepalingen voor klanten om producten die niet aan de verwachtingen voldoen terug te sturen en door producten die moeten worden gerepareerd. Deze en vele andere trends beïnvloeden de prestaties van een distributieketen en van de afzonderlijke onderdelen van de keten. Hoofdstuk 1 beschrijft verschillende van deze trends en hun invloed op de verschillende processen in een distributieketen.

Binnen de verschillende onderdelen (knooppunten) van de distributieketen spelen zich verscheidene processen af, zoals het productie-, opslag-, overslag- en verkoopproces. Het productieproces kan plaatsvinden in fabrieken. Magazijnen worden gebruikt om grondstoffen, halffabrikaten en eindproducten tijdelijk op te slaan, om zo de schommelingen tussen het productieproces en de vraag van klanten te kunnen opvangen. Producten kunnen los of gegroepeerd, in bijvoorbeeld een container, worden vervoerd. Het transport van producten naar de eindgebruiker kan plaatsvinden met behulp van verschillende transportmiddelen, zoals vrachtwagens, schepen en treinen. In overslagcentra, zoals bijvoorbeeld een containerterminal of een crossdockingcentrum kunnen producten worden overgeslagen van het ene vervoermiddel naar het andere. Uiteindelijk arriveren de producten bij de klant of in een winkel, waar ze door de klant kunnen worden gekocht. In hoofdstuk 1 worden deze knooppunten en hun logistieke activiteiten nader beschreven. In dit proefschrift concentreren we ons op twee verschillende logistieke activiteiten binnen de verschillende knooppunten. Namelijk intern transport van producten en in- en uitslag van producten uit het opslaggebied. Door de prestatie van deze logistieke activiteiten te verbeteren, kunnen prestatieverbeteringen binnen de distributieketen worden bereikt. In dit proefschrift worden nieuwe plannings- en besturingsconcepten ontwikkeld voor transportsystemen en opslagsystemen.

In het eerste gedeelte van dit proefschrift bestuderen we het interne transportproces. In geautomatiseerde omgevingen zijn automatisch geleide voertuigen beschikbaar om producten van het ene gebied naar het andere gebied te transporteren. Hoofdstuk 2 geeft een overzicht van literatuur over verschillende beslissingsproblemen die een rol spelen bij het ontwikkelen en besturen van systemen met automatisch geleide voertuigen. Een van deze beslissingsproblemen is het zodanig bepalen van het aantal benodigde voertuigen dat alle producten op tijd worden getransporteerd.

In hoofdstuk 3 beschouwen we een omgeving waarbinnen het transportproces gekoppeld is aan de andere processen. Dit betekent dat andere processen invloed hebben op het tijdstip waarop producten kunnen worden vervoerd. Niet-heffende voertuigen worden gebruikt voor het transport. Dit heeft als gevolg dat een ander systeem beschikbaar moet zijn om het product op het voertuig te plaatsen. Producten mogen niet wachten op transport. Het voertuig moet dus eerder aanwezig zijn dan het product. Aangenomen, dat voor iedere transportopdracht bekend is op welk moment het transport moet beginnen, wat de oorsprong en bestemming zijn en wat de bijbehorende rijtijd is, kan het probleem van het bepalen van het aantal voertuigen worden gemodelleerd als een netwerkprobleem. In het netwerk wordt iedere opdracht weergegeven door een knoop. Twee knopen i en j worden verbonden met een gerichte pijl (i, j) indien de opdrachten door hetzelfde voertuig kunnen worden getransporteerd. Dit is het geval als de starttijd van opdracht j groter of gelijk is aan het moment waarop opdracht i wordt afgeleverd op de bestemming plus de rijtijd van daar naar de oorsprong van opdracht j. Ook worden knopen toegevoegd voor de bron en een eindpunt. In het netwerk kunnen nu gerichte paden van de bron naar het eindpunt worden gevonden, die verschillende opdrachten met elkaar verbinden. Het minimum aantal gerichte paden, zodanig dat elke knoop in precies een pad voorkomt, correspondeert met het minimale aantal voertuigen, dat nodig is om alle opdrachten op tijd te transporteren. Het in hoofdstuk 3 ontwikkelde algoritme bepaalt een minimale stroom door het netwerk. Deze minimale stroom komt overeen met het minimale aantal voertuigen dat nodig is om alle opdrachten uit te voeren. Dit algoritme lost het probleem optimaal en in polynomiale tijd op.

