PLANNING AND CONTROL OF AGVS IN AMRF DECISION HIERARCHY

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL
ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Haluk Yılmaz
September, 1993

PLANNING AND CONTROL OF AGVS IN AMRF DECISION HIERARCHY

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL
ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Haluk Yılmaz
September, 1993

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

Selim Akturk

Assist. Prof. M. Selim Aktürk (Principal Advisor)

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

Ömer S. Benli

Assoc. Prof. Ömer Benli

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

Assoc. Prof. Cemal Dinçer

Approved for the Institute of Engineering and Sciences:

M. Baray

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

PLANNING AND CONTROL OF AGVS IN AMRF DECISION HIERARCHY

Haluk Yılmaz
M.S. in Industrial Engineering
Supervisor: Assist. Prof. M. Selim Aktürk
September, 1993

Scheduling efforts made without considering the special limitations of the material handling system might lead to infeasible results. This problem especially becomes important when the Automated Guided Vehicles (AGV) are the main material handling media due to their inherent flexibility and adaptability that increase the scheduling complexity. In this thesis, an analytical model is proposed, first, to incorporate the AGV module into the overall decision making hierarchy. A mathematical formulation is developed to include interaction between the AGV module and other modules in the system by considering the restrictions of the material handling system. A micro-opportunistic approach is proposed to solve the AGV scheduling problem. Finally, the proposed method is compared with a number of dispatching rules.

Key words: Factory Reference Models, AGVS, Opportunistic Scheduling.

ÖZET

AMRF KARAR HİYERARŞİSİ İÇİNDE OTOMATİK GÜDÜMLÜ ARAÇLARIN PLANLANMASI VE KONTROLÜ

Haluk Yılmaz
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. M. Selim Aktürk
Eylül, 1993

Malzeme taşınmasına özel sınırlamaları gözönüne almayan çizelgeleme yaklaşımları olursuz sonuçlara yol açabilir. Bu problem, özellikle Otomatik Güdümlü Araçların (OGA) ana taşıma aracı olduğu durumlarda bu araçların esneklikleri ve uyumluluklarından ötürü önem kazanmaktadır. Bu tez çalışmasında, önce OGA modülünün karar verme hiyerarşisine katılması için analitik bir model önerilmiştir. Malzeme taşınması sistemi kısıtları gözönüne alınarak, OGA modülüyle diğer modüllerin ilişkilerini sisteme dahil etmek üzere bir matematiksel formulasyon geliştirilmiştir. OGA çizelgeleme problemini çözmek için bir mikro-oportünist yaklaşım önerilmektedir. Son olarak, önerilen metot diğer sezgisel metotlarla karşılaştırılmaktadır.

Anahtar sözcükler: Fabrika Bellik Modelleri, OGA Sistemleri, oportünist çizelgeleme.

To my family

ACKNOWLEDGEMENT

I would like to express my gratitude to Assist. Prof. Selim Aktürk due to his supervision, guidance, understanding and encouragement throughout the development of this thesis. I am also indebted to Assoc. Prof. Cemal Dinçer and Assoc. Prof. Ömer Benli for showing keen interest to the subject matter and accepting to read and review this thesis.

I would like to extend my deepest gratitude and thanks to my family for their morale support and encouragment.

I greatly appreciate my friends Orhan Dağlıoğlugil, Mehmet Özkan, Selçuk Avcı, Okan Balköse, İhsan Durusoy, Elif Görgülü, Oğuz Işıklı and Alper Erdoğan for their accompany and patience in any way during my studies.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Material Handling does not receive the attention it deserves in manufacturing systems. One important reason of this fact is the difficulty in placing the material handling modules in decision hierarchy. This causes ignorance of material handling in decision making and considerable productivity losses are incurred. Automated Guided Vehicle Systems (AGVS) have gained considerable popularity because of their flexibility and Automated Guided Vehicles (AGVs) have become the main transport media especially in Flexible Manufacturing Systems. However, their flexibility makes the problem of incorporation of AGV module to the decision hierarchy even more serious.

In this study, main objective is to overcome the problem of incorporating the AGV module to the decision making hierarchy. First, the reasons of this problem are identified. As a solution to the problem, a hierarchical model is proposed. The tasks of modules in this new hierarchical model are defined. Furthermore, the AGV module's scheduling problem is formulated as a mixed integer program. However, the computational time requirements suggest developing a heuristic method for this problem. Therefore, utilizing the successful ideas of recent scheduling literature, a heuristic algorithm is developed. The experimental analysis shows that the scheduling method is quite successful.

In the next chapter, the relevant literature is studied. First, the reasons of incorporation problem and its results are analyzed in detail in the context of factory reference models. Next, the current AGV scheduling literature is discussed. The hierarchical model proposed in this study requires solution of the AGV scheduling problem based on the AMRF decision hierarchy. In order to identify the difficulty of this problem, the next item in literature review is the Time-Constrained Vehicle Routing Problem (TCVRP), which is similar to AGV scheduling problem. Computational time requirements for this problem propose that a heuristic method should be developed for AGV scheduling. The last item in literature review discusses some of the recent successful ideas in scheduling that can be used in developing a heuristic algorithm. The third chapter discusses the hierarchical model that we are proposing for incorporation problem. The resultant scheduling problem for the AGV module is also defined and a mixed integer p rogramming formulation is presented in this chapter. In Chapter 4, a heuristic method is presented for AGV scheduling problem. In Chapter 5, an experimental analysis is made. The method is compared with alternative rules. Also an ANOVA model is prepared in order to search for the factors that might affect the performance of our method. Finally, concluding remarks and future research directions are given in Chapter 6.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The development of the model presented in this study is based on a wide range of results of previous studies. In this chapter, we will mention the relevant literature that was used in the study. Different topics that will be mentioned are:

1. Factory Reference Models
2. Decision Making Hierarchy
3. Problems Related with AGVs
4. AGV Scheduling
5. Vehicle Routing
6. Opportunistic Scheduling

These different areas are all used in different parts of the model. Before reviewing these areas in the literature, the next section briefly draws the picture of where these topics were used in the study, giving the motivation and the way they are linked.

2.2 Motivation

The order of the items listed to be discussed follows the real order of areas of literature studied. Factory Reference Models are studied first, because Hierarchical Planning Systems are very appropriate for the systems requiring multilevel decision making. The second item, Decision Making Hierarchy is important in the sense that problems related to inclusion of material handling in Hierarchical Planning Systems can be identified. This item will make the motivation of the model clearer.

In the third area, Problems Related With AGVs, the AGV settings and studies in the literature will be discussed. This is for the purpose of identifying the characteristics and attributes of an AGV system in practice. These characteristics are especially important in forming the structural settings for our model as well as determining the scope of the study. Also a classification of the studies in the literature is made. AGV scheduling approaches so far do not make use of vehicle routing literature. However, by our settings, AGV scheduling and vehicle routing problems are very closely related. Because of this reason, the fourth area in the review is Vehicle Routing. This item will be utilized to identify the complexity of our problem.

The last item in the review is Opportunistic Scheduling, which is important for the purpose of developing the solution algorithm to the problem defined in our model. In this part, some recent scheduling methods are studied, by stressing the motivations of these algorithms. These algorithms construct the motivation of the algorithm that will be proposed in our study. Meantime, we will adapt a recent heuristic sequencing rule to form an AGV dispatching rule.

2.3 Factory Reference Models

In literature, Hierarchical Planning Systems are studied by many authors. Especially in systems requiring multilevel decisions, Hierarchical Planning Systems have become very popular. Typically in a production system, there are a series of decisions to be made over time. These range, at least from low level decisions such as assignment of tasks and workers to machines and ordering of jobs; to high level decisions such as amount of hiring, layoff and over-times, amount to be produced in different product groups, setting due-dates, etc. The timing requirements of these decisions change significantly. The typical and intuitive approach is decomposing the system, having smaller problems to solve which consist, more or less, decisions of same time horizon. First the high level decisions are made and put in execution. The results and output of these decisions are used to make the low level decisions.

The high level decisions are aggregated and much of required information is not known with certainty yet, including future demands, job processing times, worker availability, machine availability (because of breakdowns) and raw material availability. What is more, related with aggregation, many details are ignored at this level, to be considered in lower level decisions. For example, setups are treated as if sequence independent and product groups are used instead of individual stock keeping units.

A hierarchical system uses separate mathematical models to make decisions at each level. Solution of a higher level model is used as the parameters or constraints for the model below. From a simplistic point of view, this is the way the system would be expected to work in itself. The main difference of Hierarchical Approach is the explicit emphasis on the linkages between the modules and designing all modules in the system simultaneously so that they fit well together [7]. Hierarchical planning approaches have two important advantages as also noted by Dempster et al. [7]:

- Reducing complexity

- Coping with uncertainty

One important point of hierarchical planning systems is that, being similar in structure, they are parallel to hierarchical organizations operationally. Making use of this similarity, like organizational structures, different architectures are found in literature for hierarchical planning.

In literature, hierarchical production planning systems are studied by many authors. Hax and Meal [14] are the one of the first that propose decomposition models in order to resolve the mathematical complexity of the problem. Architectural approaches are made after these studies such as Biemans & Vissers' DEC/Philips Control Hierarchy [3] and Jackson and Johnson's AMRF Decision Hierarchy [15]. These studies propose a number of levels, which are placed in a hierarchy like an organizational hierarchy tree structure. Each level has some tasks to complete and each level can only interface either with its supervisor or subordinate.

One of these studies is made by Biemans and Vissers [3]. In their study, they form a nine level control structure, which is called DEC/Philips Control Hierarchy. The task of each level is given in Table 2.1.

For the Work-cell Level, the authors have a single controller. They do not form manufacturing cells, actually, but claim that a single controller would be sufficient even if there were. They argue that even physically new cells do not necessarily require existence of work-cell controllers, giving the example of not necessarily appointing a director when a new building is erected to a company. According to their structure, the work-cell controller (typically there is only one) coordinates the exchange of parts among workstations and tells workstations with whom they have to exchange parts. They view the transport system as a specialized instance of a workstation that executes displacement operations [3]. However the resemblance they have put is not so much valid in this case from a functional point of view. The advantages of cellular manufacturing are being lost in some sense since they are not being used to achieve a spatial decomposition to simplify the problem even further.

LEVEL	FUNCTION
Company Controller	Selects target markets and profits
Factory Controller	Negotiates exchange of raw materials and products, predicts due-dates of products
Shop Controller	Controls decoupling stocks to be able to dispatch products at their actual or expected due-date
Work-cell Controller	Schedules when, where, which operations are executed on parts, or which parts are exchanged
Workstation Controller	Determines which physical modifications parts should undergo to realize an operation
Automation Module Controller	Determines required paths of joints which describe the state of equipment
Equipment Controller	Select values of control variables, which directly reflect physical parameters
Device Controller	Issues control signals so that control variables are servo-ed by physical parameters
Sensor or Actuator	

Table 2.1: Levels and tasks of DEC/Philips Control Hierarchy

The second architecture for hierarchical planning is developed at the National Institute of Standards and Technology in the USA, for the Automated Manufacturing Research Facility, (AMRF). Quite similar to Biemans and Visser's architecture, it consists of individual modules, which have limited size functionality and complexity. Each level decomposes input commands from its supervisor into procedures to be executed at that level and subcommands to be issued to one or more subordinate modules. From top to lower levels, this process continues, which transforms to very primitive actions at the lowest level to actuate shop floor equipment. In the opposite way, status feedback is provided to supervisors by their subordinates. This ensures good performance of adaptive real-time decision making as discussed by Jackson and Jones [15].

There are five levels in AMRF architecture:

- Facility

- Shop
- Cell
- Workstation
- Equipment

The responsibilities of these five levels can be explained briefly as follows:

Facility is the highest level. The activities of facility level can be grouped in three classes of functions:

- Manufacturing Engineering: This class includes Computer Aided Design (CAD), Group Technology classification and Process Planning.
- Information Management: Activities that provide user data interfaces to support administrative and business management functions are in this class.
- Production Management: This class includes generating long range schedules, identifying production resource requirements, determining the need for additional capital investments to meet production goals, determining excess production capacity, and summarizing quality performance data.

Shop is the second level in the hierarchy. This level is responsible for:

- Conducting the production and supporting jobs on the shop floor
- Allocating resources to those jobs

Shop floor level can be considered forming with two major component modules, which are Task Manager and Resource Manager. Task Manager is responsible for capacity planning, grouping orders into batches, assigning and releasing batch jobs to cells and tracking individual orders to completion. Resource

Manager is responsible for allocating production resources to individual cells, ordering new resources and managing repair of resources at hand.

Cell is the third level. Sequencing of batch of similar parts through workstations and supervising for support services such as material handling and calibration are duties of this level. Typically, there will be jobs that require the services of one or more workstations assigned to a cell, material handling being one of these workstations. Together with jobs, certain due-date and priority data will be given. The cell must sequence these jobs and develop a schedule of start and finish times of each job at each workstation. This includes material handling requirements as well. Of course, when conflicts or delays appear in workstations, cell must re-plan, reroute and reschedule to overcome these problems. The task of a cell is quite complex, and part of the difficulty is due to shared resources like material transport devices [15].

Workstation level is the level at which activities of small integrated groupings of shop-floor equipment are directed and coordinated. Typically, a workstation in AMRF consists of a robot, a machine tool, a material storage buffer and a control computer. The controller sequences equipment level subsystems through job setup, part fixturing, cutting processes, chip removal, in-process inspection, job take-down and cleanup operations.

Equipment level is the lowest level in the hierarchy. Equipment controllers translate workstation commands into a sequence of simple tasks for that equipment. Individual equipments can be from different vendors and therefore it may be required to use different 'languages' that each one can understand. Another task of equipment controller is to monitor the execution of the tasks which is translated by means of sensors in the hardware.

As would be valid for any hierarchical structure, the nature of the problems involved in each level changes from top level to down levels in AMRF Decision Hierarchy. These changes can be listed as [15]:

- Each level must sequence through the list of jobs assigned by its supervisor and develops a schedule of tasks for its subordinates.

- The number of problems to be solved and the frequency with which they must be resolved increases dramatically
- Time available to find solutions decreases significantly
- Information used to solve the problems becomes more abundant and deterministic

The AMRF Decision Hierarchy is quite similar to that proposed by Biemans and Vissers. One difference is in the number of Work-cell controllers in the decision hierarchy. As mentioned before, we do not view having a single Work-cell controller appropriate for the system. For the Decision Hierarchy in our study, AMRF Decision Hierarchy is selected.

Mathematically, from the scheduling point of view, the scheduling problem is generally considered once for the whole system, which is usually off-line. Jackson and Jones [15] propose that each level should be responsible for generating and maintaining its own schedule, quickly and only as needed. By nature of hierarchy, the constraints imposed by the upper level should be satisfied, which are in the form of priorities among jobs, and start and finish times for jobs. As an output, a schedule of jobs assigned by the supervisor should be given to the lower level.

Remark 1: Jackson and Jones' framework accounts to the following setting: From shop level, there are job orders given to cells and move orders given to material handling module. The jobs given to each cell have due-dates and priorities (or weights) as well as release times. These times are typically computed by taking an approximate processing time for the operations. If the parts to be processed will arrive from some other cell or required material will be delivered at a certain time, there may be known release times. The cell level has to identify the workstations that will process these jobs, sequence their operations and then schedule the operations of each job on the workstations including 'Material Handling Workstation'. For the material handling module, it has to complete the move orders between work stations in a cell, as well as move orders between cells. Material Handling module has to schedule all these tasks. The resultant schedule of the cell level will form the orders for the workstations. In the other way around, any conflicts in the schedule formed will be fed back to the shop level so that the shop level can re-schedule the jobs between cells.

2.4 Decision Making Hierarchy

In the previous subsection, we discussed the possible success of hierarchical structures in controlling manufacturing processes. In this second item, we will discuss the studies in the literature that highlight the difficulty of fitting material handling modules in any hierarchical architecture.

In literature, McGinnis has a study in which he first notes the fact that fitting material handling module in any hierarchical architecture is very difficult. He explores some of the reasons of this fact and suggests an approach to resolve the problem [21]. In his study, he gives AMRF and DEC/Philips control hierarchies as reference models. For a single cell, he takes Automated Guided Vehicle System (AGVS) as a very popular technology for moving material on the factory floor. In the control structure, he takes AGVS as a special workstation as was suggested by Jackson and Jones [15] and Biemans and Vissers [3]. As every workstation, AGVS must have a dedicated controller to assign load movements to vehicles. He then identifies the problems with representation of AGVS control systems by a pure hierarchy. The problems he identifies are as follows:

PROBLEM 1: AGVs typically move on uni-directional flow paths and, although there are other ways, generally these paths are divided into segments for management of vehicle traffic. If a segment is employed by a vehicle, no vehicle is allowed to enter that segment until first one gets out. Typically, AGVS controller assigns loads to individual vehicles. For a load movement, AGV has to pass more than one segment in general. Here the problem arises. In a strict hierarchy, segment vehicle relation cannot be modeled. If segments are subordinates of vehicles, viewing vehicles assigning themselves to segments, then a segment has to subordinate more than one vehicle. In the opposite view, if vehicles subordinate segments, same problem arises. Thus, in the AGVS workstation, there cannot be strict hierarchy.

