SIMULTANEOUS SCHEDULING OF MACHINES AND THE MATERIAL HANDLING SYSTEM IN A FLEXIBLE MANUFACTURING SYSTEM

by

ÜMİT BİLGE

B.S. in I.E., Boğaziçi University, 1982

M.S. in I.E., Boğaziçi University, 1984

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor
of
Philosophy

Boğaziçi University

1991

SIMULTANEOUS SCHEDULING OF MACHINES AND THE MATERIAL HANDLING SYSTEM IN A FLEXIBLE MANUFACTURING SYSTEM

APPROVED BY

Prof Dr. Gündüz ULUSOY ....
(Dissertation Supervisor)

Doç. Dr. Nükhet YETİŞ N. Yetiş ....

Y. Doç. Dr. Gülay BARBAROSOĞLU ....

DATE OF APPROVAL ......6/8/1991......

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Prof. Dr. Gündüz ULUSOY for his invaluable guidance, continuous support , consideration, and encouragement as the supervisor of this dissertation.

I wish to express my sincere thanks to Doç. Dr. Nükhet YETİŞ for her comments and suggestions as a member of my dissertation committee.

My special thanks go to Yard. Doç. Dr. Gülay BARBAROSOĞLU for her constructive comments and kind help in reviewing the text.

I am grateful to my family without whose support this study would not have been accomplished.

ABSTRACT

The scheduling of material handling system is a crital issue in a Flexible Manufacturing System (FMS), although it has little importance in a job shop.The purpose of this dissertation is to exploit the interactions between the machine scheduling and the scheduling of the material handling system in an FMS and to integrate them by addressing them simultaneously.

In the FMS under consideration, the material transfer between machines is done by a number of identical Automated Guided Vehicles (AGVs) . Upon completing a loaded trip the AGV is designated to its next pick-up station. Therefore, the travel times of the empty trips depend on the ending and the starting points of the successive loaded trips assigned to a vehicle.This concept of sequence-dependent travel times increases the difficulty of the problem.

As a first step, the combined machine and material handling system scheduling problem is formulated as a nonlinear mixed integer programming model which turned out to be of intractable size for real-world problems. Then, the problem is decomposed into two subproblems, one having the characteristics of the machine scheduling problem while the other is a vehicle scheduling problem ,and an iterative solution procedure is developed. At each iteration, a new machine schedule, generated by a heuristic procedure, is investigated for its feasibility to the vehicle scheduling subproblem. To do this, the operation completion times obtained from the machine schedule are used to construct "time windows" for each material handling trip, and the second subproblem is handled as a "sliding time window" problem. The procedure is numerically tested on a number of example problems.

ÖZET

Malzeme taşıma sisteminin çizelgelenmesi, atelye tipi üretimde pek önem taşımadığı halde, esnek üretim sistemleri söz konusu olduğunda önem kazanmaktadır. Bu tezde, makinaların çizelgelenmesi ile malzeme taşıma sisteminin çizelgelenmesi arasındaki etkileşimlerin araştırılması ve birlikte ele alınmak suretiyle bütünleşik hale getirilmeleri amaçlanmaktadır.

Söz konusu esnek üretim sisteminde, makineler arasında malzeme taşınmasının belli sayıda birbirine eş Otomatik Kumandalı Taşıyıcı tarafından gerçekleştirildiği varsayılmaktadır. Bir taşıma işini tamamlayan taşıyıcı yeni işinin başlangıç noktasına yönelmektedir. Dolayısıyla, iki taşıma işinin arasındaki boş yolculuğun süresi, taşıyıcının arka arkaya görevlendirildiği işlerin bitiş ve başlangıç yerlerine bağlıdır. Bu, sıraya bağlı yolculuk süresi kavramı problemin zorluğunu arttırmaktadır.

Makinalar ve malzeme taşıma sistemi eş zamanlı çizelgeleme problemi, ilkin, bir doğrusal olmayan karışık tam sayı programlama modeli olarak tanımlanmıştır. Ancak, bu modelin gerçek uygulamalara olanak vermeyecek boyutlarda olduğu görülmüştür. Bunun üzerine, problem, biri makina çizelgeleme, diğeri taşıt çizelgeleme problemi özelliklerini taşıyan iki alt probleme ayrıştırılmış, ve bir döngü yöntemi geliştirilmiştir. Her çevrimde, sezgisel bir yöntemle yeni bir makina çizelgesi yaratılmakta ve bunun araç çizelgeleme alt problemine uygunluğu araştırılmaktadır. Bu yapılırken, makina çizelgesinden gelen işlem tamamlanma zamanları, taşıma işlerinin başlama zamanları için birer "zaman aralığı" oluşturmakta kullanılmakta ve ikinci alt problem "kaydırılabilir zaman aralıklı araç çizelgeleme" problemi olarak ele alınmaktadır. Yöntem, örnek problemler üzerinde sayısal olarak sınanmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

The purpose of this dissertation is to exploit the interactions between the machine scheduling and the scheduling of the material handling system in a Flexible Manufacturing System (FMS) and to integrate them by addressing them simultaneously.

An FMS consists of a set of Numerically Controlled (NC) or Computer Numerically Controlled (CNC) machines having automatic tool interchange capability, interconnected by an automated material handling system and controlled by an integrated computer system. The high level of processing flexibility created by this configuration results in negligible set-up costs and alternate routing possibilities which allow simultaneous production of small to medium batches of a variety of part types with the efficiency of mass production systems and the flexibility of job shops.

As implied by the above definition, material handling is one of the essential components in an FMS. In fact, efficient scheduling of the material handling system (MHS) is critical to the overall efficiency of the FMS.

The scheduling of MHS merits serious consideration in an FMS, although it may not be that significant in a job shop. In job shops, major portion of the manufacturing lead time is represented by the queueing time, and the transportation times are low compared to the processing times including the set-up times. Consequently, efficient scheduling of transportation devices may have little importance. In an FMS, however, the set-up times are low or non-existent and travel times can be comparable to the processing times. The complex job flow due to the large number of parts produced and possible alternate routings, and the limited buffers are further issues that make the timeliness of material supply and removal a critical consideration for productivity. In other words, the more flexible the production environment is, the more crucial the efficient scheduling of MHS becomes.

In this dissertation, an FMS where material transfer between machines is done by a number of identical Automated Guided Vehicles (AGVs) is considered, and an FMS scheduling problem where the machines and the material handling system are scheduled simultaneously is addressed.

As a first step, the combined machine and material handling system scheduling problem is formulated as a nonlinear mixed integer programming model which turned out to be of intractable size for real-world problems. However, its special structure led to another approach to the problem. Based on this structure, the problem is decomposed into two subproblems, one having the characteristics of the machine scheduling problem while the other is a vehicle scheduling problem.

The intention is to schedule operations with respect to some objective (i.e. minimization of the makespan) ,while making sure that this schedule is consistent with the constraints arising from the material handling system of the FMS. An iterative solution procedure is developed which searches a good solution to machine scheduling subproblem that is also feasible to the vehicle scheduling subproblem . At each iteration, a new machine schedule, generated by a heuristic procedure, is investigated for its feasibility to the vehicle scheduling subproblem, which is handled as a "sliding time window" problem.

The dissertation is organized as follows. Chapter 2 is a survey on FMS where basic concepts and the related research are introduced. Chapter 3 provides problem definition and the nonlinear mixed integer programming formulation. Chapter 4 introduces the subproblems. Chapter 5 describes the algorithm for the vehicle scheduling subproblem. Chapter 6 outlines the iterative solution procedure. Chapter 7 presents a simple heuristic procedure to be used for comparing the efficiency of the iterative algorithm. Chapter 8 reports numerical experimentation on small example problems. Chapter 9 states the conclusions and the directions for further research.

II. FLEXIBLE MANUFACTURING SYSTEMS,BASIC CONCEPTS AND LITERATURE SURVEY

The origins of flexible manufacturing technology lie in the desire to automate small batch manufacturing (generally smaller than 50 units). For high production volumes and output rates, transfer lines represent the most efficient method, but the fixed automation used becomes inappropriate when product variety increases and production volume decreases. However, for many industrial nations the relative importance of small batch manufacturing is surprisingly high, accounting about 10 percent of the GNP [1]* (Figure 2.1). Thereby arose the concern for a new automation concept which retains the flexibility of a classic job shop in product variety while overcoming its typical inefficiencies such as low productivity and machine utilization rates, high work-in-processs (WIP) and finished product inventories. This, together with technological improvements such as NC machines, resulted in the development of flexible manufacturing systems. First examples emerged in metal-working industry around 1970's. By 1981, the number of FMS installations in operation was about 25 in the United States, 40 in Japan, and 50 in Europe. By the beginning of 1985, the FMS population had reached an estimated 300 worldwide [2].

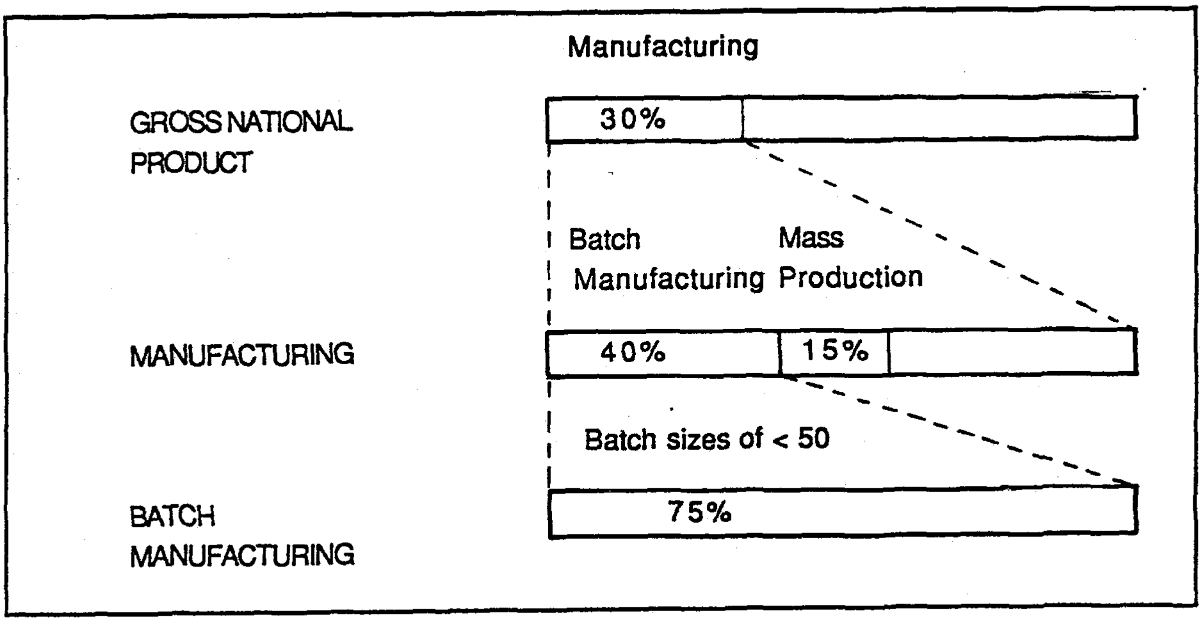


Figure 2.1 The significance of small batch manufacturing [1]

In this chapter, first, basic concepts related to FMS are defined briefly and a few FMS applications are described to make these concepts clear. Then, the OR/MS problems arising from the design and use of FMS are overviewed in conjunction with the related research in literature with special emphasis given to the FMS scheduling problem which is the scope of this dissertation.

2.1. Basic Concepts

FMS represents one of the highest levels of achievement in current technology. It incorporates many individual automation concepts into an integrated system. These concepts and technologies include:

-Flexible automation,

-Group Technology (GT),

-CNC machine tools,

-Automated material handling between machines,

-Computer control of machines (DNC) and material handling.

2.1.1. Flexible Automation

Flexible automation is an extension of programmable automation examplified by stand-alone NC machines and industrial robots. It introduces the capability of producing a variety of part types without loosing production time for changeovers. Part programs can be changed and the physical set-up (tools, fixtures, machine-settings) can be altered with no lost time. Consequently, no batching is necessary (economic batch size reduces to one) and variable mixtures of parts can be produced simultaneously.

Average machine utilization and production increases. These permit the output rate of each product to be set at its corresponding demand rate, thus reducing the WIP and final product inventories.

Furthermore, flexible automation reduces the manpower requirement: The process requires little human tending, manual labor possibly necessary only for some loading unloading and maintenance operations. In the ideal, the process could be operated continuously, 24 hours per day. The day shift might be used to organize the process (e.g. prepare raw materials, tooling, routine maintenance, etc.) so that production could continue unattended during the two night shifts.

Ability to accommodate design changes readily, and consistency of quality are other advantages of flexible automation.

However, there are also some disadvantages of this still relatively new technology: The systems are expensive and complex; expertise is difficult to find; lengthy operation and maintenance training is necessary; significant amount of software writing is required; it is difficult to integrate devices from different manufacturers; and it may take several years to implement such systems.

2.1.2. Group Technology

GT principles and procedures are used to form part families and flexible manufacturing cells. However, this grouping is a logical one, a software issue rather than a physical, hardware layout concept as in GT. The modularity and flexibility of the material handling system enables logical grouping of machines according to current production needs.

2.1.3. CNC Machine Tools

A CNC machine tool is a general purpose machine tool controlled by a dedicated computer located very near to it. Part recognition, part programs, automated tool changing, positioning of part, dimension control, failure diagnostics are controlled by this computer.

Tools for operations that can be performed by the machine are stored in a tool magazine, circular or rectangular in shape, which is mounted on one side of the machine. Tool magazines can be of two types : i) Changable, ii) Permanent [3]. The first type has a capacity of about 25 tools and can be changed with all tools loaded. The capacity for the permanent type is larger (about 60 tools). They can not be removed but the tools are exchanged. (Figure 2.2) .The magazine is indexed around so that the tool needed for the next operation can be moved in advance adjacent to the loading station- this device is called an automatic tool changer (ATC). A gripper picks up the tool that is currently on the spindle and places it in the magazine, then picks the new tool and inserts it in the machine.

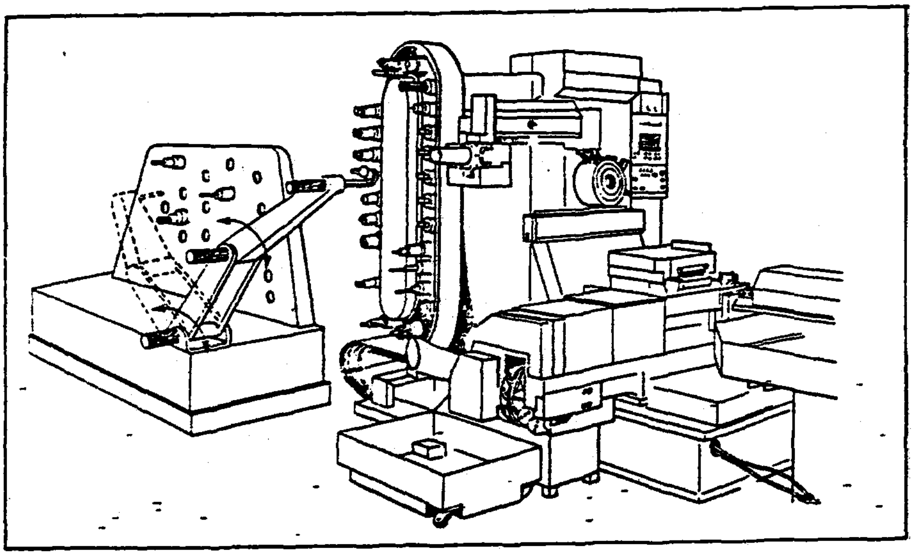


Figure 2.2 Permanent tool magazine and the concept of simple AGV/robot for tool changing [4]

Machines are also equipped with automatic pallet changers (APC) that facilitate unloading of the machined workpiece and loading of the new workpiece. (Pallets are standardized conveyance devices for the workpieces, that can be handled by the MHS and the machine tools). Figure 2.3 shows basic types of APC . Once the new pallet is mounted on the machine's table the workpiece is identified by an optical device that reads a part number or identifies the shape in some way.

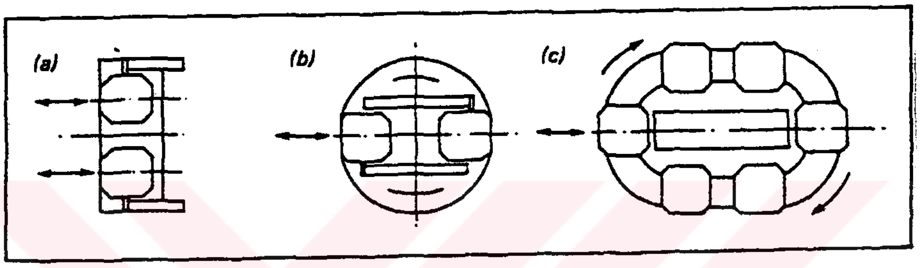


Figure 2.3 The basic types of APC : (a) parallel, (b) rotary, and (c) multiple [4]

2.1.4. Automated Material Handling System

The flexibility inherent in FMS is largely due to the MHS flexibility-that is the ability of the MHS to handle different part configurations in a number of different routes. Desired level of flexibility can be achieved through the choice of the particular layout of the MHS (i.e. line, loop, ladder, open-field or robot centered [2]), the equipment to be used (robots, conveyors, stacker-cranes, automated-guided vehicles) and their compatibility with computer control. The reader will find more about the AGV systems , in section 2.2.

2.1.5. Computer Control of Machines (DNC) and Material Handling

The operation of a flexible manufacturing system is coordinated through a central computer that gives management/manufacturing instructions, monitors the status, conducts traffic and tool control and collects information about the performance of the process.

2.2 Automated Guided Vehicle Systems

According to the Material Handling Institute (MHI), "AGV systems feature battery powered driverless vehicles with programming capabilities for path selection and positioning" [5]. The use of AGVs dates back to the 1950s and they continue to be a viable and flexible handling alternative for many low to medium flow, discrete parts handling requirements. In this section, the basic types of AGVs and several functions that must be performed to operate these systems successfully will be described briefly.

2.2.1. Vehicle Types

There are six basic types of AGVs (Figure 2.4) [6]:

-Towing Vehicles

-Unit Load Vehicles

-Pallet Trucks

-Fork Trucks

-Light Load Vehicles

-Assembly Line Vehicles

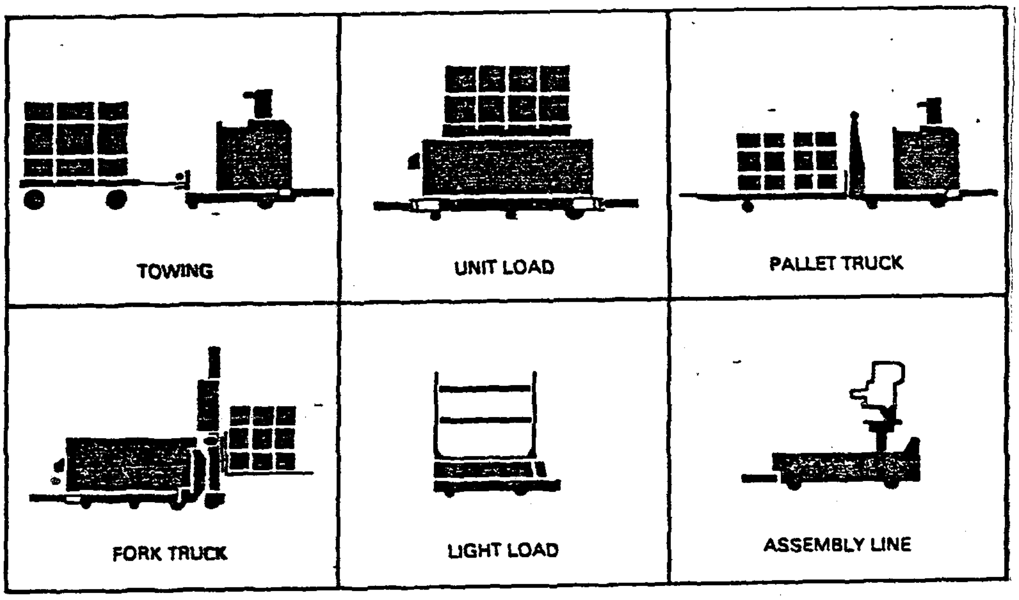


Figure 2.4 Automated Guided Vehicle types [6]

AGVs towing vehicles were the first type introduced and are still popular. Towing vehicles can pull a multitude of trailer types and have capacities ranging from 3630 to 22680 kg. Applications usually involve the bulk movement of product over large distances, into and out of warehouse areas, sometimes between buildings, outdoors.

AGVs unit load vehicles are used to move unit loads from one station to another, where distances are relatively short, but volumes are high .They are often equipped for automatic loading and unloading by means of powered rollers, moving belts, mechanized lift platforms, or other devices. Travelling speed is in the range of 30 m/min to 60 m/min. The unit load carrier is pictured in Figure 2.5.

Pallet trucks are designed to move palletized loads to and from the floor level and to eliminate the need for fixed load stands. In many applications, the vehicle is manually steered in the loading areas. A human worker backes the vehicle into the loaded pallet, uses its forks to lift the load slightly, drives it to the guide path and programs it to its destination. Then the vehicle proceeds automatically to the destination for unloading and returns empty to the loading area. The capacity ranges up to 2700 kg.

