Conference papers   School of Marketing and Entrepreneurship

2001-7

Job Shop Scheduling Problem: an Overview

Amr Arisha
Technological University Dublin, amr.arisha@tudublin.ie

Paul Young
Dublin City University

Mohie El Baradie
Dublin City University

Recommended Citation

JOB SHOP SCHEDULING PROBLEM: AN OVERVIEW

A. Arisha, P. Young, and M. A. ELBaradie

School of Mechanical & Manufacturing Engineering, Dublin City University
Dublin 9, Ireland

ABSTRACT:

The Job-shop scheduling is one of the most important industrial activities, especially in manufacturing planning. The problem complexity has increased along with the increase in the complexity of operations and product-mix. To solve this problem, numerous approaches have been developed incorporating discrete event simulation methodology. The scope and the purpose of this paper is to present a survey which covers most of the solving techniques of Job Shop Scheduling (JSS) problem. A classification of these techniques has been proposed: Traditional Techniques and Advanced Techniques. The traditional techniques to solve JSS could not fully satisfy the global competition and rapidly changing in customer requirements. Simulation and Artificial Intelligence (AI) have proven to be excellent strategic tool for scheduling problems in general and JSS in particular. The paper defined some AI techniques used by manufacturing systems. Finally, the future trends are proposed briefly.

KEYWORDS: Job Shop Scheduling, Simulation, Artificial Intelligence, Intelligent Scheduling.

1. Introduction

Over the last decade research into scheduling, particularly in its most common industrial form of job shop scheduling, has risen in importance due to the demands of industry. While much progress has been made on an academic front, doubts remain over the transfer of the technology to fit the flexibility requirements of modern production facilities. Job shop scheduling has received much attraction in the literature as the solution is made more difficult through the requirement to satisfy the conflicting demands of both batch and continuous production.

In past research, a fair number of researchers had worked on machine scheduling problem in the hope of finding optimal solution or even near optimal for complex problems. A considerable number of analytical techniques such as linear programming and "Branch and Bound" or heuristic approaches like priority rules and neighborhood methods were investigated.

In recent times, most of the studies have turned to deal with new solving techniques such as simulation and artificial intelligence techniques. These methods have proved a significant step forward in solving JSS problem with less computational effort and more powerful results. In view of this, a brief survey of these new techniques is presented on work previously carried out to investigate and solve JSS problem. Finally, future trends towards the provision of an effective JSS tool will be discussed.

2. Problem Definition

Consider a shop floor where jobs are processed by machines. Each job consists of a certain number of operations. Each operation has to be performed by a dedicated machine and requires a predefined processing time. The operation sequence is prescribed for each job in a production

recipe, imposing static constraints on scheduling. Thus, each job has its own machine order and no relation exists between the machine orders of any of two jobs. [1]

In the JSS problem a set 'J' of 'n' jobs $J_1, J_2, J_3, \ldots J_n$ have to be processed on a set 'M' of 'm' different machines $M_1, M_2, M_3, \ldots M_m$. Job J_j consists of a sequence of m_j operations $O_{j1}, O_{j2}, O_{j3},\ldots, O_{jmj}$, which have to be scheduled in this order.

Moreover, each operation can be processed only by one machine among the 'm' available ones. Operation 'O_{jk}' has a processing time 'P_{jk}'. The objective is to find an operating sequence for each machine such as to minimize a particular function of the job completion times, and in such a way that two operations are never processed on the same machine at anytime instant. [2]

An exhaustive survey on the different scheduling problems and their algorithms and complexities was presented by [3].

3. Nature of the problem

During the last decades, a tremendous number of researches include a comprehensive study of scheduling problem have been developed to find a solution to JSS problem.

The beginning of scheduling problem came just in the mid-fifties in the firm of a paper presented by Johnson (1954) [4]. In the following years, several studies have been developed to discuss JSS solution, [5, 6, 7, 8, 9, 10].

The general JSS problem is NP-hard in the strong sense (Lenstra *et. al* 1977, Gonzalez *et. al* 1978, Garey *et. al* 1979, Rinnoy *et. al* 1979, Sotskov 1991) and is probably one of the most computationally intractable combinatorial problems considered so far.

A practical proof of this intractability comes from the fact that a small example with 10 jobs and 10 machines posed by Fischer and Thompson (1963) remained open for

Figure 1: An Analysis of a scheduling problem – schematic view

over 15 years. It was solved by Carlier and Pinson (1980) as the culmination of a considerable amount of research. [2]

An approach to minimize the expected completion time for n jobs on 2 machines when the processing time for all jobs are derived from exponential distribution is discussed by [11].

The basic JSS problem is an optimization problem. It could be classified based upon the information about the production program as follow: static or dynamic, and deterministic or stochastic. [12]

Feasible schedules are obtained by permuting the processing order of operations on the machines (operations sequence) without violating the technological constraints. We have a maximum of $(n!)^m$ different solutions to a problem instance.[13] The explosive exponential growth in the number of alternative schedules with the size of problem is central to the difficulty of identifying one of these as the solution of the JSS problem.
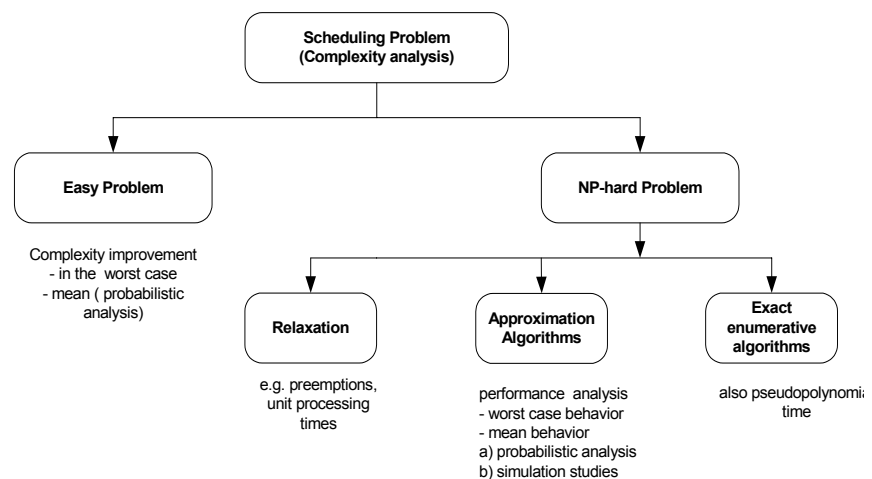
The difficulty is two folds: First, there is the problem of deciding … what characteristics should be specified for the best schedule? and second, how can such a schedule be efficiently determined?