PROBLEM 2: This problem is because of the flexibility of AGVs. AGVs can interface with different levels in the manufacturing system. Typically, AGVS is modeled as a special workstation in a cell. But an AGV may interface with some other cell, which is the way proposed by Solberg and Heim [27] model. This is, even if we view an AGVS as a workstation in a cell, quite logical. There is a material flow requirement also between cells and AGVs are the best candidates. If there is another AGV module within the shop, there will be no problem. But if, instead, the individual cell modules have the task to move material that is required for the cell, the problem arises. McGinnis notes even the possibility of interfacing of an AGV with a shop level [21]. He gives an electronic assembly plant as an example for this purpose [21]. The problem here is twofold:

Problem 2 A: One face of the problem is the ambiguity, related with the hierarchy. Who will be the 'peers' of the AGVS controller? Every shop, cell and workstation has its own controller, but AGV can interface with all of them. AGVS controller is typically a 'peer', that is at the level of a workstation controller. Thus, controller can only interface with its supervisor, which is the cell controller, and subordinates, which are individual vehicles. In order to have a direct interface with another cell or shop, they must have the same supervisor. Thus, there is a problem of where to place the AGVS controller.

Problem 2 B: The second face of the problem has to do with implementation issues. The messages of material movement requirements have to pass several levels of the hierarchy to arrive at the AGVS controller. This naturally creates delays in the system. McGinnis notes that although this delay was only a few seconds in a single load movement, it turned out to be quite significant for overall performance in the electronic system example, since there can be thousands of load movement requests [21].

McGinnis offers some modifications in the control hierarchy to solve these problems. His first modification is in the task decomposition function. He proposes that the two elements of this function, operation assignment (operation planning) and activation (operation execution) should be differentiated. Operation planning is placed in the hierarchy, but execution process is not. Any 'peer' can give the activation key for a job for which, the assignment was already made. In order to execute this assignment, both of these elements should be complete. In this way, the hierarchy problem can be solved, as well as decreasing the delays in the system.

The second modification offered by McGinnis is viewing the material handling system as a black box entity in the hierarchy. Thus, interfacing with different levels should not be conflicting to the hierarchy. But this can only be justified if the first modification is already made.

In this study, McGinnis notes the following as a research problem, which is important for our purposes:

> ...another research problem, which has attracted less attention is the problem of devising command arbitration schemes which can simultaneously optimize the material handling system and consider the urgency of specific move commands, so that the variance in execution times does not lead to much higher buffer requirements. [21]

In another study by Solberg and Heim [27], the authors first discuss some characteristics of manufacturing information. They, then, give four strategies for managing manufacturing information:

- Subsystem Optimization
- Total Integration
- Hierarchical Decomposition

- Heterarchical Decomposition

They then evaluate the performance of the heterarchical decomposition which eliminates the rigid supervisor/subordinate relationship in a hierarchical structure, and hierarchical strategies by a small batch manufacturing factory example. They make a modification and evaluate the adaptability of the strategy to this change. The original system is the one that has a single AGV system for a single shop (conflicting the general view of having one AGVS module for every cell). Then their change is addition of a new type of AGVs that will move small sized loads. As a result, the authors note similar problems identified by McGinnis for hierarchical decomposition strategy. Different layers have to be added to the hierarchy, as well as changes in the task assignment duties. The authors, identifying the strength and weaknesses of the heterarchical strategy, propose a hybrid system of these two systems and show that this system can handle such a change a lot easier. This study is important in the sense that difficulty of managing information related to material handling and adapting AGVS to the decision hierarchy is quite clear. That is the reason why their modification was in the AGVS module.

Remark 2: Placing AGVS modules in control structures is quite difficult and deserves high degree of attention.

Although not studied well in the literature, command schemes should be developed which simultaneously optimizes the material handling system and urgency of specific move commands. When the task of a cell module in AMRF hierarchy and Jackson and Jones' remark on scheduling [15] are considered, we can come up with the following remark:

Remark 3: Scheduling decisions for a shop should not be made independent of AGVS function, but the shop level should find a schedule for all the cells, including the schedule of AGVs.

2.5 Problems Related with AGVS

Automated Guided Vehicle is a driver-less machine that can be controlled by a system operator (which can also be a computer). First AGV was invented 35 years ago, which then was called 'driver-less systems'. Advances in electronics through years have led to advances in guided vehicles, giving more flexibility and capability. But the real factor in application spread is the market acceptance. Today, AGV is accepted as the standard material handling method for the Flexible Manufacturing Systems [17].

2.5.1 AGV Types and Functions

There are a number of different AGV types. These are:

- AGVS towing vehicles are the first type that was used and is still used extensively today. Towing vehicles can pull a range of trailer types with capacities from 3500 kgs to 25,000 kgs.
- AGVS unit-load vehicles are equipped with decks, which permit unit-load transportation and automatic load transfer.
- AGVS pallet trucks are designed to transport palletized loads to and from floor level, eliminating need for fixed load stands.
- AGVS fork truck is a relatively new type. This has the capability to service palletized loads both on the floor level and on fixed stands, and sometimes in a rack.
- Light-load AGVS are vehicles which have capacities of approximately 500 kgs. They are used to transport small parts, baskets, or other light loads through a small manufacturing environment, typically with a limited space.
- AGVS assembly-line vehicles are adaptations of light-load AGVS for applications involving serial-assembly processes.

With this wide range of application types, AGVs are preferred in many systems. Ray Kulwiec [29] lists the following advantages of AGVS:

- Less safety stocks, and high material control.
- More efficient use of personnel, since less operators are needed.
- Efficient work environment: AGVs allow independent loading and unloading at stations from operators.
- Flexibility: Routes can be changed, new ones can be added with great ease than other systems.
- Better use of floor space: No floor space is occupied permanently by AGVs.
- Adaptability to automation: AGVs can operate efficiently with other automated and computer controlled systems such as robots, AS/RS, conveyors, elevators, doors and automatic production machines.
- Integration within plant: AGVs can provide a link between cells in a plant which contributes to overall system integration.
- Adaptability to existing facilities: Little structural change and cost is required for constructing an AGV system to an existing plant.

Koff identifies the following technological functions that are essential in an AGVS application [17]:

1. Guidance is the way vehicles can change their directions. Guidance allows the vehicle to follow the predetermined route.
2. Routing is making the decisions in the path to follow from one point to another. There are two methods used when an AGV approaches a decision point and has to select one of the paths to go:
   (i) Frequency-select method: AGV receives one frequency for every path, selects appropriate one at decision points.

   (ii) Path switch method: When approaching a decision point, AGV activates a device which closes all but one of the path at the decision point, which is the correct one for AGV.

3. Traffic management is the way collisions with other vehicles are avoided. Of course, this is done by trying to minimize the traffic flow. Traffic management function is satisfied by the following ways:
   (i) Zone control: This is the most popular one. The layout is partitioned to segments or zones, and each zone is allowed to include at most one vehicle at any time.
   (ii) Forward sensing: The vehicle is capable of detecting the presence of another vehicle in front of it. These sensors are useless at corners.
   (iii) Combination control: The paths are separated into two and in one part, forward sensing is used. If the paths do not have so much corners, forward sensing is preferred because of less costs. In the other part, zone control is used.
4. Load transfer: is picking-and discharging of loads to and from vehicles.
5. AGVS system management is the method of controlling the system. This includes vehicle dispatching and system monitoring. Vehicle dispatching, that is selection of the vehicle to move the particular load can be accomplished in a number of different ways:
   (i) On-board dispatching
   (ii) Off-board call systems
   (iii) Remote terminal
   (iv) Control computer
   (v) Remote terminal control computer combination.

The areas of study about AGVs in the literature can be classified in parallel to AGV functions listed in the previous sub-section. However, for our purposes, we prefer to classify them on a different basis. Some issues related to AGVs are directly related to the installation (or re-design) of the system. Unless a major change in system settings, these decisions are made once for the system. We call these 'design problems'. Meantime, there are decisions to be made that are related to the working of AGVs in the installed system, which we call 'operational problems'. Now, with this classification, we have:

Design problems:

1. Determination of the number of vehicles required
2. Designing the flow path (and pick-up and drop-off points)
3. Determination of the routes to follow

Operational problems:

1. Vehicle dispatching
2. Traffic management

Design problems studied in the literature are discussed below. For the operational problems, since we are much more related to this type, we discuss this class in the next sub-section, AGV Scheduling.

2.5.2 Design Problems

As stated, these are related to decisions in the design stage. These are high level decisions that are usually made once in the implementation stage and rarely changed.

Number of Vehicles Required

Determination of the number of AGVs required is strongly related to the type of vehicles to be used. However this technical choice is not studied. Practically, the transport requirements, weight of parts, the particular manufacturing environment and the usage mode of the vehicles (only carriers vs. mobile work stations) are used to determine the type of vehicle to be used.

In determination of the number of vehicles required, the general practice is to study minimum number of vehicles required rather than optimal number of vehicles required. For the optimal number, one has to consider the time phased material pick-up and delivery requirements (the schedule of transport requests, due-dates, amounts to be transported, etc.), pick-up and drop-off area floor space capacity (and the speed of pick-up and drop-off operator as well), track congestion (shop or path blocking), and even the numbers of different types of vehicles. In addition to these factors, the cost of the vehicles has to be considered, since for a certain range, increasing number of vehicles results in increased performance even though the diminishing rate of return concept applies. As a result of these factors, the pay-back period concept should be considered. Thus it is very difficult, if possible, to determine the optimal number of vehicles.

Instead of optimal, the minimum number of vehicles required is studied by some simplifying assumptions. The following assumptions are usually made:

- static rather than dynamic system
- no floor space requirements considered
- no track congestion considered

In this respect, Maxwell and Muckstadt [20] gave a mixed integer programming formulation for determining the minimum number of vehicles required. This problem turns out to be a transportation problem. The objective in their study is to minimize the total vehicle time required for a shift. Number of vehicles is determined by dividing total vehicle time required for a shift to the capacity of a single vehicle in a shift (in hours). In their work, they also propose a dispatching procedure, and a way to measure the blocking time caused by congestion and size of shipping areas. In this way, they show that the minimum number of vehicles found by the simplified model above can be re-evaluated and result can be adjusted considering the assumptions.

In a similar work, Leung et al. [19] extend the work of Maxwell & Muckstadt to situations where there are different types of vehicles with different travelling speeds in the system. Their model is more complex as should be expected (not a transportation problem anymore) but they consider the same objective.

Flow Path Design

This problem involves determination of the aisles that will be included in the guide path of the AGV's. Ideally the flow paths for the AGVS should be determined together with the determination of the manufacturing layout. This accounts to designing the layout with performance of the AGVs' as one of the objectives. However, in practice, the general approach is either determining the guide path first, and the pick-up and drop-off points later, or taking the pick-up and drop-off points of departments as given and determining the guide path afterwards. In this respect, Gaskins and Tanchoco [11] give an integer programming formulation for determining the flow path for a given layout. Uni-directional flow is assumed and total loaded vehicle time is the objective to be minimized. In another study of the authors with Taghaboni [12], virtual flow paths (for free ranging AGVs) are determined. In this study, unloaded travels are also considered. Objective function is the sum of total distance traveled and total number of lanes. However, these are summed without weighting factors, which does not seem realistic and might cause some problems due to the different unit of measures. Their model makes use of multiple commodity flow problem. Both of the models above are quite hard to formulate and solve especially for relatively large layouts. The authors admit this fact and note presence of the simulation alternative. The particular case for which these models can be justified are flexible systems, where the flow intensity data change quite often so that only the flow rate parameters are changed in the formulation. However, a new formulation has to be made each time the layout is changed. The most difficult part of the models above is the formulation part. One more disadvantage of the two models in common is that neither of them considers the traffic or blocking aspects of the problem. Though it is quite hard to consider these in an analytical study, this can be a reason for preference of simulation study since it considers both. The recommendation about these models is that simulation should be used after the model is solved analytically by the appropriate model.

The difficulty associated with the analytical model for flow path design problem, together with some other reasons have led to different approaches in flow path design. One is designing flow paths in the form of a single loop. Single loops have a number of advantages. First, they are very simple and require very little control. Furthermore, other problems associated with the AGVS such as dispatching and traffic management are facilitated considerably when a simple loop is used as the guide path. Especially for small layouts consisting of up to six departments, single loop paths are easily used. In this respect, Bartholdi and Platzman [2], Tanchoco and Sienrich [30], and Bozer and Srinivasan [5] have studied single loop guide paths. In the study of Bozer and Srinivasan, the authors propose placing a number of disjoint simple loops instead of a traditional guide path so that the benefits of single loop is used in larger layouts. In every loop, there is a single AGV operating. In this way, traffic management problem is completely eliminated and dispatching problem is very much facilitated. Total travel distance in the system is decreased by means of many loops as well, which is an advantage against a single loop. However, the potential disadvantage of the model is that for a particular request, more than one vehicle in turn need to be used and if a single AGV fails, many requests cannot be met. What is more, the load of the AGVs is hardly balanced, since each is dedicated to operate in an isolated loop.

Route Planning

Route Planning is, in general, made after the guide path is determined. In fact, for an efficient AGVS, route planning has to be considered together with guide path designing problem. At least in determining the flow in uni-directional flow systems, the length of the resultant route depends upon the direction. However, in practice, route planning is made after the guide paths are determined. In this respect, the routes are shortest paths between the pick-up and drop-off points.

Although there is no such study, one area where the route planning can be used is identifying alternative routes. This issue is related to the traffic management task. When a particular segment of a path cannot be used for a particular amount of time because of traffic intensity, an analysis can be made comparing the time required for the alternative route(s) with the time that the present route cannot be used. In this way, considerable savings can be gained.

Remark 4: We have to define the settings in our system related with AGVs. The AGV application type in our study is a unit-load application type. The technological functions that Koff has identified, namely guidance, routing, traffic management, load transfer and system management are not significant for our purposes. Related to design problems, we assume that the AGV System already exists. We will assume that number of vehicles, the layout with pick-up and drop-off spurs and the flow path are already specified. We also assume that the flow path is uni-directional. In fact, uni-directional flow path is quite wide spread as mentioned above, and most of the design studies propose uni-directional flow for many purposes [2][5][11][12][21][30].

2.5.3 Operational Problems

These are related to the operation of the system. These decisions can be changed according to the dynamic situation of the system. For instance, the vehicle dispatching rule can be changed according to capacity of AGVs and the urgency of certain jobs in the system. These decisions are generally made on lower levels.

Vehicle Dispatching

Vehicle dispatching is simply assigning a vehicle to a job (or assigning a job to a vehicle). For reasons related to the general settings in the system, no off-line schedule is made. The schedule for each cell and the shop is made ignoring the material handling requirements. The AGV module is expected to move the loads from desired points to certain destinations, but no prior information is given to the module. This makes any off-line scheduling mechanism impossible. All the AGV module can do is try to complete the move orders on time, if impossible, giving priorities towards the objective of the system. These objectives might be meeting the due-dates as much as possible, avoiding shop blocking, increase the throughput, etc. Thus, basically, the task of AGV module in classical settings is choosing the vehicle dispatching strategy, which may be a single rule or combination of rules according to the state of the system.

The decision on the selection of two possibilities for dispatching, namely assigning jobs to vehicles or vehicles to jobs mainly depends on load of the system. Logically, if there are more vehicles available than the number of jobs waiting then vehicles should be assigned to jobs. This idea is also shown empirically by Egbelu and Tanchoco [8]. However, one should expect that in a typical AGV system, there are more jobs waiting than vehicles available in the average. Sufficiently large number of vehicles for the opposite can be hardly justified because of the high investment cost of an AGV. Actually, most of the dispatching rules studied in the literature are based on selecting jobs for vehicles, that is vehicle based rules.

The studies in the literature about the dispatching rules explore the performance of certain rules in certain settings in terms of different objectives. The well known vehicle-based dispatching rules are listed as follows by Bozer [4]:

1. Random Workstation Selection (RAND): The next load to be moved is selected randomly.
2. First Come First Served (FCFS): All the move requests are placed on a

first-in-first-out queue and the first job in this list is selected as the next load.

3. Modified First Come First Served (MODF): A variation of FCFS. In this rule, any pick-up station cannot have more than one outstanding request, that is other than first request of a point is not taken into consideration.
4. First Encountered First Served (FEFS): The vehicle follows a pre-specified route as it gets empty. Regardless of the number and time of outstanding move requests, it takes the first load on its way that it encounters.
5. Shortest Travel Time First (STTF): The next load is the one that is nearest in distance (or time) to the point where vehicle drops its previous load. Also known as vehicle-looks-for-work (VLFW).
6. Longest Travel Time First (LTTF): The next load is the one that is furthest in distance (or time) to the point where vehicle drops its previous load.
7. Maximum Outgoing Queue Size (MAXQ): The next load to be moved belongs to the pick-up station which has the maximum number of move requests.
8. Minimum Remaining Queue Space (MINQ): The next load to be moved belongs to the pick-up station which has the minimum remaining space for possible request arrivals.
9. Load Shop Arrival Time (LSAT): The next load to be moved is selected according to the time each load originally entered the shop. A load that has been in the shop for the longest time (that is earliest arrival time) is to be moved next.
10. Most Desirable Station First (MOST): The next load belongs to the station that has the largest weight. Weight is a weighted combination of mean arrival rate of move requests, proximity to current AGV location, and whether another move request exists at the delivery point of the requests. This calculation is made continuously and the previous assignment are changed if a higher weight is met.