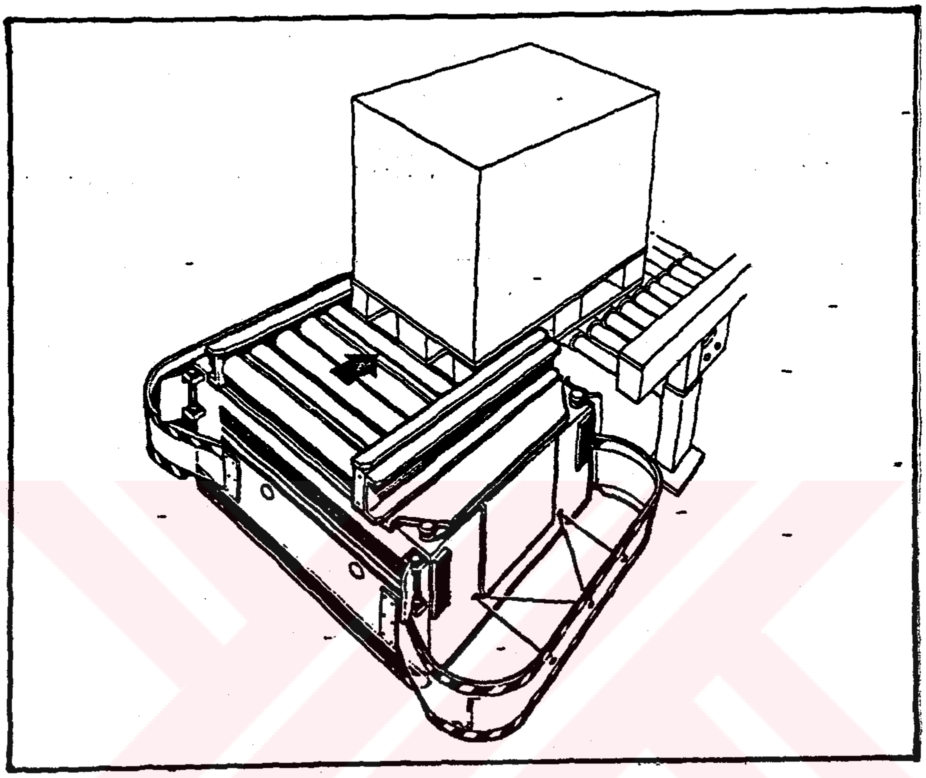


Figure 2.5 Unit load carrier [6]

Fork trucks are relatively new guided vehicles which can achieve significant vertical movement of their forks to service palletized loads at floor level, on stands and, in some cases, in racks.

Light load AGVs are variations of unit-load carriers which have capacities of about 225 kg or less and are used to transport small parts, baskets or other light loads through plants of limited size engaged in light manufacturing.

Assembly line vehicles are an adaptation of the light load AGVs for applications involving serial assembly processes.

2.2.2. Vehicle Guidance and Routing

The method by which the AGV pathways are defined and the vehicle control systems that follow the pathways are referred to as guidance systems.

The "rail-guided" AGV is the nearest device to a conveyor. It accelerates and runs fast and is suitable for heavy workpieces, but once the rails are installed they tend to be an fixed part of the plant. Furthermore, rail-guided AGVs can normally travel along straight tracks only.

"Wire-guidance" is one of the principal methods currently in use. The wires are usually embedded in a small channel cut into the surface of the floor, and a frequency generator provides a low frequency guidance signal carried in the wire. Two sensors (coils) are mounted on the vehicle on either side of the guide wire to detect the voltage, which is fed into the controller. To keep the AGV following the wire, the controller is continually actuating a steering motor to eliminate the voltage differential between the two coils.

On some AGVs, "light-guidance" is used: A light illuminating a fluorescent dye painted on the floor or tape laid on the floor. Photosensors detect the fluorescence and operate in a similar way to the scanning coils. The paint guidance system is useful in environments where electrical noise would render the guide wire system unreliable or when wire installation would be inappropriate. Light guidance has low installation costs, and routes can be changed easily, but the paint strip must be kept clear and unscratched.

Recently, off-wire technology has been introduced where vehicles are guided on software programmed paths by using various means for calculating their current position.

In "dead-reckoning", the expected position of the vehicle is calculated by the vehicle's computer based on the number of wheel rotations referenced to a starting point of motion. It is frequently used for travel off-wire for short distances-to cross regions where wires cannot be installed.

"Laser-guidance" is another method, where a scanner on the vehicle scans the area, and each triggered target transmits a bar-code by reflecting the laser-beam. The vehicle uses this information to calculate its position relative to two or more targets and compares it to the stored path program. (Figure 2.6).

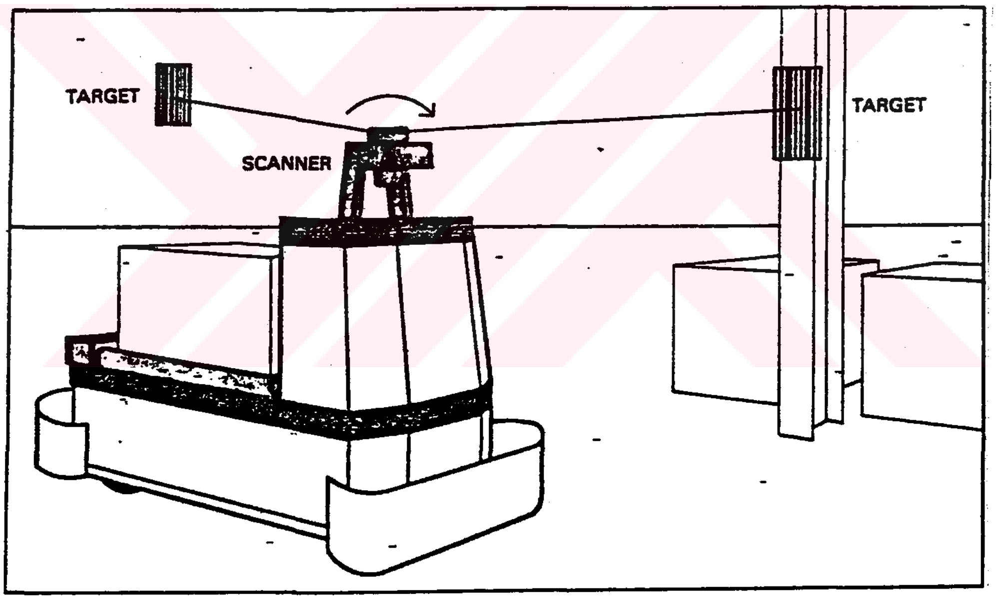


Figure 2.6 Laser guidance allows the use of AGVs without any tracks [6]

Guidance methods such as ultrasonic imaging of the surroundings, inertial navigation by utilizing gyroscopes, radio frequency communications between the moving vehicle and fixed host computer, and many others are under research.

Routing in an AGV system is concerned with the problem of selecting among alternative pathways available to a defined destination point in the system. When a vehicle approaches a branching point in which the guide path splits into two (or more) directions, a decision must be made as to which path the vehicle should take. There are two routing techniques used in commercial AGV systems:

i. Frequency select method

ii. Path switch select method

In the frequency select method, guide wires leading into the two separate paths at the branch have different frequencies. As the vehicle enters the decision point, it reads a marker on the floor that identifies the location. The vehicle, depending on its programmed destination, selects which frequency to follow, and the routing is automatically accomplished.

In the path switch select method a single frequency is used throughout the guide path layout. As the vehicle enters a decision point, it activates a floor-mounted switching device which causes one path to be turned on while the other paths are turned off. The vehicle has only one live path-the one that it has communicated to the switch device based on which stop locations it was routed to.

2.2.3. Traffic Control and Safety

The purpose of traffic control for an AGV system is to prevent collisions between vehicles travelling along the same guide path in the layout.

There are two methods in general use:

i. Zone control

ii. Forward sensing

In zone control, the AGV system layout is divided into separate zones, and only one vehicle is permitted in a given zone at a time. Zone control can be implemented by a central controller or by using individual control units for each zone which are linked together in sequence and actuated by the vehicle entering the zone.

Forward sensing involves the use of some form of sense system (optical, ultrasonic, or bumper) to detect the presence of a vehicle in front it. This method is usually used on long straight sections of path since they are not much effective at turns and convergence points where forward vehicles may not be directly in front of the sensor.

2.2.4. System Management

Managing the operations of an AGV system is primarily concerned with vehicle dispatching (e.g., to perform pick-ups and deliveries, where they are needed in a timely and efficient manner). Current dispatching methods, in increasing order of automation level, include:

i. On-board control panel
ii. Remote call stations
iii. Central computer control

Under the first method, the vehicle is dispatched to one or more pick-up/deliver (P/D) stations by the operator who programs the AGV through the control panel mounted on the vehicle. If manual load transfer is used and if the vehicle serves a few P/D points located far apart, this method may be sufficient.

However, if the load transfer is automated then no operator presence is required at P/D stations and the use of fixed control panels mounted near the various stations along the layout will allow the AGV to respond the changing demand patterns in the system. This method permits a vehicle to be stopped at a given station and its next destination to be programmed from the remote call panel.

In large factory and warehouse systems involving a high level of automation, central computer control is used to accomplish automatic dispatching of vehicles according to a preplanned schedule and/or in response to calls from various P/D stations. The central computer keeps track of each vehicle individually (including their performance statistics) and issues commands concerning their destinations and operations to perform.

To accomplish the system management function, real-time status of the AGV system is monitored by means of some form of graphic display (i.e. a CRT color graphic display). These systems alert the operator when error or abnormal conditions are detected, and allow him interfere through central terminal, if necessary.

2.3. FMS Applications

2.3.1 The SCAMP Flexible Manufacturing System

The SCAMP system [1,4] was installed in 1983, in Colchester, England. It was initially capable of producing 40 different turned components, but with an ultimate capability of several hundred different parts. The total cost of the system was about $4.5 million. There are nine machine tools arranged in a row, in one cell, but in functional groups. Apart from a shaper, each machine is served by a free-standing robot. The carry and free conveyor system consists of four rows, the two inner ones serving as a recirculating highway, with turntables at the ends, while the outer ones are used as transfer stations-to the machines on one side and for loading and unloading the system at the other. Gates are used to transfer pallets to load/unload stations, where probes read identity codes on the pallets, and on instructions from control computer stop the pallet or let it continue on its way. Parts are produced in batches of 25-100. Cycles are short (3-10 min). Inner conveyors provide a buffer. (Figure 2.7).

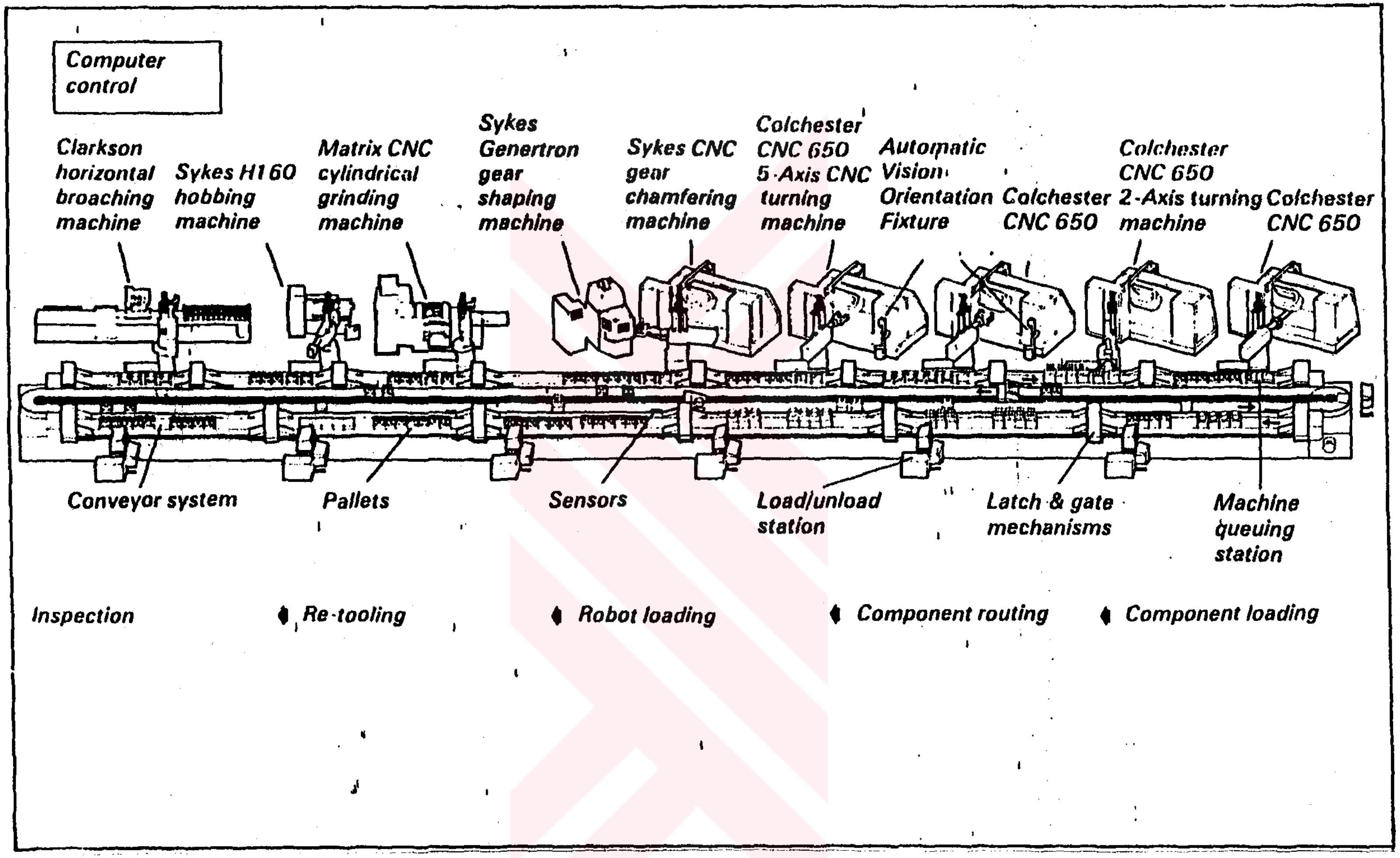


Figure 2.7 The SCAMP FMS [4]

2.3.2. The CITROEN Flexible Manufacturing System

This system installed in Citroen Constructions Mecaniques plant at Meudon near Paris is designed to machine prototype components in batches from one to about 50 [4]. Usually once one batch is complete no further batches of same type are expected, but size of all components fall within well-known limits. The system has been designed based on a minimum cycle of 10 min. Citroen installed two horizontal machining centers (a third can be added) ,served by wire-guided AGVs. A Kremlin robot is used to wash workpieces; there is a computer-controlled measuring machine and, of course, tool and pallet areas in the system. There are 56 pallets in the system to allow unmanned operation in the third shift, and in the other shifts, two men load work pieces onto pallets at six stations. Then, there is a row of 15 pallet stations, the AGV transferring these directly to the APCs at the machine as necessary. Because of the large variety of workpieces, each machining center has an ATC with 50 tools while there is a store for further 600 tools. Transfer is fully automatic, carried out by AGVs on a separate route and pick-and-place devices at the store and at the rear of each machining center (Figure 2.8).

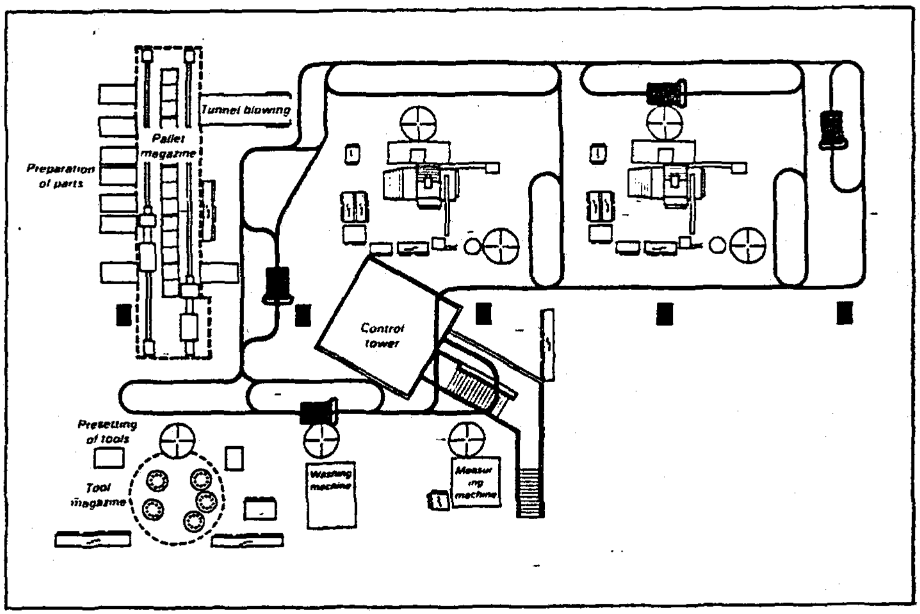


Figure 2.8 The CITROEN FMS [4]

2.3.3. VOUGHT Aero Products Division FMS

This system, located in Dallas, Texas, is for manufacturing approximately 600 different complex cubic aircraft components [2,4]. The FMS consists of eight CNC horizontal machining centers, two inspection modules and a washing machine. Average cycle time is 20 minutes, and the minimum cycle time is 5 minutes. Loading and unloading of the system is done manually at two stations. These consist of storage carousels which permit parts to be stored on pallets for subsequent transfer to the machining stations by the AGVs. There are four AGVs. The system is capable of processing a sequence of single, one-of-a kind parts in a continuous mode (Figure 2.9).

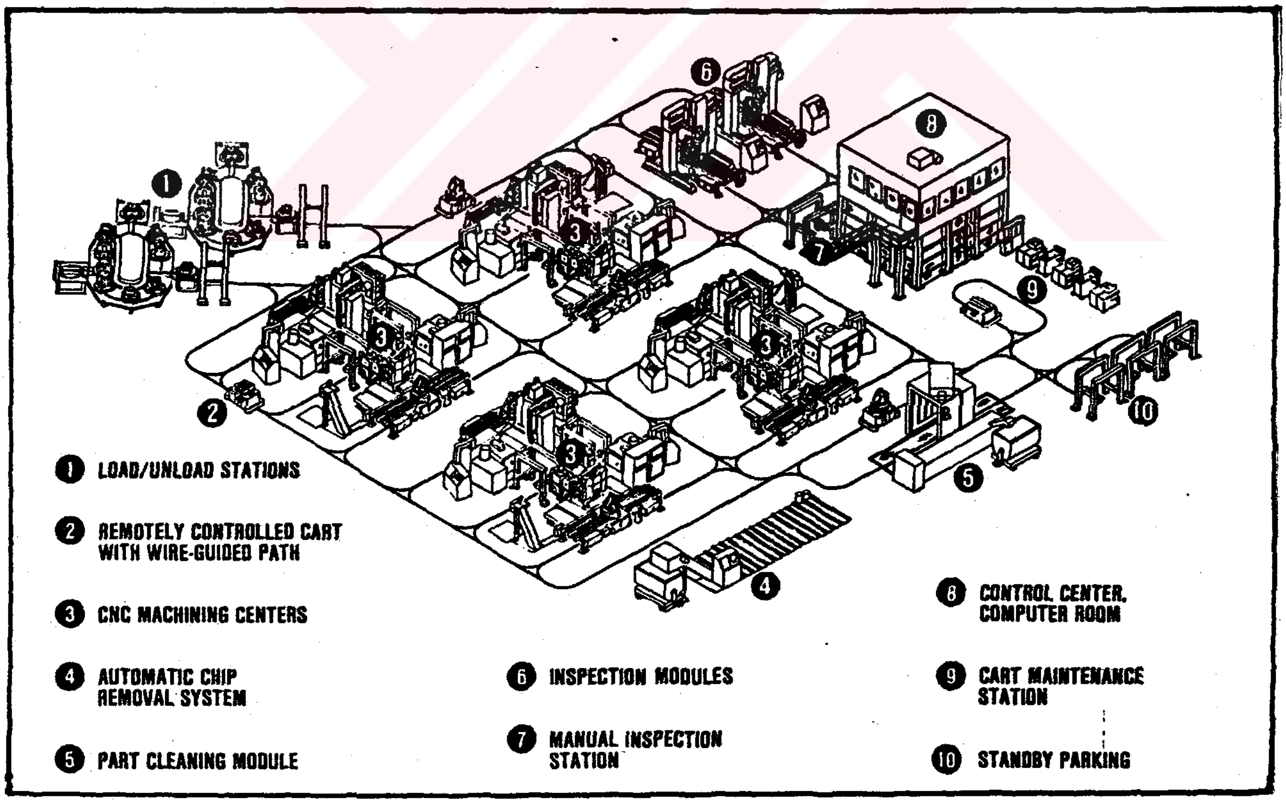


Figure 2.9 The FMS of VOUGHT Aero Products Division [2]

2.4. OR/MS Problems of FMS

Due to their intrinsic sophistication and flexibility, FMSs present more complex design and operation issues compared to conventionally equipped or fixed automation manufacturing systems. Here, various new and different OR/MS problems, that need to be addressed at different stages of an FMS's life cycle will be briefly described, in a context that unifies and summarizes the frameworks, provided by several authors (i.e. Stecke [7], Kiran and Tansel [8], Kusiak [9], Groover [2], Greenwood [1], Fry and Smith [10]).

