CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Scheduling a part of manufacturing system’s control process is necessary when a common set of resources needs to be shared to manufacture different products during the same time period. FMS scheduling has received large amount of attention, because it has the potential to dramatically decrease costs and increase throughput, thereby, profits. This chapter is focused on the literature survey conducted in the domains of FMS scheduling. The literature related to FMS scheduling problem is presented in Section 2.2. Metaheuristics used in FMS scheduling and its applications are presented in Section 2.3. The summary of the literature review is presented in Section 2.4.

2.2 FMS SCHEDULING PROBLEM

The details of the different types of FMS scheduling problems, complexities and the different types of algorithms available to solve the FMS scheduling problems are presented below.

The theory of scheduling is characterized by a virtually unlimited number of problem types. Generally, it is specified as job characteristics and optimality criterion. The job characteristics in scheduling are generally specified using parameters such as pre-emption condition, precedence relationships, processing time, release dates and due dates as shown in Table 2.1.
Table 2.1 Parameters of job characteristics in scheduling

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameter types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pre-emptive condition</td>
<td>It means that processing may be interrupted and resumed later. With pre-emption, a job or operation may be interrupted several times. If job or operation splitting is not allowed, then it is ‘without pre-emption’.</td>
</tr>
<tr>
<td>2</td>
<td>Precedence relationship</td>
<td>If the jobs or operations of a job are to be processed strictly in a specific sequence, then the jobs or operations are constrained with precedence relationship</td>
</tr>
<tr>
<td>3</td>
<td>Processing time</td>
<td>It may either be deterministic (predictable exactly before the production run) or stochastic (vary during production run)</td>
</tr>
<tr>
<td>4</td>
<td>Release dates</td>
<td>It specifies the job arrival pattern, whether it is all available at the starting of the scheduling period or arrive in future</td>
</tr>
<tr>
<td>5</td>
<td>Due dates</td>
<td>It indicates the time at which the jobs are to be delivered</td>
</tr>
</tbody>
</table>

The optimality criterion used in FMS scheduling problem falls into any one of the two types: makespan time related and due date related. The general classification of the scheduling objectives is given in Table 2.2. The scheduling problems are formulated with either single objective or multiple objectives (i.e., with more than one objective are simultaneously considered).

Table 2.2 Parameters of optimality criterion in scheduling

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameter types</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | Makespan related    | 1. Finish the last job as soon as possible  
2. Finish each job as soon as possible  
3. Maximize machine utilization |
| 2      | Due date related    | 1. Minimize the number of late jobs  
2. Minimize the total tardiness  
3. Minimize the maximum lateness of any jobs |
Classic notations used to describe Scheduling Problems:

- $FFc \mid r_j \mid \sum w_j T_j$ denotes a flexible flow shop. The jobs have release dates and due dates, and the objective is the minimization of the total weighted tardiness.

- $Pm \mid r_j, M_j \mid \sum w_j T_j$ denotes a system with $m$ machines in parallel. Job $j$ arrives at release date $r_j$ and has to leave by the due date $d_j$. Job $j$ may be processed only on one of the machines belonging to the subset $M_j$. If job $j$ is not completed in time, a penalty $w_j T_j$ is incurred.

- $1 \mid r_j, prmp \mid \sum w_j C_j$ denotes a single machine system with job $j$ entering the system at its release date $r_j$. Preemptions are allowed. The objective to be minimized is the sum of the weighted completion times.

- $1 \mid s_{jk} \mid C_{max}$ denotes a single machine system with $n$ jobs subject to sequence dependent setup times, where the objective is to minimize the makespan. It is well known that this problem is equivalent to the Traveling Salesman Problem (TSP), where a salesman has to tour $n$ cities in such a way that the total distance traveled is minimized.

- $P_j \mid prec \mid C_{max}$ denotes a scheduling problem with $n$ jobs subject to precedence constraints and an unlimited number of machines in parallel. The total time of the entire project has to be minimized.

- $Fm \mid p_{ij}=p_j \mid \sum w_j C_j$ denotes a proportionate flow shop environment with $m$ machines (i.e. $m$ machines in series), with the processing times of job $j$ on all $m$ machines identical and
equal to \( p_j \). The objective is to find the order in which the \( n \) jobs go through the system so that the sum of the weighted completion times is minimized.

- \( J_m \mid C_{\text{max}} \) denotes a job shop problem with \( m \) machines. There is no recirculation, so a job visits each machine at most once. The objective is to minimize the makespan.

- \( F Jc \mid r_j, s_{ijk} \mid \sum w_j T_j \) refers to a flexible job shop with \( c \) work centers. The jobs gave different release dates and are subject to sequence dependent set up times that are machine dependent. There is no recirculation, so a job visits each work center at most once. The objective is to minimize the total weighted tardiness.

**Computational Complexity:** In general, FMS scheduling problems belong to a broader class of combinatorial search problems. A combinatorial search problem \( \pi \) is a set of pairs \( (I, A) \), where \( I \) is an instance of a problem with specified values, and \( A \) is an answer to the instant (Mattfeld 1996). Complexity theory provides a mathematical framework in which computation problems are studied and classified as easy and hard. One of the main issued of complexity theory is to measure the performance of algorithms with respect to computational time. Time complexity function is given as: \( O[p(k)] \), where \( p \) is some polynomial and \( k \) is the input length of an instant).