The well known job based rules, on the other hand, are listed as follows by Egbelu & Tanchoco [8]:

1. Random Vehicle (RV) rule: The vehicle is selected randomly for the current job.
2. Nearest Vehicle (NV) rule: The nearest vehicle to the current job is selected.
3. Farthest Vehicle (FV) rule: The furthest vehicle to the current job is selected.
4. Longest Idle Vehicle (LIV) rule: The vehicle that has been idle longest among the idle ones is selected.
5. Least Utilized Vehicle (LUV) rule: The vehicle with smallest utilization is selected, so as to balance workload.

There are a number of studies evaluating the performance of the rules listed above. In their study, Egbelu & Tanchoco [8] analyzed the effect of different vehicle initiated and work center initiated combinations on throughput of the system. Their first result is that work center based priority rules are insignificant. Next, with infinite queue capacities, they found that modified first come first served (MFCFS) rule results in maximum throughput in the system. They conclude their study noting that the performance of dispatching rules, especially distance based ones are effected from the layout considerably. King et al. [16] compare, for a single AGV system, STTF and MOST. They found that MOST results in shorter queue lengths. As can be noticed, it is hard to identify a 'best' dispatching rule for a number of reasons. The main reason behind this fact is the difference in experimental conditions. These studies are generally empirical and for fixed layouts. Number of departments, location of the pick-up and drop-off points effect the results considerably. What is more, the workload of the system is also very important, although its effect is not noted. In relatively more loaded cases, rules minimizing unloaded travel are more likely to perform better, whereas in other cases, rules considering high priority jobs may work well. Yet, the performance measures in these studies are different and success of rules depend on the performance measure very much.

Although there are many dispatching rules studied in the literature, a general gap is easily noticed. The dispatching rules are adaptations of classical dispatching rules, most of them are either job based or resource based. However, as will be discussed in the next section, Opportunistic Scheduling, new directions in scheduling is joining the two perspectives, job based and resource based approaches in a single method. Such a method is more likely to be robust and its performance will less likely to be case dependent.

Traffic Management

Traffic Management is scheduling the vehicles so that collisions are avoided and shop blocking is minimized. For traffic management, the procedure to be used should re-schedule any vehicle that has to enter a segment or node that will be occupied by a previously assigned vehicle, if vehicles are assigned one by one. Generally, traffic management is viewed only as a technological function. By means of sensors or zone control, the collisions are prevented. However, some off-line mechanisms can be constructed so that the waiting times resulting from traffic congestion are decreased. One way is providing an alternative route for the vehicle to follow to reach its destination (as mentioned in Route Planning, alternative routes should be considered). In scheduling, there are two similar works in this area by Walker et al. [32], and Taghaboni and Tanchoco [31]. Traffic management is related to other problems as well. For instance, if the guide path is a virtual guide path, where there are more than one lane between nodes, the traffic management process needs fewer controls and is simpler. In bi-directional flow case, the waiting time should be larger because control of segments is necessary, whereas, in a uni-directional case, control of nodes is sufficient. If loaded and unloaded travel speeds are not significantly different and pick-up and drop-off points are outside the flow path on a uni-directional layout, the importance of traffic management becomes ignorable. Actually, since most of the layouts studied in the literature are uni-directional, ignoring traffic problems even in an off-line scheduling method becomes reasonable.

As stated before, the AGV studies can be grouped in two classes: Design and Operational problems of AGVS. Much of the current studies are related to design part of the problem. Only few studies are made about the operational part and the studies are narrow in the sense that all assume similar settings. As noted, the studies assume that the move orders for AGVs will not be known a priori, and not deterministic. With a quite myopic point of view, material handling part of the overall system is under-emphasized. The operational decisions in the system are made with no feedback from the Material Handling subsystem. The orders are forwarded to Material Handling "module". The studies try to identify well responding dispatching rules for these settings so that certain objectives are aimed to be improved. Thus material handling is, to an extent, perceived as an independent manufacturing subsystem, having a little impact on the performance of the overall system, which might conflict with the observation that Material Handling makes 30-70% of the total manufacturing cost [29].

2.6 Vehicle Routing

In literature, there is a vast amount of studies on vehicle routing. Although none of the AGVS research deals with this problem or any solution approach, this problem is important for our study.

Vehicle Routing Problem (VRP) is the problem of serving a set of clients with known demands by a fixed fleet of vehicles of limited capacity. The objective is to minimize the total route length by serving each client exactly once. VRP has received much attention in literature.

Time-Constrained Vehicle Routing Problem (TCVRP) [18] or Vehicle Routing and Scheduling Problem with Time Windows (VRSPTW) [28], is a generalization of VRP. In this problem, each client has a time window in which he must be served. Time windows are very common in practice; such as bank delivery problems, industrial refuse collection problems, dial-a-ride problems, and school bus routing and scheduling problems. However, this problem has received very little attention compared to VRP.

VRP itself is NP-hard and by restriction, VRSPTW is also NP-hard. Savelsbergh has shown that even finding a feasible solution to this problem is NP-hard [28]. Thus, from a computational complexity perspective, these problems are quite difficult [28]. For optimal solution to the problem, there are a number of studies using special structures. Even these studies require large computational times. The first optimization method for the problem is developed by Kolen et al. [18] and requires more than two minutes of CPU time (at Vax 11/785) for a problem with 15 clients. Thus, practically, this problem cannot be solved optimally for a large number of clients.

Because of the computational requirements, many heuristic methods have been proposed for VRPSTW. These can be classified as follows:

- Savings Heuristics
- (Time-oriented) Nearest Neighbor Heuristics
- Insertion Heuristics
- Time-oriented Sweep Heuristic

Solomon [28] has made a computational study to compare the performance of these heuristics. He found out that an insertion heuristic performed best. The 100 customer problem took from 2.4 seconds to 24.7 seconds in experiments and the insertion heuristic took 24.7 seconds on average (on DEC-10).

Remark 5: The current AGV scheduling methods are only based on a set of dispatching rules as discussed in previous section. However, if the move requests can be brought in a deterministic form, an off-line schedule can be made. The AGV scheduling problem is very similar in many respects to VRSPTW. We can utilize the literature on vehicle routing to show the computational complexity of the AGV scheduling problem with time windows.

2.7 Opportunistic Scheduling

In scheduling literature, scheduling based on bottlenecks, which is called bottleneck approach, has been started with OPT approach [13]. OPT decomposes scheduling to three basic strategies:

1. Determine the bottleneck resource
2. Schedule to use the bottleneck resource most efficiently
3. Schedule the remainder of the resources up to the bottleneck

Goldratt's idea of scheduling on the bottlenecks has become quite popular. One extension is the shifting bottleneck algorithm by Adams et al. [1]. The idea of shifting bottleneck method comes from nonlinear programming. In this approach, the sequences on all machines but one is held fixed and for each machine a solution is found. The idea is that the overall solution is one of or at least close to one of these solutions.

One important advance in scheduling literature is combining the two general perspectives in one method. The first perspective is the job based one, which tries to meet the due-dates of jobs. The idea of bottlenecks has brought the resource based perspective, which tries to maximize the utilization of resources so that the increased flow in bottleneck resources increases the overall system performance. These two perspectives are, in general, conflicting. The term called bottleneck dynamics stems from considering these two views simultaneously. In addition to costs related to completion times of jobs (i.e. penalty costs), there is an opportunity cost of using the resource, since it may become bottleneck. There are several heuristic methods developed that consider these two costs simultaneously and among them, Rachamadagu & Morton's (RM) [22] sequencing rule is found to be very successful. RM heuristic evaluates the priority of each job by simultaneously considering the time left to its due-date and amount of time that job will occupy in the resource:

$$\pi_j = (w_j/p_j) * exp(-S_j/(k * p_{av}))$$

where π_j, w_j and p_j are the priority, weight and processing times of job j, respectively, S_j is the slack time (time between due-date and possible completion, which can also be negative, corresponding to amount of tardiness) of job j, k is a constant and p_{av} is the average processing times of candidate jobs. A single priority is calculated for each job and the job having largest priority value is selected as the next job to be processed. Dynamism stems from the ability to select only one job at a time and evaluating other jobs once more at each selection. The time dimension of bottlenecks refers to the idea that a certain resource need not be a bottleneck for the whole planning horizon, advantages of a job based perspective can be used for those periods that the resource is not a bottleneck. What is more, scheduling a bottleneck resource for a certain time period may create other bottlenecks for other periods. Thus, switching from one perspective to the other is likely to result in better schedules. This ability to switch from one perspective to another is called Opportunistic Scheduling. Well known OPIS methods are first examples of opportunistic scheduling [26]. In the earlier approaches, ability to switch between perspectives is limited and large subproblems are solved before the decision to switch. These methods are called Macro-opportunistic approaches. Recently Sadeh has developed a Micro-opportunistic approach, which decides on the perspective after assignment of every operation [25].

Dynamic view of bottlenecks is used by other authors as well. Recently, Muscattola [23] has developed *Conflict Partition Scheduling* concept, in which he repeatedly identifies the bottleneck conflicts and instead of solving it, he imposes constraints into the problem, which decreases the search effort and the criticality of the bottleneck is only decreased.

Experiments in all studies show that methods that base on the dynamism of bottlenecks concept are superior in many performance measures [9][25][23].

Remark 6: In our model, we come up with a scheduling problem. However, none of the successful methods mentioned above can be applied directly. In our proposed algorithm, we make use of ideas from RM dispatching rule, micro-opportunistic scheduling and Conflict Partition Scheduling to develop an off-line scheduler for AGVS.

2.8 Conclusion

In this chapter, we discussed the relevant literature that might be utilized in our model development. We had a number of remarks after each area that determines the main lines of our study. From the first two areas in the review, we realize that material handling module is excluded from shop level and cell level decision making directly. We identified the reasons of this fact as the difficulties related to hierarchical structure. Ideally, scheduling decisions of the shop and cell level should not be made independent of the material handling modules, taking movement requirements directly into account. In this respect, there is no study in the literature that brings a hierarchical model in which material handling decisions are included to ordinary scheduling decisions. This is the main motivation of this study. We aim to develop a model addressing this necessity.

In the third area of the review, we first defined the limits of our study with respect to current literature on AGVs. Namely, this study deals with the operational part of AGV literature. The layout, pick-up and drop-off points of departments, the flow-path, and the number of vehicles are assumed to be known. The guide path is uni-directional. The part of the problem we are dealing with is the ordinary task of the AGV module; scheduling of jobs to vehicles justifying the traffic feasibility. For this part of the problem in the literature, it was noted that current literature dealt only with on-line scheduling, and only in the form of adaptations of well known priority rules. The fact that there is no off-line scheduling method studied is based on the inability to include these decisions in scheduling functions. A major gap in the literature is that there is no dispatching rule that is paralel to new research in scheduling. The recent successful priority rules may be used in AGV dispatching as well, so that methods that are successful in all cases are developed. We will adapt the bottleneck dynamics concept to AGV scheduling problem.

In the fourth area, we studied vehicle routing problem in the literature. Although this problem has many application areas, there is no study that uses vehicle routing methods in AGV scheduling. Of course this is, again, related to inability to make off-line schedules for AGV module. If off-line scheduling were being made, vehicle routing methods could be used. The review showed that the main objective in the vehicle routing with time windows problem is minimization of total distance traveled. In our overall conjecture, this objective is not appropriate for our study. However, it should be noted that these problems are very difficult and that independent of the objective, even finding a feasible schedule is NP hard and optimum seeking techniques required very large computation times. Therefore, we are opt to use a heuristic method for our problem.

The last item in our review is opportunistic scheduling. Upon the necessity to use a heuristic method for the off-line scheduling problem, the new directions in job shop scheduling are reviewed. This part of the review showed that there are very good approaches that can be used in developing an off-line scheduling method as well as for adapting an on-line dispatching rule that will consider job criticality and utilization of vehicles simultaneously, showing robust performance in very different settings.

In the next chapter, a hierarchical model will be proposed and analyzed in detail, which also discusses the inclusion of AGV module to the decision making hierarchy.

Chapter 3

PROBLEM FORMULATION

3.1 Introduction

In Chapter 2, we reviewed the literature and made a number of remarks which will help us in developing our model.

In the previous chapter, the reasons behind the difficulty of placing AGVS Module in the decision making hierarchy are noted first. Then, the results of this exclusion in the hierarchical systems are discussed. Inability to include AGVS module decreases the performance of the overall scheduling system, since degree of stochasticity is relatively high (due to not considering the specific limitations of the AGVs). Furthermore, material handling cost due to AGV module is larger because effective scheduling cannot be made. In this chapter, we present a hierarchical model so that these deficiencies related to AGV model can be overcome.

In the literature review, one problem related to AGVS was identified to be the disaggrement on the level of AGVS. In our model, AGVS module can interface with the shop and cell levels as simultaneously. In the next section, this model is discussed in detail. The AGV module formed in this model deals with the scheduling tasks related to AGVS module. The off-line scheduling problem that AGV module faces is stated in section 3.3, which is formulated as a mixed integer program. The typical size of this problem and computational requirements are discussed in section 3.3, as well. The large size of the problem suggests that a heuristic method should be used, which will be discussed in the next chapter. Finally, this chapter ends with concluding remarks.

3.2 The Scheduling Model

The two important deficiencies of placing the AGV module into the hierarchy directly (either as a cell or as a workstation) are noted as the large amount of information flow across levels, and the disagreement in the level. AGV module receives move orders from both shop and cell levels. To overcome both of these difficulties, we propose that AGV module is treated as a special module that interfaces with both the cell and shop levels. Essentially, AGV module appears at the cell level, however, it can interface with all the cells belonging to the shop directly (as if a workstation of each cell). This model can be structured as in Figure 3.1.

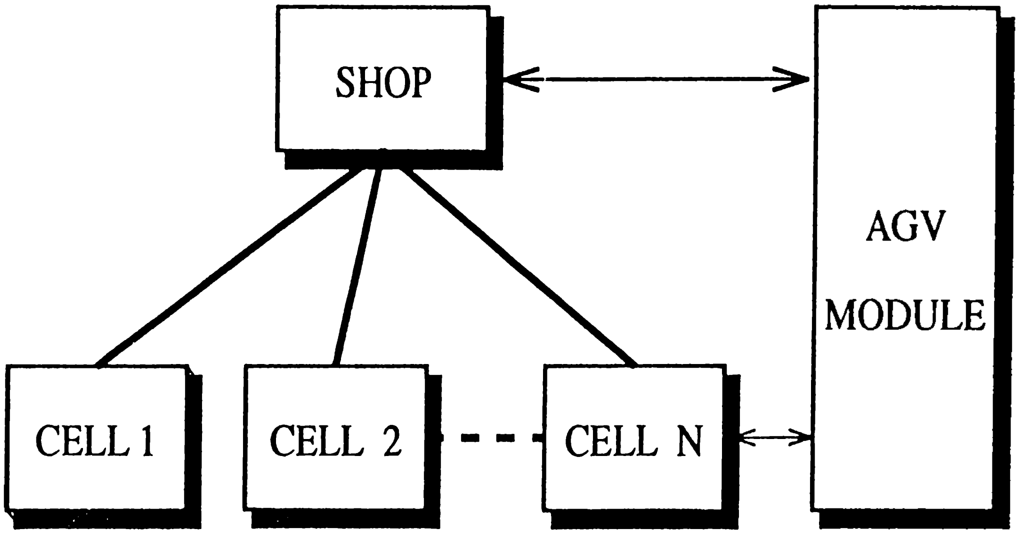

Figure 3.1: Proposed Hybrid Model

In this framework, the AGV scheduling problem can be included to the overall shop scheduling problem. Of course the overall problem is not solved by the shop level, which would be practically very hard. The AGV module is treated as if a 'black box', so that instead of trying to solve the overall problem, the problem is decomposed. Each module solves its own problem. However, in order to have a feasible solution for the overall problem, the linkages between modules are very important. Figure 3.2 shows the functional relationship between modules in the hybrid model. The tasks of each level in the model is analyzed in the following three subsections.