2.4.1. FMS Design Problems

Design problems can be studied in two stages:

1. Preliminary conception phase,
2. Decisions during implementation phase.

1.Preliminary stage: During this stage the decision for the acquisition of the FMS is made. The actual need for an FMS should be verified by a thorough analysis of facts such as production volume range, minimum number of machines required, tolerance limits on work, etc. An economic and investment analysis shows whether FMS is a preferable alternative. Such economic issues are discussed in [11,12,13,14,15,16,17,18,19] .Furthermore, the level of flexibility that is required is a preliminary question whose answer leads into the selection of the technology to be used. Different types of flexibilities are defined and described in [20,21,22,23,24,25]. In general, more flexibility will be both more expensive and more difficult to manage. FMS types range from somewhat inflexible (i.e. a flexible transfer line having a fixed process flow) to very flexible FMSs having a widely varying process flow, even for parts of identical part type. This decision will help to specify the amount of automation that will be included in the system.

Initial specification of what is desired to be produced requires part family formation, development of process plans, determination of production volumes. All these and the physical characteristics of work eventually specify the types, numbers, and capacity (both tool magazine capacity and processing time capacity) of the machine tools, as well as the type and capacity of the material handling system. Sarin and Chen [26] and Lee et al. [27] present optimization models for manufacturing system selection.Amount of variations in process routings is a key issue in selecting the layout type.

Hierarchy among computers and type of control required (CNC/DNC) in different aspects of production (operations, material handling, tool loading, etc.) have to be determined, also.

2.Implementation Stage: At this stage, mainly the following are carried out:

i. Layout is determined;

ii. The number of pallets so as to specify maximum number of parts that will be in the system is determined;

iii. The number and design of fixtures is determined;

iv. Vendors are selected;

v. Manpower requirements are determined;

vi. Software are developed.

Queueing network models have been useful in handling design problems, to suggest several possible configurations. They model an FMS at an aggregate level of detail. These models can take into account the interactions and congestion of parts competing for the same limited resources and the uncertainty and dynamics of FMS providing adequate estimates of steady-state performance measures and evaluation insights. Solberg [28] used a simple, single-class closed queueing network to model an FMS and developed a program called CAN-Q. Buzacott and Shantikumar [29] applied open queueing network approach to discuss the impact of selected parts and storage system type and capacity on FMS performance.

After the rough-cut analysis in preliminary design, models that allow more detail take over in order to fine tune the actual design parameters. Simulation is the most flexible tool, allowing as much detail as desired to mimic reality. However, it can be expensive and time consuming.

Further readings on design issues are [30,31,32,33,34,35].

2.4.2. Aggregate Planning in FMS

The basic issue in an FMS aggregate planning phase is the integration of the FMS planning and control with the factory-wide production planning and control philosophy, ie. MRP.See [36,37,38,39].

In an FMS, aggregate planning issues are different from the classical systems in a number of ways:

i. Many decisions are considered at lower levels of hierarchy in an FMS environment. For example, product-mix decisions are not in this level.

ii. Forecasting is less significant due to system flexibility.

iii. Workforce factor can be eliminated from the model.

iv. Technological parameters (i.e. cutting speed, feed rate) can be varied, which means that production capacity is a decision variable and production cost is also a function of production capacity.

2.4.3. FMS Set-up Problems

Although there is negligible set-up time associated with consecutive operations as long as all required tools have previously been placed into the machine tool's limited capacity tool magazine, still some planning and system set-up is necessary to determine which cutting tools should be placed in which tool magazine and then to load these tools into the magazine. Those set-up decisions that have to be made and implemented before the system can begin to manufacture parts, are called FMS set-up problems. These are short-term planning problems referring to a segment of the master schedule and can be classified as follows [40] :

1.Part type selection: From the list of part types for which production orders of various sizes exist, choose a subset for simultaneous manufacture in the forthcoming period.

2.Machine pooling (called "machine grouping" by Stecke [40]): Partition the machines of each type into groups such that each machine in a particular pool will be identically tooled and capable of performing the same operations. This creates alternative routes for some part types and decreases the probability that a part will be blocked having no machine available for the next operation

3.Production ratios: Determine the relative production ratios at which the selected set of part types should be kept over time in the system.

4.Resource allocation: Allocate the limited number of pallets and fixtures of each type among selected part types.

5.Machine loading: Allocate the operations and associated tools of the selected set of part types among the (pooled) machines subject to technological and capacity constraints.

While some researches address these set-up problems simultaneously by developing mathematical models that embrace some or all five subproblems (see Kiran and Tansel [41], Rajagopalon [42]), many others propose hierarchical or sequential solution procedures.

For the first planning problem, Whitney and Gaul [43] suggest a heuristic approach to select a batch of parts for concurrent production, on the basis of resource requirements in comparison to aggregate resources available. Kusiak [44] proposes a coding system based on the similarity between parts to select part types. Hwang [45] notes the differences between part family problem in GT and FMS. None of the clustering method used in GT, takes into account constraints coming from the system, such as the number of tool magazine slots, the due-dates of each part and the total processing time of each tool. O'Grady and Menon [46] apply a multiple criteria framework that incorporates some resource constraints of an existing FMS to choose a compatible subset of candidate orders for processing.

In general, two methods have been applied to determine the relative production ratios and allocation of resources (problems (3) and (4)). First one is the mean value analysis (MVA) which assumes random operation times and analyzes a queueing network model. The second model is a Petri net- a deterministic model that can also analyze some transient and steady-state effects. See Dubois and Stecke [42]. Both models provide the steady-state expected production for each of several part types and bounds on requirements to operate the FMS efficiently.

Stecke and Solberg [48], studied the pooling problem on an aggregate level by means of a closed queueing model, to maximize expected production. Their analysis revealed that having fewer groups and unbalancing the number of machines in each group as much as possible were better. On the detailed level, Stecke [49], gives a non-linear mixed integer programming formulation for the pooling problem which includes technological and capacity constraints and maximizes pooling on the basis of aggregate results.

The machine loading problem has interested many researchers. Stecke and Solberg [50] simulated an existing FMS to test alternative loading policies. Detailed nonlinear mixed integer programming formulations with different loading objectives, each applicable to different types of flexible manufacturing systems are presented in Stecke [49]. A machine loading model was formulated by Kusiak [51] for minimizing total processing cost which also considered limited tool slot capacity and tool life. Ammons et al. |52] developed a bi-criteria model to minimize part movements and work load imbalance for the loading problem and suggested use of heuristics. Heuristic algorithms are also suggested by Stecke and Talbot [53], but no performance evaluation or computational experience has been reported.

Mereno and Ding [54] developed some goal oriented heuristics for the FMS loading and part type selection problems. Lashkari et al. [55], Sarin and Chen [56], and Shanker and Rajamarthandan [57] introduced other mathematical formulations. Queueing network models have also been used to address machine loading problems. See Stecke and Solberg [48] and Stecke and Morin [58].

2.4.4. FMS Scheduling Problems

FMS Scheduling problems are concerned with the operation of the system after it has been set-up during the planning stage. The first problem to address is to determine the appropriate policy to input the parts of the selected part types into the FMS. Then, applicable algorithms to schedule the operations of all parts through the system have to be determined.

The scheduling problem in FMS has been an intriguing subject to many researches who basically had two approaches: optimization and artificial intelligence (AI).

One of the first report on scheduling FMSs was published by Hitz [59], [60]. He developed and solved a mathematical programming formulation for the specific type of manufacturing system called a flexible machining line. His periodic release strategy was explored in Erschler et al. [61]. Chang and Sullivan presented a binary formulation of the scheduling problem for a flexible machining system and developed two-phase suboptimal algorithm to solve it. The relative performance of this method is investigated in Chang et al. |62] using simulation. Afentakis [63] showed that under certain assumptions the flexible machining system can be transformed into a flow line type system . Network flow formulation for scheduling a flexible manufacturing module is discussed in Finke and Kusiak [3]. A two level algorithm for scheduling an FMS consisting of a machining and an assembly system is presented in Kusiak |64]. Kimemia and Gerschwin [65] formulated the scheduling problem as a continuous-time dynamic program, considering both the rate of demand and machine failures. Akella, Choong and Gerschwin [66] developed a hierarchical scheduling policy where in the higher level short-term production rates are computed based or machine status and demand deviation information and the lower level determines the actual times at which parts are loaded into the system. They test the performance of their method using simulation . Nasr and Elsayed [67] present a mixed integer programming model for minimizing the mean flow time in a machining system with alternative machine tool routings. Sawik [68] formulates the production scheduling of an FMS as a multi-level integer program and presents a period-by-period heuristic to minimize the maximum lateness. Erschler and Roubellat [69] propose a constraint based analysis method where item release times and due-dates, routings and limited resources are considered as constraints for real-time production scheduling. Chan and Bedworth [70], and Yao and Pei [71] also take dynamic, real time environments into consideration. Some of the other optimization based readings are [72,73,74,75,76].

Some of the AI based scheduling papers published to date are by Thesen and Lei [77], Lin and Chung [78], Shen and Chang [79], Subramanyam and Askin [80], and Park et.al [81]. Morton and Smunt proposed the PATRIARCH system for a machining system which incorporates strategic planning, capacity planning, scheduling and dispatching [82]. Shaw and Whinston [83] developed and implemented in LISP a planning and scheduling system which is able to determine routes and schedule jobs in a stochastic machining environment. Ben-

Arien [84] designed an expert system for scheduling a machining and assembly system, where some of the machined parts were designated for assembly, and reported on simulation results for seven different scheduling policies .

2.4.5. FMS Control Problems

FMS Control Problems are defined as those associated with the continuous monitoring of the system, keeping track of production to be certain that production requirements and due-dates are being met as scheduled. These cover issues such as handling machine tool and other breakdowns, maintenance, inspection, traffic control, tool life and process monitoring and data collection etc., for which policies should be determined during the design phase and implemented during real-time. Several researches have stressed the need for coordination between scheduling and control activities and indicated that utilization of AI would provide the level of intelligence required by scheduling function and facilitate the integration of scheduling with control [85,86,87]. Dawis and Jones [88] describe on-line simulation as a tool for real-time production scheduling and control .

2.4.6. Research Related to AGV System Scheduling

In spite of the large number of scheduling papers published, as most of them are limited to scheduling parts and machines only, the literature on the scheduling of the material handling system, particularly AGVs, is relatively sparse.

Since the optimal routing and scheduling methods chosen will determine the final design specifications of the AGV system, an iterative procedure of design and simulation of operation will be necessary before implementation, and majority of the work in literature has been in this respect [89,90,91,92].

The role of vehicle dispatching in system design was discussed by Maxwell and Muckstadt [93], who addressed the problem of finding minimum number of vehicles for an assembly system in which demands for material handling occur periodically at the workstations. Egbelu and Tanchoco [94] evaluated various dispatching rules for the AGVs via a simulation model applied to a particular layout. Another simulation study to test the relative performance of various dispatching rules is by Sabuncuoglu and Hammerzheim [95]. Performance bounds for the first-encountered first-served rule were derived by Bartholdi and Platzman [95] for a simple closed-loop configuration. Han and Mcginnis [97] present a real time routing heuristic for a material handling transporter in a manufacturing system where a number of workstations are arranged along a straight track on which the transporter can travel in either direction.

In all of these studies, the scheduling of the material handling system is considered independent of the schedules developed at the machines (see Bozer [5]). Raman, Talbot and Rachamadagu [98] define the problem in manner which is the closest to the definition in our study. However, the integer programming formulation and the solution procedure they present rely on the basic assumption that AGVs always return to the load/unload station upon completion of their task, so that the durations of the empty trips following each task are known. This assumption allows them to consider transportation as an operation between two successive machining operations for any given job, and utilize concepts of project scheduling under resource constraints.

In our study this assumption will be relaxed. Upon completing a loaded trip the AGV is designated to its next pick-up station. Therefore, the travel times of the empty trips, or the "deadheading trips" as they are called in literature, depend on the ending and the starting points of the successive loaded trips assigned to a vehicle, and are not known until the route is specified. This concept of sequence-dependent travel times is also encountered in Han and Mcginnis |97].

III. PROBLEM DEFINITION AND THE NONLINEAR MIXED-INTEGER PROGRAMMING MODEL

In this chapter, the FMS under consideration will be decribed, basic assumptions will be stated and the problem will be defined. The mathematical programming formulation of the problem that will be presented subsequently, turns out to be a nonlinear mixed-integer model. Since this formulation is intractable for practical purposes due to the size and the non-linearity , instead of solving it , its special structure will be exploited in the remainder of this dissertation, in order to develop a tractable approach to the problem.

3.1. Problem Definition

For the FMS under consideration it is assumed that all the design and set-up issues have already been resolved.

The set of part types which will be produced during the planning period and their process plans are determined. The types and number of machines are known. Machine loading, that is allocation of tools to machines and assignment of operations to machines, is made. Furthermore, it is assumed that the routing of each part type is available before scheduling decisions. This assumption can be made valid without restricting oneself to cases where a serial precedence relationship among operations exists and no alternative machine assignments are made, either by selecting one of the possible routings based on considerations of technological feasibility and processing efficiency, or by formulating the set-up phase problems in a manner that can also handle the routing decisions [41]. In this way, for each part type a certain routing can be used for off-line scheduling purposes while possible alternative routings may be considered during real-time control. Finally, pallets and other necessary equipment are allocated to parts.

Material transfer between machines is done by a number of identical AGVs on a given layout. Their movement is accomplished using one of the guidance methods referred to in section 2.2.2. Loading and unloading times are known. Data on travel speeds for loaded and unloaded vehicles are available so that travel times may be computed, i.e. using shortest time paths. Two kinds of trips are performed by vehicles. In a loaded trip, a part is taken from the machine where an operation has been completed to the machine where its next operation is assigned. A deadheading trip is an empty trip between two loaded trips of a vehicle whose duration is sequence-dependent.

A static environment where all jobs are available for processing is considered. A job may be defined as a batch of parts belonging to one of the part types selected to be produced simultaneously during a set-up period. Batch sizes are equal to pallet capacity and batch splitting is not allowed when loading a vehicle or a machine. The processing time for each job is deterministic and known. Preemption of operations or trips are not allowed. Such things as machine failure or downtime, scraps, rework, blocking of vehicles and vehicle dispatches for battery changer are ignored here and left as issues to be considered during real-time control.

Now, the scheduling problem under study can be defined as follows: Given the FMS described above determine the starting and completion times of operations of each job and the trips between work stations together with the vehicle assignment according to the objective of makespan minimization.

3.2 Integer Programming Formulation

Let J be the set of jobs available for processing. For each job j in J , let n_j be the number of operations to be performed.Let $n = \sum_{j \in J} n_j$ and let $I = \{1,2...,n\}$. I is the index set of operations. The first n_1

elements of I are associated with job 1, the elements $n_1+1,\ldots n_1+n_2$ with job 2, etc. Or, $I_j=\{N_j+1,\ N_j+2,..N_j+n_j\}$ is the set of indices in I associated with job j in J,where N_j is the total number of operations of the jobs indexed before j. Operations for each job are numbered such that the successor operation has an index higher than that of its predecessor. The last operation of each job (N_j+n_j) represents the trip to unload station after process is completed. Since the routing for each job is known, for each index in I the corresponding machine index is also known and in the formulation given below, the machine index is not used explicitly. For each operation i in I, there is a corresponding loaded trip i whose destination is the machine where operation i is assigned. K is the number of AGVs.

The notation is given below:

p_i : processing time of operation i.

t_i : travel time of the loaded trip i including loading and unloading times.

$\tilde{t}_{ih}$: travel time of the deadheading trip starting at the machine performing operation i and ending at the machine performing operation h.

c_i : completion time of operation i.

T_i : completion time of loaded trip i.

$$q_{rs} : \begin{cases} 1, & \text{if } c_r \text{ is less than } c_s; \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ih} : \begin{cases} 1, & \text{if a vehicle is assigned for the} \\ & \text{deadheading trip between trip i and trip h;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{0h}: \begin{cases} 1, & \text{if a vehicle is assigned for trip } h \text{ as its first trip;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i0}: \begin{cases} 1 & \text{if a vehicle is assigned for trip } i \text{ as its last trip;} \\ 0, & \text{otherwise.} \end{cases}$$

The constraints of the machine and materials handling system scheduling problem are as follows. First constraint is the precedence constraint for the operations of the same job.

$$c_i - c_{i-1} \geq p_i + t_i \qquad \forall\ i,\ i\text{-}1\ \varepsilon\, I_j,\ \ j\ \varepsilon\ J \tag{3.1}$$

The second set of constraints is a variation of the classical job-shop MIP non-interference constraint:

$$(1+M\tilde{t}_{rs})c_r \geq c_s + p_r - Mq_{rs} \tag{3.2}$$

$$(1+M\tilde{t}_{rs})c_s \geq c_r + p_s - M(1-q_{rs})$$
$$\forall\ r\varepsilon I_j,\ s\varepsilon I_k \text{ where } j,\ k\varepsilon J,\ j\neq k$$

Whenever operations r and s belonging to different jobs are assigned to the same machine, $\tilde{t}_{rs}$ will be zero. Then, q_{rs} will force one of the two inequalities to be inactive. The remaining active equation prevents the succeeding operation from being completed earlier than the sum of the preceding operation's completion time and succeeding operation's processing time. On the other hand, if operations r and s are assigned to different machines then $\tilde{t}_{rs}$ will be different than zero and both inequalities will hold trivially.

The third and fourth constraints assure that all operations are loaded and unloaded once, respectively:

$$\sum_{i \in \{0\} \cup I} x_{ih} = 1 \qquad \forall\, h \in I \qquad (3.3)$$

$$\sum_{i \in \{0\} \cup I} x_{hi} = 1 \qquad \forall\, h \in I \qquad (3.4)$$

The fifth constraint limits each vehicle to enter the system at most once:

$$\sum_{h \in I} x_{0h} \leq K \qquad (3.5)$$

The sixth is the constraint to keep the number of vehicles in the system consistent:

$$\sum_{h \in I} x_{0h} - \sum_{h \in I} x_{h0} = 0 \qquad (3.6)$$

The seventh constraint states that operation i can start only after the trip to load it is completed:

$$T_i \leq c_i - p_i \qquad \forall i \in I \qquad (3.7)$$

The last two constraints are related to the starting times of the trips. Together they state that trip i cannot start before the maximum of the completion time of the previous operation and the deadheading trip to the previous operation.

$$T_i - t_i \geq c_{i-1} \qquad \forall i,\ i-1 \in I_j,\ j \in J \qquad (3.8)$$

$$T_i - t_i \geq \sum_{h \in \{0\} \cup I} x_{hi}(T_h + \tilde{t}_{h,i-1}) \qquad \forall i, i-1 \in I_j,\ j \in J \qquad (3.9)$$

The objective to be considered is makespan minimization via minimizing the maximum of the completion time of the last operations for all jobs. Therefore, the complete formulation is:

Minimize Z

Subject to (1), (2), (3), (4), (5), (6), (7), (8), (9)

and

$$Z \geq c_{nj} \qquad \forall\ j \varepsilon J \qquad (3.10)$$

$$c_i \geq 0 \qquad \forall\ i \varepsilon I$$

$$x_{ih} = 0,\ 1 \qquad \forall\ i,h\ \varepsilon\ IU\{0\}$$

$$q_{rs} = 0,\ 1 \qquad \forall\ r,\ s \varepsilon I$$

With slight changes in constraint (3.10) other objective functions are possible. For instance,

$$Z \geq \sum_{j=1}^{|J|} c_{nj}\ /\ |J| \qquad (3.10a)$$

minimizes mean flow time, and

$$Z. \geq c_{nj} - D_j \qquad \forall\ j \varepsilon J \qquad (3.10b)$$

minimizes maximum lateness where D_j is the due-date for job j.

However, this formulation is intractable due to its size and nonlinearity , thus no solution procedure will be sought for it. Rather, the problem formulated above will be solved by an iterative heuristic procedure that tries to incorporate its combinatorial nature, on the basis of insight gained through this formulation. On the other hand, a quick and good upper bound will be obtained by a simple, single-pass heuristic procedure. However, it is not easy to find a tight lower bound, because either the problem generated for this purpose is too hard, or it ignores such important features that the resulting lower bound is of no use. At any rate, such a lower bound would not be of great help as is the case with a good upper bound which provides a feasible solution.

IV. MACHINE AND VEHICLE SCHEDULING SUBPROBLEMS

When the mixed integer programming formulation in Chapter 3 is studied closely, it is seen that the problem is in fact a combination of two subproblems, a machine scheduling problem and a vehicle scheduling problem. These two subproblems interact through two constraints, (3.7) and (3.8).