4. Solving Techniques for the Scheduling Problem

Over the last 20 years or so, several techniques have been developed by researchers to deal with the scheduling problem. These techniques can be grouped under the following headings: Traditional Techniques and Advanced Techniques.

4.1. Traditional Techniques

Traditional Techniques can be classified under two main categories, i.e. Analytical Techniques and Heuristic Techniques. Figure 2.

The general approach of the analytical methods is to consider the problem in its total system form of scheduling 'n' jobs on 'm' machines. The relative lack of success of this approach in providing a general optimization method of wide applicability, has led to a switch in the focus of attention from the total system to a more simple decomposed subsystem view of the problem; in which the job shop is considered to be a series of interrelated single machine scheduling problems [1].

Attempts to bridge the gap between heuristic approaches and optimization approach have also been undertaken (e.g. Fisher (1986), Adams (1998) and Luh (1990)). In Adams, the solution is provided by ' local optimization'. However, schedule evaluation could only be achieved through "selective enumeration". Recently, The Lagrangian relaxation technique has been used by Fisher (1986) to obtain a more efficient enumeration method for a class of JSS problems. More recently, a technique to obtain near-optimal solution for parallel identical machines has been used by [13]. A shifting bottleneck heuristic as one successful research for decomposing the job shop into sub-problems is presented by [14].

LEKIN Software, as reported in [15], was designed as a tool with the main purpose of introducing the main scheduling theory and demonstrating the capabilities of several traditional techniques.

A concise survey on main Analytical and Heuristic Techniques that have been used to deal with JSS problem is provided in tables (1), (2) respectively.
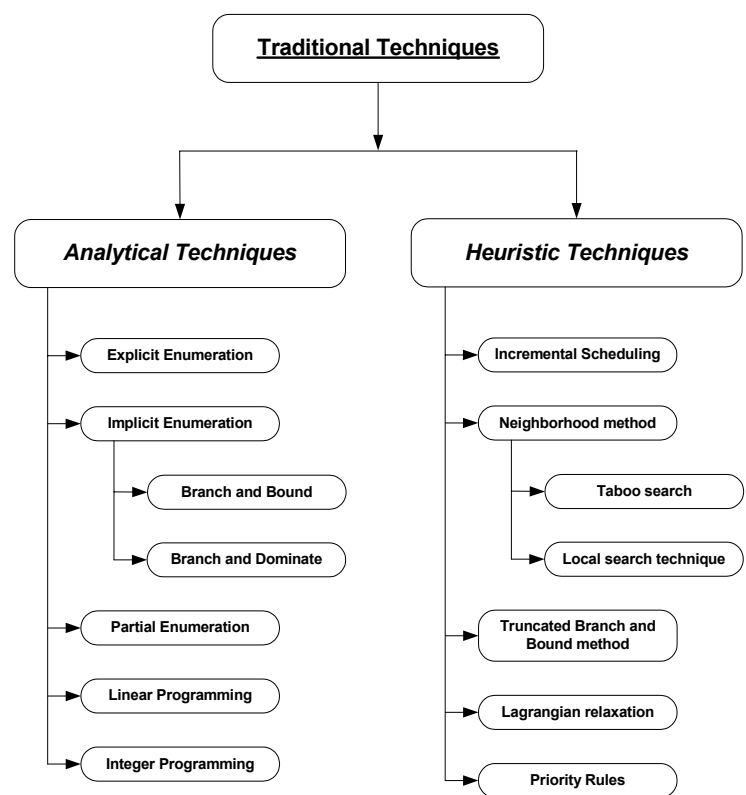