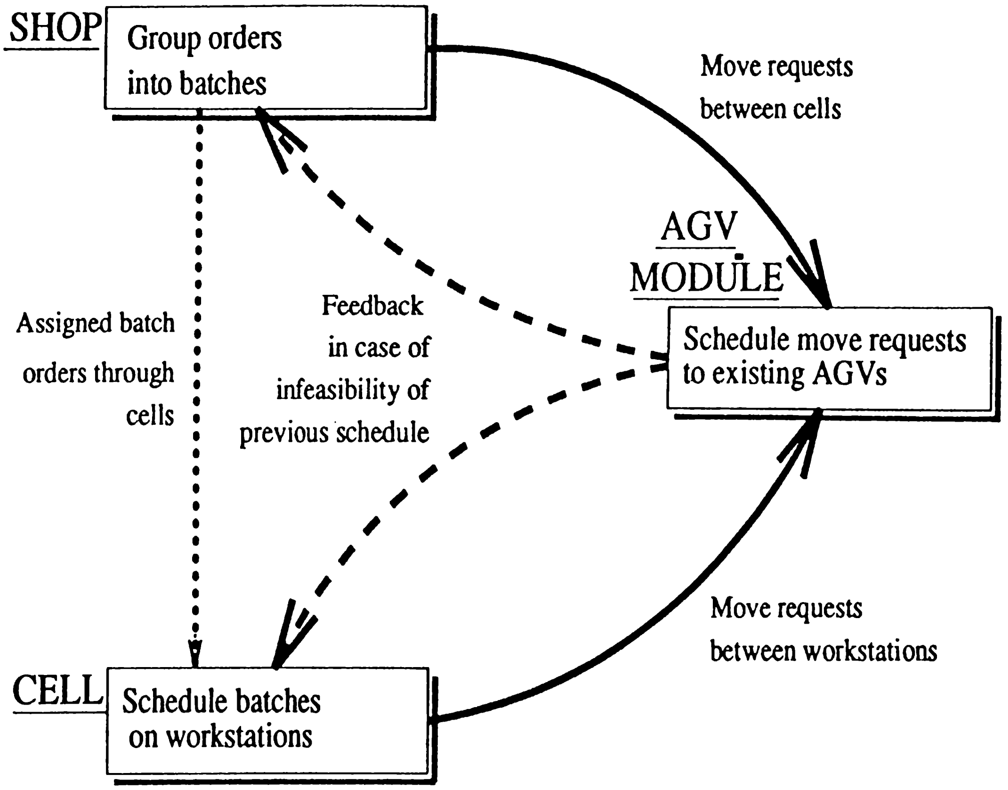

Figure 3.2: Functional Relations in Hybrid Model

3.2.1 Shop Level

The shop level is responsible for grouping orders to batches and assigning batches to cells. Depending upon the feedback from cell levels, shop level can change its assignment or batching decisions. With our model, before assigning batches to cells, shop level has to include transport requirements into its problem, which can be done as follows. When comparing a schedule, the tasks of each cell is determined as well as the transport requirements between cells. These transport orders are typically in the form of a release time and a due-date for each move request. Meantime, cells receive the orders from the shop. Each cell makes its own schedule and determines the move orders between its workstations. Meantime, AGV module, having collected all the move orders in the shop, some between the cells (received from shop) and others between workstations (received from cells), tries to make its own schedule, i.e. schedules the move orders on the existing vehicles. The schedule of each cell and AGV module is conducted to shop level. Shop level is responsible for the overall feasibility of these schedules. In case of any infeasibility, it is shop level's task to resolve the conflicts by alternative assignments and schedules. Thus only difference for the shop level is preparing a more accurate schedule for the whole shop, since in the absence of material handling decisions, more slack times were being inserted for material movement times. The scheduling problem to be solved is not more complex, but for the planning horizon, it has to be solved more than once, if a conflict arises.

3.2.2 Cell Level

The Cell level is responsible for scheduling the jobs to workstations. In our model, the only difference is that the cells conduct the move requests before the start of the planning horizon. With some approximate time requirements for material movement, each cell prepares an initial schedule. Similar to those of shop level, a release time and due-date for each move order is determined. The AGV module determines its own schedule after all the move requests are received. In case of infeasibility in some of the move orders, signals are given to the related cells, so that cells can revise their schedules, and conduct to AGV module again. These bring no additional complexity to the cell scheduling problem, except, instead of having a single pass, more than one passes are performed, if need arises.

3.2.3 AGV Module

With the new task definitions of cell and shop levels, the AGV module can make an off-line schedule since all the move orders are known. AGV module receives job orders between cells and within cells in the form of time windows, in which the corresponding move request has to be completed. These move requests represent an overall schedule for the shop. AGV module should try to find a feasible schedule so that the overall schedule stays feasible. However, if there is no feasible schedule, then the AGV module should form a new schedule such that related modules can revise their schedules with as little effort as possible. In this respect, the objective of the AGV module's scheduling problem should be minimum amount of deviation from the given time windows. There are two possible deviations that should be penalized. One is earliness, that is requiring an earlier start time of time window. This corresponds to earlier release of the load to be moved at the point of pick-up. On the other hand, the other deviation is tardiness, which is a later ending time for the window. This corresponds to a later delivery to the load's drop-off point.

In the next section, the problem that AGV module has to solve will be defined and formulated.

3.3 AGV Scheduling Problem

The AGV module takes the release times of loads at certain points (corresponding to completion times of processes) and due-dates of these loads at some other points (corresponding to starting times of next process). The output of the module will be a schedule of these move orders on vehicles. But the release times and due-dates may not be satisfied due to additional constraints imposed by the AGVS. The infeasible move orders are fed back to the related levels.

The physical settings of the cell are identified in Remark 4 in page 22. We have a known layout for the system, with spurs for pick-up and delivery for each work station. The guide path is assumed to be uni-directional. There is a fixed number of vehicles in the shop. The objective is finding a schedule that minimizes total deviation, since both earliness and tardiness account for deviation from the best schedule, which is the only objective. Secondary objectives such as minimization of total distance travelled should be effective only in the case of a feasible schedule. Since the number of vehicles is fixed, the cost of additional distance travelled corresponds to the variable operating cost. In fact, these two objectives are closely related. Thus, secondary objectives are ignored. However, for feedback purposes, the utilization levels of vehicles should be conveyed to shop level so that the need for extra vehicles can be identified for long range investment decisions.

The assumptions made in our study can be listed as follows:

1. There are N (fixed) move requests (corresponding to a planning horizon) and M (fixed) vehicles. The workstations that the loads will be taken from and to, the release times of loads at pick-up points and due-dates of loads at delivery points are known.
2. The layout of the cell with the pick-up and delivery spurs of each workstation is known.
3. The uni-directional guide path is known.
4. The loads are unit loads, therefore each request requires a devoted vehicle and one vehicle is sufficient for a load request.
5. At joints of the guide path, very small segments are used for traffic management so that time losses at the intersection points can be ignored.
6. All vehicles are identical in speed and capacity.
7. Earliness and tardiness in the schedule have equal penalty even though they can be differentiated.
8. All jobs are of the same weight

9. The vehicles can be anywhere desired at time zero.

Now that we have defined the problem to be solved by the AGV module, we can present the mathematical formulation of the model in the next section.

3.4 Mathematical Model

The problem defined in Section 3.2 can be modeled as a mixed integer program as follows:

$$MIN \quad \sum_{i=1}^{N}(T_i + E_i)$$

$$S.T. \quad \sum_{j=1}^{M}\sum_{k=1}^{L} X_{i,j,k} = 1 \qquad \forall i = 1,..,N \qquad (1)$$

$$\sum_{i=1}^{N} X_{i,j,k} \leq 1 \qquad \begin{array}{l}\forall j = 1,..,M\\ \forall k = 1,..,L\end{array} \qquad (2)$$

$$S_{i,j,k} + E_{i,j,k} \geq R_i.X_{i,j,k} \qquad \begin{array}{l}\forall i = 1,..N\\ \forall j = 1,..,M\\ \forall k = 1,..,L\end{array} \qquad (3)$$

$$F_{i,j,k} - S_{i,j,k} \geq t_1(i).X_{i,j,k} \qquad \begin{array}{l}\forall i = 1,..,N\\ \forall j = 1,..,M\\ \forall k = 1,..,L\end{array} \qquad (4)$$

$$S_{i,j,k} \geq F_{l,j,k-1} + Y_{l,i}.t_2(l,i) \qquad \begin{array}{l}\forall i,l = 1,..N\\ \forall j = 1,..M\\ \forall k = 1,..,L\end{array} \qquad (5)$$

$$Y_{l,i} \geq X_{l,j,k-1} + X_{i,j,k} - 1 \qquad \begin{array}{l}\forall i,l = 1,..,N\\ \forall j = 1,..,M\\ \forall k = 1,..,L\end{array} \qquad (6)$$

$$F_{i,j,k} - T_{i,j,k} \leq D_i.X_{i,j,k} \qquad \begin{array}{l}\forall i = 1,..,N\\ \forall j = 1,..,M\\ \forall k = 1,..,L\end{array} \qquad (7)$$

$$T_i = \sum_{j=1}^{M}\sum_{k=1}^{L} T_{i,j,k} \qquad \forall i = 1,..,N \qquad (8)$$

$$E_i = \sum_{j=1}^{M}\sum_{k=1}^{L} E_{i,j,k} \qquad \forall i = 1,..,N \qquad (9)$$

where the parameters are

R_i = release time of job i;

D_i = due-date of job i;

$t_1(i)$ = time required for moving the i^{th} load from its pick-up point to drop-of point;

$t_2(l,i)$ = time required for moving from the drop-off point of job l to pick-up point of job i;

M = number of vehicles;

N = number of jobs to be assigned.

L = a sufficiently large number, which limits the number of jobs that a vehicle can be assigned to.

L can be selected as N for confidence. If one wants to limit number of jobs allowed for a vehicle, that limit can be taken as L.

The decision variables are:

$$X_{i,j,k} = \begin{cases} 1, & \text{if job } i \text{ is processed as the } k^{th} \text{ job of vehicle } j \\ 0, & \text{otherwise} \end{cases}$$

$$Y_{i,l} = \begin{cases} 1, & \text{if job } l \text{ is processed by the same vehicle after job } i \\ 0, & \text{otherwise} \end{cases}$$

$S_{i,j,k}$ = starting time of job i processing by vehicle j as its k^{th} job

$E_{i,j,k}$ = earliness associated with starting of job i by vehicle j as its k^{th} job

$F_{i,j,k}$ = finishing time of processing job i by vehicle j as its k^{th} job

$T_{i,j,k}$ = tardiness associated with completion of job i by vehicle j as its k^{th} job

E_i = earliness of job i

T_i = tardiness of job i

The objective is earliness plus tardiness summed over all the jobs. This corresponds to total absolute deviation from the overall schedule. Clearly, a non-positive objective function corresponds to a feasible schedule. First set of constraints ensure that one and only one vehicle is assigned for each job. Second set of constraints ensure that any order of a particular vehicle is not assigned to more than one job. Third constraint set is the definition of $E_{i,j,k}$. Fourth set is offsetting the loaded travel times on the basis of jobs. Fifth set is offsetting the unloaded travel time on the basis of vehicles. Sixth constraint set is the definition of $Y_{i,l}$, which shows whether the two jobs are completed by the same vehicle consecutively or not. Seventh set is the definition of $T_{i,j,k}$. Eighth and ninth sets are definitions of T_i and E_i , respectively.

As it can be observed from the model, the mixed integer programming formulation requires a quite large number of constraints and variables. For example, number of binary variables is equal to

$$N^2 - N + N * M * T.$$

Furthermore, number of constraints is

$$5 * N * M * T + M * T + 3 * N.$$

Since, for constant M, T has to be proportional to N, the actual number of constraints is also proportional to N^2. Thus, number of binary variables and number of constraints increase rapidly when number of jobs increases. For instance, for a problem of 30 move requests and 5 vehicles, if T is selected to be $5 * N/M$, there will be 5370 binary variables and 22740 constraints. If number of jobs is doubled, we will have 21540 binary variables and 90480 constraints. Since a typical problem can have more than 100 jobs, it is very difficult to solve the mixed integer program in a reasonable computation time.

Meantime, the problem defined is very similar to Time Constrained Vehicle Routing Problem, discussed in Chapter 2. Actually, by adding the loaded travel time required for a particular job to unloaded travel times from every job to this particular job; our problem becomes identical to TCVRP except objective functions. TCVRP tries to minimize the distance travelled, whereas our objective is minimum deviation from time windows. In this respect, TCVRP potentially assumes that the problem is feasible. However, despite this difference, the results found for TCVRP can be used to justify that optimal solution would require a large computation time. Actually, noting that a 15 client (corresponding to 15 jobs) TCVRP took 2 minutes, normal sized instances of our problem (e.g. $N = 100$) should be expected to require long computation times.

Therefore, the mathematical program cannot be used practically to solve our problem. As in TCVRP, heuristic methods should be developed that can solve the problem in reasonable computation times without large deviation from the optimal solution.

3.5 Conclusion

In this chapter, we first defined our model for incorporating AGV module to the AMRF Decision Hierarchy. We identified the problem that the AGV Scheduler has to solve. We also gave a mixed integer programming formulation of the problem. But the resultant model requires too many binary variables and constraints, so it is practically very difficult to solve.

This result is not surprising, since the problem is a scheduling problem in the end and it is natural to have difficulty in obtaining an optimal solution. The best alternative is finding a heuristic method and that is what is done in this study. In the next chapter, we will present a heuristic method for this problem. In subsequent chapters, the performance of that model will be compared with the existing heuristics in the literature.

Chapter 4

SCHEDULING ALGORITHM

4.1 Introduction

In Chapter 3, we proposed a hierarchical scheme for incorporating AGVS to the AMRF decision hierarchy. Many problems related to hierarchy of AGVS were eliminated by this scheme. Incorporating AGVS in scheduling task was proposed by means of an AGVS module which scheduled all the move requests for AGVs throughout the shop. This scheduling problem was formulated as a mixed integer program in chapter 3. At Section 3.4, the computational difficulty of solving the problem was discussed. In this chapter, a heuristic method will be proposed for this problem. In section 4.2, we discuss the fundamentals of our heuristic method for the problem. Then, the method is described in more detail in section 4.3. In the next chapter, experimental design and computational results will be given.

4.2 Fundamentals of the algorithm

In Section 3.2, the scheduling task of the shop level in the AMRF decision hierarchy was explained. Furthermore, a framework for including material handling requirements in the scheduling task was also proposed. Besides the regular job scheduling problem, AGV scheduling problem was identified. Our approach is to decompose the overall scheduling problem into two problems and solve them interactively by different modules. The assumptions made in the problem were listed in Section 3.3.2. The scheduling problem can be described as follows: There are M identical AGVs and N move requests to be completed by these vehicles. Each move request requires one and only one vehicle (i.e. jobs can neither be split nor merged) that will take the load from the known pick-up point to known drop-off point. The layout is given, so are the distances between these points. Each job has known release times (corresponding to completion of process in the pick-up work station) and due-dates (corresponding to start of process in th e drop-off work station). The objective is to have a minimum total earliness plus tardiness, or to minimize the total deviation from the schedule given by the AMRF scheduling module.

Our problem described above can be considered in a scheduling framework as follows: There are N jobs with known processing times to be processed on M identical parallel machines. Each job has a release time and a due-date between which the jobs can be processed without penalty. Requiring an earlier start time than release time (i.e. earliness) or completing after due-date (i.e. tardiness) is undesired and results in a penalty. Thus objective is to minimize the total deviation. A sequence dependent setup time is required between two consecutive operations on a machine. Jobs cannot be split. The proposed micro-opportunistic scheduling algorithm (MOSA) is described in detail in the following section.

4.3 The scheduling heuristic

In this section, we will explain our heuristic method for the problem described in Section 4.2. We will start this section by noting the key ideas of our heuristic that we have adapted to our problem from different studies in the literature in subsection 4.3.1. In subsection 4.3.2, we will explain our model in detail.

4.3.1 Key ideas in the algorithm

In Section 2.3.5, we mentioned some studies in scheduling literature. The OPT approach [13] brought the idea of bottlenecks but from a static perspective. However the idea that bottlenecks should be viewed from a dynamic perspective resulted in very good methods like OPIS [26] and MICRO-BOSS [25].

In our problem, since all vehicles are identical and parallel, there is no bottleneck resource. However, the idea of opportunistic scheduling and micro-opportunistic scheduling can still be adapted to our problem as well. One can approach our problem from two simple points of view. One approach might be a job-based approach. We could identify the priorities among jobs. A good measure of priority would be the 'slack times' for jobs, which is the due-date of the job minus its release time and time required for transportation. Clearly, a job with relatively little slack is more likely to have penalties in a schedule. The job-based approach might try to schedule the jobs with high priorities first. This is similar to the idea used in the OPT and ISIS [10] in the sense that the jobs that are more likely to result in penalty can be viewed as bottleneck resources. However, using only this idea ignores the unloaded movement times of vehicles which correspond to sequence dependent set-up times in scheduling framework. A schedule based solely on job priorities does not consider the unloaded travel times. But actually, since total loaded travel time requirement is fixed, the utilization of vehicles is in fact determined by these unloaded travels and if a large utilization of vehicles is required, it will be very difficult to schedule the jobs with lower priorities resulting in many conflicts. Therefore, giving priority to those jobs that are more likely to result in penalty is a good idea, but is not sufficient per se.

Another view might be a vehicle-based one. We could try to minimize the unloaded travel times so that jobs will have more opportunities to be scheduled. But clearly, taking only the distance requirements to schedule jobs ignores the priorities between jobs. The resultant schedule is likely to have some penalty because of this. So a vehicle-based approach eliminates the disadvantage of a job-based one but has other disadvantages since it disregards job priorities.