The machine scheduling problem given by (3.1),(3.2),(3.3) and (3.10) is in form of an $n/m/G/C_{max}$ problem, a well known NP-hard problem [99]. The remaining part has the form of a vehicle scheduling problem (VSP), where a set of transportation tasks (loaded trips), each with origin g_i, destination h_i, earliest finish time, ef_i, latest finish time, lf_i, and duration t_i, are to be scheduled. A survey for VSP can be found in Kusiak [100].

Given a solution for the machine scheduling problem-i.e. by one of the heuristic methods described in French [101]- the completion times for operations, c_i, dictate the earliest and latest finish times for the trips to load them such that

$$ef_i = c_{i-1}+t_i$$

$$lf_i = c_i-p_i$$

and, therefore the constraints (3.7) and (3.8) become

$$ef_i \leq T_i \leq lf_i \tag{4.1}$$

Thus, it remains to see whether a set of T_i can be found that is feasible to (4.1) as well as (3.4),(3.5),(3.6), and (3.7). If the answer is negative, then the machine schedule should be altered in some way and the search for a feasible vehicle schedule (with respect to the new machine schedule) restarts. Once a vehicle schedule that is consistent with the constraints resulting from the machine schedule is obtained, then a solution to the original simultaneous machine and vehicle scheduling problem has been found. Iterations will be carried on in order to find a better solution in terms of the makespan. After a given number of iterations the procedure will terminate with the best solution obtained thus far. Therefore, if the original problem is decomposed into two subproblems in this way, an iterative procedure which provides the following, is required:

i - an algorithm that generates machine schedules,

ii - an algorithm that finds a feasible solution to the VSP given the machine schedule,

iii - an iterative structure that links the two and facilitates the search for a good solution.

These three items will be developed in the subsequent sections.

4.1. Machine Schedule Generation

In machine schedule generation two algorithms are used. First is based on Giffler and Thompson's active schedule algorithm, and the second is its variation that produces non-delay schedules[101]. The only modification made in both algorithms is that, here the loaded trip times are considered as well as the processing times. Notation and the algorithms are given below:

t - Iteration number.

P_t - Partial schedule of the (t-1) scheduled operations each represented by (i,m_i,c_i) where m_i is the corresponding machine and c_i is the scheduled completion time.

S_t- Set of schedulable operations at stage t, whose predecessors have been scheduled and are contained in P_t.

est_i- Earliest time that operation i in S_t could be started, $est_i=\max\ \{c_{i-1}+t_i,\ C_M\}$, where (i-1) is the predecessor operation i in its job, t_i is the duration for the trip to load it, M is the machine where i will be processed, and $C_M=\max\ \{c_j |\ j\ \varepsilon\ P_t,\ m_j=M\}$.

eft_i- Earliest time that operation i in S_t could be finished, $eft_i=est_i-p_i$ where p_i is the processing time of i.

Algorithm 1. (Giffler and Thompson)

Step 1. Let t=1 and P_1 be empty. S_1 will be the set of all operations with no predecessors.

Step 2. Find $eft^* = \min\ \{eft_i,\ i\ \varepsilon\ S_t\}$ and the machine M^* where eft^* occurs. If there is a tie for M^* choose arbitrarily.

Step 3. Choose an operation j in S_t such that

i) it requires M^*, and

ii) $est_j<eft^*$.

Step 4. Create P_{t+1} by adding operation j, (j,M^*,eft_j), to P_t. Create S_{t+1} replacing operation j in S_t by the operation that directly follows it in its job. (unless j is the last operation of its job). Increase t by 1. If t<nm, total number of stages, go to step 2. Otherwise STOP.

Algorithm 2. (Non-delay)

Step 1. Let t=1 and P_1 be empty. S_1 will be the set of all operations with no predecessors.

Step 2. Find est* = min {est_i, i ε S_t} and the machine M* where est* occurs. If there is a tie for M*, choose arbitrarily.

Step 3. Choose an operation j in S_t such that

i) it requires M*, and

ii) est_j= est*.

Step 4. Same as algorithm 1.

In Step 3 of both algorithms, in order to make the choice for the operation to be scheduled (in case a choice exists) a number of rules have been used. These are

(1) MWKR- Most Work Remaining,

(2) LWKR-Least Work Remaining,

(3) SPT- Shortest Processing Time.

4.2. Vehicle Scheduling with Time Windows

It has already been noted that the operation completion times generated by the machine scheduling subproblem can be used to construct time windows for the completion times of the material handling trips as in (4.1) Equivalently, one can write time-windows $[a_i,b_i]$ for the trip starting times ST_i, such that earliest pick-up time, $a_i = c_{i-1}$,

latest pick-up time, $b_i = c_i - p_i - t_i$,

and $a_i \leq ST_i \leq b_i$. (4.2)

Here, each trip i is associated with loading a processing operation i. Its origin is the machine where the previous operation i-1 is processed, and destination is the machine where operation i will be processed. Therefore it cannot start before an earliest-pick-up time, a_i, determined by the completion time of the previous operation i-1, and it should start before a latest-pick-up time, b_i, so that operation i can be completed at its scheduled time.

The vehicle scheduling problem with time windows has been encountered in several contexts in literature: e.g. school busing [102,103],scheduling of urban transit, passenger trains, airline fleets. The same problem structure also occurs in the assignment of n jobs to m parallel machines where jobs have earliest start and due-date constraints. Bratley, Florian and Robillard [104] and Gertsbakh and Stern [105] have examined this context.

Orloff [106], who presented a first attempt at an optimal approach to the problem, formulated the VSP as a network problem where trips are defined as nodes, and deadheading time is defined on feasible intertrip arcs. A route is a sequence of trips and intertrip arcs carried out by the same vehicle, and the problem is to determine routes and schedules for all the trips so as to minimize the number of vehicles and travel costs while respecting network and scheduling constraints. He proposed to approach the problem as a set partitioning problem by explicitly enumerating all routes satisfying the scheduling constraints. However, he noted that the number of resulting variables is very large even for a medium-sized problem.

Desrosiers, Soumis and Desrochers [102] presented a formulation with a reduced number of variables. In their formulation

P is the set of trips and I is the set of intertrip arcs, and t_{ij} is the duration of the intertrip arc. An intertrip arc is considered only if it is possible to undertake trip j after trip i while obeying the time window constraints (i.e. $a_i+t_i+t_{ij}\leq b_j$). The network used by the vehicles is defined by the set of nodes $N=P\cup\{s\}\cup\{k\}$ and the set of arcs $A= I \cup \{s\}\times P \cup P\times\{t\}$ where s and t represent the entrance to and exit from the depot respectively. Each vehicle leaves the depot once. Flow variables, x_{ij}, take value 1 when arc (i,j) is used by a vehicle and continuous time variables, ST_i, are associated with starting time of each trip. Variable travel costs and fixed vehicle costs are denoted by f_{ij}, such that

$$f_{ij}=\begin{cases} cg & \text{if } i=s \text{ or } j=s \text{ with } cg>>t_{ij} \\ t_{ij} & \text{otherwise.} \end{cases}$$

Their formulation for vehicle scheduling with time windows (TWP) is as follows:

$$\textbf{TWP:}\quad \min \sum_{i,j\varepsilon A} f_{ij}\, x_{ij} \qquad (4.3)$$

$$\text{st.}\quad \sum_{j\varepsilon N} x_{ij} = \sum_{j\varepsilon N} x_{ij} =1 \qquad i\varepsilon P \qquad (4.4)$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i,j)\varepsilon A \qquad (4.5)$$

$$x_{ij}>0 \Rightarrow ST_i+t_i+t_{ij} \leq ST_j \qquad (i,j)\varepsilon I \qquad (4.6)$$

$$a_i\leq ST_i\leq b_i \qquad i\varepsilon P \qquad (4.7)$$

The first part of the formulation (4.3-4.5) is a minimum cost flow problem. Constraint (4.6) is a scheduling constraint to ensure compatibility of routes and schedules. The time windows are represented by (4.7).

Dantzig-Wolfe decomposition and column generation are the techniques used to solve this problem. The master problem, which includes trip constraints, is solved by a simplex algorithm. The subproblems, which include the scheduling constraints and the flow constraints, are solved as shortest-path problems with time windows.

Desrosiers, Soumis and Sauve in their further work on this problem. compared two branch-and-bound algorithms, where scheduling constraints were relaxed and branching was done either on flow variables or by dividing the time windows [103].

Later, their work was extended by Ferland and Fortin [107], who added another level of flexibility by allowing the time windows of trips to slide . This is called "vehicle scheduling with sliding time windows".

The "sliding time windows" concept exactly conforms to our vehicle scheduling subproblem, because given a solution to the machine scheduling subproblem with a particular operation sequence on each machine and a particular makespan, C_{max}, operation scheduled times are ,in fact, earliest completion times, EC_i . Then, latest completion times, LC_i ,can be computed by a backward pass. Therefore, the operation completion times ,c_i ,can slide between the EC_i and LC_i without changing C_{max}.

$$EC_i \leq c_i \leq LC_i.$$

However, this will, in turn, cause the time windows for the corresponding trip starting times ,defined as in (4.2) , to slide, also.

We reformulate our subproblem using this concept as a vehicle scheduling problem with sliding time windows (STWP). Additional notation used and the formulation are given below:

AS : the set of admissible intertrip (deadheading) arcs
: $\{(i,j) | \ i=j, \ EC_{i-1}+t_i+t_{ij} \leq LC_j - p_j - t_j\}$

AS' : the set of admissible arcs including the arcs entering and leaving the depot s.
: $AS \cup \{(i,s),(s,i) : 1 \leq i \leq n\}$

PS : the set of successors of trip i, $\{j : (i,j) \varepsilon AS\}$

BS : the set of predecessors of trip i, $\{j : (i,j) \varepsilon AS\}$

STWP: min $\sum_{(i,j)\varepsilon AS'} f_{ij} x_{ij}$

st.

$$\sum_{j \varepsilon PS_i} x_{ij} = 1 \qquad 1 \leq i \leq n$$

$$\sum_{j \varepsilon BS_i} x_{ji} = 1 \qquad 1 \leq i \leq n$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i,j) \varepsilon AS'$$

$$x_{ij} > 0 \Rightarrow ST_i + t_i + t_{ij} \leq ST_j \qquad (i,j) \varepsilon AS$$

$$\begin{aligned} a_i &= c_{i-1} \\ b_i &= c_i - p_i - t_i \qquad 1 \leq i \leq n \end{aligned}$$

$$a_i \leq ST_i \leq b_i \qquad 1 \leq i \leq n$$

$$EC_i \leq c_i \leq \ < LC_i \qquad 1 \leq i \leq n$$

An heuristic solution procedure for this subproblem will be developed in the next chapter.

V. VEHICLE SCHEDULING WITH SLIDING TIME WINDOWS

In this chapter; a heuristic approach for the vehicle scheduling problem with sliding time windows is developed. The approach relies heavily on the notion of the 'opportunity cost' of linking two trips to reduce the overall cost; which was introduced by Ferland and Fortin [106] who studied the problem in the context of school busing.

The procedure starts with a set of starting times feasible for the sliding time window problem (STWP). Optimal solution to the associated problem with fixed starting times (FP) is found and the resulting opportunity costs are computed. After identifying pairs of trips (i,j) offering good opportunity costs for reducing the overall cost, ways of modifying the starting times to allow them to be linked are searched. These modifications should be such that the slidden time windows and the starting times will obey the related constraints in the STWP. Furthermore, the modifications should preserve the routes in the optimal solution for the problem on hand. Hence, this is a feasible solution for the next (FP) for which the optimal value can only decrease. The procedure is repeated until the number of routes in the optimal solution is less than or equal to the number of vehicles available. If the procedure stops without being able to find any pairs (i,j) with negative opportunity cost that can be linked using modifications, then this means that the heuristic cannot find a feasible solution in terms of number of vehicles within the given makespan. The original makespan is increased, new time windows are constructed and the procedure is continued with the new STWP.

In the next section, the vehicle scheduling problem with fixed starting times is reviewed. Then the notion of opportunity cost of linking two trips by modifying their starting times is introduced.

The following section explains how the starting times will be modified within the time windows, under the restrictions mentioned above . The case where makespan is increased is described ,next. Then, the steps of the algorithm are stated. Finally, the algorithm is illustrated on a small example.

5.1. Vehicle Scheduling With Fixed Starting Times (FP)

A mathematical formulation of the problem with fixed starting times will be derived using the precedence graph (N,A) characterized as follows:

(i) With each trip i associate a node $i \varepsilon N$. A node $s \varepsilon N$ is associated with the depot.

(ii) An arc (i,j), exists (or trip j can follow trip i), if $ST_i + t_i + t_{ij} \leq ST_j$.

Furthermore, for each trip i, introduce arcs (s,i) and (i,s) allowing i to be the first or the last trip of a route.

Thus, the set of admissible arcs is

$$A=\{(i,j): ST_i+t_i+t_{ij} \leq ST_j\} U \{(s,i), (i,s): 1 \leq i \leq n\},$$

the set of successors of trip i is

$$P_i= \{j:(i,j) \varepsilon A\},$$

and the set of predecessors of trip j is

$$B_j= \{i:(i,j) \varepsilon A\}.$$

Then, the problem can be formulated as a minimum cost flow problem defined on this precedence graph:

$$\text{FP:} \quad \min \sum_{(i,j)\varepsilon A} f_{ij}\, x_{ij}$$

s.t

$$\sum_{j\varepsilon P_i} x_{ij}=1 \qquad i=1..n,$$

$$\sum_{i\varepsilon B_j} x_{ij}=1 \qquad j=1..n,$$

$$\sum_{i=1}^{n} (x_{si}-x_{is}) = 0$$

$$0 \le x_{ij} \le 1 \qquad (i,j)\varepsilon A$$

Given an optimal solution of this problem,

(i) if $x_{si}=1$, then trip i is the first trip of a route;
(ii) if $x_{is}=1$, then trip i is the last trip of a route;
(iii) if $x_{ij}=1$, then then trip j follows trip i in a route.

If the number of routes in this optimal solution is equal to or less than the number of vehicles available, then the given set of starting times is feasible for the original problem and we have a vehicle schedule and routing compatible with the machine schedule on hand. If, on the other hand, this is not the case, then the starting times should be modified.

(FP) can be solved by the simplex algorithm which will also generate simplex multipliers associated with the constraints of the problem. If π_i denotes the dual variable associated with node $i\varepsilon N$, then it is well known that the relative cost of an arc $(i,j)\varepsilon A$ is given by the relation

$$\tilde{f}_{ij} = f_{ij} - \pi_i + \pi_j .$$

This result will be used to define the opportunity cost of linking two trips.

5.2. Opportunity Cost

Suppose that given the starting times as specified, trip j cannot follow trip i (i.e., $(i,j) \notin A$). The opportunity cost of introducing arc (i,j) in A, will be a measure of the influence on the value of the objective function of modifying ST_i or ST_j to induce the possibility of linking trip j to trip i.

Assuming that an optimal solution for (FP) together with optimal multipliers associated with nodes in N is available, the opportunity cost of arc (i,j) may be defined in the form of the relative cost, as

$$\tilde{f}_{ij} = t_{ij} - \pi_i + \pi_j \, .$$

Recall that $f_{ij}=t_{ij}$ for $1 \leq i \leq n$, $1 \leq j \leq n$.

If $(i,j) \notin A$ and $\tilde{f}_{ij}>0$, then relative to the actual optimal basis, the objective function of (FP) cannot be reduced even if trip j could be linked to trip i. But, if $\tilde{f}_{ij}<0$ and arc (i,j) can be introduced into A in such a way that the basic optimal solution on hand remains a basic feasible solution for the new problem (FP) where the non-basic variable x_{ij} has a negative relative cost, then the objective function might decrease.

A decrease in the objective value is not guaranteed even if this variable becomes basic since there might be degeneracy. However, it is interesting to analyze the opportunity of linking task j to i if $\tilde{f}_{ij}<0$. Consequently, we state the following condition:

CONDITION A: A pair $(i,j) \notin A$ is considered for linking if

1. The opportunity cost of linking j to i is negative, and
2. i and j belong to different vehicle routes.

Note that, the latter condition is necessary because if i and j belong to same vehicle route and if j precedes i in this route, then link $(j,i) \varepsilon A$, and there will be no way of preserving this route if the new link is introduced. If i precedes j in the vehicle route, link (i,j) already exists in A.

So far, we have been using notions introduced by Ferland and Fortin [106]. But developments from this point on will be specific to our problem since the structure of time windows and their interrelationships are quite different from those in context of school busing.

5.3. Starting Time Modification Procedure

The specification of feasible modifications in starting times is more complex in the context of simultaneous machine and vehicle scheduling, because time windows are not independent from each other.

If a completion time c_i is increased, then the completion times for the next operation of the same job and for the next operation assigned to the same machine with operation i may be influenced, and this influence may be reflected to other operations downstream. For this reason, specification of the adjustments is not straightforward.

In order to comprehend the way a modification in some c_i affects other trips, the set of trips will be partitioned in two ways. The first, partition is according to the jobs. That is, the set of trips T is partitioned into L sequences where L is the number of jobs, $L \leq n$, and J_l is the sequence of trips related to job l, i.e.

$$T = J_1 \cup J_2 \cup \ldots \cup J_L$$

In other words, J_l gives the serial precedence relationship among operations of job l.

The second partition is according to the machines. This time, the set of trips T is partitioned into M sequences, where M is the number of machines, $M \leq n$. Then, T_m is the sequence of trips related to machine m, i.e.

$$T = T_1 \cup T_2 \cup \ldots \cup T_M$$

In this way, a trip i belongs to T_m, if its associated operation i has been assigned to machine m, and T_m gives the sequence of operations assigned to m.

5.3.1. Modifications for Fixed Time Windows

Let's first analyze the case where the starting times are slid without increasing any of the completion times, that is for fixed time windows, i.e. $a_i \leq ST \leq b_i$.

Assume that the trips i and j belong to different vehicle routes in the optimal solution to (FP) and j cannot be linked to trip i according to the ST_i and ST_j on hand; i.e., $ST_i+t_i+t_{ij}>ST_j$. Denote by S_i and S_j the routes including i and j, respectively. If trip j is to be linked to trip i, then trip i should be completed earlier at $(ST_i+t_i-\Delta^-_i)$, and trip j should start later at $(ST_j+\Delta^+_j)$ such that

$$ST_i+t_i-\Delta^-_i+t_{ij} \le ST_j+\Delta^+_j$$

or,

$$\Delta^-_i+\Delta^+_j \ge (ST_i+t_i+t_{ij})-ST_j. \qquad (5.1)$$

If S_i and S_j are to be preserved, then Δ^-_i and Δ^+_j depend on the adjustments allowed by other trips before trip i in S_i and after trip j in S_j. Then, Δ^-_i and Δ^+_i, the maximal adjustments for ST_i for given c_i, can be computed recursively for the tasks in the sequences as follows:

$$\Delta^-_i=\begin{cases} ST_i-a_i & \text{if } i \text{ is the first trip of } S_i \\ ST_i-\max\{a_i,\ ST_\mu + t_i + t_{\mu i} - \Delta^-_\mu\} & \text{if } \mu \text{ is the immediate predecessor of } i \text{ in } S_i. \end{cases} \qquad (5.2)$$

$$\Delta^+_i=\begin{cases} b_i-ST_i & \text{if } i \text{ is the last trip of } S_i \\ \min\{b_i,\ ST_\beta + \Delta^+_\beta - t_{i\beta} - t_i\} - ST_i & \text{if } \beta \text{ is the immediate successor of } i \text{ in } S_i. \end{cases} \qquad (5.3)$$

Let us denote by P_i the set including trip i and trips in S_i completed before trip i, and by N_j the set including trip j and the trips in S_j completed after trip j. The trips i and j form a candidate pair to be linked only if adjustments in one set do not arise modifications in the other one. That will be true if the following condition holds:

CONDITION B: (i,j) forms a candidate pair to be linked if

(1) N_j does not contain any predecessors of k in $J_{l(k)}$ for any k in P_i,

(2) Any r in N_j (and its successors in $J_{l(r)}$) does not precede in machine sequence any predecessors of k in $J_{l(k)}$ for any k in P_i.

PROOF: (i,j) can be linked if (5.1) is satisfied. It will be shown that if the above condition does not hold it is not possible to find such Δ^-_i and Δ^+_j that satisfies (5.1) and preserves the routes on hand.

(1) Assume that the reverse of part 1 is true, or $r \in N_j$ precedes $k \in P_i$. Then,

$$c_r \leq c_{k-1}.$$

From the time windows, we have

$$ST_r + \Delta^+_r \leq c_r - p_r - t_r, \text{ and } \quad c_{k-1} \leq ST_k - \Delta^-_k.$$

Therefore,

$$ST_r + \Delta^+_r < c_{k-1} \leq ST_k - \Delta^-_k.$$

There existed a link (k,i) which should be preserved to preserve S_i, so we must have

$$ST_k - \Delta^-_k + t_k + t_{ki} \leq ST_i - \Delta^-_i$$

or,

$$\Delta^-_i \leq -ST_k + \Delta^-_k - t_k - t_{ki} + ST_i.$$

There also existed a link (j,r) which should be preserved to preserve S_j, so Δ^+_j should be such that

$$ST_j + \Delta^+_j + t_j + t_{jr} \leq ST_r + \Delta^+_r$$

or

$$\Delta^+_j \leq ST_r + \Delta^+_r - t_j - t_{jr} - ST_j$$

Then,

$$\Delta^-_i + \Delta+_j \leq ST_r + \Delta^+_r - (ST_k - \Delta^-_k) - t_j - t_{jr} - t_k - t_{ki} + ST_i - ST_j.$$

Hence

$$\Delta^-_i + \Delta^+_j < ST_i - ST_j < ST_i + t_i + t_{ij} - ST_j$$

which contradicts (5.1).