Figure 2 : Traditional Techniques

Table 1: Analytical Techniques

MAIN FEATURES	LIMITATIONS
1. Explicit Enumeration - In this method, they generate a complete enumeration tree. The leaves of the tree represent all feasible solutions. The path from the root to a leaf of minimal makespan represents an optimal solution. - [8, 16] offered an algorithm which created an active schedule with respect to disjunctive arcs. - A computer program algorithm, which deals with optimal job sequence (90x90) took 2.5 hours on 386 PC, 16 MHz). [11]	- The remaining difficulty is the size of search tree generated. Since we have a maximum of $(n!)^m$ solutions to consider. - The limitation of algorithm [8], [16] is that it presents procedure relations that cannot be determined before a schedule is constructed. Also, it is not adequate to capture sequence dependent set-up and tear down time in every case (White *et. al* 1990).
2. Implicit Enumeration - The strategy of implicit enumeration attempts to minimize an objective function without considering every possible solution. Implicit enumeration schemes examine increasingly smaller subsets of feasible solutions until these subsets definitely do not contain improved solutions.	- All implicit enumeration approaches for the determination of an optimal schedule appear to be susceptible to the combinatorial nature of these problems, when they are tested with multiple-resources (more than 50 activities). [8]
2.1. Branch and Bound - Branch and Bound Algorithms cut branches from the enumeration tree and therefore reduce the number of generated nodes substantially. An optimal solution can be found by systematically examining the subsets of a feasible solution. - Several different algorithms exist for JSS [7,8] and have been applied to flow scheduling [51].Survey on Branch and Bound methods.[17]	- The limitation of this algorithm is that the makespan is the only criterion, which can be evaluated. - The efficiency of the technique depends very much on the efficiency of the lower bound. The more efficient the bound, the smaller the amount of enumeration of the solution tree that needs to be carried out. - It is impractical to enumerate even this reduced set of alternatives because it is too large.
2.2. Branch and Dominate - Similar to Branch and bound but differs in the pruning approach. If there is a set of conditions at a node which mean that the schedules will be inferior to the best schedule at some other node, then the first node may be eliminated from further consideration. In this way the second node dominates the first.	- Using dominance conditions may shorten the search sufficiently such that a reduction in overall computational requirements is obtained [8, 9]. It is still impractical to enumerate this reduced set.
3. Partial Enumeration - The optimal schedule has been shown to always be in a subset of feasible schedules, termed 'active'. This identification of such active feasible schedules has been used. [19] Recently, Shifting Bottleneck algorithm is considered as a good step in partial enumeration. [12]	- This method to define active and semi-active schedules helps to reduce a computational work somewhat. However, there is still a need to generate a high number of schedules to get the optimal one. The problem complexity increases with more machines and jobs.
4. Linear Programming - The particular attraction, from the model building point of view, of linear programming is that highly efficient program codes are available which can deal with very large problems involving many variables and constraints. - It is fair to say that, in some cases the specific nature of the problem allows certain simplifying approximations that permit a solution by Linear Programming.	- Linear Programming can often be used as a practical technique, but only if the problem conforms entirely to the requirements of the approach. - The main shortcoming in that most of the real manufacturing planning and scheduling do not behave linearly in most cases, even after the simplifications. - Moreover, some or most of constraints in practice cannot be represented as linear. For example, the specification that either machine A or B may be used to process job j.
5. Integer Programming - To overcome some of the limitations of linear programming integer variables may be used. This complication the solution requiring the use of less efficient algorithms. Integer Programming formulations of the JSS problem have been reported. [20]	- The present integer programming codes available are over-stretched even by very small JSS problem formulations. Even allowing for possible developments and improvements in integer programming computer codes, it would appear that as a general method for solution of JSS problems it is a non-starter.

Table 2: Heuristic Techniques

MAIN FEATURES	LIMITATIONS
1. Incremental Scheduling - Incremental schedule building starts with an empty timeline and a set of tasks to be scheduled. The basic idea behind incremental scheduling is to choose the next task to be scheduled and to place that task on the timeline so that no constraints will be violated. The placement algorithm may be very simple or very involved, attempting some degree of optimization. - This process repeats until either all tasks have been scheduled, or there are still tasks that remain to be scheduled but no times. In this latter case, the scheduler has effectively reached a dead-end.	- Some systems halt at this point of dead-end, presenting an incomplete solution to the user. Others attempt to free the scheduler from the dead-end condition by undoing some previously made decision. Incremental scheduling can degenerate into brute-force trial-and-error searches. Since this is a computationally intractable alternative, incremental scheduling systems tend to be either slow or poorly optimizing, or both.
2. Neighborhood method - Neighborhood search techniques begin with any feasible schedule, adjust this somewhat, check whether the adjustment has made any improvement. Continuing in this cycle of adjusting and testing until an improvement measure is achieved. Two related concepts, which are the basis of this method, are the neighborhood sequence and the neighborhood generating mechanisms for these sequences [21].	- The search procedure of this family of algorithms terminates with a sequence that is a local optimum. Unfortunately, there is in general no way to guarantee or even know if the terminal sequence is also a global optimum. However, few experiments (Spachis & King, 1979) indicated that, fundamental neighborhood search algorithm described above, is fairly reliable as a general-purpose heuristic procedure (Baker 1976).
2.1. Taboo search - Taboo search approaches produce good results in reasonable runtime. Taillard [22] applied this global optimization technique to the JSS and showed that it is typically more efficient than the shifting bottleneck procedure and simulated annealing implemented by Lenstra (1992). Taillard provides optimal solution for some identified problem with shorter computational time for more complex problems.	- Requires large memory, as subsets of the solution path are kept in memory [12]. - Another crucial aspect is the maintenance of the taboo list using variable taboo list length and cycle detection mechanisms which prevent cycling around a number of neighboring solutions.
2.2. Local search technique - Simulated annealing [23] and taboo search techniques (Widmer 1989, Dell'Amico *et.al.* 1993,) are the main local search techniques that have been tested on the JSS problem. In both cases, the neighborhood structure is based on scheduling arrangement. [19]	- In comparison with other heuristic methods both techniques yield quite consistently good solutions. - Simulated annealing is comparatively much more time consuming that taboo search on difficult instances.
3. Truncated Branch and Bound method - One of the most efficient approximate methods proposed so far is probably the shifting bottleneck procedure developed by Adams et al (1988). [24] Its main idea is following: Starting with the initial JSS problem, they optimally sequence one by one the machines, using Carlier (1982) algorithm for the one machine problem. The order in which the machines are sequenced depends on a bottleneck measure associated with them.	- This procedure is embedded in a second heuristic of an enumerative type, for which each node of the search tree corresponds to a subset of sequenced machines. In comparison to other algorithms, it is less efficient as each time a new machine is sequenced, they attempt to improve all previous scheduled machines in long re-optimization steps.
4. Lagrangian relaxation - Scheduling methodologies based on Lagrangian relaxation have proved to be computationally efficient and have provided near optimal solutions to identical parallel machine scheduling problems. It has been applied to schedule job shops, which include multiple machine types, generic precedence constraints and simple routing considerations. [13]	- It can be applied to some cases in machine scheduling and under certain conditions. - The results are not guaranteed in complex JSS problems. - It provides near optimal in case of identical machine scheduling.