The scheduling algorithms to generate optimal schedules for different production environments are traditionally developed by using optimization and approximation approaches (King 1975). The scheduling algorithms can be classified as shown in Table 2.3.
### Table 2.3 Classification of scheduling algorithms

<table>
<thead>
<tr>
<th>Class</th>
<th>Types of algorithms</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization</td>
<td>Enumerative procedure</td>
<td>It is based on enumerating all combinations, sorting the feasible set and selecting the best solution from the feasible set.</td>
</tr>
<tr>
<td></td>
<td>Mathematical programming</td>
<td>These are the formulation with a set of equation that represents the constraints and the objective criteria.</td>
</tr>
<tr>
<td></td>
<td>Branch-Bound algorithm</td>
<td>It is based on the idea of intelligently enumerating the feasible solutions with lower and upper bounds.</td>
</tr>
<tr>
<td></td>
<td>Priority rules</td>
<td>These are simple sequencing rules that specify the queue discipline.</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>These algorithms rely on rules of thumb. Any approach without formal guarantee of performance can be considered heuristic.</td>
</tr>
<tr>
<td>Approximation</td>
<td>Local search</td>
<td>These algorithms use the concept of neighbourhood search for better solutions within the neighbours and move towards optimal.</td>
</tr>
<tr>
<td></td>
<td>Evolutionary algorithms</td>
<td>Based on the recognition that evolution, with its principles of mutation and selecting, represents an efficient process for solving hard optimization problems.</td>
</tr>
</tbody>
</table>

Stecke (1985) described the problems and decisions that have to be addressed during the design, planning, scheduling and control of an FMS. Rachamadugu and Stecke (1994) classified, compared and reviewed the flexible manufacturing system scheduling procedures. The choice of appropriate scheduling criteria is discussed. Basnet and Mize (1994) presented a review of literature concerning the scheduling methodology under six categories: mathematical programming, multi-criteria decision-making, heuristic oriented, control theoretic, simulation and artificial intelligence. Articles emphasizing many methodological perspectives are critically reviewed. Chan et al (2002) reviewed a number of papers about the scheduling study of FMS by simulation, from general studies to multi criteria approaches, and to artificial intelligence approaches.
2.2.1  Job and Tool Scheduling

Giffler and Thompson (1960) developed an enumerative procedure to generate all active schedules for the general ‘n’ jobs and ‘m’ machines problems. Greenberg (1968) formulated a mixed integer programming model for the general ‘n’ jobs, ‘m’ machines problems and suggested a branch and bound technique to reduce the formulation as a series of non-integer linear programming problem. Dispatching rules have been applied consistently to scheduling problems and its procedures are designed to provide good solutions to complex problems in real-time. The term dispatching rule, scheduling rule, sequencing rule, or heuristic are often used synonymously (Baker 1974, Panwalkar and Iskander 1977, Blackstone et al 1982). Stecke and Solberg (1982) investigated the performance of dispatching rules with five loading policies in conjunction with sixteen dispatching rules in the simulated operation of an actual FMS. Iwata et al (1982) reported on a set of decision rules to control FMS that is to select machine tools, cutting tools and transport devices in a hierarchical framework.

Kusiak (1985) discussed FMS in its broadest sense to include fabrication, machining and assembly, and have done a brief structural taxonomy of FMS. Chang et al (1985) reported a two-step method for scheduling parts using simulation. The suggested procedure was compared with some dispatching rules including Shortest Processing Time (SPT), Longest Processing Time (LPT), First-come first-served (FCFS), Most Work Remaining (MWKR) and Least amount of Work Remaining (LWKR). The performance measure for this comparison was mean flow time. Shanker and Tzen (1985) have formulated the loading and dispatching problem in a random flexible manufacturing system using integer programming and suggested heuristics methods for solution. Buzacott and Yao (1986) presented a comprehensive review of the analytical models developed for the design and
control of FMS. Stecke (1986) addressed grouping and loading problems by formulating as nonlinear integer programs. Selladurai et al (1986) developed in the C language, visual interactive computer graphics simulation software for analyzing dynamic scheduling of an FMS.

Sarin and Chen (1987) approached a loading problem of FMS from the view point of machining cost and proposed an integer programming formulation. Wilson (1987) suggested a methodology to formulate and solve a set of sequencing problems using integer programming. Lashkari et al (1987) developed a mathematical formulation for the loading problem of FMS consider with refixturing and limited tool availability. Chang et al (1989) reported on heuristics based beam search technique designed to solve the random FMS scheduling problem. Han et al (1989) proposed a non-linear integer programming model for the setup and scheduling problem in a specific type of FMS wherein all the machines are of the same type, and tools are borrowed between machines and form the tool crib is needed. Lee and Jung (1989) formulated a part selection and allocation problem using goal programming and considered the goals of meeting the production requirements, balancing of machine utilization and minimization of throughput time of parts.

Gershwin (1989) used the notion of temporal decomposition to propose a mathematical programming framework for analysis of production planning and scheduling. Chang and Sullivan (1990) addressed a scheduling problem of a dynamic job shop, in which Giffler and Thompson algorithm is used to generate all active feasible schedules and pick the optimal schedule. Co et al (1990) have suggested a four pass approach based on integer programming method to solve the balancing, loading and tool configuration problems of a random FMS. Srinivasan et al (1990) addressed the problem of grouping of parts using clustering methods and integer programming.
Hutchison et al (1991) developed a mathematical formulation of a random FMS scheduling problem, to minimize the makespan as a mixed integer 0-1 programming model. Yih and Thewn (1991) provided a semi-martkov decision model for a real time scheduling problem to determine the sequence of parts within a FMS. Co (1992) used mathematical programming tools to investigate the benefits of restricting production flexibility of FMS and suggested a methodology for streamlining the material flow. Gyampah et al (1992) conducted a simulation study on the performance of various strategies for tool allocation and part type selection.