(2) Assume that the reverse of part 2 is true.That is, we have an $r \varepsilon N_j$, an x that succeeds r in $J_{l(r)}$,a $k \varepsilon P_i$,and a y that precedes k in $J_{l(k)}$ such that x precedes y in $T_{m(y)}$. From time windows, we have

$$ST_r + \Delta^+{}_r \leq c_r - p_r - t_r \quad \text{and} \quad c_{k-1} \leq ST_k - \Delta^-{}_k$$

Since r precedes x, $c_r \leq c_x - p_x - t_x$, and

$$ST_r + \Delta^+{}_r \leq c_x - p_x - t_x - p_r - t_r .$$

Since x precedes y on $T_{m(y)}$, $c_x \leq c_y - p_y$, and

$$ST_r + \Delta^+{}_r \leq c_y - p_y - p_x - t_x - p_r - t_r .$$

Lastly, since y precedes k, $c_y \leq c_{k-1}$, and therefore

$$ST_r + \Delta^+{}_r < c_{k-1} \leq ST_k - \Delta^-{}_k .$$

As shown in the first part of the proof, in order to preserve links (k,i) and (j,r) ,we must have

$$\Delta^-{}_i + \Delta^+{}_j \leq ST_r + \Delta^+{}_r - (ST_k - \Delta^-{}_k) - t_j - t_{jr} - t_k - t_{ki} + ST_i - ST_j,$$

which is not possible,and hence, contradicts (5.1).□

Given that Condition B holds , if modifications of type Δ^+ are applied to trips in N_j, then modifications of type Δ^- can be applied to trips in P_i, because their time windows are not affected by the preceding modifications of type Δ^+.

5.3.2. Adjustments with Sliding Time Windows

Now, we can specify the adjustments to preserve the optimal routes an hand, if some c_i are modified. There are two constraints on the amount of time by which Δc_i^+, by c_i can be increased, when considered solely. First one is the latest completion time of operation i, LC_i, i.e.

$$\Delta c_i^+ \leq LC_i - c_i$$

The second constraint comes from the vehicle routes on hand. Given a vehicle route S_i, for each trip i on S_i we can determine an earliest handling time, EH_i, and a latest handling time, LH_i. The earliest time EH_i, at which the vehicle can begin to execute trip i is

$$EH_i = \begin{cases} 0 & \text{if i is the first trip of } J_{l(i)} \text{ and } S_i, \\ \max \{EC_{i-1}, (EH_{i-1}+t_{i-1}+p_{i-1}), (EH_\mu+t_\mu+t_{\mu i})\} & \text{if i has a predecessor (i-1) on } J_{l(i)} \text{, and } \mu \text{ on } S_i. \end{cases} \tag{5.4}$$

The latest time LH_i, at which the vehicle can begin to execute trip i is

$$LH_i = \begin{cases} (LC_i-p_i-t_i) & \text{if is the last trip in } J_{l(i)} \text{ and } S_i, \\ \min \{(LC_i-p_i-t_i), (LH_{i+1}-p_i-t_i)\ (LH_\beta-t_{i\beta}-t_i)\} & \text{if i has a successor (i+1) on } J_{l(i)} \text{ and } \beta \text{ on } S_i. \end{cases} \tag{5.5}$$

Hence, we have $EH_i \leq ST_i \leq LH_i$, and $a_i \leq ST_i \leq b_i$. To preserve the sequence S_i, the intersection of the two time intervals should not be empty, or the following constraints should be satisfied :

$$b_i \geq EH_i, \text{ and } a_i \leq LH_i.$$

From (4.2), if c_i is increased, then a_i is unchanged while b_i increases to $b_i' = b_i + \Delta c_i^+$, and $b_i' \geq EH_i$ is still valid. On the other hand,

a'_{i+1} increases to $a'_{i+1} = a_{i+1} + \Delta c_i^+$, and $a'_{i+1} \leq LH_{i+1}$ should be satisfied. Therefore, the second constraint on Δc_i^+ is

$$\Delta c_i^+ \leq LH_{i+1} - a_{i+1}$$

Both constraints can be combined into one in the following from:

$$\Delta c_i + = \min \{(LC_i - c_i), (LH_{i+1} - c_i)\} \qquad (5.6)$$

Given an increase in completion time of operation i, completion time of the next operation of the same job will be affected if

$$c_i + \Delta c_i^+ > c_{i+1} + p_{i+1} - t_{i+1}$$

where (i+1) is the successor of i in $J_{l(i)}$ and l(i) is the job task i belongs to. And if this is the case, then the induced increase in the completion time of operation (i+1) will be

$$\Delta c_{i+1}^{ind+} = c_i + \Delta c_i^+ + p_{i+1} + t_{i+1} - c_{i+1} \qquad (5.7)$$

which may in turn cause increases in completion times of its successors (for both partitions).

On the other hand, Δc_i^+ will affect the completion time of the next operation of the same machine if

$$c_i + \Delta c_i^+ > c_k - p_k$$

where k is the successor of i in $T_{m(i)}$ and m(i) is the machine sequence operation is assigned to. Then, once again, the induced increase Δc_k^{ind+} should be checked for resulting changes in completion times of its successors (for the job and the machine).

$$\Delta c_k^{ind+} = c_i+\Delta c_i^{+}+p_k - c_k \tag{5.8}$$

Therefore, when computing Δc_i^+, all influenced trips should be considered,and it should be checked whether the feasibility of the routes that they belong, are preserved.

Let us assume that the completion time of an operation i is increased by Δc_i^+, given by (5.6). If i has successors (i+1) and (i+2) on $J_{l(i)}$, and Δc_{i+1}^{ind+}, computed using (5.7), is positive, then Δc_{i+1}^{ind+} should preserve the route S_{i+2} where (i+2) takes place. This is true if

$$\Delta c_{i+1}^{ind+} \le LH_{i+2} - c_{i+1}.$$

We have,

$$\Delta c_{i+1}^{ind+}=c_i+\Delta c_i^{+}+p_{i+1}+t_{i+1}-c_{i+1}\le c_i+(LH_{i+1}-c_i)+p_{i+1}+t_{i+1}-c_{i+1} \qquad \text{from } 5.6$$

and,

$$LH_{i+1} \le LH_{i+2}-p_{i+1}-t_{i+1} \qquad \text{from } 5.5$$

therefore,

$$LH_{i+1}+p_{i+1}+t_{i+1}-c_{i+1} \le LH_{i+2}-c_{i+1}.$$

Thus, it has been verified that if Δc_i^+ is computed according to (5.6), then feasibility of routes, where trips corresponding to succeeeding operations of i on $J_{l(i)}$ take place, are preserved.

Let us now check the feasibility of the routes for the machine partition. If i has a successor k on $T_{m(i)}$, and k has a successor (k+1) on $J_{l(k)}$ and Δc_k^{ind+} , computed using (5.8), is positive, then Δc_k^{ind+} preserves route S_{k+1} , if

$$\Delta c_k^{ind+} \le LH_{k+1}-c_k.$$

Unfortunately, this does not hold in all cases and the feasibility of Δc_i^+, computed according to (5.8) should be checked by using the following recursive procedure which updates Δc_i^+ whenever an infeasibility occurs. The resulting Δc_i^+ preserves all routes.

```
Procedure CHECK P $(i, \Delta c_i^+)$
BEGIN
    IF i has a successor k on $T_{m(i)}$
        THEN
            Begin
                Compute $\Delta c_k^{ind+}$
                IF $\Delta c_k^{ind+} > 0$ THEN
                    Begin
                        IF k has a successor (k+1) on $J_{l(k)}$
                          THEN if $\Delta c_k^{ind+} > LH_{k+1} - c_k$
                                then $\Delta c_k^{ind+} = LH_{k+1} - c_k$
                        CHECKP $(k, \Delta c_k^{ind+})$
                        $\Delta c_i^+ = c_k + \Delta c_k^{ind+} - p_k - c_i$
                    End
            End;
    IF i has a successor (i+1) on $J_{l(i)}$
        and $\Delta c_{i+1}^{ind+} > 0$
        THEN
            Begin
                CHECK $(i+1, \Delta c_{i+1}^{ind+})$
                $\Delta c_i^+ = c_{i+1} + \Delta c_{i+1}^{ind+} - p_{i+1} - t_{i+1} - c_i$
            End
END;
```

The same kind of analysis can be carried out for the case where the completion time of an operation i is decreased by Δc_i^-. By definition, the earliest completion time of operation i creates a bound on Δc_i^-, such that

$$\Delta c_i^- \leq c_i - EC_i.$$

Also, if the sequence S_i is to be preserved, the adjusted time window after modifying c_i should satisfy

$$b_i' \geq EH_i \quad \text{and} \quad a_i' \leq LH_i$$

as in the previous case.

When c_i is decreased, from (4.2), a_i is unchanged, a_{i+1} decreases to $a_{i+1}' = a_{i+1} - \Delta c_i^-$, still satisfying $a_{i+1}' \leq LH_{i+1}$, but b_i decreases to $b_i' = b_i - \Delta c_i^-$ and $b_i' \geq EH_i$ puts another bound on Δc_i^-.

$$\Delta c_i^- = \min\{(c_i - EC_i), (c_i - p_i - t_i - EH_i)\} \tag{5.9}$$

However, this modification Δc_i^- may cause changes in completion times of upstream operations, and once again, they have to be checked for feasibility of the corresponding routes.

Given Δc_i^-, the induced decrease in the completion time of its preceding operation (i-1) will be

$$\Delta c_{i-1}^{ind-} = c_{i-1} - (c_i - \Delta c_i^- - t_i - p_i) \tag{5.10}$$

and, if positive it should preserve route S_{i-1} by obeying

$$\Delta c_{i-1}^{ind-} \leq c_{i-1} - p_{i-1} - t_{i-1} - EH_{i-1}.$$

Now, it will be verified that, if Δc_i^- is computed according to (5.9); then feasibility of routes can be preserved for the job partition, that is for the routes where trips corresponding to preceding operations of i take place.

$$\Delta c_{i-1}^{ind-} = c_{i-1} - c_i + \Delta c_i^- - t_i - p_i \leq c_{i-1} - c_i + (c_i - p_i - t_i - EH_i) + t_i + p_i \qquad \text{from (5.9)}$$

and , $$EH_i \geq EH_{i-1} + p_{i-1} + t_{i-1} \qquad \text{from (5.4)}$$

therefore, $$c_{i-1} - EH_i \leq c_{i-1} - (EH_{i-1} + p_{i-1} + t_{i-1})$$

which is the desired result.

However we cannot verify the same result for the machine partition. Given Δc_i^-, the induced decrease in the completion time of its preceding operation k on the same machine will be

$$\Delta c_k^{ind-} = c_K - (c_i - \Delta c_i^- - p_i), \tag{5.11}$$

but it is not guaranteed that this will preserve the route S_k.

Therefore, a recursive procedure, similar to the previous one, is introduced for computing Δc_i^-, which initially takes the value given by (5.9).

```
Procedure CHECK M (i, $\Delta c_i^-$)
BEGIN
    IF i has a predecessor k on $T_{m(i)}$
        THEN
            Begin
                Compute  $\Delta c_k^{ind-}$
                IF  $\Delta c_k^{ind-} > 0$
                    THEN Begin
                        IF  $\Delta c_k^{ind-} > c_k - p_k - t_k - EH_k$
                            THEN  $\Delta c_k^{ind-} = c_k - p_k - t_k - EH_k$
                        CHECK M (k, $\Delta c_k^{ind-}$ )
                        $\Delta c_i^- = c_i - p_i - c_k + \Delta c_k^{ind-}$
                    End
            End;
    IF i has a predecessor (i-1) on $J_{l(i)}$
        and $\Delta c_{i-1}^{ind-} > 0$
        THEN Begin CHECK (i-1, $\Delta c_i^{ind-}$ )
                $\Delta c_i^- = c_i - p_i - t_i - c_{i-1} + \Delta c_{i-1}^{ind-}$
            End
END;
```

Now , we are in a position to modify c_i and slide time windows while keeping the optimal routes feasible, so that we can find expressions for maximal starting time variations $\tilde{\Delta}_i^+$ and $\tilde{\Delta}_i^-$. By analogy with the preceding section and referring to relations (5.2) and (5.3) $\tilde{\Delta}_i^+$ and $\tilde{\Delta}_i^-$ can be computed recursively:

$$\tilde{\Delta}_i^{+} = \begin{cases} b_i+\Delta c_i^{+}-ST_i & \text{if } i \text{ is the last trip of } S_i \\ \min\{b_i+\Delta c_i^{+},\ ST_\beta+\tilde{\Delta}_\beta^{+}-t_{i\beta}-t_i\}-ST_i & \\ \quad \text{if } \beta \text{ is the successor of } i \text{ in } S_i\,. & \end{cases} \tag{5.12}$$

$$\tilde{\Delta}_i^{-} = \begin{cases} ST_i-a_i-\Delta c_{i-1}^{-} & \text{if } i \text{ is the first trip of } S_i \\ ST_i-\max\{a_i-\Delta c_{i-1}^{-},\ ST_\mu-\tilde{\Delta}_\mu^{-}+t_\mu-t_{\mu i}\} & \\ \quad \text{if } \mu \text{ is the predecessor of } i \text{ in } S_i\,. & \end{cases} \tag{5.13}$$

Hence, a candidate pair i and j can be linked, i.e. arc (i,j) can be included in the network, if

$$\tilde{\Delta}_i^{-}+\tilde{\Delta}_j^{+} \geq ST_i+t_i+t_{ij}-ST_j. \tag{5.14}$$

However, these are maximal adjustments and one would prefer to make minimal adjustments when modifying the starting times and sliding the time windows, to cause minimal distortions in criteria such as flow times and tardiness, even though they are not considered explicitly in this algorithm. Therefore, one has to recalculate the adjustments for (i,j) such that the inequality that is satisfied, now holds as an equality.

5.4. Incrementing the Makespan

A candidate pair (i,j) that satisfies Conditions A and B has potential to be linked without destroying the routes on hand, if

$$EH_i + t_i + t_{ij} \leq LH_j . \qquad (5.15)$$

Otherwise, this pair (i,j) cannot be introduced into the set of admissible arcs unless the current makespan is increased to push the latest handling times forward. The amount of increase in the makespan is given by

$$LB_{ij} = EH_i + t_i + t_{ij} - LH_j . \qquad (5.16)$$

The following argument will verify that LB_{ij} indeed gives the increment in the makespan.

Let us write the term LH_j in open form:

$$\begin{aligned}
LH_j &= \min\left\{LC_j - p_j - t_j, \quad LH_{j+1} - p_j - t_j, \quad LH_\beta - t_{j\beta} - t_j\right\} \\
&= \min\Big\{LC_j - p_j - t_j, \\
&\qquad \min(LC_{j+1} - p_{j+1} - t_{j+1}, LH_{j+2} - p_{j+1} - t_{j+1}, LH_\mu - t_{j+1,\mu} - t_{j+1}) - p_j - t_j, \\
&\qquad \min(LC_\beta - p_\beta - t_\beta, LH_{\beta+1} - p_\beta - t_\beta, LH_\gamma - t_{\beta\gamma} - t_\beta) - p_j - t_j\Big\} \\
&= \min\Big\{(LC_j - p_j - t_j), (LC_{j+1} - \text{constant}_{j+1}), (LH_{j+2} - \text{constant}_{j+2}), \\
&\qquad (LH_\mu - \text{constant}_\mu), (LC_\beta - \text{constant}_\beta), (LH_{\beta+1} - \text{constant}_{\beta+1}), \\
&\qquad (LH_\gamma - \text{constant}_\gamma)\Big\}.
\end{aligned}$$

where β, μ and γ are immediate successors of j,j+1, and β in their routes, respectively, When all LH terms are recursively written in open form, all of the terms inside the parentheses will take the form (LC_{α}-$constant_{\alpha}$), α being all successors in job partitions and routes starting from j. Therefore, increasing LH_j by some constant LB_{ij} to satisfy

$$EH_i + t_i + t_{ij} = LH_j + LB_{ij}$$

means increasing the corresponding (minimum) latest completion time that determines LH_j, by LB_{ij}. This will increase the makespan by LB_{ij} and will allow (i,j) to become a feasible candidate.

In the algorithm, all candidates that are infeasible to the current makespan (i.e. those which do not satisfy 5.15) are kept in a list in ascending order of their lowerbounds, LB. Then, if it turns out that all feasible candidates fail to satisfy (5.14), instead of stopping without a solution with desired number of vehicles, it is preferred to search for a vehicle routing solution by increasing the makespan in an iterative manner.

Initially, the makespan is increased by the lowerbound of the first arc on the infeasible arc list, and all the arcs with the same lowerbound are tested in order of their opportunity costs to check whether a link is possible. If so, the algorithm continues with this makespan. Otherwise, makespan is increased once again using the lowerbound of the first arc of the list after discarding the already tested arcs. In this manner, the makespan is increased until a solution is found or makespan exceeds a given upperbound (ie., makespan of a previously obtained solution).

5.5. STW Heuristic Algorithm

Based on what has been stated up to now, the heuristic algorithm whose steps are specified below is developed.

ALGORITHM:

Step 1. Set starting times at initial values, ST, feasible to.the STWP.

Step 2. Solve the associated (FP). If the number of routes in.the optimal solution is less than or equal to the number of vehicles available then report the solution and stop.

Step 3. Prepare candidate and infeasible arc list: For all pairs (i,j) which satisfy Condition A and Condition B, if $EH_i+t_i+t_{ij}\leq LH_j$ then put (i,j) in candidate list in ascending order of opportunity; else, if makespan+$LB_{ij}\leq$UB (upperbound), then put (i,j) in infeasible arcs list in ascending order of LB.

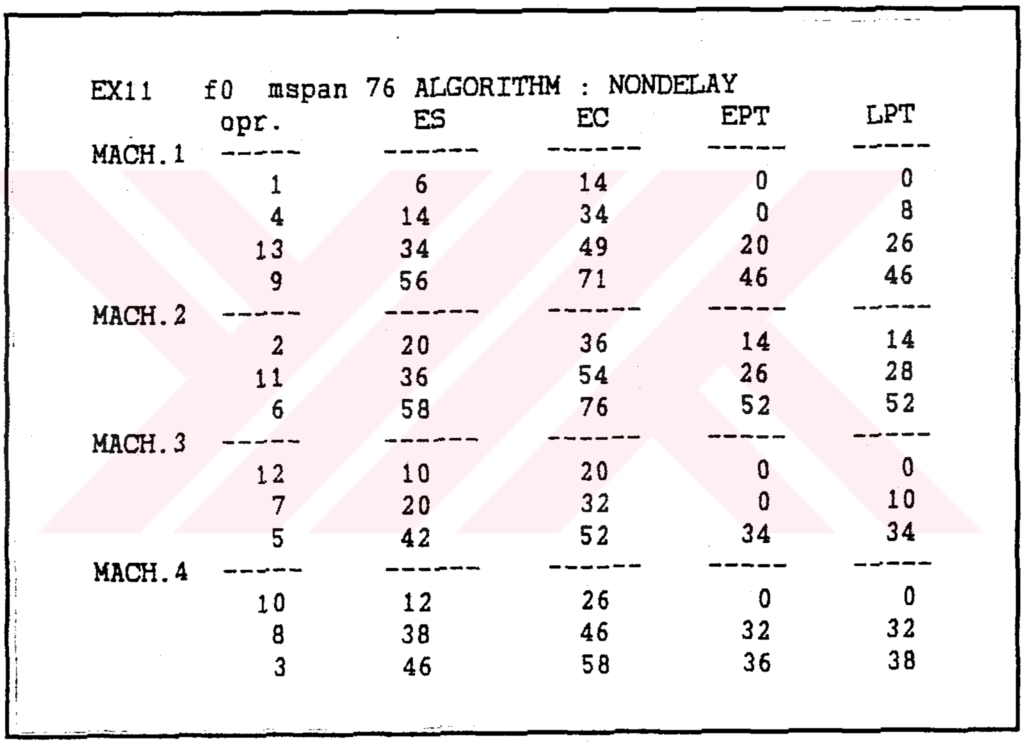
Step 4. Check candidate arcs: Take first candidate pair (i,j).Compute $\tilde{\Delta}^-_i$ and $\tilde{\Delta}^+_j$. If link is possible ((5.16) is satisfied), then make minimal adjustments in ST and c, and go to Step 2. Otherwise, repeat this step until all candidates arc tested.

Step 5. Increase makespan: Take first infeasible arc (i,j).Increase makespan by LB_{ij}.Compute $\tilde{\Delta}^-_i$ and $\tilde{\Delta}^+_j$. If link is possible adjust ST and c, and go to Step 2. Otherwise, repeat this step until all infeasible arcs are tested.