5. Priority Rules
- Priority Rules indicate how to assign a specific job to a specific machine at a given time, when a machine becomes available for process (Rowe, Jackson 1956). A lot of studies were done over these rules, [8, 7, 21, and 25]. Pinedo & Bhaskaran (1992) presents classification of basic dispatching rules.
- As [26] classified over 100 priority rules as
   I)   Simple Priority Rules.
   II)   Combination of simple Priority Rules.
   III)   Weighted Priority Index.
   IV)   Heuristic Scheduling Rules.
   V)   Other Rules.
	- Researchers have analyzed sequencing decisions jointly with other dynamic decisions, such as job release (Wein 1990a, 1992), due date (Seidmann 1981, Baker & Bertand 1981), pricing (Kleinrock 1967, Dolan 1978 and Mendelson & Whang 1990), routing (Hajek 1984, Wein 1991) and arrival rates (Ezat and ELBaradie 1992). Unfortunately, none of the rules seems to outperform any other for practical problem setting. Recently simulated annealing was also applied to deal with JSS problems as a remedy.

4.2.   Theory-Practice Gap

Unfortunately, no simple scheduling algorithm exists for the general 'n' jobs, 'm' machines in case of JSS. There is a gap between scheduling theory – as represented by analytical methods – and practice. Figure 3. This stems from the inability of theory, as so far developed, to cope adequately with the complexities of many of the real-world JSS problems.

Many researchers in the field, faced with these difficulties, have adopted the not uncommon device used by all researchers, which is if you are faced with a problem that is difficult to solve, then make simplifying assumption and approximations to reduce the problem to a form that you can hope to solve.

In case of JSS the fundamental difficulties of the real practical problem have led to simplifications on a scale, that in some cases has reduced the problem to a shadow of reality.[1] Typically this has resulted in:

1. *Emphasis on small-scale problems*
Although most of research is concerned with the general n-job, m-machine scheduling problem there has been a great deal of concentration of effort on the small-scale problem involving more or less four machines.

2. *Simplified Problem constraints*
These constraints were covered in detail by [6]. The main constraints are:

Figure 3: The Theory-Practice Gap [1]

- All jobs and their processing times are known prior to scheduling being carried out. This effectively transforms the dynamic problem to static one.
- Machines are assumed to be able to operate on only one job at a time.
- Job splitting and job lapping are only permitted in very exceptional cases.

3. *Simple objective functions*
Most of these are single parameter objective functions, which are to be optimized. Probably, the most commonly used objective functions are:
- Mean flow time.
- Total lateness.
- Number of late jobs.

It is true to say that some insight into the solution of the practical JSS problem has been derived from this research activity on simplified problems.

In practice the complexity of production planning and control in general and scheduling in particular varies from one situation to another. The degree of complexity is governed by such factors as average number of operations per product, product variety and scale and type of production.

4. *Optimization & satisfying solution*
Analytical methods are generally concerned with optimizing solutions, where the optimization is carried out with respect to some particular success criteria. It is well known that the measures of success are multi-dimensional. For example, makespan time may be important but so are work in progress, machine utilization, labor utilization, delivery performance, … etc.
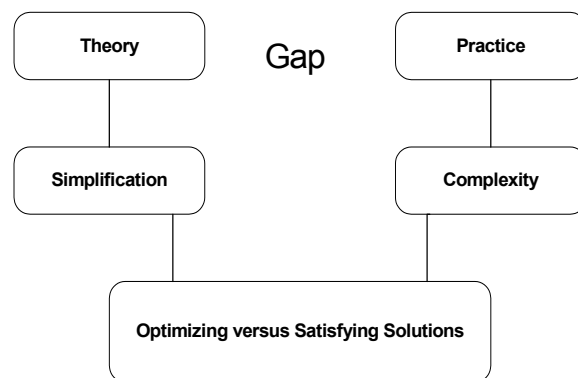
The Relative importance of each will vary from company to company, and indeed may change over time within a company. Probably, the most rational objective function would be one comprising a weighted function of the various performance criteria deemed relevant, with management determining subjectively the relative weighted of each.

In practical scheduling work, they are more concerned with establishing feasible schedules, which will provide satisfactory performance against measures of success such as flow time, delivery dates, utilization, ..etc.

Heuristic methods of scheduling, by definition give rise to satisfying solutions, and the advanced techniques focused on satisfying solution more comprehensive.

4.3.   Advanced Techniques

▪ **Simulation**

Simulation has proven to be an excellent strategic tool for the enterprise. However, it can be used as a day-to-day tactical tool on the shop floor.

Simulation can be applied to many aspects of manufacturing systems, however two areas stand out in particular [11]:

1. In Job shop, the simulation of dispatching rules and the assessment of the effect of different rules on the shop's ability to meet delivery dates and utilize the machines.
2. In flow lines to try to minimize the loss of output.

However, Simulation has been applied to more advanced systems in manufacturing such as FMS, Automation, … etc.

The first application of simulation was, computer simulation studies of different priority rule have been carried out. For example, Le Grande and Bulkin *et.al.* (1963), and Elmaghraby & Cole (1963) [27], applied their control of the production at Western Electric.

Other investigations such as [28, 7, 29, 30, 31], they have experimented with computer simulation models of hypothetical job shops in which assumptions are made about the mechanism for generating job arrivals and processing times, while [32] establishes an economic evaluation of job shop dispatching rules. The priority dispatching rules in job shops with assembly operations and random delays has been studied by [33] followed by more comprehensive study with [34] in a fabrication/assembly shop.

ElBaradie [25, 35] showed how to use the computer aided-simulation as a tool for the optimization of the flow shop scheduling versus different priority rules, followed by further study to the effect of various priority rules on minimizing multiple criteria. The objective of all these simulation experiments has been to evaluate and determine efficient and effective scheduling rules that may be generally applied in practice.