Een uitbreiding van dit probleem wordt behandeld in hoofdstuk 4. In dit geval is het transportproces losgekoppeld van de andere processen door buffers. Opdrachten die beschikbaar zijn voor transport worden in een buffer geplaatst. Deze buffer kan zijn gepositioneerd op bijvoorbeeld de grond of aan het einde van een conveyor. Door ruimtebeperkingen en om vertragingen te voorkomen heeft elke buffer een beperkte capaciteit. Voordat een volgend product in een volle buffer kan worden geplaatst, moet eerst de oudste opdracht uit de buffer worden getransporteerd. Als gevolg hiervan heeft elke opdracht een laatste moment waarop het transport kan aanvangen. Dit moment, samen met het moment waarop een opdracht in de buffer wordt geplaatst, resulteert in een tijdsvenster voor elke opdracht waarbinnen een voertuig moet starten met het transporteren van de opdracht. Door dit tijdsvenster te discretiseren ontstaat voor elke opdracht een eindig aantal mogelijke transportmomenten. In hoofdstuk 4 zijn twee verschillende oplossingsmethoden ontwikkeld om het minimaal aantal voertuigen te bepalen voor deze situatie. Het probleem kan worden geformuleerd als een geheeltallig programmeringsprobleem. Het doel is om het aantal gerichte paden zodanig te minimaliseren dat elke opdracht met precies een startmoment in precies een gericht pad voorkomt. Daarnaast kan het probleem worden geformuleerd als een setpartitioningprobleem. Een nieuwe zoekmethode is ontwikkeld om alle mogelijke combinaties van opdrachten (gerichte paden) te vinden, die door een voertuig kunnen worden getransporteerd. Met deze gerichte paden kan een incidentiematrix worden opgesteld, met welke het minimale aantal voertuigen kan worden bepaald.

In het tweede gedeelte van dit proefschrift staat het inslag- en uitslagproces centraal. We beschouwen een opslaggebied waar eenheidsladingen, zoals pallets of containers, tijdelijk worden opgeslagen. Opslagsystemen voeren inslag- en uitslagopdrachten uit. Een van de operationele problemen is het bepalen van de volgorde waarin opdrachten worden verwerkt, zodanig dat de rijtijd van het systeem minimaal is. In hoofdstuk 5 wordt een literatuuroverzicht gegeven van de literatuur over de volgordebepaling van inslag- en uitslagopdrachten bij geautomatiseerde systemen. Deze systemen werken in een of meerdere gangen. Elke gang heeft een of meerdere punten waar producten worden overgeladen van het opslagsysteem naar andere systemen en vice versa. In de literatuur wordt nauwelijks aandacht besteed aan gangen met meerdere van deze punten. In de praktijk worden veelal meerdere inslag- en uitslagpunten gebruikt om de verschillende stromen in het systeem te scheiden en om een hogere doorstroming te verkrijgen. Een voorbeeld hiervan is het opslaggebied in een containerterminal. Containers komen beschikbaar voor in- en uitslag aan twee kanten van het opslaggebied, namelijk aan de zeekant en aan de landkant. Daarnaast is weinig aandacht besteed aan de bepaling van de volgorde van opdrachten indien een opslagsysteem meerdere gangen bedient. Dit komt in de praktijk voor indien opslagtijden veel groter zijn dan doorlooptijden in het systeem.

In hoofdstuk 6 wordt een gecombineerd probleem bestudeerd, namelijk de volgordebepaling van inslag- en uitslagopdrachten voor een opslagsysteem, dat meerdere gangen bedient. Een algoritme is ontwikkeld dat in snelle rekentijd een optimale volgorde bepaalt van inslag- en uitslagopdrachten in een enkele gang met twee inslag/uitslagpunten. Het doel hierbij is om de leegrijtijden van het systeem zodanig te minimaliseren dat alle opdrachten worden afgehandeld door het systeem dat begint en eindigt aan vooraf gespecificeerde kanten van de gang. Het probleem kan worden geformuleerd als een toewijzingsprobleem. Niet gekoppelde kringen in de oplossing worden op optimale wijze gekoppeld. Dit resulteert in een optimale oplossing van het genoemde probleem. Onder de aanname dat een systeem eerst een gang afhandelt voordat de overgang naar een nieuwe gang wordt gemaakt, kan het resultaat van deze eerste stap worden gebruikt voor het oplossen van het probleem met inslag- en uitslag in meerdere gangen.

Een dynamischprogrammeringsprobleem is ontwikkeld om de volgorde te bepalen van inslag- en uitslagopdrachten in meerdere gangen voor een systeem dat een route start in een vast begin- en eindpunt. Het doel is om een gerichte route te vinden waarbij wordt aangegeven in welke volgorde de gangen moeten worden behandeld, aan welke kant van elke gang wordt begonnen, aan welke kant van elke gang wordt geëindigd en in welke volgorde de opdrachten in een gang worden afgehandeld. Dit wordt zodanig gedaan, dat alle opdrachten worden afgehandeld en dat de leegrijtijd van het systeem wordt geminimaliseerd. Er wordt aangetoond dat het dynamischprogrammeringsalgoritme het probleem in polynomiale tijd optimaal kan oplossen.