In Section 2.3.5, we mentioned opportunistic scheduling approaches such as OPIS [26] which provides the ability to dynamically switch between the job centered and resource centered perspectives. The idea of opportunistic scheduling would correspond to combining the vehicle-based and job-based perspectives in a single algorithm in which the perspective is changed dynamically. Our algorithm is, in this respect, an opportunistic scheduling algorithm. We will combine these two perspectives.

One other study that was mentioned in Section 2.3.5 was a recent study of Sadeh [25] in which he used the term micro-opportunistic scheduling first. In his study, Sadeh presents an algorithm that has the ability to change its perspective very frequently (after each operation schedule) and his algorithm outperformed the macro-opportunistic schedulers such as the one proposed by Adams et al. [26]. MOSA is called a micro-opportunistic one because we also have the ability of changing our perspective as we schedule each job.

Another idea is taken from Muscettola's Conflict Partition Scheduling concept, which imposes some constraints on scheduling the parts on the bottleneck resource, instead of directly scheduling any bottleneck. This idea can be used in our problem as well. Instead of strictly scheduling the jobs to operations, we impose intervals for each job in which it will be scheduled in the end, to the assigned vehicle. These intervals get narrow in time, after each assignment, and at the end, the exact schedule is formed. This gives a large amount of flexibility to our scheduler, since the number of alternatives is held as large as possible for the jobs that will be scheduled in later stages. MOSA is described below.

4.3.2 The Algorithm

In this section, we will present our micro-opportunistic scheduling algorithm. Our algorithm consists of five basic steps:

- Step 0: Identify time slots from release times and due-dates of each job.

- Step 1: If all the jobs are scheduled then stop, else, determine slot loads from the set of all unscheduled jobs. Determine the most contended slot.
- Step 2: Schedule the jobs in the most contended slot to M temporary vehicles (as if no previous schedule was made).
- Step 3: Merge the temporary schedule to the existing schedule. Check if the temporary schedule has to be joined with a later or earlier slot's schedule, because of conflicts. If not go to Step 1. If there is only one side conflict (an earlier or a later), go to Step 4, else (there are two conflicts, both earlier and later) go to Step 5.
- Step 4: Start with the earlier of the two slots, identify and append the jobs between the two slots to this slot. Finally assign two temporary assignments. Go to Step 1.
- Step 5: Start with the earlier conflicting slot, identify and schedule the jobs between this slot and current slot. Fit the current slot to the existing schedule. Identify and append the jobs between the current slot and later conflicting slot. Finally fit the later slot to this combination. Go to Step 1.

Step 0 is traced only once at the start. The release time and due-dates of each job can be viewed as points in time. We eliminate repetitions (e.g. one job's due-date may be equal to one other's release time) and every interval formed between consecutive such points is called a 'time slot'. We can have at most $2 * N$ slots (0 being first point in time). These slots will be used to determine priority of time intervals.

Individual job weights are determined by the following expression:

$$W_i = \frac{1}{D_i - R_i - t_1(i)} \quad \text{if } D_i > R_i + t_1(i)$$

Thus, those jobs with a larger time window are assigned a lower weight. In determination of slot loads, these weights are used.

In Subsections 4.3.3.1 through 4.3.3.5, the steps of the algorithm will be explained in detail.

4.3.3 Determination of slot loads

Slots are used to determine the time intervals that are more likely to result in early or tardy jobs. If those intervals are identified and scheduled first, the large load in the interval can be successfully dispersed to earlier and later time periods. Our algorithm selects these intervals as a starting point. Thus instead of giving priority to particular jobs, we give priority to time intervals.

Slot loads are determined as follows:

$$SL_k = \sum_{i=1}^{N} I_{i,k} * W_i$$

where the indicator function $I_{i,k}$ is defined by

$$I_{i,k} = \begin{cases} 1, & \text{if slot } k \text{ is included in the interval } (R_i, D_i) \\ 0, & \text{otherwise} \end{cases}$$

Figure 4.1 shows determination of slot loads from job weights. For each slot, the jobs that have a smaller release time and a larger due-date are identified (note that no job can have a release time or due-date in between the starting and ending times of the slot). Next, the weights of all such jobs are summed to find the total load of the slot. Having a large load for a slot accounts for a high probability of penalty in scheduling the jobs in that slot due to the conflicts that might arise between jobs. For instance if a slot has a job with zero slack, then its load will automatically be very large. Actually, such a job means that the job should be started at its release time to prevent it from a penalty. Starting to schedule from such a slot has clearly the advantage that we fix an operation on which we have almost no choice. We decrease our alternatives for later stages in one sense by imposing new constraints. From another view, we try to eliminate possibility of a penalty. We might have a penalty if we had already scheduled some other jobs using a forward scheduling rule and had to append the job with zero slack to the existing schedule.

At the beginning, all the jobs will be unscheduled. The slot loads will make up a profile in the form of a step function. The first slot to start scheduling is the one with highest load. In further iterations, some of the jobs will be

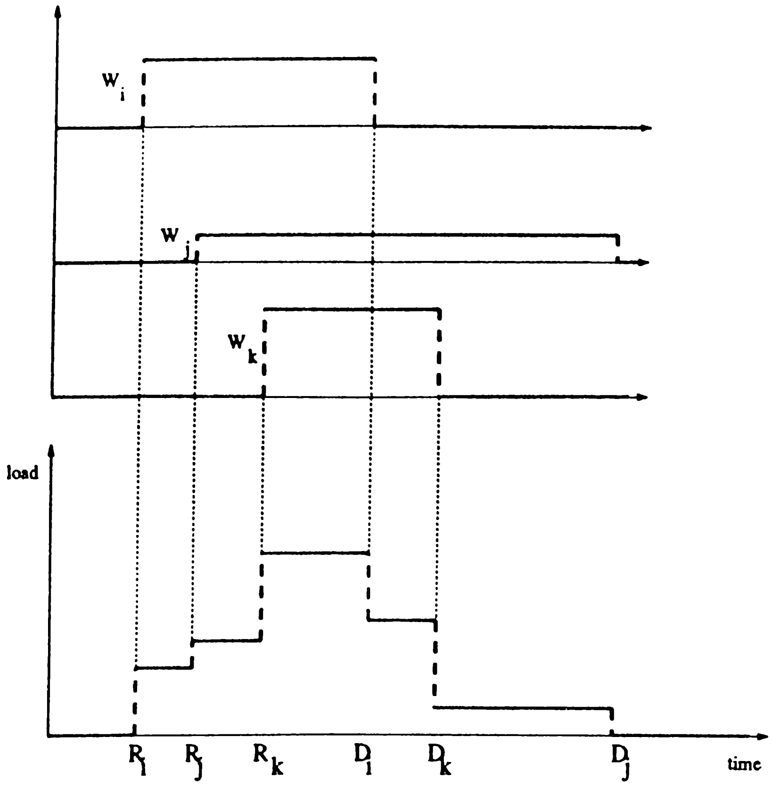

Figure 4.1: Individual job weights and slot loads

scheduled and these load profiles will be re-computed at each iteration. Clearly, the loads of slots will decrease in time.

4.3.4 Scheduling the jobs in the most contended slot

In Step 1, we determined the most contended slot for the current iteration. In Step 2, the jobs in this slot will be scheduled to M temporary vehicles. In scheduling the jobs in the current slot, we follow two different approaches. This step of the algorithm can be described as follows in more detail:

- Step 2.a: Identify the jobs in the slot as 'current set', sort them in decreasing weight and form the groups of jobs that have the same weight. Assign first job on the list at the earliest time possible to the first vehicle, compute the 'slack' of the vehicle as the amount that the assigned jobs can be shifted to the right without incurring penalty.

- Step 2.b: For the most critical unscheduled group in the current set, check whether one of these jobs is the nearest one in distance (among all unscheduled ones) to either the first pick-up point or the last drop-off point of a vehicle on the temporary schedule so that no penalty incurs. If so, select that job to be scheduled next and go to Step 2.d; else, go to step 2.c.

- Step 2.c: Start with the first unscheduled group and search a job that cannot be scheduled without incurring any penalty to those scheduled before. If there is more than one in a group, choose the one with largest minimum distance requirement to those scheduled before. If there is no unscheduled job in the whole list that might cause infeasibility, then select the job with the largest minimum distance requirement in the first unscheduled group. Schedule this job to a new vehicle at the earliest possible time, compute the 'slack' of the vehicle. Go to Step 2.e.

- Step 2.d: Schedule the selected job next to the scheduled job on the same vehicle just before or just after the scheduled one depending upon the decision made at Step 2.b. Update the 'slack' of the vehicle. Go to Step 2.e.

- Step 2.e: Repeat Steps 2.b, 2.c and 2.d until all the vehicles are used (at least once) or all the jobs in the slot are scheduled. If there are unscheduled jobs yet, then continue, else stop.

- Step 2.f: For each unscheduled job, select the 'best' vehicle and 'direction' (i.e. the side of the vehicle to append) using priority 1.

- Step 2.g: Select the next job to be appended: For each job, compute priority 2 for the 'best' vehicle and 'direction'. Among all the unscheduled jobs select the job with highest priority 2.

- Step 2.h: Insert the selected job to its 'best' vehicle in its 'direction' (i.e. to the left or right of the block of jobs in the vehicle). Update the slack of the vehicle. Update the 'best' vehicles and 'directions' of unscheduled jobs.

- Step 2.i: Repeat Steps 2.f, 2.g, 2.h until all jobs are scheduled. When all the jobs in the slot are scheduled, stop.

Priorities 1 and 2 use the ideas similar to RM dispatching rule. In addition to a single 'forward' priority, a 'backward' priority is defined as well. The two priority 1 functions, for left and right, respectively are:

$$\pi^L_{1,j} = \frac{1}{t_1(i) + t_2(i,l)} * exp\left(\frac{ST_j + SLACK_j - (R_i + t_1(i) + t_2(i,l))}{2 * k * t_{av}}\right)$$

and

$$\pi^R_{1,j} = \frac{1}{t_1(i) + t_2(r,i)} * exp\left(\frac{D_i - (ET_j + t_2(r,i) + t_1(i))}{2 * k * t_{av}}\right)$$

where l is the pick-up point of first, r is the drop-off point of last job in the vehicle, R_i and D_i are release time and due-date of job i, ST_j, ET_j and $SLACK_j$ are the starting time, ending time and slack of vehicle j, respectively, k is a constant and t_{av} is the average travelling times of jobs. Because of the objective, weights of jobs are all taken identical. With this form, a vehicle with larger feasibility interval (i.e. slack) and requiring smaller unloaded travel time is given higher priority. Meantime, the preferred side of the vehicle is chosen. The slack of the vehicle is used to offset the starting time to latest possible start time. Note that the urgency of the jobs is not considered in these functions because what is being selected is not the job, but the vehicle. Thus among the $2 * m$ possibilities, the particular vehicle and side combination that is preferred most is selected.

For selecting the next job to be scheduled, the following priority 2 functions $\pi^L_{2,j}$ and $\pi^R_{2,j}$ are used, which are similar to $\pi^L_{1,j}$ and $\pi^R_{1,j}$:

$$\pi^L_{2,j} = \frac{1}{t_1(i) + t_2(i,l)} * exp\left(\frac{(R_i + t_1(i) + t_2(i,l)) - (ST_j + SLACK_j)}{2 * k * t_{av}}\right)$$

and

$$\pi^R_{2,j} = \frac{1}{t_1(i) + t_2(r,i)} * exp\left(\frac{ET_j + t_2(r,i) + t_1(i) - D_i}{2 * k * t_{av}}\right)$$

For a particular job, the appropriate one of these two functions is determined from the 'direction' stored for that job and the 'best' vehicle is used in these functions. Now that the next job is to be selected, the priority function (priority 2) gives higher priority to the more critical job (i.e. having smaller slack).

Notice that the priorities of jobs are re-calculated after each job assignment. Only the jobs that have included in the load of the slot are considered and the priority among these jobs is determined by the weights of jobs. Starting with one vehicle, as long as there are other vehicles, all the vehicles are tried to utilize as early as possible to increase the number of alternatives in later stages. But a new vehicle is utilized only when there is a job that would require penalty with other vehicles or that is very far from other vehicles in distance so that it cannot be justified to schedule to other vehicles. Meantime, high priority jobs that require minimum distance to those scheduled before are searched and appended, if there is any. Thus, for this first part, the job criticality and resource utilization is considered simultaneously. Meantime, in the second part, the priority rule combines the two perspectives successfully as shown in Chapter 5. In this respect, Step 2 is an opportunistic approach. Since the perspective decision is made after each assignment, this step follows a micro-opportunistic approach.

The assignments and schedules made in Step 2 is not fixed. Starting with the first job of a vehicle, the job is scheduled to start in a certain interval, rather than a fixed point. The first job is assigned to the interval that it will result in no penalty in Step 2.c, that is the interval starting with the release time of the vehicle and ending with the due-date minus movement time. This assignment is preserved in each iteration; for a block of jobs on a vehicle, there is an interval that the block can be placed anywhere within. In the algorithm, interval assignments are made by assigning the jobs of a vehicle at the earliest possible time and using a 'slack'. As a new job is to be appended to a vehicle as in either Step 2.d, or 2.g or 2.h; the ability to shift the jobs that are already in the vehicle is also considered. The job is scheduled to start at its earliest possible time and the vehicle's slack is taken into account. If the new job cannot be shifted to right as much as the other jobs in the vehicle, the slack is decreased so that all the jobs in the vehicle can be shifted in that amount to the right. This continues in the same fashion till all the jobs in the slot are scheduled. At the end, for each vehicle, all the jobs in that vehicle are assigned as early as possible but the whole has a slack which corresponds to amount of time the jobs in the vehicle may be shifted to the right. Actually, from a logical point of view, the jobs are scheduled to intervals rather than fixed points. These intervals will be used in further steps and when joining two temporary vehicle blocks, the blocks are held in the form of constraints rather than fixed schedules.

4.3.5 Block Merging Process

In Step 2, we form a block of the schedule on temporary vehicles in the form of intervals. When making these schedules, the previous vehicle schedules were not taken into consideration. In Steps 3, 4 and 5, the conflicts between the new block and those formed before are resolved. An approach of first solving Step 2, then applying Steps 3, 4 and 5 correspond to a decomposition approach. We first ignore the other loads of vehicles and solve the scheduling problem, then we try to impose the previous assignments in the form of constraints to the existing solution. In Step 2, the jobs are scheduled to temporary vehicles. In steps 3, 4 and 5, in case of a conflict, each temporary vehicle is assigned to a vehicle from the previously scheduled block and two vehicles are 'merged'. Note that a temporary block corresponds to a slot and a slot can have a conflict either with the latest of the earlier slots or earliest of later slots in time. Thus there can be three alternatives; which are, no conflict case, one conflict case and two conflicts case. Step 3 identifies the case for the current block. Clearly, if there is no conflict, there is nothing to do, but to implement the current schedule and return to Step 1 to identify and schedule another slot. Step 4 and Step 5 solve the one conflict and two conflict cases, respectively. When the conflicts are resolved, we return to Step 1, again.

The term conflict can be defined as follows: Two blocks are said to have a conflict if the latest completion time of all jobs in the earlier block exceed the earliest completion time of all jobs in the later block. This corresponds to overlapping of partial schedules, in other words, non-interference constraints for vehicles are violated. Thus for the current temporary block, in Step 3, the earlier and later slots are checked, independent of one other. The next step is determined by the number of conflicts identified. In case of a single conflict, not only the two blocks are merged, but also the jobs that are between the two conflicting slots are scheduled to these slots. Clearly, two conflicting slots do not have to cover all the jobs between them. Rather, there can be jobs that were not included in either of the slots, which have a release time later than the earlier slot's ending time but a due-date earlier than the later slot's starting time. If these jobs are not taken into account at the moment, they are likely to create a very high penalty in further stages, because these two slots will be merged and these jobs will have to be scheduled earlier or later than the merged slot, which will be too early or too late, resulting in penalties.

We can have a single conflict case in two ways. The current block is either earlier or later. These two cases are not handled in different ways, but rather, the blocks are assumed to have equal importance without a differentiation on the temporary one. Step 4 can be explained by following steps:

- Step 4.a: Set the earlier of the conflicting blocks at its earliest possible time to start. Sort these jobs in increasing completion time. Identify the jobs in between these two blocks. Sort them in decreasing weight as defined in section 4.3. Select the first job in the list.
- Step 4.b: For the selected job, identify the vehicles that this job can be appended without any penalty. If there is more than one such vehicle, select the one with minimum distance requirement and append selected job to that vehicle. If there is no vehicle without penalty requirement,

select the one requiring minimum penalty and append the job to that vehicle. Revise the slack of the vehicle.