Simulation has been a popular methodology (Paul 1990) with a broad range of applications (Mott 1993). The use of simulation software has become widely accepted as a tool for the improvement and enhancement of the performance of a manufacturing system in general. Simulation is also accepted as the tool for the evaluation of the manufacturing system in operations using " What-If" Scenarios prior to doing any harm in real life.

Current tools make it relatively simple to build a simulation model for planning and scheduling. Using this model, through the definition and application of the rules used to assign work to the available resources, the scheduler can be sure that all of the combinations and exceptions are considered and the production objectives satisfied. More recently the tracking and reporting of this process has been integrated within the software, and hence simulation-based scheduling has become the start point in solving the scheduling problems. Improvements in simulation software can help to find efficient way to shorten the time needed to get the optimal scheduling.

There are many different manufacturing oriented simulation packages in the market. Each package carries their strengths and weaknesses. Some packages focus on ease of use and compromise flexibility, while others focus on flexibility and are more difficult to use. In addition, some packages have been developed for specific industries or systems. As most of manufacturing systems have some unique intricacy, the best packages allow the user to combine easy to use constructs with more flexible, lower level constructs.

A comprehensive list of criteria structured in a hierarchical framework for evaluating simulation software is shown in Figure 4. [36]
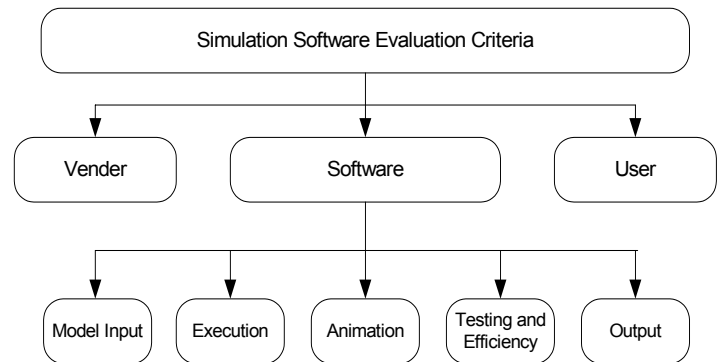
Figure 4: Software Selection Criteria [36]

Some of the more popular manufacturing oriented packages include Arena, AutoMod, ProModel, Extend, Frontstep, Interval-Logic, GPSS, AutoMod, Mathlab (Simulink), and WITNESS.

- **Artificial Intelligence**

The need for rapid solution prompted researchers to use AI techniques such as Knowledge-Based Systems (KBS), Expert Systems (ES), Neural Networks (NNs), Case-Based Reasoning (CBR), Genetic Algorithms (GA), Fuzzy Logic and any combination of these techniques [37]. In AI term, JSS is a planning problem with special characteristic [38]. Figure 5 illustrates some applications of AI in scheduling to deal with JSS problem.

- **Repair Based Methods**

In general, scheduling appears to be an excellent application area for repair-based methods. Supporting evidence comes from previous work on other real-world scheduling applications by Zweben (1990), Biefeld and Cooper (1991), and Kurtzmann (1988). Each of these projects uses iterative improvement methods that can be characterized as repair-based.

There are several reasons why repair-based methods are well suited to scheduling applications. Unexpected events can require schedule revision; in which case, dynamic rescheduling is an important issue [39]. Repair-based methods can be used for rescheduling in a natural manner. Second, most scheduling applications involve optimization, at least to some degree, and repair-based methods are also naturally extended to deal with such issue [38].

Empirically, it has been shown that this approach rapidly converges to solutions, and, guided by preferences, converges to highly optimized solutions. Another critical attribute of the Repair-Based approach is that it is extraordinarily well suited to rescheduling. Rescheduling typically takes less time than the initial schedule generation, as it requires fewer repairs [40].

It has been pointed out, there are real-world scheduling problems where humans find repair-based methods very natural [41].

Hybrid Systems:
- Szelke and Markus (1997) combined machine learning techniques and CBR to solve shop floor scheduling problems.
- Kim (1998) used an integrated approach of inductive learning and competitive NNs for developing multi-objective FMS schedulers. The FMS scheduler can assist to make a decision in real time, while satisfying multiple objectives desired by the operator.
- Lee (1998) combined GAs and machine learning to develop a JSS system. [49]
- Soon and Souza (1997) [17] have explored the possibility of using a hybrid approach employing neural networks and simulation technology to solve the detailed scheduling problem (a known class of NP-complete problems) in a manufacturing cell.
- Wang and Zheng presents (May 2001) a parallel easily implemented hybrid optimization framework, which reasonably combines Genetic Algorithm (GA) with Simulated Annealing (SA) and apply it to JSS problem. [29]

Fuzzy Logic :
- Yu (1999) proposed approach to FMS with Multi-criteria based on Fuzzy inference.
- The proposed system performed very robustly with respect to shop workload for all used performance measures. [29]