Mukhopadhyay et al (1992) suggested a heuristic solution to the loading problem of developing the concept of essentially ratio for the objective of minimizing the system unbalance and thereby maximizing the throughput. The proposed heuristic is tested on ten problems and the results show that the algorithm developed is very reliable and efficient. Kim and Yano (1992) developed an iterative algorithm to solve the part type selection, machine grouping and loading problems with an overall objective of maximizing system throughput. A few researchers have done extensive survey on the tool management issues of automated manufacturing systems and emphasize that the tooling considerations should be included in evaluating the performance of FMS systems (Veeramani et al 1992). In general, the objective of tool scheduling is to coordinate the flow of parts and tools to minimize unnecessary delays. Without an effective tool scheduling technique, the system may not operate as planned in the planning and part scheduling stages (Roh and Kim 1997). It is shown that the added cost of tool automation can bring great improvements in economic efficiency (Kahator and Leung 1994).

Spano et al (1993) reviewed the work on the design of FMS in the areas of facilities design, material handling systems design, control systems
design and scheduling. Nascimento (1993) showed with CPU time and solution quality, that the Giffler and Thompson algorithm which was originally designed for the traditional job shop model can still provide good results for an FMS Model. Toker et al (1994) proposed an approximation algorithm for the ‘n’ jobs, ‘m’ machines resource constrained job shop problem and introduced two lower bounds to adjust the schedules. Although, the evolution of FMS/CIM has complicated the issue, because of its complex nature of working; it will be best suited to produce variety of parts with flexibility of even small lots (Jawahar et al 1998b). Jain and Elmaraghy (1997) proposed an algorithm, which revises only those operations that must be rescheduled and is used in conjunction with the existing scheduling methods to improve the efficiency of flexible manufacturing systems.


McPherson and White (1998) studied a new class of cyclic scheduling problem, the periodic flow line, characterized by a perpetual stream of jobs arriving periodically in finite batches. The majority of research conducted on the scheduling problem of FMS generally assumes a part movement policy. Researchers using this part movement approach have focused on a number of issues that are specifically related to the flow of work pieces (Rahimifard and Newman 1997). The part movement approach can be justified in many cases, where the cost of parts is very high in comparison
with the cost of cutting tools required. However in many batch machining applications, the cost of cutting tools is a significant proportion of the total production cost, sometimes as much as 25–30% (Akturk and Onen 1999). As the budget for tool purchases is limited, it is necessary to determine a tool copy configuration that can give the best system performance for a given limited tool budget (Jun et al. 1999). Therefore, by adopting appropriate tooling strategies and economizing on the tooling cost, large reductions in the total cost is feasible (Gray 1993).

Tiwari and Vidyarthi (2000) proposed a GA based heuristic to solve machine loading problems in a FMS. In this work, assumption was made to have individual tool box for each machining centres. Kumar and Shanker (2000) proposed a genetic algorithm for FMS part type selection and machine loading problems with a variety of objectives. Vidyarthi and Tiwari (2001) formulated a fuzzy-based approach to address the machine-loading problem in a FMS. The objectives are minimization of system unbalance and maximization of throughput, whereas the systems technological constraints are posed by availability of machining time and tool slots. Saravana Sankar et al. (2003) developed a scheduling procedure for a specific FMS to maintain its flexibility and thereby to maintain the intended performance measures using genetic algorithm.

Watson et al. (2003) developed a model of problem difficulty for tabu search in the job shop scheduling problem, borrowing from similar models developed for SAT and other NP-complete problems. Kumar et al. (2003) presented an effective method of scheduling jobs in flexible manufacturing system through the use of an ant colony optimization technique. Chan (2004) studied the effects of different levels of operation flexibility and various dispatching rules on the performance of a flexible manufacturing system. The system performance considered is mean flow

Prabaharan et al (2006) used two heuristics PDRA and SA algorithm for generating optimal schedules for a specific manufacturing environment. Saravana Sankar et al (2006) designed a GA based multi objective evolutionary algorithm equipped with a mechanism to parallely generate diverse optimal solutions for scheduling the optimization of a FMS. Biswas and Mahapatra (2007) applied a swarm optimization approach for machine loading problem in FMS to avoid premature convergence with the objective of minimization of system unbalance. Saravanan and Noorul Haq (2007) applied a scatter search technique for scheduling optimization of flexible manufacturing system by minimizing the idle time of the machine and also minimizing the total penalty cost for not meeting the due date concurrently. The application of metaheuristics to solve job and tool scheduling problem in FMS needs attention.

2.2.2 Routing and Dispatching of AGV

Egbelu and Tanchoco (1984) presented the heuristic rules for dispatching automatic guided vehicles in a job shop environment. Mathematical programming has been used to formulate and solve some of the design, planning and scheduling problems. The traditional flow path problem for AGVs has been modeled as mathematical programming problems by Gaskins and Tanchoco (1987), Kaspi and Tanchoco (1990), Sinreich and Tanchoco (1993) and Seo and Egbelu (1995) among others. The assumptions made are (a) the layout of the machines is known, (b) travel is unidirectional
and (c) the p/d points are known. Mufit Ozden (1988) investigated the effect of several key factors related to the automated guided vehicles on the overall performance of a flexible manufacturing system through a simulation programme. Mahadevan and Narendran (1990) addressed the key issues involved in the design and operation of AGV-based material handling systems for an FMS. The problems arising from multi-vehicle systems are analyzed, and strategies for resolving them are examined using analytical and simulation models.

Kim and Tanchoco (1991) presented a conflict-free and shortest time algorithm for routing AGVs in a bidirectional path network based on Dijkstra’s shortest path algorithm. Bozer and Srinivasan (1991) developed an analytical model of a single vehicle operating in a closed loop that represents the individual loop in a tandem AGV system and analyzed the throughput capacity. Simchi-Levi and Berman (1991) discussed a sequencing problem to minimize flow time in a network. Algorithms were developed to control a single AGV in a manufacturing system. Usher et al (1988) and Kouvelis et al (1992) developed heuristics to solve the traditional unidirectional flow path design problem.