- Step 4.c: Sort the vehicles again in increasing completion time. If there is any other job, select the next job in the list, go to Step 4.b. If not, select the first vehicle in the list.
- Step 4.d: For the selected vehicle, sort the unmerged vehicles of the later block in increasing possible start times (ending time of selected vehicle offset by unloaded travel time required to later vehicle's pick-up point). Select the first vehicle in the list to merge with the selected vehicle of earlier block.
- Step 4.e: If two vehicles can be merged without incurring any penalty, schedule the group at the earliest time possible and update slacks to a single slack. Else shift the schedule of the vehicle having smaller number of jobs on it either to the left (if it is the earlier one) or to the right (if it is the later one), while holding the other one fixed. Schedule the two parts of schedules at those times with zero slack.
- Step 4.f: Mark the two vehicles. If there are still vehicles to merge, select the next vehicle from the earlier group, go to step 4.d. Otherwise, stop.

The approach in appending the jobs is different than that in Step 2 in the sense that jobs are directly scheduled to the block. The highest priority is given to feasibility, then to distance. First, feasible alternatives are checked, if there are any, priority is given to distance requirements. One could follow strictly the same approach as in Step 2, that is first schedule all the jobs in between to temporary vehicles and then try to match these with the conflicting blocks. But the difference in the structures suggest a different approach as above. First, number of jobs identified in this step are likely to be very little, the economics of step 2 is hardly made. Next, the objective of this step should be to fit the jobs to conflicting blocks, since the two blocks are already conflicting.

Note that the two conflicting blocks are not differentiated in Step 4 above.

Step 5 solves the two conflict case, using Step 4 twice. First, the earlier conflicting block is taken as the first block; the temporary block is the second and the two blocks together with jobs in between are merged, exactly as Step 4. Then this merged block is taken as the first block, the later of conflicting blocks as second block and these two blocks are merged together with the jobs in between. In this way, all the conflicts are resolved.

An iteration is finished with Step 4. Next step is Step 1 again, which identifies the next slot to be scheduled. The algorithm continues in this fashion until all the jobs are scheduled.

4.4 Conclusion

The complexity of the problem to be solved by the AGVS module was mentioned in the previous chapter. The size of the mathematical program suggested using a heuristic method as a solution methodology. In the previous sections, the key ideas and the method was explained in detail. The heuristic method proposed utilized three ideas of scheduling literature, namely, idea of joining two conflicting perspectives in a single function of Rachamadagu and Morton [22], idea of ability to change the perspective at each job assignment of Sadeh [25], and idea of imposing constraints instead of fixed assignments of Muscettola [23]. Actually, all of these ideas, devised for job shop, have potential difficulties for direct adaptation. Our problem involved, in the scheduling terminology, sequence dependent set-ups, which cannot be ignored. Meantime there are time windows and the objective is related to deviation from time windows. Therefore, none of the ideas can be applied directly. To support the success of the proposed method, the method will be compared with other approaches in literature in many different experimental settings that should reflect the actual shop floor conditions in the next chapter.

Chapter 5

EXPERIMENTAL ANALYSIS

5.1 Introduction

In Chapter 3, the mathematical formulation of the AGV scheduling problem was given. Because of the complexity of the problem, the micro-opportunistic scheduling algorithm was proposed for the AGV scheduling problem in Chapter 4. In this chapter, performance of the algorithm is analyzed. An experimental design is made so that the computational analysis and analysis of variance can be performed.

5.2 Experimental Design

The scheduling algorithm proposed in Chapter 4 was developed for the purpose of off-line scheduling of AGVs. In Chapter 2, it was noted that the AGV dispatching rules studied in the literature are not robust in the sense that their success highly depends on the experimental settings. Because of this fact, performance of the proposed algorithm in different settings is quite important. This requires a careful selection of factors and levels in the experimental design.

5.2.1 Choice of Factors and Levels

For the purpose of spanning an extensive amount of different settings, the factors to be used in the experimental analysis should be determined carefully. In order to include sufficiently many cases in the experiments, any variable that is likely to be significant for the performance of the algorithm should be included. Some of these factors may, of course, turn out to be ineffective eventually.

The first factor, that is included in the experiments is factor A, number of move orders to be scheduled, which is called number of jobs and denoted by N. The structure of the heuristic method suggests the possibility that number of jobs may effect the performance negatively. Thus small and large problems should be included in the design.

Second factor, factor B is determined to be the physical layout. As was discussed in Section 2.5, studies in the literature show the significance of layout in the success of different dispatching heuristics. One would expect a robust algorithm to give good results in all layouts. However, the problem with layout is the inability to express it in the experiment. Specifically, what makes the layout significant in the performance needs to be discussed. Different layouts can be handled by means of two measures, the average and standard deviation of distances in the layout. Average, by itself, is not likely to be very much significant since it only brings a scale effect to the problem. Meantime, the standard deviation of distances can be significant since the selections in scheduling become more critical. In the experimental design, however, instead of generating distances from scratch, layouts similar to those studied in the literature are used. The reason behind this fact is that standard deviation of layout does not have a clear practical meaning. What is more, effect of number of departments can easily be combined, which cannot, and should not be isolated from dispersion of distances. The size of the layouts should be different so that the standard deviation of the dispersion of the distances can be manipulated as desired.

One important factor that can effect the performance of the system is the tightness of time windows for jobs. As was discussed in Section 2.5, performance of job-based and resource based dispatching rules depends highly on this factor. This factor can be expressed in the experimental design by the total work content rule. The length of the time window can be held as a certain multiple of movement time required for that move order. This multiple, denoted by K, is viewed to be a factor, factor C, so that different tightness cases are handled. This setting has, in fact, its source on the overall shop model described in Chapter 3. When the schedules for the cells and workstations are being prepared, approximate times have to be prepared for move orders. The logical way for this task is taking a certain multiple of the required movement time as the length of the time window, which is known as the total work content rule (T.W.K.).

Finally, one other factor that was different in current studies is factor D, number of vehicles, which is denoted by M. Range of number of AGVs studied is from 2 to 8 and number of vehicles can effect the performance significantly. Thus, it is included as a factor to the experimental design. The factors included in the study are listed in Table 5.1.

Factor	Levels		
Number of jobs	100	150	200
Layout	Small	Medium	Large
T.W.K. coefficient	2.0	4.0	6.0
Number of vehicles	2	5	8

Table 5.1: Factors and levels in the experimental design

The other variables in the system are treated as fixed parameters. One of them is the release time distribution. Release times of jobs are assumed to be distributed uniformly in the interval between zero and expected makespan. The distribution of departments for the jobs is also assumed to be uniform, that is every department is equally likely to be the pick-up or drop-off point of a move order. This assumption is based on the fact that there is neither queue size limit, nor traffic problem in the model, which would justify introducing some critical departments into the model. Number of departments in the layout is considered as part of the layout factor, thus is not independently manipulated. The fixed parameters are listed in Table 5.2.

Fixed Parameters	Distribution
Release Time of jobs	Uniform(0, $\frac{2*t_{av}*N}{M}$)
Pick-up and Drop-off departments	Uniform(0, Number of Departments)

Table 5.2: Fixed Parameters in experimental design

In order to be able to observe any quadratic relation between the factors and the performance, every factor has three levels in the design. These levels are chosen so that most of the practical settings are spanned. Thus, some of the levels might be even beyond the practical limits.

The three levels in factor A, number of jobs, are determined as 100, 150 and 200. These levels are chosen with pilot runs and the difference in computation time and objective is found to be sufficient for $N = 100$ and $N = 200$. The medium level is chosen to be the average, in principle.

For factor B, the three levels are chosen to be three different layouts, being small, medium and large. The standard deviation of the layouts are chosen to cover a wide range, in parallel to size of layout (i.e. σ is small in small layout, large in large).

Tightness of time windows, factor C has again three levels. This factor clearly requires more pilot runs for determination of its levels. In setting the range of this factor, at one extreme, the windows should not be too loose so that a zero-penalty solution to the problem can be found. At the other side, a too tight time window can hide the difference of good and bad schedules since every schedule will be likely to generate high penalties. As the result of pilot runs a reasonable range is determined to be 2 at minimum and 6 at maximum. For convenience, the medium level is chosen to be the average, 4.

The three levels for factor D, number of vehicles are determined to be 2, 5 and 8 for small, medium and large levels, respectively. These numbers are determined for practical justification. The large level, being at 8 could be beyond the practical maximum for a typical AGVS. However, for the sake of robustness, this is required.

5.2.2 Alternatives for Comparison

For comparison purposes, four alternatives are selected. Since the computational time to obtain an optimal solution is very large, as discussed in Chapter 3, the alternatives are heuristic methods. The heuristic rules that are included in comparison are:

- First Come First Served (FCFS) or Earliest Release Time (ER)
- Shortest Travel Time First (STTF)
- Earliest Due-date (EDD)
- Rachamadagu-Morton (RM)

The first two rules are included in the list of Section 2.5. STTF is a resource based, while FCFS is a job based rule. The third rule is not studied in the literature because most of the studies do not assume due-dates for the jobs. However, since main objective in our problem is lateness, EDD rule has the potential to give good results. The forth alternative is the adaptation of RM rule which is explained below. No vehicle-based dispatching rule is included in alternatives, because, our typical problems are quite loaded and such a rule cannot be a reasonable alternative. Of course, since these are all forward dispatching rules, lateness of these rules can only be in the form of tardiness, no job can be assigned before its release time.

5.2.2.1 A Dynamic AGV Dispatching Rule

In section 2.4, it was noted that the performance of the existing dispatching rules depended highly on the particular settings of the problem. One important reason of this fact is the inability in the rules to consider the trade-off between the urgency of certain loads waiting transportation and effective utilization of the vehicles. These two objectives correspond to the job and resource based perspectives of the scheduling literature. In section 2.6, a priority rule that considers these two perspectives simultaneously, namely the RM dynamic dispatching rule was discussed. In this rule, at the completion time of each job, the next job to be processed is determined as the one with largest priority. The priority of each job is dynamic, since it depends on the current time. The priority of each job is determined by the following formula, which was also given in section 2.6:

$$\pi_j = \frac{w_j}{t_j} * exp\left(\frac{-S_j}{k * t_{av}}\right) \tag{5.1}$$

where, w_j and t_j are the weight and processing times of job j, S_j is the slack time (expected time left between the due-date and the completion time, which can also be negative, corresponding to amount of tardiness) of the job, k is a constant and t_{av} is the process time average of the jobs waiting to be served. The first factor in the expression corresponds to giving relatively higher priority to those jobs requiring shorter processing times. This corresponds to opportunity cost of using the resource. Meantime the second factor corresponds to giving higher priority to those jobs that are (more) likely to result in more penalty. For AGV problem, this priority rule can be adapted to be:

$$\pi_j = \frac{w_j}{t_1(i) + t_2(l,i)} * exp\left(\frac{t_{now} + t_1(i) + t_2(l,i) - D_i}{2 * k * t_{av}}\right) \tag{5.2}$$

In AGV problem, the opportunity cost of an assignment is not related only to the moving time of that load, but also to the time the AGV will spend by unloaded travel, from its current location to the pick-up point of the load. Thus, in the first factor, the unloaded travel time is also included. Similarly, for the second factor, the completion time of the candidate job should include the time required for unloaded travel. This correction in the second factor suggests that, for normalizing the slack time, a denominator of $2 * t_{av}$ should be used. Actually, this corresponds to treating the unloaded and loaded travel times as a single process. Assuming that loaded and unloaded travel times have the same average, this average can be taken as t_{av} in the formula. As was noted by the authors [22], the factor k may take a value between 1 and 3. In the experimental design, it is taken to be 2, as is suggested by the authors [22].

The dispatching rule should run as follows: As a vehicle completes its current job movement, the next job for this vehicle is selected to be the job having largest priority at that current time (completion time of previous move). At the next completion time, the priorities for the jobs should be computed again, the highest priority job should be selected, and so forth till all the orders are scheduled.

5.2.3 Response Variable

As discussed in Chapter 3, because of the task of the AGV module in the hierarchical system, the performance measure is the total deviation from time windows (in terms of earlier start or later finish times). The total distance travelled or utilization of the vehicles are not directly included in the objectives function, however, as is easy to interpret, there is a strong correlation between all these measures. On the other hand, no measure related to average queue sizes nor shop locking is considered, again, because of our objective. Our interest is in the task of AGV module, not the shop itself. These limits should be considered in related scheduling module, not in AGV module. Thus, our only objective is the minimization of amount of deviation, i.e. lateness. However, for justification, the CPU times of the methods included in comparison is taken as a response variable as well.

5.2.4 Type of Experimental Design

As is discussed in Section 5.2.1, the performance of the algorithm we are proposing has to be robust, which requires experimenting it at a wide range. For this reason all the factors that have potential to affect the performance are included in the model. One aim of experimental analysis is to identify the effect of each individual factor or any combination of factors on the performance of the algorithm. Because of this, the experimental design selected for our problem is a full-factorial design. Since there are four factors and three levels, our experiment is 3^4 factorial design, which corresponds to eighty one treatment combinations. Number of replications for each combination is taken as five. Therefore, there are 405 different runs.

5.2.5 Computational Comparison

In order or compare these five methods (four rules listed and our algorithm), all these rules are coded in Turbo Pascal and run on PC-486. Five sets of eighty one problems are generated using the parameters as stated. For the same problem, five methods are applied and total lateness and computational time required are recorded. The result of individual runs are included in Appendix A.1 through A.5. Table 5.3 indicates the summary lateness results.

The summary results show that our method outperforms all the others. Average penalty of our method is around 72% of the second best, which is RM. A paired t-test is performed to see whether our algorithm is significantly better than others. The results showed that our method outperformed others in 0.5 % significance level. Furthermore, as far as the minimum and maximum results of the methods are concerned, minimums found in the runs are equal while our algorithm's maximum is about 53% of that of RM, which is the second minimum after our's. These give a good indication that our algorithm is also robust to changes in settings.

For comparison of other rules, first of all, no rule is dominated, i.e. each rule performs best in at least one run. For general success of other rules, RM was the second best. As a surprising result, STTF, although not significantly, performed better than RM in one set. Clearly, STTF is a job based heuristic and it does not consider the urgency of jobs. In this respect, noting the general

Replication No.		ER	EDD	STTF	RM	MOSA
Rep. 1	Aver.	1353.2	1690.0	353.3	348.3	227.9
	Min.	0	0	0	0	0
	Max.	8666	8881	2201	2944	1391
Rep. 2	Aver.	1401.2	1751.3	353.0	345.1	263.8
	Min.	0	0	0	0	0
	Max.	9604	9305	2239	2509	2206
Rep. 3	Aver.	1437.4	1832.6	389.5	381.5	261.8
	Min.	5	0	1	0	0
	Max.	12973	10250	2716	2790	1531
Rep. 4	Aver.	1504.8	2164.4	403.1	365.4	267.3
	Min.	0	0	0	0	0
	Max.	17096	24831	3859	2571	1432
Rep. 5	Aver.	1461.8	1657.9	446.1	360.7	268.5
	Min.	0	0	0	0	0
	Max.	20065	15047	6532	4461	2381
OVERALL	Aver.	1431.7	1819.2	389.0	360.2	257.9

Table 5.3: Total Deviation Comparison

success of RM heuristic, one would expect RM to outperform STTF in every case. When the individual runs are analyzed, number of times one performs better than the other are not very different. One possible reason of this result is that, the problems, in general, are quite loaded. A vehicle based method is likely to result better in these cases. Actually the two other rules, ER and EDD, both job-based, have poor results. The adapted RM rule, including urgency of some jobs, performed similar to STTF, but it performed better than in all five replications. Furthermore, RM rule cannot balance the two perspectives as well as our method does. Although not included in our study, some additional runs with lower vehicle loads showed that RM and STTF still perform very well, but RM beats STTF significantly. Thus we can conclude that the particular problems are so loaded for vehicles that resource based perspectives become successful.

When the individual results are analyzed, it turns out that number of times our method outperforms the others is only 209 of 405 (and there are 9 ties) despite the 28% difference in the overall. The reason for this is that, our algorithm performs significantly better than others especially in the high load cases. Thus, the average results seem better than individual comparisons.

As for the computational time requirements, the runs are taken on a PC-486. Table 5.4 indicates the summary computation time requirements.

	CPU times (sec.)				
	ER	EDD	STTF	RM	MOSA
Aver.	0.10	0.09	0.11	0.19	8.59
Min.	0.04	0.05	0.03	0.06	1.04
Max.	0.17	0.17	0.22	0.28	39.00

Table 5.4: Computation times comparison

Time requirements show that our method requires considerably larger time than the other rules. However, this should be expected since the other four rules are only dispatching rules. Although the computation time is relatively large, normatively, it is acceptable for such a planning decision. The maximum time required in the runs is 39 seconds for a problem with 200 jobs. Amount of savings in lateness can compensate the computation time requirements for our model.

5.3 Analysis of Variance

In order to observe the effects of factors chosen in the performance measures, an ANOVA model is performed. As stated before, each factor has three levels in order to be able to observe a quadratic relation, if exists. Yates algorithm is used to determine which factors and factor combinations are significant for performance measures. A factor or combination is assumed to be insignificant if it is not significant at 25% level.