Neural Networks (NNs):
- Neural Networks have been successfully applied to the solution of constrained optimization problems.
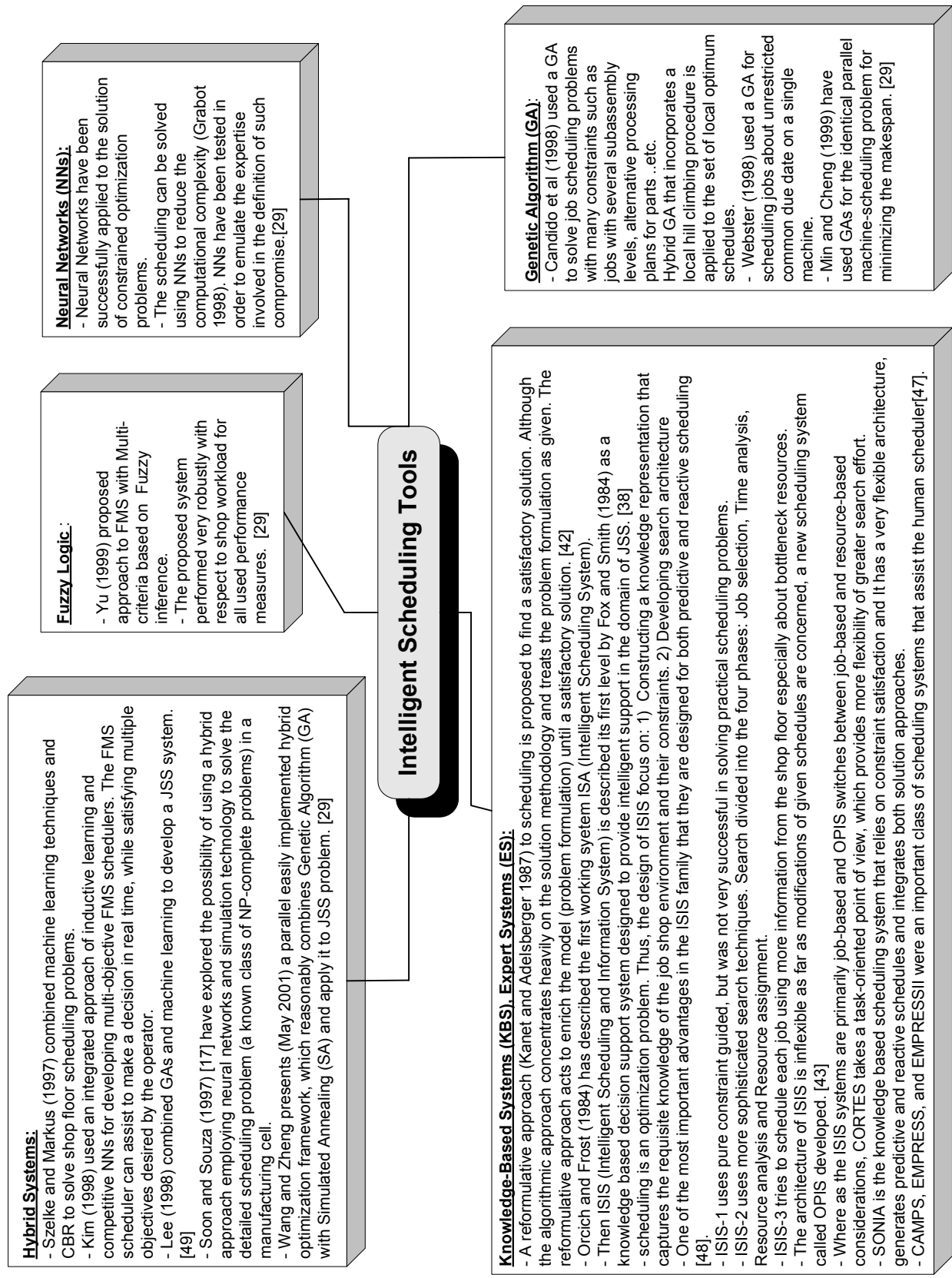- The scheduling can be solved using NNs to reduce the computational complexity (Grabot 1998). NNs have been tested in order to emulate the expertise involved in the definition of such compromise.[29]

Genetic Algorithm (GA):
- Candido et al (1998) used a GA to solve job scheduling problems with many constraints such as jobs with several subassembly levels, alternative processing plans for parts ..etc.
Hybrid GA that incorporates a local hill climbing procedure is applied to the set of local optimum schedules.
- Webster (1998) used a GA for scheduling jobs about unrestricted common due date on a single machine.
- Min and Cheng (1999) have used GAs for the identical parallel machine-scheduling problem for minimizing the makespan. [29]

Intelligent Scheduling Tools

Knowledge-Based Systems (KBS). Expert Systems (ES):
- A reformulative approach (Kanet and Adelsberger 1987) to scheduling is proposed to find a satisfactory solution. Although the algorithmic approach concentrates heavily on the solution methodology and treats the problem formulation as given. The reformulative approach acts to enrich the model (problem formulation) until a satisfactory solution. [42]
- Orcich and Frost (1984) has described the first working syetem ISA (Intelligent Scheduling System).
- Then ISIS (Intelligent Scheduling and Information System) is described its first level by Fox and Smith (1984) as a knowledge based decision support system designed to provide intelligent support in the domain of JSS. [38]
- scheduling is an optimization problem. Thus, the design of ISIS focus on: 1) Constructing a knowledge representation that captures the requisite knowledge of the job shop environment and their constraints. 2) Developing search architecture [48].
- One of the most important advantages in the ISIS family that they are designed for both predictive and reactive scheduling
- ISIS-1 uses pure constraint guided, but was not very successful in solving practical scheduling problems.
- ISIS-2 uses more sophisticated search techniques. Search divided into the four phases: Job selection, Time analysis, Resource analysis and Resource assignment.
- ISIS-3 tries to schedule each job using more information from the shop floor especially about bottleneck resources.
- The architecture of ISIS is inflexible as far as modifications of given schedules are concerned, a new scheduling system called OPIS developed. [43]
- Where as the ISIS systems are primarily job-based and OPIS switches between job-based and resource-based considerations, CORTES takes a task-oriented point of view, which provides more flexibility of greater search effort.
- SONIA is the knowledge based scheduling system that relies on constraint satisfaction and It has a very flexible architecture, generates predictive and reactive schedules and integrates both solution approaches.
- CAMPS, EMPRESS, and EMPRESSII were an important class of scheduling systems that assist the human scheduler[47].