Sabuncuoglu and Hommertzheim (1992b) attempted to investigate the performances of machine and AGV scheduling rules against the mean flow-time criterion. Tanchoco and Sinreich (1992) tackled the unidirectional single loop AGV flow path problem by using mixture of integer programming techniques and rules to develop solution procedures. The single loop flow path problem involves identifying a single loop that passes through all the p/d points. Control of a single loop system is relatively simple. Yim and Linn (1993) addressed a petri-net-based simulation which was used to investigate the effect of different dispatching rules on the FMS performance. Mahadevan and Narendran (1993) described the development of an analytical model for
estimation of the number of AGVs. The ability of model to estimate the vehicle requirements under various conditions have been tested. Ulsoy et al (1993) attempted to make scheduling of AGVs an integral part of the overall scheduling activity in an FMS environment.

Krishnamurthy et al (1993) presented a mixed 0-1 integer programming model to formulate AGV routing and dispatching problem. The objective of the model is to minimize the total travel time in a network with bi-directional paths. Lin et al (1994a) studied the load routing problem (LRP) and proposed a two-phase approach to solve it. Phase I consists of an analytical approach based on the steady state. The output of phase I provide an adequate estimate and recommendation of routing decisions of an operational system. Phase 2 is a simulation model which accepts the results from phase 1 to realize the impact of another operational performance such as the queue length of each station. Lin et al (1994b) proposed a task-list time-window algorithm to find a shortest travel time for routing control of a tandem automated guided vehicle system. Ganesharajah and Sriskandarajah (1995) studied the operational issues of scheduling, dispatching and routing of AGVs in various flow path layouts.

Bilge and Ulusoy (1995) proposed a time window approach to the simultaneous scheduling of machines and material handling system in an FMS. Taghaboni and Tacnhoco (1995) developed an intelligent real-time controller for free ranging AGVs. Banerjee and Zhou (1995) used a genetic algorithm based approach to solve the facility layout design with single loop material flow path configuration. Chen (1996) discussed the problems of the planning and control of automated guided vehicles in manufacturing systems. A mixed integer programming model is developed to minimize the total material handling cost in manufacturing systems. Langevin et al (1996) proposed a dynamic programming based method to solve exactly instances
with two vehicles and solved the combined problem of dispatching and conflict free routing. Akturk and Yilmaz (1996) proposed an algorithm to schedule vehicles and jobs in a decision-making hierarchy based on mixed integer programming; however, this approach is applicable only for problems with a small number of jobs and vehicles.

Ulusoy and Bilge (1997) addressed the problem of simultaneous scheduling of machines and a number of identical AGVs in a FMS so as to minimize the makespan. Computational results show that the GA developed is an effective solution method for this problem. Kim et al (1997) discussed how to estimate the mean response time, the utilization and the cycle time of AGV for a delivery order. A procedure is suggested in order to determine the home location of idle vehicles in a way of minimizing the mean response time for an arbitrary delivery order. Jawahar et al (1998a) attempted to link the operation of automated guided vehicles with the production schedule and suggested a heuristic algorithm that employs vehicle dispatching rules (vdrs) for conflict resolution. Oboth et al (1999) presented a heuristic method to solve the dispatching and routing problems but not simultaneously. Scheduling is performed first and a sequential path generation heuristic (SPG) is used to generate conflict free routes.

Seo and Egbelu (1999) developed a manufacturing planning methodology for an AGV-based automated manufacturing system by simultaneously dealing with material processing and transportation functions. Soylu et al (2000) investigated the automated guided vehicle routing problem. The objective is to find the shortest tour for a single, free-ranging AGV that has to carry out multiple pick and deliver requests. An artificial neural network algorithm based on Kohonen's self-organizing feature maps is developed to solve the problem. Gademann and Van de Velde (2000) addressed the problem of determining the home positions for ‘m’ automated
guided vehicles (AGVs) in a loop layout. Qiu et al (2002) described the emergence of the problem of AGV scheduling and routing and also pointed out the fertile areas for future study of AGV.

Gaur et al (2003) studied a particular type of vehicle routing problem that arises in the context of scheduling an automatic guided vehicle in the flexible manufacturing paradigm. Levitin and Abezgaouz (2003) considered the case when loads are placed in flat pallets and each new picked up pallet is loaded on the top of batch of pallets already carried by the AGV. To avoid use of excessive space and time needed to reorder pallets in the batch, the loading–unloading procedures should be performed in accordance with last-in-first-out (LIFO) rule. Rajagopalan et al (2004) tackled the problem of simultaneously locating the pick-up/drop-off (p/d) points along the periphery of a cell and determining the flow path for an AGV based material handling system. A mathematical model to determine the p/d points as well as flow path for a bi-directional AGV is presented. Singh and Tiwari (2004) presented a framework for an AGV dispatching system based on an object oriented approach using the unified modeling language (UML), and the development of a dispatching algorithm to facilitate a human controller to dispatch efficiently a fleet of AGVs in response to calls from any shop floor or machine operator.

Shalaby et al (2006) proposed a partitioning algorithm for designing tandem AGV systems. The proposed algorithm serves a number of objectives; minimizing the total handling cost, minimizing the maximum workload in the system and minimizing the number of between-zones trips. Lee and Srisavat (2006) investigated the interaction between manufacturing system constructs and the operation strategies in a multiple-load automated guided vehicle system when AGVs in a system can carry two or more loads. Correa et al (2007) proposed a hybrid method designed to solve a problem of
dispatching and conflict free routing of automated guided vehicles in a flexible manufacturing system. The application of metaheuristics for routing and dispatching of AGV in FMS needs attention.