5.3.1 Effect on Total Lateness

All four factors are found to be significant for the total penalty measure. For number of jobs, both the linear and quadratic effect are significant at 1%. So, total penalty of the problem increases more than linearly with number of jobs. Total penalty should clearly be proportional to number of jobs, thus such a relation is expected. Similarly, both linear and quadratic effects of layout are significant at 1% level for total lateness. Layout, by means of distances, brings a scale to the problem. In this case, an increase in total penalty is to be expected. The factor K is significant (at 1% level) only linearly. Tightness of time windows should directly effect total lateness. However, one could expect that a quadratic effect also occurs because the difficulty of the problem increases very much by smaller time windows. Finally, number of vehicles has only a linear significance (at 1%).

For the combination of factors, the results showed that only the layout - time window tightness combination is significant (at 1% level) for total lateness. One explanation for this result might be as follows. Additional to its direct effect, layout effects the release times of jobs. However, this effect becomes significant only with tighter time windows. Release times determine number of available jobs in a certain time. For loose time windows, this has no clear effect, however, with tight time windows, this effect becomes significant. Table

5.5 shows significance levels of factors and significant combinations for total lateness.

Factors	Significance Levels		
	Linear	Quad.	Total
Number of jobs	1%	1%	1%
Layout	1%	1%	1%
Time window tightness	1%	*	1%
Number of vehicles	1%	*	1%
Layout-time window tightness	1%	25%	1%

Table 5.5: Significant factors for total lateness

5.3.2 Effect on Computation Time

All of the factors are significant for computation time except the time window tightness (each at 1% level). Number of jobs and number of vehicles have quite clear effects since amount of search effect increases significantly. Layout's effect on computation time can be because of the increase in the number of departments.

For combination effects, it was found that, the number of jobs - layout (at 5%), layout - number of vehicles (at 5%) and number of jobs - number of vehicles (at 1%) combinations are significant. Table 5.6 shows the significant factors for computation time.

5.4 Conclusion

In this chapter, an experimental analysis is conducted. A 3^4 full factorial design is used corresponding to 81 runs. With five replications, 405 runs are taken in total. The four dispatching rules and our algorithm are compared on the basis of two performance measures. For total lateness measure, our

Factors	Significance Levels		
	Linear	Quad.	Total
Number of jobs	1%	*	1%
Layout	1%	*	1%
Time window tightness	25%	*	*
Number of jobs	1%	10%	1%
Number of jobs-layout	1%	*	5%
Number of jobs-number of vehicles	1%	10%	1%
Layout-number of vehicles	1%	1%	5%

Table 5.6: Significant factors for computation time

method outperformed others and yielded a 72% penalty of that of RM heuristic. Especially in high load cases, our algorithm yielded very good results. Although we cannot compare it with optimum results, we can conclude that our method performs significantly better than all dispatching rules. For robustness, the results are even better. Of the 405 runs, minimum of all rules were 0. In the maximums, our method yielded a 53% maximum of that of RM, which is the second best. The results of individual runs are also parallel to this overall result. Therefore, we can conclude that our method is robust (at least relative to all dispatching rules) to changes in the experimental settings.

In order to justify our algorithm, we should show that its computation time requirements are acceptable. For this reason, we also made a comparison of computation time requirements of the five alternatives compared. As would be expected, computation time requirements of our model are considerably higher than those of the alternatives. However, when the maximum time requirement for a 200 job problem is considered, one should conclude that 39 seconds of computation time is quite acceptable, since the logical planning horizon for such a decision making problem is much longer.

In order to identify the effects of individual factors and their combinations in our performance measures, an ANOVA study is made. For lateness, as can be expected, all four individual factors are found to be significant, whereas, only the layout-time window combination is significant among factor combinations.

These results are very important. When the Shop and Cell Levels are making their schedules, they should determine a value of K such that neither results directly in a zero penalty so that overall makespan is very large, nor in too much penalty so that it becomes very hard to obtain an overall feasible schedule. Our results show that the three other factors affect the value for K, as well as their pairwise combinations.

Chapter 6

CONCLUSION

In this study, the problem of incorporating the AGVS module to the decision making hierarchy is analyzed. The reasons and results of this difficulty are identified. In attempt to overcome this difficulty, a hierarchical model for AGV module is proposed. The AGV scheduling problem that is to be solved by AGV module in this model is formulated as a mixed integer program. The size and complexity of this problem suggested using a heuristic method. Utilizing some recent ideas of successful scheduling methods, a heuristic method is developed.

In order to investigate the performance of the method, an experimental analysis is made. As for alternatives, no optimal seeking method is chosen since computation times are quite large, instead four dispatching rules are selected. The results showed that our method outperformed the successful AGV dispatching rules. Its computation time requirements were, of course, higher that those of alternatives, but were still acceptable for such a planning problem. Furthermore, in order to observe the significance of certain factors in performance and computation time of our method, we applied a statistical test. It was found that every factor that we have included had a significant effect on performance of our method. These result showed the importance of system settings in overall decision making. The cell and shop levels should choose the tightness of time windows in their schedule by considering many factors including the layout, number of move orders in the planning horizon, number of vehicles, number of departmen ts in the system, etc.

For future research, there are mainly two areas that needs further analysis. First, as was noted, the number of vehicles is assumed to be an input for AGVS module. This is because of the fact that the decision on the number of vehicles is a high level decision and it should be made at the design stage of an AGVS system. However, number of vehicles required for a shop has to be re-evaluated. The changing conditions in a shop may justify buying new vehicles or shifting some vehicles across different shops. In order to facilitate these high level decisions, AGV module should provide the required feedback to the plant level. Construction of this mechanism should be studied for further research. With the present form, the feedback information may be in form of average penalty incurred in the shop. However, since different scheduling settings may be in effect in different shops, this is not sufficient per se.

Another future research topic would be the cell and shop levels. Since the scheduling tasks of these levels have not been changed considerably, we did not include those tasks in the study. However, in order to obtain an overall feasible schedule more easily, a mechanism that jointly determines the time window tightness for jobs would be beneficial. Furthermore, the action to take when a feasibility occurs in a cell or shop itself is not studied in detail either. A successful mechanism for scheduling tasks in the proposed model would decrease the search effort for an overall feasible solution.

Appendix A

Individual runs for total lateness

Nomenclature

N	Number of jobs
L	Layout
K	Time window tightness
M	Number of vehicles
ER	Earliest Release
EDD	Earliest Due-date
STTF	Shortest Travelling Time First
RM	Rachamadagu Morton priority rule
MOSA	Micro-Opportunistic Scheduling Approach

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
1	1	1	1	888	1076	361	405	212
1	1	1	2	126	223	69	73	54
1	1	1	3	102	266	56	63	114
1	1	2	1	336	292	105	135	63
1	1	2	2	65	24	27	26	12
1	1	2	3	32	0	14	18	3
1	1	3	1	450	823	99	34	44
1	1	3	2	6	24	1	3	25
1	1	3	3	13	0	11	11	9
1	2	1	1	1097	1593	583	790	276
1	2	1	2	82	119	82	100	69
1	2	1	3	294	231	133	178	110
1	2	2	1	2950	1360	411	417	158
1	2	2	2	64	65	35	29	44
1	2	2	3	6	2	6	6	3
1	2	3	1	69	90	19	19	57
1	2	3	2	20	0	0	0	1
1	2	3	3	0	8	6	6	0
1	3	1	1	1500	1831	845	816	356
1	3	1	2	501	728	481	379	463
1	3	1	3	454	524	292	284	165
1	3	2	1	3379	1294	842	854	164
1	3	2	2	790	1299	142	214	188
1	3	2	3	260	714	111	96	42
1	3	3	1	767	2233	285	299	370
1	3	3	2	30	0	33	14	0
1	3	3	3	0	0	0	0	0

Table A.1: Total Deviation for Replication 1

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
2	1	1	1	3025	2877	837	685	405
2	1	1	2	561	374	210	232	195
2	1	1	3	156	331	95	95	48
2	1	2	1	353	1093	171	197	155
2	1	2	2	235	650	46	40	35
2	1	2	3	21	35	19	23	256
2	1	3	1	334	171	33	28	59
2	1	3	2	50	101	12	2	4
2	1	3	3	28	0	5	10	29
2	2	1	1	4546	4477	1422	1701	791
2	2	1	2	1616	1337	469	472	273
2	2	1	3	228	455	171	137	49
2	2	2	1	739	1019	382	238	198
2	2	2	2	486	1524	57	13	106
2	2	2	3	206	223	24	34	83
2	2	3	1	1394	4089	412	313	448
2	2	3	2	174	614	31	22	65
2	2	3	3	20	0	3	3	6
2	3	1	1	7774	8313	2239	2589	1470
2	3	1	2	2073	3482	788	698	738
2	3	1	3	2043	2171	478	879	642
2	3	2	1	7485	8881	1688	1408	1481
2	3	2	2	90	1806	54	53	193
2	3	2	3	143	93	49	95	46
2	3	3	1	2541	5525	777	367	509
2	3	3	2	243	1353	101	72	27
2	3	3	3	86	17	15	54	64

Table A.1.Total Deviation for Replication 1 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
3	1	1	1	5092	8292	1075	1134	579
3	1	1	2	2982	3101	409	420	279
3	1	1	3	1040	924	253	267	178
3	1	2	1	1110	1172	293	211	79
3	1	2	2	1562	1418	122	83	98
3	1	2	3	46	91	28	35	25
3	1	3	1	706	1380	146	112	65
3	1	3	2	22	297	7	9	22
3	1	3	3	41	152	5	2	23
3	2	1	1	8666	6541	2053	2213	742
3	2	1	2	1544	2353	663	506	281
3	2	1	3	1547	1258	222	240	296
3	2	2	1	5492	7577	1222	735	588
3	2	2	2	248	1480	99	53	155
3	2	2	3	137	164	72	44	114
3	2	3	1	602	1667	224	131	179
3	2	3	2	172	493	36	23	43
3	2	3	3	108	250	15	0	33
3	3	1	1	7918	7868	1951	2123	1244
3	3	1	2	5156	2534	1093	1246	1097
3	3	1	3	4276	4206	755	872	2206
3	3	2	1	2095	4909	813	494	368
3	3	2	2	2845	3547	100	157	366
3	3	2	3	595	1488	206	117	57
3	3	3	1	4256	6982	918	845	751
3	3	3	2	248	583	141	117	48
3	3	3	3	174	339	38	38	76
AVERAGE				1353.2	1690.0	353.3	348.3	227.9
MINIMUM				0	0	0	0	0
MAXIMUM				8666	8881	2201	2944	1391

Table A.1. Total Deviation for Replication 1 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
1	1	1	1	726	948	542	963	183
1	1	1	2	358	509	207	200	147
1	1	1	3	130	101	128	135	34
1	1	2	1	155	249	71	57	20
1	1	2	2	19	60	17	17	9
1	1	2	3	31	0	9	5	13
1	1	3	1	320	446	71	15	89
1	1	3	2	5	36	0	0	0
1	1	3	3	9	7	5	3	10
1	2	1	1	1473	1832	676	682	406
1	2	1	2	1402	1389	439	401	419
1	2	1	3	252	408	106	181	361
1	2	2	1	1873	1887	275	247	265
1	2	2	2	201	599	76	34	42
1	2	2	3	119	245	48	30	15
1	2	3	1	1258	497	165	115	65
1	2	3	2	49	81	6	19	9
1	2	3	3	0	0	0	0	0
1	3	1	1	6766	7710	2162	2019	830
1	3	1	2	910	916	267	348	504
1	3	1	3	164	706	107	133	453
1	3	2	1	269	553	166	146	213
1	3	2	2	317	696	65	69	23
1	3	2	3	77	32	60	45	631
1	3	3	1	1304	1889	173	372	97
1	3	3	2	68	336	38	55	12
1	3	3	3	29	8	22	29	0

Table A.2: Total Deviation for Replication 2

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
2	1	1	1	2118	2009	590	565	341
2	1	1	2	954	1053	399	479	343
2	1	1	3	381	490	161	149	329
2	1	2	1	1845	3419	328	294	334
2	1	2	2	35	89	21	22	46
2	1	2	3	19	45	22	8	23
2	1	3	1	261	1322	62	127	113
2	1	3	2	140	278	34	20	44
2	1	3	3	57	28	19	10	11
2	2	1	1	4588	6110	1223	1214	814
2	2	1	2	1188	1142	408	446	281
2	2	1	3	939	1272	226	316	424
2	2	2	1	1447	1758	520	416	372
2	2	2	2	15	128	35	50	77
2	2	2	3	44	59	30	26	35
2	2	3	1	499	2814	135	37	88
2	2	3	2	27	93	6	10	30
2	2	3	3	14	0	2	3	56
2	3	1	1	7867	8084	1264	1493	957
2	3	1	2	1166	1311	448	497	513
2	3	1	3	412	1032	307	359	184
2	3	2	1	1155	3405	1468	504	258
2	3	2	2	880	1465	193	171	106
2	3	2	3	178	253	26	24	110
2	3	3	1	3803	5571	1048	1289	543
2	3	3	2	196	515	60	72	29
2	3	3	3	128	471	38	45	147

Table A.2. Total Deviation for Replication 2 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
3	1	1	1	4177	3038	983	1150	701
3	1	1	2	926	1664	319	342	378
3	1	1	3	394	554	219	225	235
3	1	2	1	3067	4504	538	795	244
3	1	2	2	131	162	32	34	95
3	1	2	3	43	165	42	53	58
3	1	3	1	1294	2379	322	277	303
3	1	3	2	400	776	86	40	159
3	1	3	3	56	61	4	5	19
3	2	1	1	8208	5236	1709	1303	1275
3	2	1	2	1194	2583	679	581	483
3	2	1	3	463	1191	367	302	308
3	2	2	1	3633	7858	597	621	402
3	2	2	2	426	1275	93	60	348
3	2	2	3	74	139	39	36	134
3	2	3	1	2534	4876	556	798	492
3	2	3	2	165	261	69	22	47
3	2	3	3	41	27	12	1	10
3	3	1	1	9604	9305	2239	2509	2206
3	3	1	2	7423	5451	1253	1381	860
3	3	1	3	1594	3010	647	525	705
3	3	2	1	10611	7249	1294	892	781
3	3	2	2	1270	2077	309	201	495
3	3	2	3	1458	2232	209	192	122
3	3	3	1	4450	7992	677	542	664
3	3	3	2	1134	1345	288	76	137
3	3	3	3	81	77	59	62	101
AVERAGE				1401.2	1751.3	353.0	345.1	263.8
MINIMUM				0	0	0	0	0
MAXIMUM				9604	9305	2239	2509	2206

Table A.2. Total Deviation for Replication 2 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
1	1	1	1	527	694	305	317	173
1	1	1	2	156	236	74	92	45
1	1	1	3	47	186	50	50	27
1	1	2	1	491	1364	159	113	93
1	1	2	2	90	205	49	50	25
1	1	2	3	53	76	6	7	5
1	1	3	1	295	174	86	51	20
1	1	3	2	36	12	7	5	175
1	1	3	3	9	16	1	0	1
1	2	1	1	3018	4211	1043	1394	588
1	2	1	2	588	1065	207	309	231
1	2	1	3	160	194	95	142	77
1	2	2	1	1426	2637	177	281	289
1	2	2	2	159	456	32	21	54
1	2	2	3	10	5	10	10	21
1	2	3	1	925	1803	506	412	170
1	2	3	2	13	0	6	10	46
1	2	3	3	20	5	1	0	11
1	3	1	1	3791	3446	1945	1925	543
1	3	1	2	565	1195	211	342	314
1	3	1	3	227	905	123	123	212
1	3	2	1	2315	2512	838	600	757
1	3	2	2	423	679	92	156	170
1	3	2	3	102	63	79	64	3
1	3	3	1	1002	2461	250	93	116
1	3	3	2	300	126	34	19	16
1	3	3	3	81	103	41	31	28

Table A.3: Total Deviation for Replication 3

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
2	1	1	1	2894	3972	676	678	428
2	1	1	2	436	346	205	210	152
2	1	1	3	574	546	154	172	148
2	1	2	1	2099	2330	685	665	322
2	1	2	2	157	238	45	37	40
2	1	2	3	16	34	25	35	31
2	1	3	1	481	562	67	55	83
2	1	3	2	8	27	5	1	18
2	1	3	3	55	0	9	3	0
2	2	1	1	2937	4038	1269	1051	698
2	2	1	2	1594	1858	463	530	408
2	2	1	3	899	1309	185	164	183
2	2	2	1	3649	6116	528	574	349
2	2	2	2	268	383	57	67	152
2	2	2	3	17	23	7	15	25
2	2	3	1	1323	1034	202	147	145
2	2	3	2	49	362	2	10	14
2	2	3	3	5	0	5	5	13
2	3	1	1	5107	4469	2618	2364	1063
2	3	1	2	1397	2073	782	809	674
2	3	1	3	449	1021	379	343	196
2	3	2	1	4948	4499	1017	659	665
2	3	2	2	897	1204	160	99	160
2	3	2	3	358	140	82	49	54
2	3	3	1	5322	10250	737	464	1386
2	3	3	2	733	625	85	45	14
2	3	3	3	477	866	86	77	49