Figure 5: Intelligent Scheduling Tools

5. Future Trends

- The trend of current scheduling technology is towards a combination of the three common approaches; operations research-based, simulation-based and AI-based.
- AI techniques show the most promise in providing an effective tool to solve JSS problems.
- Rodd (1992) and Kopacek (1999) state that integration of JSS solvers into the manufacturing system will be the next main task. Some progress has already been made on the integration of computerized process planning and scheduling (Aldakhilallah and Ramesh 1999) and (Morad and Zalzala 1999) using GAs [45].
  Intelligent agents, which co-ordinate localized AI systems distributed throughout the manufacturing plants and business are seen as a key solution for integration.
  The two component which make up such a system are the agent, which selects the most appropriate priority rule from the local shop conditions in real time, and a simulation environment which performs the scheduling using this rule [46].
- The World Wide Web represents one of the most important challenges in the emerging information society. Improvements in web-simulation technology can allow both the clients and the consultant to work with models more efficiently and effectively over the web without having to be physically in the same location.
- Virtual manufacturing is another new approach to Manufacturing. It requires a robust information model for products, processes and production systems. The decreasing costs of hardware have made virtual environment increasingly popular and are used in many fields. VM can provide details and information about, process, production and shop floor control to be shared over network (Lin 1995).

6. Conclusion

Throughout this paper, the development of the solution to the JSS problem, a hard combinatorial optimization problem of practical relevance, has been described. Following brief evaluation of the JSS problem and its nature of this problem. The solution techniques are classified into two main folds: Traditional Techniques and Advanced Techniques.

A concise survey of the huge research effort devoted to the JSS problem over last 30 years is presented. Considerable progress has been made in the traditional techniques towards an efficient resolution of the JSS problem. Nevertheless, this remains one of the most difficult combinatorial problems to date and always arouses new research interest.

New outlook on the scheduling problem, since the 1980s, as a topic of research has garnered the attention of significant AI research. Much of the successful work to date has been based on the use of constraints to guide the search process.

In this paper, a sample of distinguished works in AI approach towards solving JSS problem have been provided. Simulation has proven to be an excellent technique in search of even more widespread acceptance than it currently has. Simulation has developed into a useful tool to evaluate and enhance the performance of the manufacturing systems using " What-if" scenarios prior to implementing the solution in real life.

However, there are still many areas where more research is needed, including the integration of the system into manufacturing installations, the use of intelligent agents and application to virtual manufacturing.

References
[1]   King, J.R. "The Theory-Practice Gap in Job Shop Scheduling". *The Production Engineer*, March 1976.

[2]   Pinson, E., "The Job Shop Scheduling Problem: Concise survey and some recent Developments", Scheduling Theory and its applications Edited by P. Chretienne, E.G. Coffman Jr., J.K. Lenstra and Z. Lim, 1995 *John Wiley & Sons Ltd*.

[3]   Herrmann, J. W. and Snowdon, Jane L. "A Classification of Static Scheduling Problems". *Complexity in Numerical Optimization* 1993, pp 203-253.

[4]   Johnson, S. M. " Optimal Two and Three-Stage Production Schedules with Setup Times Included". *Naval Research Logistics Quarterly*, Vol. 1, No. 1, 1954.

[5]   Smith, W. E. " Various Optimizers for Single Stage Production". *Naval Research Logistics Quarterly*, Vol. 13, No. 1, 1956.

[6]   Jackson, J. R. " Simulation Research on Job Shop Production". Naval Research Logistics Quarterly, Vol. 4, 1957, 287-295.

[7]   Conway, R. W., Maxwell, W.L. and Miller, I. W. " Theory of Scheduling ". *Addison Wesley*, Reading, Massachusetts, 1967.

[8]   Baker, Kenneth " Sequencing and Scheduling". *John Wiley & Sons Inc.*1998, ISBN: 0-9639746-1-0.

[9]   Rinnooy Kan A. H. G. " Machine Scheduling Problems: Classification, Complexity and Computations". *Nijhoff, The Hague*, 1976.

[10] Brooks, G.H., and White, C.R. " An Algorithm for finding optimal or near optimal solutions to the production scheduling problem". Journal of Industrial Engineering, Vol. 16, No. 1, 1965.

[11] Agha, Mujanah Ezat, " Simulation of Production scheduling in Manufacturing Systems", M.Sc., School of Mechanical & Manufacturing, *Dublin City University*, 1993.

[12] Mattfeld, Dirk C. " EVOLUTIONARY SEARCH AND THE JOB SHOP, Investigation on Genetic Algorithms for Production Scheduling". *Physica-Verlag*, A Springer-Verlag company, 1996, ISBN: 3-7908-0917-9.

[13] Hoitomt Debra J., Luh, Peter B., and Pattipati, Krishna R. " A Practical Approach to Job Shop Scheduling Problems". *IEEE Transactions on Robotics and Automation*, Vol.9, No.1, Feb. 1993.

[14] Pinedo, Michael, Singer, Marcos " A Shifting Bottle neck Heuristic for Minimizing the Total Weighted Tardiness in a Job Shop". *Naval Research Logistics*, Vol. 46, 1999, pp 1-17.

[15] Pinedo, Michael. " Operations scheduling with applications in manufacturing and services". *Irwin/McGraw-Hill*, 1998. ISBN: 0072897791.

[16] Giffler, B. and Thompson, G. L. " Algorithms for Solving Production Scheduling Problems". *Operations Research*, Vol. 8, No. 4, 1960.

[17] Brucker, P., Jurisch, B., Sievers, B. (1994) " A Branch and Bound Algorithm for Job Shop Scheduling Problem". *Discrete Applied Mathematics*, Vol. 49, pp 107-127.

[18] Brooks, G., and White, C. " An Algorithm for finding optimal or near optimal solutions to the production scheduling problem". *J. of Industrial Engineering*, Vol. 16, No. 1, 1965.

[19] Grabowski J., Nowicki E. and Zdrzalka S. " A Block Approach for Single Machine Scheduling with Release dates and Due Dates". *European Journal of Operations Research*, Vol. 26, 1986, pp 278-285.

[20] Wagner, H. M. " An Integer Linear Programming Model for Machine Scheduling". *Naval Logistics Quarterly*, Vol. 6, No. 2, 1959.