### 2.2.3 Task Scheduling of AGV

Chen and Talavage (1982) proposed A* search algorithm with minimax criterion and heuristic rules to solve the optimal task scheduling for FMS. Stecke and Morin (1985) used mathematical programming and established the optimality of balanced workload for certain types of FMS’s. Berrada and Stecke (1986) have proposed efficient branch and bound procedure for solving the loading problem with objective of workload balancing. Cleveland and Smith (1989) investigated the use of GA’s in scheduling a multi-stage flow line with non-standard characteristics. Juliff (1993) has developed an interesting multi chromosome GA for multi-dimensional scheduling problems. Key dimensions of the solution are represented on different chromosomes and found this method superior to a single chromosome representation. Prakash and Chen (1995) presented the results of a simulation study of a flexible manufacturing system. The FMS considered for the present study includes six machining centers capable of performing a variety of tasks, an automated guided vehicle based material handling system and a single input, single output storage-retrieval system connected to the manufacturing system by conveyors. Time spent for parts in the system is considered as one of the main criteria of the system performance. Schultz and Mertens (1997) compared the GA with an expert system approach and priority rules and indicated that GA generally produces satisfactory schedules. The performance of GA depends on the run time i.e., population size and number of generations.

Kim and Hwang (1999) developed a heuristic procedure for the control of materials flows and automated guided vehicles in manufacturing
job shop system. The procedure is based on the idea of workload balancing; it tries to balance the workload among machines as well as the workload between the machines and the transporters. Tuan Le-Anh and René de Koster (2004) addressed most key related issues including guide-path design, estimating the number of vehicles, vehicle scheduling, idle-vehicle positioning, battery management, vehicle routing, and conflict resolution. The application of metaheuristics for task scheduling of AGV in FMS needs attention.

2.2.4 Single Machine Scheduling in FMM

The single machine total weighted tardiness problem has been tackled by enumerative algorithms: branch and bound algorithm Shwimer (1972), Picard and Queyranne (1978), Potts and Wassenhove (1985) and dynamic programming algorithms Schrage and Baker (1978) to generate exact solutions that are guaranteed to be optimal. But the branch and bound algorithms are limited by computational times and the dynamic programming algorithms are limited by computer storage requirements, especially when the number of jobs is more than about fifty. Thereafter, the problem has been extensively studied by heuristics. Kusiak and Ahn (1992) provided a list of non-due date related dispatching rules and due date related dispatching rules and introduced Most Dissimilar Resources (MDR) rule. The MDR rule is found suitable for maximization of resource utilization.

Kondakci et al (1994) have presented a new branch and bound algorithm for minimizing the total tardiness in single machine scheduling problems. Kim and Yano (1994) have contributed in terms of developing heuristics for single machine scheduling problems to minimize the total tardiness. Sarper (1994) examined two criteria against four dispatching rules under three system utilization levels and five due date assignment levels.
James and Buchanan (1997) developed a new method of finding good quality solutions to this scheduling problem by using the concept of a compressed solution space, based on a binary representation of the early/tardy scheduling problem, and tabu search. Crauwels et al (1998) compared the performance of a number of local search heuristics that have the binary representation such as descent methods, simulated annealing, threshold accepting, tabu search and GA for total weighted tardiness problems with forty jobs, fifty jobs, and one hundred jobs. Binary encoded GA performs very well and requires comparatively little computation time. Binary encoded GA is also a viable alternative to other heuristic methods, especially in view of its small maximum relative deviations and modest computation time.

Brucker et al (1999) developed a branch and bound algorithm for solving single machine scheduling problem with positive and negative time lags between jobs. Mazzini and Armentano (2001) have developed a heuristic for minimizing total earliness and tardiness cost in a single machine scheduling problem with distinct ready times and due dates. Mondal and Sen (2001) suggest an algorithm to solve the single machine weighted earliness/tardiness penalty problem with a common due date. This algorithm uses a graph search space. Wan et al (2002) proposed an approach to combine a tabu search procedure and an optimal timing algorithm for solving the problem with distinct due windows. Shin et al (2002) presented a tabu search algorithm that schedules ‘n’ jobs to a single machine in order to minimize the maximum lateness of the jobs.

Maheswaran and Ponnambalam (2003) discussed with an investigation on the total weighted tardiness of the single machine scheduling problems. The problems are solved by a heuristic procedure namely backward and forward heuristics. Ventura and Radhakrishnan (2003) use a lagrangian relaxation procedure that utilizes the subgradient algorithm to tackle the problem. Feldmann and Biskup (2003) address the restrictive common due


Cheng et al (2008) developed a model for single machine scheduling of multi operation jobs without missing operations to minimize the total completion time and the objective is to minimize the total completion time. Biskup and Herrmann (2008) developed a model for single machine scheduling against due dates with past sequence-dependent setup times and
the objective is to minimize the due date. Sidhoum et al (2008) developed optimization models for lower bounds for the earliness/tardiness scheduling problem on parallel machines with distinct due dates, considering the parallel machine scheduling problem in which the jobs have distinct due dates with earliness and tardiness. Ji and Cheng (2008) considered the parallel machine scheduling by deteriorating jobs with an objective function of minimizing the total completion time.


2.2.5 Production and MHS Scheduling

Kimemia and Gershwin (1985) reported on an optimization problem that optimizes the routing of the parts in an FMS with the objective of maximizing the flow while keeping the average in-process inventory below a fixed level. The machines in the cell have different processing times for an
operation. Jaikumar and Solomon (1990) studied a manufacturing system, integrated with a central warehouse by means of AGVs. The jobs are returned to the central warehouse after each manufacturing operation. The job and AGV scheduling problem in a two-level hierarchical optimization is considered. Sabuncuoglu and Hommertzheim (1992a) proposed a dispatching algorithm that considers the availability of both machine and AGV to select a job for loading. The performance of the proposed methodology is compared with a few simple dispatching rules by using the mean flow time and mean tardiness criteria through a simulation test.