In het derde gedeelte van dit proefschrift is beschreven hoe de ontwikkelde plannings- en besturingsconcepten kunnen worden toegepast in een van de knooppunten in een distributieketen, namelijk een containerterminal. In een containerterminal worden containers overgeslagen van het ene transportsysteem naar het andere. Containers arriveren over het water op schepen. Kadekranen worden gebruikt om de containers te lossen van het schip. Geautomatiseerde of bemande voertuigen zorgen voor het interne transport van de containers naar het opslaggebied. Meestal worden containers tijdelijk opgeslagen, voordat verder transport richting de bestemming wordt uitgevoerd. Geautomatiseerde bovenloopkranen, bemande kranen of straddle carriers zorgen voor de inslag en uitslag van containers in het opslaggebied. Om containers op een schip te laden, wordt het omgekeerde proces uitgevoerd. De verschillende processen in een containerterminal en de gebruikte transport- en opslagsystemen worden in detail beschreven in hoofdstuk 7 van dit proefschrift. Verschillende beslissingsproblemen treden op in een containerterminal. Een overzicht van gerelateerde literatuur is eveneens beschreven in hoofdstuk 7.

Het is duidelijk, dat een containerterminal het transportproces en het inslag- en uitslagproces een rol van betekenis spelen. In hoofdstuk 8 wordt een containerterminal bestudeerd waarin geautomatiseerde niet-heffende voertuigen worden vergeleken met heffende voertuigen. De modellen van de hoofdstukken 3 en 4 kunnen worden toegepast om de minimale hoeveelheid benodigde niet-heffende en heffende voertuigen in een containerterminal te bepalen. Met behulp van een simulatiestudie wordt inzicht verkregen in het verschil in aantallen heffende en niet-heffende voertuigen, die nodig zijn om alle containers op tijd te transporteren. Een van de resultaten is het feit dat ongeveer 30% minder voertuigen nodig zijn indien, in plaats van niet-heffende voertuigen, gebruik wordt gemaakt van heffende voertuigen. Dit resultaat geldt voor een containerterminal met buffers met een capaciteit van twee containers.

In hoofdstuk 9 wordt het inslag- en uitslagproces van containers in en uit het opslaggebied behandeld. De modellen uit hoofdstuk 6 kunnen worden toegepast om de volgorde te bepalen van inslag- en uitslagopdrachten bij straddle carriers en geautomatiseerde bovenloopkranen. Het grootste verschil tussen beide systemen is dat een straddle carrier over slechts een rij containers kan rijden en dat een geautomatiseerde bovenloopkraan over meerdere rijen containers tegelijk kan bewegen. Met behulp van een simulatiestudie wordt inzicht verkregen in de routelengte en leegrijafstanden van straddle carriers en geautomatiseerde bovenloopkranen. Doordat een straddle carrier een langere afstand van de ene rij naar de andere rij moet afleggen, is de gemiddelde afstand die door een straddle carrier wordt afgelegd langer dan de afgelegde afstand van bovenloopkranen. Daarnaast is een gevoeligheidsanalyse uitgevoerd om de gevolgen voor leegrijafstanden te meten als veranderingen in aantallen containers, aantal rijen en aantal locaties in elke rij optreden. Door het aantal opdrachten te variëren blijkt, dat de afstand die leeg wordt afgelegd, afneemt ten opzichte van de totale afstand indien het aantal opdrachten toeneemt. Dit kan worden verklaard uit het feit dat opdrachten relatief dichter bij elkaar liggen indien meer containers worden ingeslagen en uitgeslagen.

Curriculum Vitae

Iris Vis was born in 1974 in Leidschendam, The Netherlands. She attended the secondary school 'Alfrink College' in Zoetermeer, The Netherlands, from 1986 to 1992. She studied mathematics, with specialisation Operations Research, at the University of Leiden. In 1997 she graduated on her Master's thesis 'Kasvoorraadbeheer bij aangesloten banken' (cash-balancing problems at banks). In december 1997 Iris Vis won the 'AXA Leven Scriptieprijs 1997' for the best Master's thesis on financial services. Next, she started as a Ph.D. candidate at the Faculteit Bedrijfskunde/Rotterdam School of Management of the Erasmus University Rotterdam. She performed research on planning and control concepts for material handling systems in nodes in supply chains. Iris Vis has given presentations and seminars on this subject in both Europe and North America. Several of her articles have appeared in scientific and professional journals and conference proceedings. She has been a visiting scholar for three months in 1999 at the School of Industrial and Systems Engineering at the Georgia Institute of Technology in Atlanta, United States of America. During her Ph.D. period, she has given a course in which students actively have searched for reductions in throughput times in real supply chains. Furthermore she has given computer courses on simulation. From May 1, 2002, she will be working as an assistant professor logistics at the faculty of Economics and Business Adminstration of the Vrije Universiteit Amsterdam.

ERASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT
ERIM Ph.D. SERIES
RESEARCH IN MANAGEMENT

Title: **Operational Control of Internal Transport**
Author: J. Robert van der Meer
Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. R. Dekker
Defended: September 28, 2000
Series number: 1
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-004-6

Title: **Quantitative Models for Reverse Logistics**
Author: Moritz Fleischmann
Promotor(es): Prof.dr.ir. J.A.E.E. van Nunen, Prof.dr.ir. R. Dekker, dr. R. Kuik
Defended: October 5, 2000
Series number: 2
Published: Lecture Notes in Economics and Mathematical Systems, Volume 501, 2001, Springer Verlag, Berlin
ISBN: 3540 417 117

Title: **Optimization Problems in Supply Chain Management**
Author: Dolores Romero Morales
Promotor(es): Prof.dr.ir. J.A.E.E. van Nunen, dr. H.E. Romeijn
Defended: October 12, 2000
Series number: 3
Published: ERIM Ph.D. series Research in Management
ISBN: 90-9014078-6

Title: **Layout and Routing Methods for Warehouses**
Author: Kees Jan Roodbergen
Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. J.A.E.E. van Nunen
Defended: May 10, 2001
Series number: 4
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-005-4

Title: **Rethinking Risk in International Financial Markets**
Author: Rachel Campbell
Promotor(es): Prof.dr. C.G. Koedijk
Defended: September 7, 2001
Series number: 5
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-008-9

Title: **Labour flexibility in China's companies: an empirical study**
Author: Yongping Chen
Promotor(es): Prof.dr. A. Buitendam, Prof.dr. B. Krug
Defended: October 4, 2001
Series number: 6
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-012-7

Title:**Strategic Issues Management: Implications for Corporate Performance**
Author: Pursey P.M.A.R. Heugens
Promotor(es): Prof.dr.ing. F.A.J. van den Bosch, Prof.dr. C.B.M. van Riel
Defended: October 19, 2001
Series number: 7
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-009-7

Title: **Beyond Generics; A closer look at Hybrid and Hierarchical Governance**
Author: Roland F. Speklé
Promotor(es): Prof. dr. M.A. van Hoepen RA
Defended: October 25, 2001
Series number: 8
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-011-9

Title: **Interorganizational Trust in Business to Business E-Commerce**
Author: Pauline Puvanasvari Ratnasingam
Promotor(es): Prof.dr. K. Kumar, Prof.dr. H.G. van Dissel
Defended: November 22, 2001
Series number: 9
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-017-8

Title: **Outsourcing, Supplier-relations and Internationalisation: Global Sourcing Strategy as a Chinese puzzle**
Author: Michael M. Mol
Promotor(es): Prof.dr. R.J.M. van Tulder
Defended: December 13, 2001
Series number: 10
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-014-3

Title: **The Business of Modularity and the Modularity of Business**
Author: Matthijs J.J. Wolters
Promotor(es): Prof.mr.dr. P.H.M. Vervest, Prof.dr.ir. H.W.G.M. van Heck
Defended: February 8, 2002
Series number: 11
Published: ERIM Ph.D. series Research in Management
ISBN: 90-5892-020-8

Planning and Control Concepts for Material Handling Systems

Planning and Control Concepts for Material Handling Systems discusses the planning and control of material handling systems in nodes in supply chains. Material handling systems consist of the equipment, personnel, information, materials and related planning and control systems. Automated Guided Vehicles take care of internal transport of loads. Various new planning concepts are developed to determine the number of vehicles required to transport all loads in time. Storage and Retrieval Machines take care of the storage and retrieval of loads in a storage area. A new dynamic programming algorithm is introduced to sequence storage and retrieval requests for a single machine in a number of parallel aisles with transfer stations at both ends of each aisle. Simulation studies are performed to test these new planning and control concepts within the context of a container terminal.

ERIM

The Erasmus Research Institute of Management (ERIM) is the research school in the field of management of Erasmus University Rotterdam. The founding partners of ERIM are the Rotterdam School of Management / Faculteit Bedrijfskunde and the Rotterdam 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 focuses 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 Ph.D. program in management.

ERIM ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

www.erim.eur.nl

ISBN 90-5892-021-6