Step 6. A solution with makespan≤UB cannot be obtained with this machine sequence. Stop.

5.6 Illustration of the STW Heuristic Algorithm

The STW Heuristic Algorithm will now be illustrated through an example problem (EX11) of 4 machines,5 jobs (13 operations) ,and 2 vehicles.The complete data of problem can be found in the appendices A and B (Job set1 and Layout 2). The machine schedule generated by applying Algorithm 2 with SPT is displayed in Figure 5.1. The makespan, C_{max} is 76.



```
EX11    f0   mspan 76 ALGORITHM : NONDELAY
         opr.          ES          EC         EPT        LPT
MACH.1   -----       ------      ------      -----      -----
            1           6          14           0          0
            4          14          34           0          8
           13          34          49          20         26
            9          56          71          46         46
MACH.2   -----       ------      ------      -----      -----
            2          20          36          14         14
           11          36          54          26         28
            6          58          76          52         52
MACH.3   -----       ------      ------      -----      -----
           12          10          20           0          0
            7          20          32           0         10
            5          42          52          34         34
MACH.4   -----       ------      ------      -----      -----
           10          12          26           0          0
            8          38          46          32         32
            3          46          58          36         38
```

Figure 5.1 The machine schedule generated for the example problem by Algorithm 2 (SPT).

ITERATION 1: The trip starting times, ST, are set at their earliest pick-up times, a_i, as initial values (STEP 1) and the first fixed problem , FP, is solved, to yield the vehicle schedule in Figure 5.2 (STEP 2). Since the vehicle schedule requires 5 vehicles (cost=10016) , the algorithm continues. All candidate arcs are checked, however none could be linked (STEPS 3 and 4). So, the makespan is increased by 3,to check the infeasible arc (10,7), which is then included to the FP (STEP 5).

```
   ITERATION : 1  SEED FILE : f0  COST :  10016
   veh  opr   ept  lpt  st  comp   lc   eh   lh

    1     1     0    0    0   14   14    0    0
         11    26   28   26   54   58   26   30
          3    36   38   36   58   76   36   38
          6    52   52   52   76   76   52   52

    2     4     0    8    0   34   34    0    8
          2    14   14   14   36   40   14   16

    3     7     0   10    0   32   37    0   15
         13    20   26   20   49   61   20   26
          5    34   34   34   52   52   34   34

    4    10     0    0    0   26   32    0    4
          9    46   46   46   71   76   46   51

    5    12     0    0    0   20   25    0    5
          8    32   32   32   46   51   32   37




  increase makespan  by 3
  include arc    10   7

   ITERATION : 2  SEED FILE : f0  COST :   8022
  increase makespan  by 7
  include arc    10   4

   ITERATION : 3  SEED FILE : f0  COST :   6036
  include arc     2  13

   ITERATION : 4  SEED FILE : f0  COST :   6036
  include arc     2  11

   ITERATION : 5  SEED FILE : f0  COST :   6032
  increase makespan  by 4
  include arc     8  11

   ITERATION : 6  SEED FILE : f0  COST :   6024
  increase makespan  by 14
  include arc     1  10
```

Figure 5.2 The vehicle schedule generated at the first iteration of the STW Heuristic,and the arcs included in the first six iterations

ITERATION 2: When the FP is solved a vehicle schedule with 4 routes (Cost=8022) is obtained. Once more, it turns out that none of the candidate arcs can be linked,and the makespan is increased by 7, to include arc (10,4).

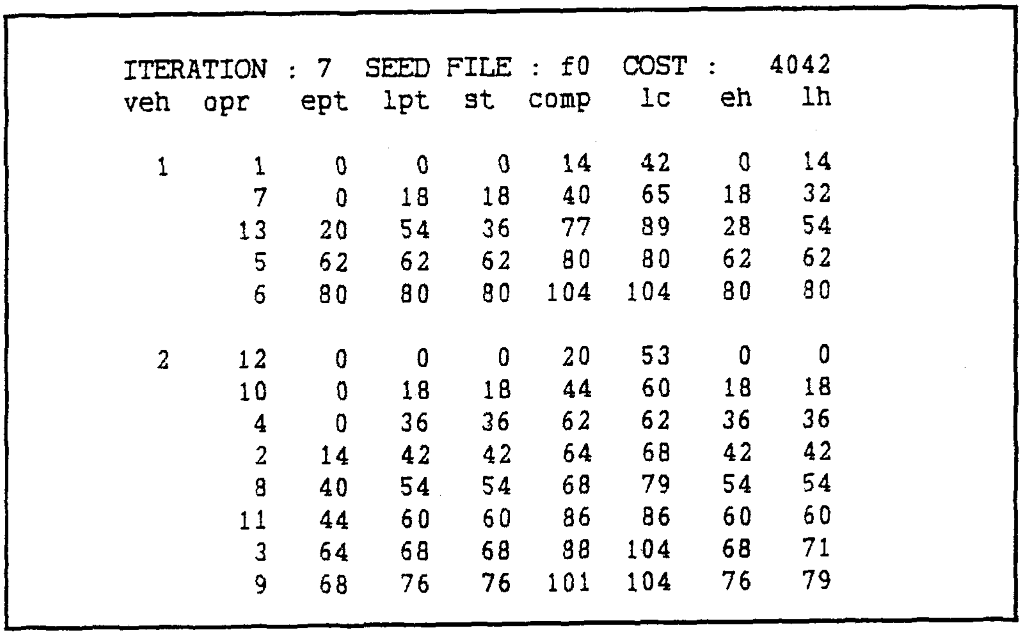
ITERATION 3: FP generates a vehicle schedule with 3 routes (Cost=6036).Candidate arc (2,3) is included in the network.

ITERATION 4: Solution of the new FP is another 3 routed vehicle schedule with the same cost. This time, candidate arc (2,11) is included.

ITERATION 5: Another vehicle schedule with 3 routes is generated, but the cost is reduced to 6032. Candidate arc list is empty , therefore the makespan is increased by 4, and the infeasible arc (8,11) is introduced.

ITERATION 6: FP yields another vehicle schedule with 3 routes (Cost=6024). Since all candidate arcs fail the test, the makespan is increased by 14, and arc (1,10) is joined the network.

ITERATION 7: Finally, FP yields a vehicle schedule with 2 routes ,only (Cost=4042). The algorithm terminates with the solution presented in Figure 5.3. The resulting makespan is 104 .



```
ITERATION : 7  SEED FILE : f0  COST :   4042
veh  opr    ept  lpt  st  comp   lc   eh   lh

 1      1     0    0    0   14   42    0   14
        7     0   18   18   40   65   18   32
       13    20   54   36   77   89   28   54
        5    62   62   62   80   80   62   62
        6    80   80   80  104  104   80   80

 2     12     0    0    0   20   53    0    0
       10     0   18   18   44   60   18   18
        4     0   36   36   62   62   36   36
        2    14   42   42   64   68   42   42
        8    40   54   54   68   79   54   54
       11    44   60   60   86   86   60   60
        3    64   68   68   88  104   68   71
        9    68   76   76  101  104   76   79
```

Figure 5.3 The solution obtained in the seventh iteration

VI. ITERATIVE SOLUTION PROCEDURE

In this chapter, an iterative search procedure in which the machine scheduling and the vehicle scheduling subproblems discussed above, are embedded will be developed. The procedure describes:

(1) A way to move to another machine schedule after employing STW Heuristic to a seed machine schedule, in search of a better solution to the overall machine and vehicle scheduling problem.

(2) A rule to stop the search procedure.

The flowchart of the iterative procedure is given in Figure 6.1. Initially, the set of machine schedules, SS, is empty. An upperbound, UB, is determined. This can be the makespan of a known solution to the original problem (i.e. obtained by a simple heuristic such as the one that will be described in the next chapter), or a big number. Then, an initial machine schedule, SCH_1, is generated using one of the algorithms adapted from Giffler and Thompson in section 4.1. To this seed schedule, STW heuristic algorithm is applied. If the resulting makespan (after incrementing, if necessary) is less than the current upperbound, then a better solution has been found and this becomes the new upperbound. Then, new machine schedules who are neighbors of the seed on hand, are generated and included in SS. This neighborhood concepts will be defined in section 6.1. If there are any schedules in SS, whose makespan are greater than or equal to the already found solution, UB, than these are excluded from SS, thus from further consideration. The seed schedule is also removed from SS, since it has completed its mission. After such eliminations, if SS is still non-empty, then the next iteration starts with a new seed schedule chosen from SS. The procedure is repeated until one of the two stopping criteria is satisfied, either reaching the maximum iteration number, or SS becoming null. The remaining question of how to choose the new seed when SS has more than one element ,will be discussed in section 6.2.

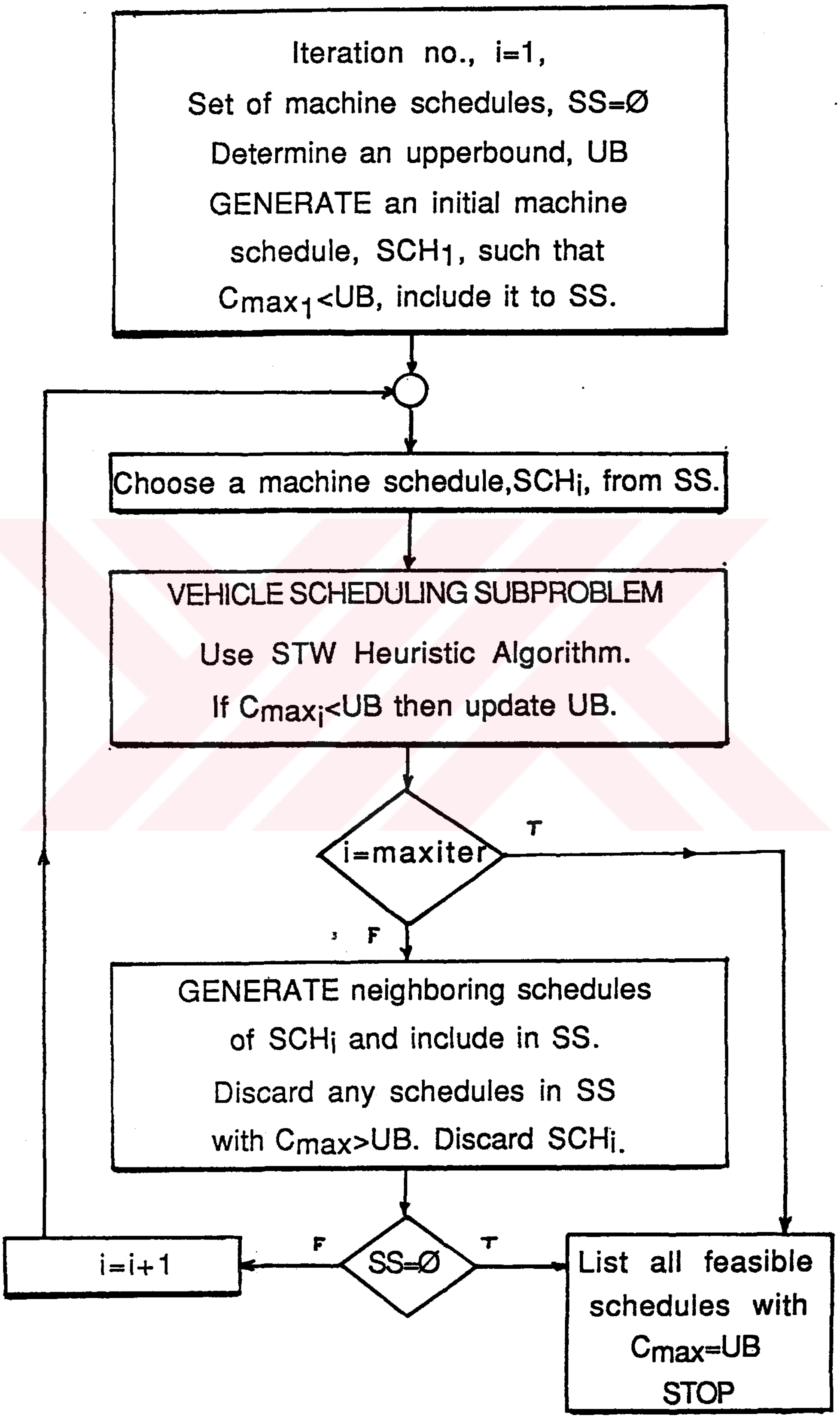


Figure 6.1 Flowchart of the iterative solution procedure

6.1.Neighborhood of a Seed Machine Schedule

At each iteration of the machine schedule generation algorithms of section 4.1., a choice for the operation to be scheduled next is made in Step 3. If the choice made at any one iteration during the generation of the seed schedule on hand is changed, one ends up with a different sequence of operations, and this schedule will be called a neighboring schedule of the seed.

To change the choice for an operation, simply a constraint will be introduced on its earliest starting time, which prevents that particular operation to be selected at that stage.

To be more specific, assume that at some iteration p of the Algorithm 1 of section 4.1., the choice is for operation j to be scheduled on machine m, and that the last operation previously scheduled on m is i. Then, the condition that $est_j < eft^*$ has been satisfied in Step 3 of the algorithm where,

$$eft^* = \min_{k \varepsilon T_m} \left(\max(c_{k-1} + t_k,\ eft_i) + p_k\right),$$

and T_m denotes the set of operations that are yet to be processed on machine m.

If est_j is increased such that it is at least equal to the minimum earliest finishing time among others on T_m, then the condition of Step 3 can no longer be satisfied by operation j in this iteration. Thus, the following constraint will be introduced on est_j:

$$est_j = eft_{k \neq j}^* = \min_{k \varepsilon T_m,\ k \neq j} \left(\max(c_{k-1} + t_k,\ eft_i) + p_k\right)$$

For each machine m, $|T_m|-1$ choices are made, therefore $|T_m|-1$ neighbors can be generated. The complete neighborhood consists of

$\sum_{m=1}^{M} |T_m| - M$ such schedules, each generated by introducing a constraint on one earliest starting time before reapplying the algorithm.

When a schedule generated in this fashion becomes the seed, its neighbors will be created by using the intersection of the seed's constraint with the new constraints. This is illustrated in Figure 6.2.

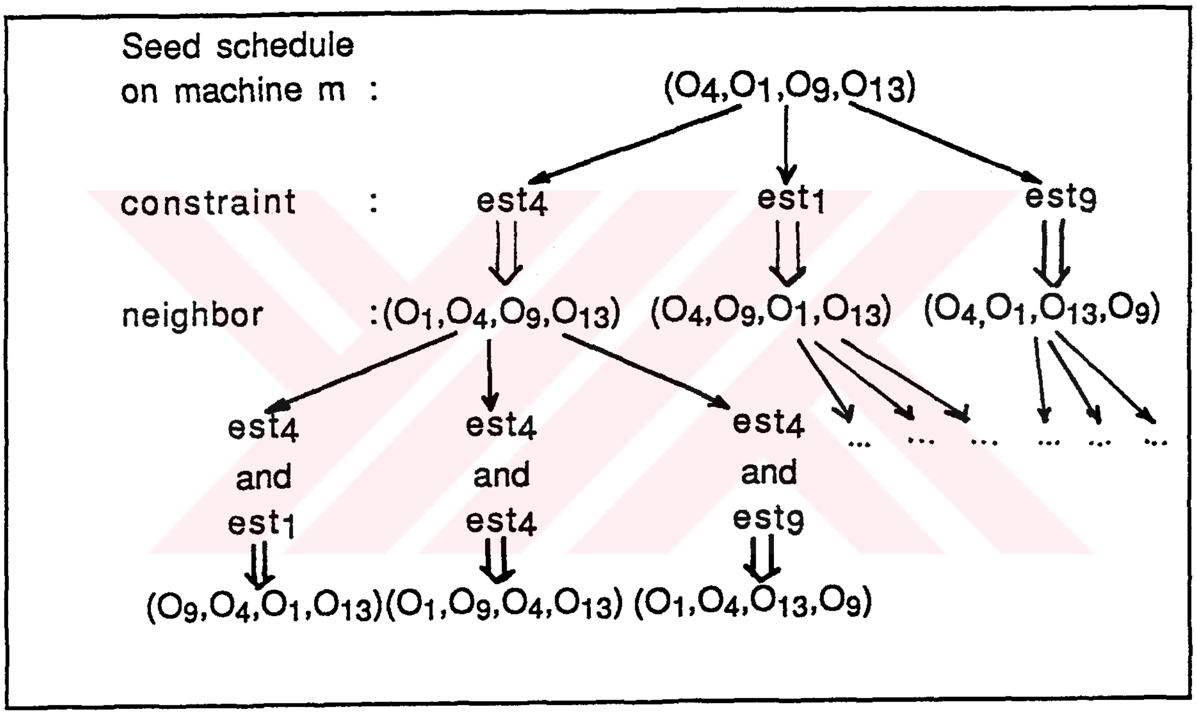


Figure 6.2 Illustration of neighborhood generation(O_i-operation no.).

6.2.Selection of the New Seed Machine Schedule

Two different rules have been used for selecting the new seed:

Rule 1-Minimum Makespan Rule: This is a simple rule based on selecting the machine schedule with the minimum makespan in SS. Although a better machine schedule in terms of machine schedule makespan does not guarantee a better solution to the original machine and vehicle scheduling problem (since the slack times between operations created by a sequence with a longer makespan may fit better to the required handling times in between), one would not like to stick to a sequence with an unnecessarily long makespan, either. So, this is accepted as a logical rule to use during the search.

Rule 2-Maximum Machine Conflict: In developing rule 2, it was expected to make use of some information gained during the STW phase. The concern was to use this information as a feedback in identifying the way the seed schedule should be altered to obtain a possibly better solution.

Recall Condition B in section 6.3.1, used for identifying candidate pairs (i,j) to be linked. There, part (i) is a constraint resulting from the precedence relationships among operations of jobs, which is the data of the original problem, whereas part (ii) is a constraint resulting from the sequence of operations on machines, that is, from the particular machine schedule under consideration at that time.

In this rule, the number of times part (ii) is violated by the sequence on a particular machine , preventing some high opportunity pair (i,j) to be linked, is counted, and used as a penalty on that machine . These numbers are called "machine conflicts". With the expectation that changing the sequence on the machine which caused higher conflict might lead to a better solution, these machine conflicts are assigned as weights to the neighboring schedules of the seed. All the neighbors that are generated by constraining earliest starting time of operations belonging to the same machine receive the same weight: the machine conflict number of that particular machine. Then, the machine schedule with the highest weight among the schedules in SS, is chosen as the new seed. Ties are broken on the basis of minimum makespan.

VII. A SIMPLE HEURISTIC PROCEDURE

Before carrying out numerical experimentation with the proposed iterative algorithm, a simple heuristic procedure will be developed in this chapter so as to be used as a basis of comparison for the numerical results that will be presented in the next chapter. This method can also be employed to obtain a quick initial solution to the problem, which then can be used as an upperbound at the beginning of the iterative solution procedure.

This simple heuristic procedure is again a variation of Giffler and Thompson's active schedule algorithm referred to in section 4.1. This time vehicle availability is also considered during the calculation of the earliest starting times of operations. Vehicle selection is based on the simple rule of selecting the vehicle which arrives first. In addition to those defined in section 4.1., the following definitions will be required for this algorithm.

K denotes the set of vehicles, and k is the vehicle index. In the partial schedule, P_t, each scheduled operation is now represented by (i,m_i,c_i,k_i), where k_i is the vehicle which is assigned for the corresponding loaded trip i. A vector V_t describes the status of vehicles at the beginning of each iteration t, where each vehicle is represented by (k,μ_k,v_k). The last assignment of vehicle k is given by $\mu_k \varepsilon P_t$, and v_k is the time vehicle k becomes available. Now, the earliest time an operation i in the set of schedulable operations, S_t, could be started, est_i, can be redefined as follows:

$$est_i = \max \{ c_{i-1} + t_i,\ c_M, \min_{k \varepsilon K} (v_k + t_{\mu_k i}) + t_i \}$$

The first two terms account for the job and machine availabilities as before, and the last term takes care of the vehicle availability.

With this definition of est_j, both Algorithm 1 (active schedules) and Algorithm 2 (non-delay schedules) of section 4.1 will remain unchanged, except for step 4, where P_{t+1}, and V_{t+1} will be created in the following manner:

P_{t+1} : include operation j, (j, M^*, eft_j, k^*) to P_t, where k^* is the vehicle which reaches the origin of trip j first, which satisfies

$$k^* = \min \ (v_k + t_{\mu_k j}).$$

V_{t+1} : update the status of vehicle k^* as $(k^*, j, v_{k^*}^{t+1})$, where

$$v_{k^*}^{t+1} = \max \ (c_{j-1}, v_{k^*}^{t} + t_{\mu_k j}) + t_j.$$

Again, the three dispatching rules mentioned before , namely MWKR, LWKR, and SPT, will be used for tie breaking in Step 3 of both algorithms. Therefore, there are six versions of this simple heuristic (three active schedule generating and three non-delay schedule generating), and the best result out of these will be compared to the results of the proposed iterative algorithm.