Karabtlik and Sabuncuoglu (1993) introduced a beam search based algorithm for the simultaneous scheduling of machines and AGVs. The assumptions made are vehicles always return to the load / unload station after transferring a load. Lee and DiCesare (1994) formulated the integrated production and material handling scheduling in an off-line job shop context. Two shortest path routes (in opposite directions) exist between every pair of machines. Routes are constraining resources. Sawik (1996) presented a multi-level decision model for simultaneous scheduling of machine and vehicle in an FMS. Smith et al (1999) considered not only material handling activities but also explicitly loading and unloading activities for off-line scheduling. Kim et al (1999) proposed a new dispatching algorithm for an efficient operation of AGVS. It utilizes the information of work-in-process in buffers and travel times of AGVs. The performance of the algorithm is compared with some well-known dispatching rules in terms of the system throughput through simulation. Lee (1999) presented a contribution where the material handling activities that influence the schedule take place between the load/unload station and the machine. Dispatching strategies are proposed and evaluated for rail guided vehicles in a loading / unloading zone of an FMS.

Noorul Haq et al (2003) discussed the multi level scheduling decisions of a FMS to generate realistic schedules for the efficient operation
of the FMS. Abdelmaguid et al (2004) has presented a new hybrid genetic algorithm for the simultaneous scheduling problem for the makespan minimization objective. The hybrid GA is composed of GA and a heuristic. The GA is used to address the first part of the problem that is theoretically similar to the job shop scheduling problem and the vehicle assignment is handled by a heuristic called vehicle assignment algorithm (VAA). Jerald et al (2006) proposed the two approaches such as genetic algorithm and adaptive genetic algorithm used for scheduling both parts and AGVs simultaneously in an FMS environment. Vis (2006) described the research perspectives in the design and control of AGV systems in distribution, transshipment and transportation systems.

Reddy and Rao (2006) addressed the simultaneous scheduling problem as a multi objective problem in scheduling with conflicting objectives and solved by non-dominated sorting evolutionary algorithms. Subbaiah et al (2009) addressed the problem of simultaneous scheduling of machines and two identical automated guided vehicles in a FMS so as to minimize makespan and mean tardiness using sheep flock heredity algorithm. The application of metaheuristics for production and material handling system scheduling problem in FMS needs attention.

In traditional optimization techniques, the following limitations are observed:

- Traditional optimization techniques start with a single point.
- The convergence to an optimal solution depends on the chosen initial random solution.
- The results tend to stick with local optima.
- These are not efficient when practical search space is too large.
These techniques follow a deterministic rule.

These techniques are not efficient in handling the multi-objective functions.

Most traditional techniques that give optimal solutions apply to problems of very small size.

These techniques require excessive computation time and are not practical for use on a daily basis.

Also, modeling the techniques is a difficult task.

These limitations urge the researchers to implement metaheuristic techniques in application domains. Scheduling problems are proved to be NP-hard types of problems and are not easily or exactly solved for large sizes. The application of metaheuristics technique to solve such NP hard problems needs attention.

2.3 METAHEURISTICS

Metaheuristics techniques comprise a variety of methods including optimization paradigms that are based on evolutionary mechanisms such as biological genetics and natural selections. A metaheuristic is described as an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high quality solutions. It may manipulate a complete or incomplete single or a collection of solutions for every iteration. The subordinate heuristics may be high or low level procedures, or a simple local search or just a construction method. The metaheuristics that are not designed specifically for a particular problem but are considered general approaches that can be tuned for any problem.

While these methods provide many characteristics that make it the method of choice for the researchers in problem domain, the most important
reasons are, the paradigms use direct ‘fitness’ information instead of functional derivatives and use probabilistic, rather than deterministic transition rules. This overcomes the problem of getting stuck in local optima prevalent with deterministic rules. There are several possible classifications of metaheuristics but one is commonly used in single solution approaches and population based approaches. Single solution methods are Basic local search, Tabu search, Simulated annealing, Variable neighbourhood search and others. Population based methods include Genetic algorithm, Particle swarm optimization, Ant colony optimization, Scatter search, Memetic algorithm etc.

2.3.1 Genetic Algorithm

David Goldberg (1989) defined GA as that Genetic algorithms are the search algorithms based on the mechanics of natural selection and natural genetics. GA combines the survival of fitness among the string structure with a structured, yet randomized information exchanges to form a search algorithm with some of the innovative flair of human search.

Michalewicz (1996) defined GA as that, GA’s are stochastic algorithms whose search methods model some natural phenomena: Genetic inheritance and Darwinian strive for survival. Each cell of every organism of a given species carries a certain number of chromosomes. Chromosomes are made of units of genes, which are arranged, in a linear succession. Every gene controls the inheritance of one or several characters. Genes of certain characters are located at certain places of the chromosome, which are called string positions. Each chromosome would represent a potential solution to a problem. An evaluation process run on a population of chromosomes corresponds to a search through a space of potential solutions. Such a search requires balancing two objectives.
i) Exploiting the best solution and

ii) Exploring the search space

The steps of genetic algorithm are described below and the flowchart is shown in Figure 2.1.

Step 1 : Generate random population of n chromosomes.

Step 2 : Evaluate the fitness $f(x)$ of each chromosome $x$ in the population.

Step 3 : Create a new population by repeating following steps until the new population is complete.

Step 4 : Select two parent chromosomes from a population according to the fitness.

Step 5 : With a crossover probability, crossover the parents to form a new offspring (Children). If no crossover was performed, offspring is an exact copy of parents.

Step 6 : With a mutation probability, mutate new offspring at each position in chromosome.

Step 7 : Place new offspring in a new population.