[21] French, S. "Sequencing and Scheduling". An introduction to the mathematics of the job shop, *Ellis Horwood series*, 1982.

[22] Taillard, Eric D., " Parallel Taboo search Techniques for the Job Shop Scheduling Problem". *ORSA Journal on Computing*, Vol.6, No.2, Spring 1994.

[23] Van Laarhoven P.J.M., Aarts E. H. L. and Lenstra J. K. " Job Shop Scheduling by Simulated Annealing". *Operations Research*, Vol. 40, 1992, pp 113-125.

[24] Adams, J., Balas E. and Zawack, D. " The Shifting Bottleneck Procedure for Job Shop Scheduling". *Management Science*, Vol.34, 1988, pp 391-401.

[25] Ezat, A. M. and ElBaradie, M. " A Computer Aided-Simulation for the Optimization of the Flow-Shop Scheduling Vs. Different Priority Rules". *Proceedings of the Ninth Conference of the Irish Manufacturing Committee (IMC-9)*, Sep. 1992, pp 139-150.

[26] Panwalker, S.S., Iskander, Wafik " A Survey of Scheduling Rules". *Operations Research*,

Vol.25, No.1, January-February 1977.

[27] Elmaghraby, S. W. and Cole, R. T. " On the Control of Production in Small Job Shops". *Industrial Engineering*, Vol. 14, No. 4, 1963.

[28] Dzielinski, B. P. and Baker, C. T. " Simulation of Simplified Job Shop". Management Science, Vol. 6, No. 3, Apr. 1960.

[29] Gere, W." Heuristics in Job Shop Scheduling". *Management Science*, Vol.13, No.3, 1966.

[30] Ellon, S. and Cotterill, D. J., " A Modified SI Rule in Job Shop Scheduling". The *International Journal of Production Research*, Vol.7, No.2, 1968.

[31] Hollier, R. H., " A Simulation Study of Sequencing in Batch Production". Operation Research Quarterly, Vol. 19, No. 4, 1986.

[32] Jones, C. H. " An Economic Evaluation of Job Shop Dispatching Rules". Management Science, Vol. 20, No. 3, 1973.

[33] Sculli, D. " Priority Dispatching Rules in Job Shops with Assembly Operations and Random Delays". OMEGA 8, 227-234, 1980.

[34] Sculli, D. and Tsang, K. K. " Priority Dispatching Rules In A Fabrication/Assembly Shop". *Mathl Comput. Modeling*, Vol. 13, No. 3, 1990, pp. 73-79.

[35] Ezat, A. M. and ELBaradie, M. " Flow-Shop Scheduling: Effect of Various Priority Rules on Minimizing Multiple Criteria". *30th International MATADOR Conference (UMIST)*, 1993.

[36] Nikoukaran, J., Hlupic, Vlatka and Paul, Ray J. " Criteria For Simulation Software Evaluation". *Proceeding of the 1998 Winter Simulation Conference (WSC)*, pp 399-406.

[37] Meziane, Vadera, Kobbacy and Proudlove " Intelligent Systems in Manufacturing: Current Developments and Future Prospects". *Integrated Manufacturing System*, Vol. 11, No. 4, 2000.

[38] Fox, Mark S. and Zweben, Monte " Intelligent Scheduling". *Morgan Kaufmann Publishers INC*., 1994, ISBN: 415-392-2665.

[39] Zweben, M., Deale M. and Gargan, R. " Anytime Rescheduling". Proceedings of the *Workshop on Innovative Approaches to Planning, Scheduling, and Control*, Calif. 1990.

[40] Rosenthal, Don and Johnston, Mark " A Look Under The Hood". *Interval Logic corporation*, 1997, http://www.interval-logic.com.

[41] Biefeld, E., and Cooper, L. " Bottleneck Identification Using Process Chronologies". Proceedings IJCAI- 91. AAAI press, Menlo Park, Claif.1991.

[42] Kanet, J. and Adelsberger, H. " Expert Systems in Production Scheduling". *European Journal of Operational Research*, Vol.29, 1987, pp 51-59.

[43] Blazewicz, J., Ecker, Klaus H., Schmidt, Guenter and Weglarz, Jan " Scheduling in Computer and Manufacturing Systems". *Springer Verlag*, Berlin 1994. ISBN: 3540580492.

[44] Meht, Arvind and Rawles, Ian " Business Solution using WITNESS". *Proceeding of the 1999 Winter Simulation Conference (WSC)*, 230-233.

[45] Morad, N. and Zalzala, A.M.S. " Genetic Algorithms in Integrated Process Planning and Scheduling". *Journal of Intelligent Manufacturing*, Vol. 10, No. 2, 1999, pp 169-79.

[46] Aydin, M., and Öztemel, E. " Dynamic Job Shop Scheduling Using Reinforcement Learning Agents". *Robotics and Autonomous Systems*, Vol. 33, Issue 2-3, Nov. 2000.

[47] Atabakhsh, H. " A Survey of Constraint-based Scheduling Systems using an Artificial Intelligence Approach". Artificial Intelligence in Engineering, Vol.6, No.2, 1991.

[48] Fox, M.S. and Smith, S.F. " ISIS – A Knowledge-Based System for Factory Scheduling". *Expert Systems*, Vol. 1, No. 1, 1984, pp 25-49.

[49] Lee, C.Y., Piramuthu, and Tsai, Y.K." job Shop Scheduling with Genetic Algorithm and Machine Learning". *Int. Journal of Production Research*, Vol. 35, No. 4, 1998, pp 1171-91.

[50] Lawrence M. Wein and Chevalier, Philippe B. " A Broader view of the Job Shop Scheduling Problem". *Management Science*, Vol. 38, No.7, July 1992.

[51] Brown, A. P. G. and Lomnicki, Z.A. " Some applications of the Branch and Bound Algorithm to the Machine Scheduling Problem". Operational Research Quarterly, Vol.17, 1966.