VIII. NUMERICAL STUDY

The iterative algorithm was implemented in PASCAL on IBM-PC2 70 386 and tested first on a set of 22 problems to compare the behavior of different schedule generation and seed schedule selection rules, and the efficiency of the algorithm against the simple heuristic developed in the previous chapter. Then, 10 additional problems are solved to exploit the interaction between the layout configuration and the process routes on the performance of the algorithm. In this chapter, the example problems will be introduced and the numerical results will be reported.

8.1. Example Problems

The main concern while generating the example problems is for the relative magnitudes of the processing times with respect to the travel times, since this study assumes that they are of comparable order of magnitude. Furthermore, in order to have a meaningful distribution of travel times among machines, first, four different layout examples are generated (Figure 8.1). These are basic layout types encountered in literature [2] such as loop or ladder or their combinations. The distance data for each layout in terms of travel times are given in Appendix A.Then, five different job sets with different processing sequences, machine routes and process times are generated and given in Appendix B. From different combinations of these five job sets with the four layouts, 22 example problems are generated, which are described in Table 8.1. In all these problems there are 4 machines and 2 vehicles.The last column on Table 8.1 indicates the ratio of average travel time to average process time for each problem. From here onward, these problems will be sorted in increasing order of $\bar{t}_{ij}/\bar{p}_i$ and presented in three groups. In the first group there are four problems whose ratios are below 20%. The second group consists of nine problems whose ratios are around 50-60%, and the last group contains another nine problems whose ratios are over 70%.

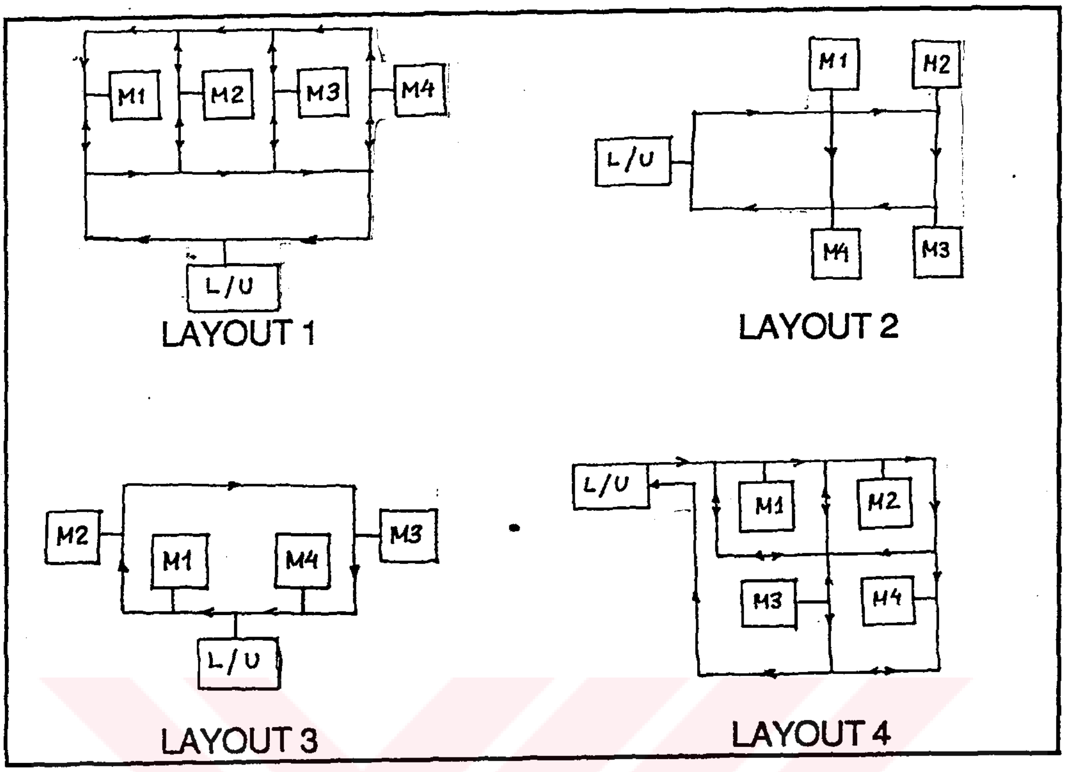


Figure 8.1 The layout configurations used in generating the example problems

8.2. Numerical Study

As a first step, the example problems are solved using the simple heuristic procedure of Chapter 7. Six versions of the algorithm (active or non-delay schedule generation versus three dispatching rules) are applied, and the best solution is recorded to be used later on in making comparisons against the results of the proposed iterative technique. Table 8.2 exhibits these results. There, it is observed that, among active schedule generating versions, MWKR rule performs better, while among non-delay schedule versions SPT rules produces better results most of the time. Based on this observation, with the iterative solution procedure, only these two schedule generation strategies will be used, that is Algorithm 1 with MWKR, and Algorithm 2 with SPT (section 4.1).

TABLE 8.1 Description of example problems

((*) : process times multiplied by two,travel times halved)
((**) : process times multiplied by three,travel times halved)

PROBLEM	Description	$\bar{t}_{ij} / \bar{p}_i$
EX 10	Job Set 1, Layout 1*	0.15
EX 11	Job Set 1, Layout 1	0.59
EX 12	Job Set 1, Layout 2	0.47
EX 13	Job Set 1, Layout 3	0.52
EX 14	Job Set 1, Layout 4	0.74
EX 20	Job Set 2, Layout 2*	0.12
EX 21	Job Set 2, Layout 1	0.61
EX 22	Job Set 2, Layout 2	0.49
EX 23	Job Set 2, Layout 3	0.54
EX 24	Job Set 2, Layout 4	0.77
EX 30	Job Set 3, Layout 3*	0.13
EX 31	Job Set 3, Layout 1	0.59
EX 32	Job Set 3, Layout 2	0.47
EX 33	Job Set 3, Layout 3	0.51
EX 34	Job Set 3, Layout 4	0.74
EX 40	Job Set 4, Layout 4**	0.19
EX 41	Job Set 4, Layout 1	0.91
EX 42	Job Set 4, Layout 2	0.73
EX 43	Job Set 4, Layout 3	0.80
EX 44	Job Set 4, Layout 4	1.14
EX 51	Job Set 5, Layout 1	0.86
EX 54	Job Set 5, Layout 4	1.08

TABLE 8.2 Solutions to different versions of simple heuristic procedure.

PROBLEM	Active SPT	Active MWKR	Active LWKR	Nondelay SPT	Nondelay MWKR	Nondelay LWKR	BEST
EX 20	159	195	173	148*	165	163	148
EX 30	224	222	226	153*	167	186	153
EX 10	206	172	162	126*	151	126	126
EX 40	272	229	236	197*	197*	200	197
EX 12	139	110	120	101	85*	101	85
EX 32	136	113	127	90*	90*	108	90
EX 22	101	122	94	80*	97	89	80
EX 33	147	112	129	94*	94*	117	94
EX 13	129	94*	139	106	106	106	94
EX 23	100	110	104	90*	98	92	90
EX 11	141	108*	154	134	124	134	108
EX 31	129	132	162	125*	125*	125*	125
EX 21	122	122	126	139	106*	137	106
EX 42	121	108	133	103*	105	136	103
EX 14	172	146	172	152	144*	152	144
EX 34	154	130*	173	135	135	157	130
EX 24	153	143	153	131*	141	139	131
EX 43	149	108*	142	127	127	108*	108
EX 51	136	113	137	111*	113	111*	111
EX 41	177	143*	182	169	158	169	143
EX 54	141	129*	141	137	133	137	129
EX 44	199	164	199	178	163*	178	163

The results of these two, under the seed selection rule of "maximum machine conflict" (RULE 2) are presented in Table 8.3. In more than two thirds of the time, Algorithm 2 is superior. Then, for each problem, another strategy is tested by choosing the schedule generation method which performed better, and changing the seed selection rule to "minimum makespan" (RULE 1) . The results, which are displayed on the fifth column of Table 8.3, show that RULE 1 performs at least as good as RULE 2 with the only exceptional case of EX51. This implies that the use of RULE 2 which is more complicated than RULE 1 is not justified.

In the sixth column of the table, percentage of improvement achieved by using the iterative algorithm with RULE 1 over the best simple heuristic result is presented. The results indicate that there is a substantial benefit in using the iterative technique, especially as the ratio of average travel time to average process time gets higher. Actually, for the three groups of problems the average percentage of improvement are 3.8%, 7.1%, and 16.8%, respectively.

Finally, the last column of the table indicates the iteration number in which each solution for RULE 1 was obtained. It is observed that 60% of the solutions were found in less than or equal to 3 iterations, 86% in less than or equal to 7 iterations, and the expected number of iterations required is 5. In all these problems the maximum iteration number allowed was 15. Therefore, one may conclude that it is not necessary to carry on too many iterations, and a relatively small maximum iteration number can be chosen as the stopping criterion. Another observation related to the improvement percentage is that it has no relation with the iteration number.

TABLE 8.3 Summary of results for the iterative solution procedure.

PROBLEM	Best Simple Heuristic	Max. machine conflict		Minimum makespan	% impr.	iter. no.
		ALG 1	ALG 2			
EX 20	148	159	143	143	3.4	2
EX 30	153	166	149	149	2.6	2
EX 10	126	152	126	126	0	1
EX 40	197	197	179	179	9.1	2
EX 12	85	83	82	82	3.5	7
EX 32	90	88	88	88	2.2	7
EX 22	80	86	80	80	0	8
EX 33	94	86	86	86	8.5	7
EX 13	94	84	86	84	10.6	2
EX 23	90	92	90	86	4.4	14
EX 11	108	101	96	96	11.1	2
EX 31	125	105	107	105	16.0	5
EX 21	106	114	106	98	7.5	6
EX 42	103	94	93	93	9.7	1
EX 14	144	109	108	108	25.0	2
EX 34	130	116	118	116	10.8	3
EX 24	131	124	117	116	11.5	7
EX 43	108	95	100	95	12.0	2
EX 51	111	89	90	90	18.9	3
EX 41	143	125	118	118	17.5	12
EX 54	129	99	103	99	23.3	1
EX 44	163	134	126	126	22.7	3

The interpretation of these results are , in fact, quite intuitive. The iterative technique may yield improvement over the simple heuristic basically in two ways:

(1) Its schedule generation scheme, resembling to some extent a branch and bound scheme, may bring out good machine schedules, both in terms of makespan, and in terms of shuffled operations so as to create suitable slack times in between for handling ;

(2) The simple heuristic can be regarded as a real-time schedule generation scheme which uses a dispatching rule coupled with a selection rule without a look-ahead nature. Therefore, the vehicle assignment at any iteration affects the subsequent decisions and may inflate the makespan. On the other hand, the STW Heuristic takes into account the combinatorial nature of the problem, anticipating all of the flow requirements created by a particular machine schedule and making vehicle assignments accordingly.

When the travel times are low relative to process times, both of these effects will be diminished. Because of the long process times, the machine schedules generated will be more likely to have suitable slack times to absorb the low deadheading times required in between; thus, the difference between the C'_{max} at the end of the STW Heuristic will probably be close to original C_{max} of the machine schedule. This will create tight upper bounds for the schedule generation scheme , where any neighboring machine schedules with $C_{max} \geq UB$ are discarded. As a consequence, the number of machine schedules generated during the procedure will be less. Actually, in all of the four problems in the first $\bar{t}_{ij}/\bar{p}_i$ group , the procedure have stopped due to the second stopping criterion; that is, the set of machine schedules became empty. In the other two groups, on the other hand, nearly all problems completed the maximum number of iterations allowed.

Secondly, with relatively low travel times it will be less often that the vehicle availability becomes a constraint during the simple heuristic procedure, so the adverse effect of the myopic vehicle selection will be less important. Furthermore, any improvement obtained through the use of STW heuristic in this respect will be lower in magnitude, since the magnitudes of travel times are low, any way.

This explains why it is beneficial to use the iterative technique instead of a real-time scheme such as the one introduced as the simple heuristic; and how the performance improves as the ratio of the average travel time to the average process time gets higher. However, there may be other factors that influence the performance of the procedure.

TABLE 8.4 Comparison of layouts

LAYOUT 1		LAYOUT 2		LAYOUT 3		LAYOUT 4	
PROBLEM	impr.	PROBLEM	impr.	PROBLEM	impr.	PROBLEM	impr.
EX11	11.1%	EX12	3.5%	EX13	10.6%		
EX21	7.5	EX22	0	EX23	4.4		
EX31	16.0	EX32	2.2	EX33	8.5		
EX41	17.5	EX42	9.5	EX43	12.0	EX14	25.0%
EX51	18.9					EX24	11.5
						EX34	10.8
						EX44	22.7
						EX54	23.3

Reorganizing the results in Table 8.3 as in Table 8.4,additional remarks can be made. There, the problems are grouped according to their layouts, still retaining their $\bar{t}_{ij}/\bar{p}_i$ groups. It is observed that, while high improvements are achieved on layouts 1 and 4, improvements obtained on layouts 2 and 3 are less.

Furthermore, the improvement levels attained by job sets 1, 4 and 5 are higher than those of job sets 2 and 3. When Figure 8.1, and Appendices A and B are analyzed for studying the characteristics of those layouts and job sets, it is observed that layouts1 and 4 are more complex with respect to layouts 2 and 3. Furthermore, job sets 2 and 3 consist of jobs that have simple process routes (i.e., they visit machines in the order they appear on the layouts), whereas job sets 1 and 4 contain process routes where returning to a pre-visited or passed-by machine is required.

Before drawing any conclusions, a few additional examples are solved to clearify and support these observations, since the composition of the available examples is not appropriate for carrying out statistical analysis . For this purpose, layouts 2 and 3, and job sets 2 and 3 are selected. Presently, those job sets and layouts make quite smart combinations with respect to the flow of materials. Layouts are modified by interchanging the locations of two of the machines, in order to suppress the smoothness of the flow. In this way, problem pairs , for which the $\bar{t}_{ij}/\bar{p}_i$ ratios are the same, but material flow characteristics are different , are obtained. In layout 2A, machines M1 and M4, and in layout 3A, machines M2 and M4 are exchanged, and the travel time matrices in Appendix A are reorganized, respectively. The resulting problems are EX22A, EX23A, EX33A, and EX32A. Furthermore, in order to investigate the same effect on problems of low $\bar{t}_{ij}/\bar{p}_i$ ratios, another set of examples is generated, where the travel times of layouts 2A and 3A are halved, and process times of job sets 2 and 3 are multiplied by two. These problems are called EX220A, EX230A, EX330A, and EX320A. They are solved using Algorithm 2 and Rule 1, as well as the simple heuristic , and displayed in Table 8.5, together with their matching problems.

TABLE 8.5 New examples to test the effect of material flow pattern on the performance of the procedure.

PROBLEM	$\bar{t}_{ij}/\bar{p}_i$	BEST SIMPLE HEURISTIC	ITERATIVE PROCEDURE	% IMPR.
EX22	0.49	80	80	0
EX22A		99	97	2.0
EX23	0.54	90	88	2.2
EX23A		116	102	12.1
EX33	0.51	94	86	8.5
EX33A		136	112	17.6
EX32	0.47	90	86	4.4
EX32A		117	99	15.4
EX20	0.12	148	143	3.4
EX20A		153	147	3.9
EX230	0.13	149	146	2.0
EX230A		156	142	6.4
EX30	0.13	153	149	2.6
EX30A		168	152	9.5
EX320	0.12	153	148	3.3
EX320A		168	149	11.3
$t_{stat}=4.5$,	Prob($t \leq 3.5$)= 0.995 with seven degrees of freedom Null hypothesis ($\mu_A - \mu_B = 0$) is rejected			

Paired t-test reveals that different flow patterns significantly affect the performance of the algorithm. With smooth flow patterns where the process routes and the layout configurations suit each other well, the criticality of the vehicle selection is reduced,and therefore the effects of the shortsighted approach are less important. However, as the flow pattern becomes more complex, the performance of the simple heuristic declines, and more benefits are expected from the iterative procedure .

When these new examples are considered together with those in Table 8.3, each in its corresponding $\bar{t}_{ij}/\bar{p}_i$ group, the improvement percentage for each group becomes 5.2%, 8.5%, and 16.8%, respectively.

Also, even when the travel times are low compared to process times, if the jobs to be processed create an inconvenient material flow on the layout, the use of the iterative technique is strongly recommended.

IX. CONCLUSION AND FURTHER CONSIDERATIONS

Material handling is one of the essential components in an FMS. Since only a small proportion of job throughput time is actually devoted to processing, and it is unlikely to achieve significant increase in the processing speed, it is really the efficiency of the material handling system that is critical for the overall efficiency of the FMS. Indeed, any improvement in material handling has great potential in reducing inventory and maximizing throughput.

In previous studies, the subject of scheduling the material handling system has generally been set out either as comparison of various vehicle dispatching rules on a particular layout, or in relation with the design phase. Its coordination and integration with the machine scheduling during the scheduling phase of the FMS has not received much attention.

This dissertation is an attempt to exploit the interactions between these two aspects of scheduling, and to coordinate them by addressing them simultaneously. The iterative algorithm created for this purpose has two important features: first, it anticipates the complete set of flow requirements for a given machine schedule and makes vehicle assignments accordingly, as opposed to a real-time dispatching scheme that uses no information other than the move request queue. Secondly, its search mechanism makes the AGV scheduling an integral part of the scheduling activity, actively participating in the specification of the off-line schedule, rather than just reacting to it.

The impacts of relative magnitudes of processing and travel times ,and the complexity of the material flow pattern defined by the interaction between the process routes and the layout configuration are analyzed. The results presented in section 8.2 demonstrate significant benefits of using the proposed iterative algorithm instead of the simple heuristic introduced which can be regarded as a real-time schedule generation scheme that uses a dispatching rule coupled with a shortsighted vehicle selection rule. The iterative algorithm promises improvement in scheduling especially in environments where cycle times are short and travel times are comparable to them, or where the layout and the process routes do no suit each other.

Furthermore, the heuristic Sliding Time Window algorithm developed in this study is a tool on its own, whose applicability to other problems with similar characteristics should be investigated as further research. A few such potential areas of application may be:

-resource constrained project scheduling when resources have sequence dependent set-up times,

-scheduling on parallel machines with sequence dependent set-ups, where operations have ready-times and due-dates,

-dial-a-ride problems where requested trip start times are given as time intervals.

Another important issue is the real time considerations. Whether or not the iterative procedure can be adapted such that it becomes usable in real-time dynamic scheduling is an open question for research. Also, making use of the insight gained through the performance of this algorithm and some of its notions as "time windows" and "machine conflict" might be interesting in developing a new procedure for the real-time case.

APPENDIX A- TRAVEL TIME DATA USED IN EXAMPLE PROBLEMS

LAYOUT 1 :	L/U	M1	M2	M3	M4
L/U	0	6	8	10	12
M1	12	0	6	8	10
M2	10	6	0	6	8
M3	8	8	6	0	6
M4	6	10	8	6	0

LAYOUT 2 :	L/U	M1	M2	M3	M4
L/U	0	4	6	8	6
M1	6	0	2	4	2
M2	8	12	0	2	4
M3	6	10	12	0	2
M4	4	8	10	12	0

LAYOUT 3 :	L/U	M1	M2	M3	M4
L/U	0	2	4	10	12
M1	12	0	2	8	10
M2	10	12	0	6	8
M3	4	6	8	0	2
M4	2	4	6	12	0

LAYOUT 4 :	L/U	M1	M2	M3	M4
L/U	0	4	8	10	14
M1	18	0	4	6	10
M2	20	14	0	8	6
M3	12	8	6	0	6
M4	14	14	12	6	0

APPENDIX B- Data for the job sets used in example problems

JOB SET 1

JOB	1			2			3			4		5	
OPERATION	1	2	3	4	5	6	7	8	9	10	11	12	13
MACHINE	M1	M2	M4	M1	M3	M2	M3	M4	M1	M4	M2	M3	M1
PROCESS TIME	8	16	12	20	10	18	12	8	15	14	18	10	15

JOB SET 2

JOB	1		2		3		4			5			6		
OPERATION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MACHINE	M1	M4	M2	M4	M1	M3	M2	M3	M4	M1	M2	M4	M1	M2	M3
PROCESS TIME	10	18	10	18	10	20	10	15	12	10	15	12	10	15	12

JOB SET 3

JOB	1		2		3		4		5				6			
OPERATION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MACHINE	M1	M3	M2	M4	M1	M2	M3	M4	M1	M2	M3	M4	M2	M3	M4	M1
PROCESS TIME	16	15	18	15	20	10	15	10	8	10	15	17	10	15	8	15

APPENDIX B (con't)- Data for the job sets used in example problems

JOB SET 4

JOB	1			2			3				4				5				
OPERATION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
MACHINE	M4	M1	M2	M3	M2	M4	M2	M3	M1	M3	M2	M4	M1	M2	M1	M2	M4	M2	M3
PROCESS TIME	11	10	7	12	10	8	7	10	9	8	7	8	12	6	9	7	8	10	8

JOB SET 5

JOB	1			2			3			4		5	
OPERATION	1	2	3	4	5	6	7	8	9	10	11	12	13
MACHINE	M1	M2	M4	M1	M3	M2	M3	M4	M1	M4	M2	M3	M1
PROCESS TIME	6	12	9	18	6	15	9	3	12	6	15	3	9

BIBLIOGRAPHY

1. Greenwood, N.R. , Implementing Flexible Manufacturing Systems, Macmillan Education Ltd, 1988.

2. Groover, M.P., Automation, Production Systems and Computer Integrated Manufacturing, Prentice-Hall International, Inc., 1987.