Step 8 : Use new generated population for a further run of algorithm.

Step 9 : If the end condition is satisfied, stop and returns the best solution in current population and goto step 2.
Figure 2.1 Flowchart for genetic algorithm

Sridhar and Rajendran (1994) addressed a GA for part family grouping and scheduling the parts within the part families in a flow line based manufacturing cell. The objectives have been considered are makespan and total flow time. Chen et al (1995) proposed a GA based heuristic for flow shop problems with makespan as the criterion and compared the efficiency of the proposed GA heuristic with the other heuristics. Stockton and Quinn (1995) proposed a GA for an aggregate production scheduling problem.
Weindahl and Garlichs (1994) proposed a GA, for the decentral production scheduling of assembly system. The advantages of the GA compared to classical optimization technique include shorter computing times and high quality or goal attainment in the solution found. Sridhar and Rajendran (1996) evaluated the proposed GA by comparing it with a HC heuristic to the problem of scheduling in flow shops and cellular manufacturing system with multiple objectives and indicated that the proposed GA yielded much better quality solution than HC heuristic. It is inferred that the GA is an adaptive search method that can be used to tackle hard combinatorial optimization problem.

2.3.2 Simulated Annealing Algorithm

Kalyanmoy Deb (2002) defined SA algorithm as the method that resembles the cooling process of molten metals through annealing. At high temperatures, the atoms in the molten metal can move freely with respect to each another, but as the temperature is reduced, the movement of atoms gets restricted. The atoms start to get ordered and finally form crystals having the minimum possible energy. However, the formation of the crystal depends on the cooling rate. If the temperature is reduced at a faster rate, the crystalline state may not be achieved at all; instead the system may end up in a polycrystalline state, which may have a higher energy state than the crystalline state. Therefore, to achieve the absolute minimum state, the temperature needs to be reduced at a slow rate. The process of slow cooling is known as annealing in metallurgical parlance. SA simulates this process of slow cooling of molten metal to achieve the minimum function value in a minimization problem. The cooling phenomenon is simulated by controlling a temperature like parameter introduced with the concept of the Boltzmann probability distribution. According to the Boltzmann probability distribution, a system in thermal equilibrium at a temperature $T$ has its energy distributed probabilistically according to the equation (2.1).
\[ P(E) = \exp(-\frac{E}{kT}) \]  

(2.1)

where \( k \) is the Boltzmann constant. This expression suggests that a system at a high temperature has almost uniform probability of being at any energy state, but at a low temperature it has a small probability of being at a high energy state. Therefore by controlling the temperature \( T \) and assuming that the search process follows the Boltzmann probability distribution, the convergence of an algorithm are controlled using the metropolis algorithm.

At any instant, the current point is \( x^{(t)} \) and the function value at that point is \( E(t) = f(x^{(t)}) \). Using the metropolis algorithm, the probability of the next point being at \( x^{(t+1)} \) depends on the difference in the function values at these two points or on \( \Delta E = E(t+1) - E(t) \) and is calculated using the Boltzmann probability distribution equation (2.2).

\[ P(E(t+1)) = \min \{1, \exp(-\frac{\Delta E}{kT}) \} \]  

(2.2)

If \( \Delta E \leq 0 \), this probability is one and the point \( x^{(t+1)} \) is always accepted. In the function minimization context, this makes sense because if the function value at \( x^{(t+1)} \) is better than that at \( x^{(t)} \), the point \( x^{(t+1)} \) must be accepted. When \( \Delta E > 0 \), which implies that the function value at \( x^{(t+1)} \) is worse than that at \( x^{(t)} \). According to the metropolis algorithm, there is some finite probability of selecting the point \( x^{(t+1)} \) even though it is a worse than the point \( x^{(t)} \). However this probability is not the same in all situations. The probability depends on relative magnitude of \( \Delta E \) and \( T \) values. If the parameter \( T \) is large, this probability is more or less high for points with largely disparate function values. Thus, any point is almost acceptable for a large value of \( T \). For small values of \( T \), the points with only small deviation in function value are accepted.
The steps of SA algorithm are described below and the flowchart is shown in Figure 2.2.

**Figure 2.2 Flowchart for SA algorithm**

Step 1: Choose an initial point $x^{(0)}$, a termination criterion $\varepsilon$. Set $T$ as a sufficiently high value, number of iterations to be performed at a particular temperature $n$ and set $t=0$.

Step 2: Calculate a neighbouring point $x^{(t+1)} = N(x^{(t)})$. Usually, a random point in the neighbourhood is created.
Step 3: If $\Delta E = E(x^{(t+1)}) - E(x^{(t)}) < 0$, set $t = t + 1$; Else create a random number ($r$) in the range (0,1). If $r \leq \exp(-\Delta E/kT)$, set $t = t + 1$; Else goto step 2.

Step 4: If $(x^{(t+1)} - x^{(t)}) < \varepsilon$ and $T$ is small, Terminate. Else goto Step 2.


### 2.3.3 Ant Colony Optimization Algorithm

The ant system is a new kind of co-operative search algorithm inspired by the behavior of colonies of real ants. The blind ants are able to find astonishing good solutions to shortest path problems between food sources and home colony. The medium used to communicate information among individuals regarding paths, and decide where to go, was the pheromone trials. A moving ant lays some pheromone on the moving path, thus marking the path by the substance. While an isolated ant moves essentially at random, it can encounter a previously laid trail and decide with high probability to follow it, and also reinforcing the trail with its own pheromone. The collective behavior that emerges in a form of autocatalytic behavior where the more the ants following a trail, the more attractive that trail becomes for being followed. There is a path along which ants are walking from nest to the food source and vice versa. If a sudden obstacle appears and the path is cut off, the choice is influenced by the intensity of the pheromone trails left by proceeding ants. On the shorter path more pheromone is laid down.
Ant colony optimization algorithm can be applied for the continuous function optimization algorithm. Hence, the domain has to be divided into a specific number of “R” randomly distributed regions. These regions are indeed the trial solutions and act as local stations for the ants to move and explore. The fitness of these regions are first evaluated and sorted on the basis of fitness. Totally a population of ants explores these regions; the updating of the regions is done locally and globally with the local search and global search mechanism respectively.