Table A.3. Total Deviation for Replication 3 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
3	1	1	1	2726	3534	570	669	440
3	1	1	2	485	723	355	367	202
3	1	1	3	263	461	175	187	65
3	1	2	1	2933	1818	538	441	316
3	1	2	2	110	508	50	27	41
3	1	2	3	42	34	14	13	22
3	1	3	1	1710	1849	206	219	308
3	1	3	2	89	36	21	10	13
3	1	3	3	306	30	5	3	6
3	2	1	1	8538	7562	2376	2060	1029
3	2	1	2	2758	3476	800	786	521
3	2	1	3	597	1245	285	271	248
3	2	2	1	7059	7322	1764	2257	932
3	2	2	2	885	1587	87	81	70
3	2	2	3	110	357	10	9	64
3	2	3	1	1151	1354	182	127	163
3	2	3	2	967	3261	21	36	23
3	2	3	3	31	53	13	15	0
3	3	1	1	6718	7552	2716	2790	1531
3	3	1	2	1461	3309	902	1045	770
3	3	1	3	2921	3189	651	942	936
3	3	2	1	5054	7036	1154	1317	794
3	3	2	2	522	4712	309	248	269
3	3	2	3	594	1521	211	131	65
3	3	3	1	12973	9572	896	771	535
3	3	3	2	1202	2486	175	79	92
3	3	3	3	248	656	107	56	77
AVERAGE				1437.4	1832.6	389.5	381.5	261.8
MINIMUM				5	0	1	0	0
MAXIMUM				12973	10250	2716	2790	1531

Table A.3. Total Deviation for Replication 3 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
1	1	1	1	1232	860	442	458	406
1	1	1	2	129	255	115	115	104
1	1	1	3	52	68	52	52	89
1	1	2	1	2419	3664	638	620	310
1	1	2	2	28	11	31	15	73
1	1	2	3	77	19	30	28	9
1	1	3	1	102	362	71	47	26
1	1	3	2	13	5	4	10	18
1	1	3	3	20	0	6	1	2
1	2	1	1	3955	5262	1660	1521	1020
1	2	1	2	326	244	164	115	114
1	2	1	3	490	750	241	209	183
1	2	2	1	857	772	309	147	187
1	2	2	2	81	268	20	30	26
1	2	2	3	28	30	9	10	24
1	2	3	1	339	679	54	17	74
1	2	3	2	86	85	23	14	22
1	2	3	3	0	4	0	0	0
1	3	1	1	3440	5678	1198	1030	616
1	3	1	2	1333	2549	567	634	500
1	3	1	3	202	219	197	207	119
1	3	2	1	1953	4025	263	362	413
1	3	2	2	555	852	153	147	92
1	3	2	3	115	104	35	73	43
1	3	3	1	2165	2877	546	296	362
1	3	3	2	232	86	134	58	26
1	3	3	3	50	71	50	9	0

Table A.4: Total Deviation for Replication 4

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
2	1	1	1	2843	2583	823	775	444
2	1	1	2	946	703	143	196	240
2	1	1	3	479	381	126	99	180
2	1	2	1	1429	3324	252	186	119
2	1	2	2	42	100	8	4	20
2	1	2	3	27	61	7	7	15
2	1	3	1	259	314	127	91	92
2	1	3	2	131	232	9	15	34
2	1	3	3	3	2	6	4	0
2	2	1	1	14177	17509	3859	2529	1432
2	2	1	2	2406	2358	431	491	424
2	2	1	3	341	557	207	213	228
2	2	2	1	1567	2306	584	629	570
2	2	2	2	469	924	128	120	164
2	2	2	3	52	184	54	32	26
2	2	3	1	1165	1315	278	81	113
2	2	3	2	41	14	28	6	99
2	2	3	3	8	0	4	0	4
2	3	1	1	9966	12311	2194	2055	1221
2	3	1	2	936	2668	576	565	465
2	3	1	3	291	977	400	423	270
2	3	2	1	3099	3610	1006	1127	507
2	3	2	2	244	455	59	79	35
2	3	2	3	64	91	70	55	119
2	3	3	1	4901	7727	627	521	310
2	3	3	2	1067	727	32	47	24
2	3	3	3	143	46	66	31	54

Table A.4. Total Deviation for Replication 4 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
3	1	1	1	2301	1829	792	805	389
3	1	1	2	377	413	279	257	271
3	1	1	3	361	306	165	180	220
3	1	2	1	1333	1838	375	276	313
3	1	2	2	63	386	27	15	47
3	1	2	3	40	37	42	36	11
3	1	3	1	3018	5248	274	508	247
3	1	3	2	115	438	23	6	102
3	1	3	3	7	55	4	8	27
3	2	1	1	4771	5129	2005	2097	948
3	2	1	2	1058	1082	469	560	328
3	2	1	3	699	1062	264	293	280
3	2	2	1	5583	5175	1077	770	519
3	2	2	2	226	324	83	67	93
3	2	2	3	14	30	11	9	38
3	2	3	1	1255	3391	680	471	451
3	2	3	2	2109	2598	4	80	171
3	2	3	3	32	6	30	0	54
3	3	1	1	4432	5628	2480	2571	1235
3	3	1	2	3623	2525	878	1036	838
3	3	1	3	1461	2092	736	678	1199
3	3	2	1	17096	24831	1340	1346	879
3	3	2	2	2156	4128	528	284	152
3	3	2	3	322	1686	61	66	134
3	3	3	1	1639	12841	721	474	398
3	3	3	2	346	853	165	75	97
3	3	3	3	78	107	25	25	144
AVERAGE				1504.8	2164.4	403.1	365.4	267.3
MINIMUM				0	0	0	0	0
MAXIMUM				17096	24831	3859	2571	1432

Table A.4. Total Deviation for Replication 4 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
1	1	1	1	1504	1390	456	543	266
1	1	1	2	334	653	152	160	66
1	1	1	3	101	139	101	98	53
1	1	2	1	828	658	207	141	131
1	1	2	2	57	86	21	37	31
1	1	2	3	19	19	19	19	16
1	1	3	1	61	200	9	7	16
1	1	3	2	6	16	9	5	3
1	1	3	3	0	0	0	0	16
1	2	1	1	1167	1778	539	721	297
1	2	1	2	1041	1296	443	402	302
1	2	1	3	381	355	243	194	168
1	2	2	1	2631	1152	375	373	143
1	2	2	2	108	221	29	47	45
1	2	2	3	11	12	13	13	42
1	2	3	1	1611	2638	251	193	454
1	2	3	2	19	28	10	19	12
1	2	3	3	120	0	7	0	13
1	3	1	1	1897	2495	1443	1557	766
1	3	1	2	931	942	372	520	338
1	3	1	3	757	893	398	401	272
1	3	2	1	1592	1009	608	496	182
1	3	2	2	192	457	58	123	59
1	3	2	3	133	169	86	91	69
1	3	3	1	1109	2250	204	118	185
1	3	3	2	782	272	103	43	3
1	3	3	3	43	22	12	0	65

Table A.5: Total Deviation for Replication 5

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
2	1	1	1	1282	1464	631	652	459
2	1	1	2	552	729	247	238	153
2	1	1	3	186	222	135	111	76
2	1	2	1	1237	1878	6532	311	363
2	1	2	2	412	833	73	73	96
2	1	2	3	27	77	33	17	17
2	1	3	1	426	766	47	81	44
2	1	3	2	1	143	11	11	19
2	1	3	3	48	49	2	8	0
2	2	1	1	5595	4914	987	1210	555
2	2	1	2	1116	2011	493	488	592
2	2	1	3	1202	1253	425	513	209
2	2	2	1	4253	5663	566	603	559
2	2	2	2	450	865	85	68	95
2	2	2	3	183	127	14	19	9
2	2	3	1	797	1365	163	70	154
2	2	3	2	153	1749	18	0	120
2	2	3	3	30	0	11	6	0
2	3	1	1	8499	7328	1940	1820	962
2	3	1	2	2257	3387	924	896	591
2	3	1	3	237	315	237	234	453
2	3	2	1	2132	4619	817	619	383
2	3	2	2	808	1968	183	186	228
2	3	2	3	199	493	78	76	407
2	3	3	1	1488	2278	516	499	316
2	3	3	2	225	420	33	46	42
2	3	3	3	201	287	33	53	26

Table A.5. Total Deviation for Replication 5 (continued)

Factors				Alternatives				
N	L	K	M	ER	EDD	STTF	RM	MOSA
3	1	1	1	2337	2767	1042	1095	644
3	1	1	2	1102	1381	1033	459	480
3	1	1	3	1366	1416	262	234	422
3	1	2	1	907	1195	254	271	181
3	1	2	2	496	975	37	33	21
3	1	2	3	475	717	62	50	19
3	1	3	1	871	2195	215	141	84
3	1	3	2	107	116	30	13	7
3	1	3	3	66	196	0	18	20
3	2	1	1	7370	8050	1410	1322	644
3	2	1	2	1191	2029	650	635	427
3	2	1	3	568	907	272	369	544
3	2	2	1	1256	2860	586	537	337
3	2	2	2	332	660	104	70	147
3	2	2	3	314	1344	46	46	65
3	2	3	1	7037	4510	483	228	269
3	2	3	2	26	32	19	18	36
3	2	3	3	1	0	0	0	33
3	3	1	1	20065	15047	4208	4461	2381
3	3	1	2	10584	6797	1531	1964	1750
3	3	1	3	1933	2333	654	764	846
3	3	2	1	5284	8060	1034	638	787
3	3	2	2	944	1541	303	195	244
3	3	2	3	178	65	98	97	41
3	3	3	1	1414	2277	299	232	255
3	3	3	2	740	2384	79	74	64
3	3	3	3	13	83	25	26	96
AVERAGE				1461.8	1657.9	446.1	360.7	268.5
MINIMUM				0	0	0	0	0
MAXIMUM				20065	15047	6532	4461	2381

Table A.4. Total Deviation for Replication 5 (continued)

Bibliography

[1] Adams, J., Balas, E. , Zawack, D., 'The Shifting Bottleneck Procedure for Job Shop Scheduling', *Management Science*, Vol.34, No 3, pp.391-401, 1988.

[2] Bartholdi, J. J. III and Platzman, L. K., 'Decentralized Control Of Automated Guided Vehicles On A Single Loop', *IIE Transactions,* Vol. 21, No 1, pp.76-81, 1989.

[3] Biemans, F.P. and Vissers, C.A., 'Reference Model for Manufacturing Planning and Control Systems', *Journal of Manufacturing Systems,* Vol.8, No.1,1989.

[4] Bozer, Y.A., 'Guided Vehicle Systems: Information/Control System Implications of Alternative Design and Operation Strategies', *Nato ASI Series, Advanced Information Technologies for Industrial Material Flow Systems,* Vol.F53, (Shimon N. Nof and Colin L. Moodie, ed.) Springer-Verlang, Berlin Heidelberg, 1989.

[5] Bozer, Y.A. and Srinivasan, M.M. 'Tandem Configurations For AGV Systems Offer Simplicity And Flexibility', *Industrial Engineering,* Vol.21, No.2, pp.23-27, 1989.

[6] Co, C.G. and Tanchoco, J.M.A., 'A review of research on AGVS Vehicle Management', *Engineering Costs and Production Economics*, Elsevier, Vol.21, pp.35-42, 1991.

[7] Dempster, M.A.H., Fisher, M.L., Jansen, L.,Lageweg,B.L., Lenstra, J.K. and Rinnooy Kan, A.H.G., 'Analytic Evaluation of Hierarchical Planning Systems', *Operations Research,* Vol.29, No.4, pp.707-716, 1981.

[8] Egbelu P.J. and Tanchoco, J.M.A., 'Characterization of Automatic Guided Vehicle Dispatching Rules', *International Journal of Productions Research,* Vol.22, No. 3, pp.359-374, 1984.

[9] Fox, M.S. 'Constraint Guided Scheduling - A Short History of Research at CMU', *Computers in Industry,* Elsevier, No.14, pp.79-88, 1990.

[10] Fox, M.S. and Smith, S., 'ISIS: A Knowledge-Based System For Factory Scheduling', *International Journal of Expert Systems,* Vol.1, No.1,pp.25-49, 1984.

[11] Gaskins, R.J. and Tanchoco, J.M.A., 'Flow Path Design For Automated Guided Vehicle Systems', *International Journal of Production Research,* Vol.25, No.5, pp.667-676, 1987.

[12] Gaskins, R.J., Tanchoco J.M.A. and Taghaboni F. 'Virtual Flow Paths For Free Ranging Automated Guided Vehicles On A Single Loop', *IIE Transactions,* Vol.21, No.1, pp.76-81, 1989.

[13] Goldratt, E., 'Optimized Production Timetables: A Revolutionary Program For Industry', *APICS* 23th *Annual International Conference Proceedings,* pp.172-176.

[14] Hax, A.C. and Meal, A.C., 'Hierarchical Integration of Production Planning and Scheduling', *Studies in Management Science,* Vol.1, Logistics, (M.A. Geisler, ed.), North Holland, New York.

[15] Jackson, R.H.F. and Jones, A.W.T. 'An Architecture for Decision Making in the Factory of the Future', *Interfaces,* Vol.17, No.6, pp.15-20, 1987.

[16] King, R.E. and Wilson C., 'A Review of Automated Guided Vehicle Systems Design and Scheduling,' *Production Planning and Control,* Vol.2, No.1, pp.44-51, 1991.

[17] Koff, G.A., 'Automatic Guided Vehicle Systems: Applications, Controls and Planning', *Material Flow*, Vol.4, pp.3-16, 1987.

[18] Kolen, A.W.J., Rinnooy Kan, A.H.G. and Trienekens, H.W.J.M., 'Vehicle Routing With Time Windows', *Operations Research*, Vol.35, No.2, pp.266-274, 1987.

[19] Leung, L.C., Khator, S.K. and Kimbler, D.L., 'Assignment of AGVS with Different Vehicle Types', *Material Flow*, Vol.4, pp.65-72, 1987.

[20] Maxwell, W.L. and Muckstadt, J.A. 'Design of Automated Guided Vehicle Systems', *IIE Transactions,* Vol.14, No.2, 1982.

[21] McGinnis, L.F., 'Some Issues in Controlling Integrated WIP Material Handling Requirements', *Nato ASI Series, Advanced Information Technologies for Industrial Material Flow Systems,* Vol.F53, (Shimon N. Nof and Colin L. Moodie, ed.) Springer-Verlang, Berlin Heidelberg, 1989.

[22] Morton, T.E., Rachamadagu, R.V. and Vepsalainen, A.P.J., 'Accurate Myopic Heuristics for Tardiness Scheduling', Working Paper, No. 36-83-84, GSIA, Carnegie Mellon University, 1984.

[23] Muscettola, N., 'Scheduling by Iterative Partition of Bottleneck Conflicts', Technical Report, CMU-RI-TR-92-05. The Robotics Institute, Carnegie Mellon University, 1992.

[24] 'Reference Model of Production Systems', Version 1, Digital Equipment Corporation and Nederlandse Philips Bedrijyen BV, 1987.

[25] Sadeh, N., 'MICRO-BOSS: A Micro-Opportunistic Factory Scheduler', *Carnegie Mellon University,* Technical Report, CMU-RI-TR-91-22, 1991.

[26] Smith, S.F., Ow, P.S., Muscettola, N., Potvin, J.Y. and Matthys, D.C., 'An integrated framework for generating and revising factory schedules', *Journal of Operational Research Society,* Vol.41, No 6, pp.539-552, 1990.

[27] Solberg, J.J. and Heim, J.A., 'Managing Information Complexity in Material Flow Systems', *Nato ASI Series, Advanced Information Technologies*

for Industrial Material Flow Systems, Vol.F53, (Shimon N. Nof and Colin L. Moodie, ed.) Springer-Verlang, Berlin Heidelberg, 1989.

[28] Solomon, M.M., 'Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints', *Operations Research,* No.35, pp. 254-265, 1987.

[29] Sule, D.R., 'Manufacturing Facilities, Location, Planning and Design', 1st ed., PWS-Kent, 1988.

[30] Tanchoco, J.M.A. and Sienrich, D., 'OSL, Optimal Single Loop Guide Paths for AGVS', *International Journal of Production Research*, Vol.30, No.3, 665-681, 1992.

[31] Taghaboni, F. and Tanchoco, J.M.A., 'A LISP Based Controller for Free-Ranging Automated Guided Vehicle Systems', *International Journal of Production Research,* Vol.26, No.2, pp.173-188, 1988.

[32] Walker, S.P., Premi, S.K., Besut, C.B. and Broadbent, A.J., 'Free-Ranging AGV and Scheduling System', *Proceedings of the Third International Conference on AGVS,* IFS Conferences Ltd., Stockholm, Sweden, October 1985.

VITAE

Haluk Yılmaz is born in 1969 in Konya. He is graduated from Ankara Fen Lisesi in 1987. He studied Industrial Engineering in Bilkent University. He received M.S. in Industrial Engineering from Bilkent University. Now, he is working in Koç Holding in Istanbul. His research interests include AGVS, scheduling, hierarchical planning and quality management. He is a member of IIE and TIMS.