3. Finke, G., Kusiak, A., "Network Approach to Modelling of Flexible Manufacturing Modules and Cells", RAIRO-APII, Vol 19, No 4, 359-370,1985.

4. Hartley, J., FMS At Work, IFS Ltd, UK, 1984.

5. Bozer, Y.A., "Guided Vehicle Systems: Information/Control System Implication of Alternative Design and Operation Strategies", Advanced Information Technologies for Industrial Material Flow Systems, S.Y.Nof, C.L. Moodie (eds.), Nato ASI Series, Springer-Verlag, 417-436, 1989.

6. Miller, R.K., Automated Guided Vehicles and Automated Manufacturing, Society of Manufacturing Engineers, 1987.

7. Stecke, K.E., "Design Planning, Scheduling, and Control Problems of Flexible Manufacturing Systems", Annals of Operations Research, 3, 3-12, 1985.

8. Kıran, A.S., Tansel, B.C., "A Framework for Flexible Manufacturing Systems", 1985 Annual International Industrial Engineering Conference Proceedings, Institute of Industrial Engineers, 446-450, 1985.

9. Kusiak, A., "Application of Operational Research Models and Techniques in Flexible Manufacturing Systems", EJOR, 24, 336-345, 1986.

10. Fry, T.D., Smith,A.E. "FMS Implementation Procedure: A Case Study" , IIE Transactions, 21, No.3, 288-293, 1989.

11. Michael, G.J., Millen, R.A., "Economic Justification of Modern Computer-Based Factory Automation Equipment: A Status Report", FMS:Operations Research Models and Applications, K.E.Stecke, R.Suri (eds.), Baltzer AG, 25-34, 1985.

12. Burstein, M.C., Talbi,M., "Economic Evaluation for the Optimal Introduction of Flexible Manufacturing Technology Under Rivalry", FMS: Operations Research Models and Applications, K.E.Stecke, R.Suri (eds.), Baltzer AG ,81-112 ,1985.

13. Burstein, M.C. "Finding the Economical Mix of Rigid and Flexible Automation for Manufacturing", Proc. of the Second ORSA/TIMS Conf.on FMS, K.E.Stecke, R.Suri (eds.) Elsevier, 43-54, 1986.

14. Fine, C.H., Freund, R.M., "Economic Analysis of Product-Flexible Manufacturing Systems Investment Decisions", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E. Stecke, R.Suri (eds.), Elsevier, 55-68, 1986.

15. Gaiman, C. "The Strategic Decision to Acquire Flexible Technology", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E.Stecke, R.Suri (eds.), Elsevier, 69-82, 1986.

16. Srinivasan, V., Millen R.A.,"Evaluating FMSs a Strategic Investment", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E.Stecke, R.Suri (eds.),Elsevier, 83-94, 1986.

17. Primrose, P.L., Leonard,R., "Conditions Under Which Flexible Manufacturing is Financially Viable", FMS: Methods and Studies, A.Kusiak, (ed.), Elsevier, 355-366,1986.

18. Miltenburg, G.J., Krinsky, I., "Evaluating Flexible Manufacturing Systems", IIE Transactions, 19, No.2, 222-233, 1987.

19. Pollard, P.L., Tapscott, R.L., "Investment Evaluation Methodology", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E.Stecke, and R.Suri, (eds.), Elsevier, 47-52, 1989.

20. Kusiak, A., "FMS: A Structural Approach", Int. Journal of Prod. Res., 23, No.6, 1057-1073, 1985.

21. Falkner, C.H., "Flexibility in Manufacturing Plants" Proc. of the Second ORSA/TIMS Conf. on FMS, K.E.Stecke, R.Suri, (eds.), Elsevier, 95-106, 1986.

22. Carter, M.F. "Designing Flexibility into Automated Manufacturing Systems", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E. Stecke, R.Suri, (eds.), Elsevier, 107-118, 1986.

23. Mandelbaum, M., Buzacott,J.A., "Flexibility and Its Use: A Formal Decision Process and Manufacturing View", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E.Stecke, R.Suri, (eds.), Elsevier, 119-130, 1986.

24. Kumar, V., "On Measurement of Flexibility in FMS: An Information-Theoretic Approach", Proc. of the Second ORSA/TIMS Conf. on FMS , K.E.Stecke, R.Suri, (eds.), Elsevier, 131-144, 1986.

25. Chung, C., Chen, I.,"A Systematic Assessment of the Value of Flexibility for an FMS", Proc. of the Second ORSA/TIMS Conf. on FMS K.E.Stecke, R.Suri, (eds.), Elsevier, 27-32, 1989.

26. Sarin, S.C., Shen,C.S., "A Mathematical Model for Manufacturing System Selection", FMS: Methods and Studies, A.Kusiak, (ed.), North-Holland, 99-112, 1986.

27. Lee, H.F., Srinivasan, M.M., Yano, C.A., "An Algorithm for the Minimum Cost Configuration Problem in Flexible Manufacturing Systems", FMS: Methods and Studies, A.Kusiak, (ed.), North-Holland, 85-90, 1989.

28. Solberg, J.J., "Optimal Design and Control of Computerized Manufacturing Systems", Proc. AIEE Systems Engineering Conf., 138-147, 1976.

29. Buzacott, J.A., Shantikumar, T.G., "Models for Understanding FMS", AIIE Transactions, 12, No.4, 339-350, 1980.

30. Browne, J., Chan,W.W., Rathmill, K., "In Integrated FMS Design Procedure", Annals of Opr. Res., 3, 207-237, 1985.

31. Whitney, C.K., Suri,R., "Algorithms for Part and Machine Selection in FMS", Annals of Opr.Res., 3,239-261, 1985,

32. Dietrich, B.L., March,B.M., "An Application of Hybrid Approach to Modelling a FMS", Annals of Opr.Res., 3,263-276, 1985.

33. Dallery, Y., Frein,Y., "An Efficient Method to Determine the Optimal Configuration of a FMS", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E.Stecke, R. Suri,(eds.), Elsevier, 269-282, 1986.

34. Afentakis, P., Millen,R.A., Solomon, M., "Layout Design for FMS: Models and Strategies, Proc. of the Second ORSA/TIMS Conf. on FMS, K.E. Stecke, R. Suri,(eds.), Elsevier, 221-232, 1986.

35. Mellichamp, J.M., Kwon, O. Wahab, A.F., "FMS: Designer: An Expert System for FMS Design", Int. Jour. of Prod.Res., 28, 2013-2024, 1990.

36. Villa, A., Rosetto, S., "Towards a Hierarchical Structure for Production Planning and Control in FMS", Modelling and Design of FMS, A. Kusiak, (ed.), Elsevier, 209-228, 1986.

37. White, C. "Production Scheduling and MRP for Flexible Manufacturing", Modelling and Design of FMS ,A. Kusiak, (ed.), Elsevier, 229-248, 1986.

38. Mazzola, J.B., "Heuristics for the FMS/MRP Rough-Cut Capacity Planning Problem", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E.Stecke, and R. Suri (eds.), Elsevier, 119-126, 1989.

39. Chun, S., Lee, T., "A Heuristic Method for Solving an FMS Master Production Scheduling Problem, Proc. of the Third ORSA/TIMS Conf. on FMS, K.E.Stecke, and R.Suri (eds.), Elsevier, 127-132 1989.

40. Stecke, K.E., "A Hierarchical Approach to Solving Machine Grouping and Loading Problems of FMSs", EJOR, 24 369-378, 1986.

41. Kıran, A.S., Tansel, B.C., "Mathematical Programming Models for FMSs", Working Paper 86-01, Univ.of Southern California, Dept. of Ind. and Systems Eng., 1986.

42. Rajagopalan S., "Formulation and Heuristic Solutions for Parts Grouping and Tool Loading in Flexible Manufacturing Systems", Proc. of the Second ORSA/TIMS Conf. on FMS, K.E.Stecke, R.Suri (eds), Elsevier, 311-320, 1986.

43. Whitney, C.K., Gaul, "Sequential Decision Procedures for Batching and Balancing in FMSs", Annals of Opr. Res., 3, 301-316, 1985.

44. Kusiak, A., "The Part Families Problem in FMS", Annals of Opr. Res., 3,279-300, 1985.

45. Hwang, S., "A Constraint-Directed Method to Solve the Part Selection Problem in FMS Planning Stage", Proc. of the Second ORSA/TIMS Conf.on FMS, K.E. Stecke, R.Suri (eds), Elsevier, 297-310, 1986.

46. O'Grady, P.J., Menon, U., "Loading a Flexible Manufacturing System", Int. Journal of Prod. Res., 25, 1053-1068, 1987.

47. Dubois, D., Stecke, K.E., "Using Petri-Nets to Represent Production Processes", Proceedings of the 22 nd IEEE Conf. on Decision and Control, San Antonio, TX, 1983.

48. Stecke, K.E., Solberg, J.J., "The Optimality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multiserver Queues", Operations Research, 33, 882-910, 1985.

49. Stecke, K.E., "Formulation and Solution of Nonlinear Integer Production Planning Problems for FMSs", Management Science, 29, 273-287, 1983.

50. Stecke, K.E., Solberg, J.J. "Loading and Control Policies for a Flexible Manufacturing System", Int.Journal of Prod.Res., 19, 481-490, 1981.

51. Kusiak, A., "Loading Models in Flexible Manufacturing Systems", Flexible Manufacturing: Recent Developments in FMS, Robotics, CAD/CAM, CIM", A., Ahmad, Raouf, S.I., (eds.), Elsevier, 119-132, 1985.

52. Ammons, J.C., Lofgren, C.B., Mc Ginnis, L.F., "A Large Scale Machine Loading Problem in Flexible Assembly", Annals of Opr. Res., 3, 319-322, 1985.

53. Stecke, K.E., Talbot, F.B., "Heuristics for Loading FMSs", Flexible Manufacturing: Recent Developments in FMS, Robotics, CAD/CAM,CIM", A.Raouf, and S.I.Ahmad., (eds.), Elsevier, 73-86, 1985.

54. Moreno, A.A., Ding, F., "Goal Oriented Heuristics for the FMS Loading (and Part Type Selection) Problems", Proc. of the Third ORSA/TIMS Conf.on FMS, K.E. Stecke, R., Suri, (eds), Elsevier, 105-110, 1989.

55. Lashkari, R.S., Dutta, S.P., Padhye, A.M., "A New Formulation of Operation Allocation Problem in FMSs: Mathematical Modeling and Computational Experience", Int. Journal of Prod.Res., 25, 1267-1283, 1987.

56. Sarin, S.C., Chen, C.S., "The Machine Loading and Tool Allocation Problem in a Flexible Manufacturing Systems", Int. Journal of Prod.Res., 25, 1081-1094, 1987.

57. Shanker, K., Rajamarthandan, S.,"Loading Problem in FMS: Part Movement Minimization", Proc.of the Third ORSA/TIMS Conf. or FMS, K.E.Stecke, R., Suri (eds), Elsevier, 99-104, 1989.

58. Stecke,K.E., Morin,T.L., "The Optimality of Balancing Workloads in Certain Types of Flexible Manufacturing Systems", EJOR, 20, 68-82, 1985.

59. Hitz, K.L., "Scheduling of Flexible Flow Shops-I, Report LIDS-R-879, Laboratory for Information and Decision Systems", MIT, Cambridge, MA, 1979.

60. Hitz, K.L., "Scheduling of Flexible Flow Shops-II, Report LIDS-R-380, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, 1980.

61. Erschler, J., Roubellat, F., and Thutiot, C., "Steady state scheduling of a Flexible Manufacturing System with Periodic Releasing and Flow Time Constraints, Flexible Manufacturing Systems"; Operation Research Models and Applications, K.E., Stecke, and R.Suri (eds.), J.C., Baltzer AG,333-354. 1985.

62. Chang, Y.L., Sullivan, R.S.,Bagchi, U., and Wilson, J.R., "Experimental Investigation of Real-Time Scheduling in Flexible Manufacturing Systems", Flexible Manufacturing Systems: Operations Research Models and Applications, K.E. Stecke and R.Suri (eds.), J.C.Baltzer AG, 355-378., 1985.

63. Afentakis,P., "Maximum Throughout in Flexible Manufacturing Systems",Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems", K.E., Stecke and R.Suri (eds.), Elsevier, Amsterdam, 509-520, 1985.

64. Kusiak, A., "Scheduling Flexible Machining and Assembly Systems", Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems, K.E. Stecke and R.Suri (eds.), Elsevier, Amsterdam, 521-532, 1986.

65. Kimemia, J., and Gershwin, S.B., "An Algorithm for the Computer Control of a Flexible Manufacturing System", IIE Transactions, 15, No.4, 353-362,1983.

66. Akella, R., Choong, Y., and Gerschwin, S.B., "Real-Time Production Scheduling of an Automated Cardline", Annals of Opr.Res., 3, 3-12, 1985.

67. Nasr, N., Elsayed, E.A., "Job Shop Scheduling with Alternative Machines", Int.Journal of Prod.Res., 28, 1595-1609, 1990.

68. Sawik, T., "Modelling and Scheduling of a Flexible Manufacturing System", EJOR, 45, 177-190, 1990.

69. Erschler, J., Robellat, F., "An Approach to Solve Workshop Real Time Scheduling Problems", Advanced Information Technologies for Industrial Material Flow Systems", Y.Shimon and C.L. Moodie (eds.),Nato ASI Series, Springer-Verlag, 1989.

70. Chan, D., Bedworth, D.D., "Design of a Scheduling System for Flexible Manufacturing Cells", Int.Journal of Prod.Res., 28, 2037-2049, 1990.

71. Yao, D.D., Pei, F.F., "Flexible Parts Routing in Manufacturing Systems", IIE Transactions, 22, 48-55, 1990.

72. Greene, T.J., and Sadowski, R.P., "A Mixed Integer Program for Loading and Scheduling Multiple Flexible Manufacturing cells", European Journal of Operation Research, 24, No.3, 379-386, 1986.

73. Hutchison, J., Leong, K., Snyder, D.L., and Ward, P., "Scheduling for Random Job Shop Flexible Manufacturing Systems", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E.Stecke, and R.Suri, (eds.), Elsevier, 161-166, 1989.

74. Afentakis, P., Solomon M.M., and Millen, A.R., "Minimizing the Work-in-Process in FMSs which Use Cyclic Schedules", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E. Stecke, and R.Suri (eds.), Elsevier ,167-172 ,1989.

75. Dietrich, B.L., Escudero, L.F., "On Solving a O-l Model for Workload Allocation on Parallel Unrelated Machines with Set-ups", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E. Stecke, and R.Suri, (eds.), Elsevier, 181-186, 1989.

76. Dietrich, B.L., "A Two Phase Heuristic for Scheduling on Parallel Unrelated Machines with Set-ups", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E.Stecke, and R.Suri,(eds.), Elsevier,187-192 , 1989.

77. Thesen, A., and Lei, L., "An Expert System for Scheduling Robots in a Flexible Eloctroplating System with Dynamically Changing Workloads, Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems, K.E. Stecke and R.Suri, (eds.), Elsevier, Amsterdam, 555-565, 1986.

78. Lin, L.S., and Chung, S., "A Systematic FMS Model of Real-time On-line Control and Question-Answerer Simulation Using Artificial Intelligence", Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems, K.E. Stecke, and R.Suri (eds.), Elsevier, Amsterdam, 567-580,1986.

79. Shen, S., and Chang, Y., "An AI Approach to Schedule Generation in a Flexible Manufacturing System", Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems, K.E.Stecke, and R.Suri, (eds.), Elsevier, Amsterdam, 581-592, 1986.

80. Subramanyam, S., and Askin, R.G., "An Expert Systems Approach to Scheduling in Flexible Manufacturing Systems", Flexible Manufacturing Systems: Methods and Studies, A. Kusiak, (ed.), North Holland, Amsterdam, 243-256, 1986.

81. Park, S., Raman, N., and Shaw, M.J., "Heuristic Learning for Pattern-Directed Scheduling in a FMS", Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems, K.E. Stecke, and R. Suri (eds.), Elsevier, 369-376,1989.

82. Morton, T.E., and Smunt, T.L., "A Planning and Scheduling System for Flexible Manufacturing", Flexible Manufacturing Systems: Methods and Studies, A.Kusiak, (ed.), North Holland, Amsterdam, 151- 164,1986.

83. Shaw, M.J., and Whinston, A.B., "Application of Artificial Intelligence to Planning and Scheduling", Flexible Manufacturing Systems: Methods and Studies, A.Kusiak(ed.),North Holland, Amsterdam, 223-242, 1986.

84. Ben-Arieh, D., Knowledge Based Control System for Automated Production and Assembly, Modelling and Design of Flexible Manufacturing Systems, A.Kusiak (ed.),Elsevier, N.Y., 347-368, 1986.

85. Maley, J., Ruiz-Myer, S., and Solberg, J., "Dynamic Control in Automated Manufacturing: A Knowledge Integrated Approach", Int. Journal of Prod. Res., 26, 1739-1748, 1988.

86. O'Grady, P., Lee, K., "An Intelligent Cell Control System for Automated Manufacturing, Int. Journal of Prod.Res., 1988.

87. Rabelo, L.C., Alptekin, S., "Synergy of Neural Networks and Expert Systems for FMS Scheduling", Proc.of Third ORSA/TIMS Conf. on FMS, K.E. Stecke, and R. Suri (eds.), Elsevier, 361-366, 1989.

88. Davis, W.J., Jones, A.T., "On-Line Concurrent Simulation in Production Scheduling", Proc.of Third ORSA/TIMS Conf. on FMS, K.E. Stecke, and R. Suri (eds.), Elsevier, 253-258, 1989.

89. Grosseschallau, W., Kusiak, A., "An Expert System for design of Automated Material Handling Systems (INSIMAS)", Material Flow, 2, 157-166, 1985.

90. Chang, Y., Sullivan, R.S., and Wilson, J.R., "Using SLAM to Design the Material Handling System of a FMS", Int.Journal of Prod. Res., 24,15-26, 1986.

91. Mahadevan, B., Narendran, T.T., "Design of an Automated Guided Vehicle-Based Material Handling System for a FMS", Int. Journal of Prod. Res., 28, 1611-1622, 1990.

92. Malmborg, C.J., "A Model for the Design of Zone Control Automated Guided Vehicle Systems", Int. Journal of Prod. Res., 28, 1741-1758, 1990.

93. Maxwell, W.L., and Muckstadt, J.A., "Design of Automated Guided Vehicles, IIE Transactions, 14, 2, 114-124, 1982.

94. Egbelu,P.J., and Tanchoco, J.M.A., "Characterization of Automated Guided Vehicle Dispatching Rules, Int. Journal of Prod.Res., 359-374, 22, 1984.

95. Sabuncuoğlu, I., Hommertzheim, D.L., "An Investigation of Machine and AGV Scheduling Rules in an FMS", Proc. of the Third ORSA/TIMS Conf. on FMS, K.E.Stecke and R. Suri, (eds.), Elsevier, 261-266, 1989.

96. Bartholdi, J.J., Platzman, L.K., "Decentralized Control of AGVs on a Simple Loop", IIE Transactions, 21, 79-81, 1989.

97. Han, M., Mcginnis, L.F., "Control of Material Handling Transporter in Automated Manufacturing", IIE Transactions, 21, 184-190, 1989.

98. Raman, N., Talbot, F.B., and Rachamadgu R.V., "Simultaneous Scheduling of Machines and Material Handling Devices in Automated Manufacturing". K.E.Stecke, and R.Suri, (eds.), Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems, Elsevier, Amsterdam, 455-466, 1986.

99. Lenstra, J.K., Rinooy Kan, A.H.G., " Complexity of Scheduling under Precedence Constraints", Operations Research, 26,1, 22-35, 1978.

100. Kusiak,A.,"Material Handling in Flexible Manufacturing Systems", Material Flow,2,79-95, 1985.

101. French,S.,Sequencing and Scheduling, Ellis Horwood Limited, England,1982.

102. Desrosiers, J., Soumis, F., and Desrochers, M.," Routing with time Windows by Column Generation", Publication 278, Centre de Recherche sur le Transports,Universite de Monreal,1983.

103. Desrosiers, J., Soumis, F., Desrochers, M., and Sauve, M.,"Routing and Scheduling by Branch and Bound on time Windows", G-83-16, Centre de Recherche sur le Transports,Universite de Monreal,1983.

104 Bratley, P.,Florian, M., and Robillard, P.,"Scheduling With Earliest Start and Duedate Constraints on Multiple Machines", Naval Research Logistics Quarterly,22,1,165-173,1975.

105. Gertsbakh, L.,Stern, H., "Minimal Resources for Fixed and Variable Job Schedules", Operations Research,26,175-182,1986.

106. Orloff,C.S., "Route Constrained Fleet Scheduling", Transportation Science, 10,2,149-168, 1976.

107. Ferland, J.A., Fortin, L., "Vehicles Scheduling with Sliding Time Windows", EJOR, 38,213-226, 1989.