The steps of an ACO algorithm are described below and the flowchart is shown in Figure 2.3.

Figure 2.3 Flowchart for an ACO algorithm
Step 1 : Fix the evaporation rate and number of runs

Step 2 : While (number of runs is less than required)

Step 3 : Initialize pheromone values.

Step 4 : Call random number generation function

Step 5 : Generate group of ants with different paths.

Step 6 : Call the function for calculating the objective function

Step 7 : Sort the objective function values in ascending order

Step 8 : For best sequences, update pheromone level

Step 9 : Repeat steps 4,5,6,7 and 8 till obtaining required number of runs

Step 10 : Print the best sequences and the objective function value.

Step 11 : Change evaporation rate and number of runs for next trial.


2.3.4 Particle Swarm Optimization Algorithm

A swarm of individuals exploring a large solution space can benefit from sharing the experiences gained during the search with the other individuals in the population. This social behavior has inspired the development of Particle Swarm Optimization. PSO is an evolutionary computation technique developed by Kennedy and Eberhart in 1995. Similar to Genetic algorithm (GA), PSO is a population based optimization tool, has fitness values to evaluate the population, and update the population for the optimum with random techniques. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, particles update themselves with the initial velocity. In this algorithm, the individuals are not selected to survive or die in each generation. Instead all the individuals learn from the others and adapt themselves by trying to imitate the behavior of the fittest individuals. The PSO methods adhere to the basic principles of swarm intelligence like proximity, quality, diverse response, stability and adaptability.

The steps of PSO algorithm are described below and the flowchart is shown in Figure 2.4.

Step 1 : Initialize a population of ‘n’ particles randomly
Step 2: Calculate fitness value for each particle. If the fitness value is better than the best fitness value (pbest) in history, then set current value as the new pbest.

Step 3: Choose particle with the best fitness value of all the particles at the gbest.

Step 4: For each particle, calculate particle velocity according to the equation (2.3) and equation (2.4).

\[
v[] = v[] + c_1 \times \text{rand()} \times (pbest[] - \text{present[]}) + c_2 \times \text{rand()} \times (gbest[] - \text{present[]})
\]  \hspace{1cm} (2.3)

\[
\text{present[]} = \text{present[]} + v[
\]  \hspace{1cm} (2.4)

where \(v[]\) is the particle velocity

\(\text{present[]}\) is the current particle (solution)

\(pbest[]\) and \(gbest[]\) are defined as stated before

\(\text{rand()}\) is a random number between 0 and 1.

\(c_1, c_2\) are learning factors. Usually \(c_1\) equals to \(c_2\) and ranges from 0 to 4.

Step 5: Particle velocities on each dimension are clamped to a maximum velocity \(V_{\text{max}}\). If the sum of acceleration would cause the velocity on that dimension to exceed \(V_{\text{max}}\) (specified by the user), the velocity on the dimension is limited to \(V_{\text{max}}\).
Figure 2.4 Flowchart for PSO algorithm
Sha and Hsu (2006) modified the particle position representation, particle movement, and particle velocity to better suit PSO for the job shop scheduling problem and also applied tabu search to improve the solution quality. Xia and Wu (2006) proposed a hybrid PSO algorithm for the problem of finding the minimum makespan in the job-shop scheduling environment. Ponnambalam and Kiat (2008) proposed a PSO algorithm to solve machine loading problem in a flexible manufacturing system with bicriterion objectives of minimizing system unbalance and maximizing system throughput in the occurrence of technological constraints such as available machining time and tool slots. Kashan and Karimi (2009) proposed a discrete PSO algorithm to tackle the problem of optimal assignment of jobs to machine to minimize the makespan time. Fauadi and Murata (2010) addressed binary particle swarm optimization algorithm to optimize simultaneous machines and AGVs scheduling process with makespan minimization function.

2.3.5 Research Gaps identified

- The majority of research conducted on the scheduling problem of FMS generally assumes a part movement policy.

- In most of the research works, part and tool flow as separate issues and often, the effects of one of the pair on the other is neglected.

- In AGV scheduling, less attention has been paid to considering the objective of minimization of the number of back-tracking movements.

- The application of metaheuristics for single machine scheduling problem in FMM needs attention.

- Little attention is given to integrate production and MHS scheduling.
• The traditional techniques are not efficient in handling multiple objectives. Also, they are not efficient when practical search space is too large.

• The performance of the previously implemented heuristics is suitable as long as the operating characteristics and objectives of the system remain the same.

2.4 SUMMARY

The issues in FMS scheduling problem have been identified. The use of metaheuristics to solve scheduling problem of FMS needs attention. The literature review is carried out on FMS scheduling problem and optimization techniques. Literature related to job and tool scheduling; routing, dispatching and task scheduling of AGV; single machine scheduling; production and material handling system scheduling have been reviewed. The application of metaheuristics for scheduling of job and tool in FMS needs attention. The methodologies for routing, dispatching and task scheduling of AGV in FMS need attention. The application of metaheuristics to minimize single machine total weighted tardiness problems in FMM need attention. Metaheuristic techniques for production and MHS scheduling with the objective of minimization of the number of back tracking movement need attention. The principle behind various metaheuristic approaches such as genetic algorithm, simulated annealing algorithm, ant colony optimization algorithm and particle swarm optimization algorithm are investigated.