DETERMINISTIC JOB-SHOP SCHEDULING : PAST, PRESENT AND FUTURE

A. S. Jain • S. Meeran

Department of Applied Physics and Electronic and Mechanical Engineering,

University of Dundee, Dundee, Scotland, UK, DD1 4HN

a.z.jain@dundee.ac.uk • s.meeran@dundee.ac.uk

ABSTRACT:- Due to the stubborn nature of the deterministic job-shop scheduling problem many solutions proposed are of hybrid construction cutting across the traditional disciplines. The problem has been investigated from a variety of perspectives resulting in several analytical techniques combining generic as well as problem specific strategies. We seek to assess a subclass of this problem in which the objective is minimising makespan, by providing an overview of the history, the techniques used and the researchers involved. The sense and direction of their work is evaluated by assessing the reported results of their techniques on the available benchmark problems. From these results the current situation and pointers for future work are provided.

KEYWORDS:- Scheduling Theory; Job-Shop; Review; Computational Study;

1. INTRODUCTION

Current market trends such as consumer demand for variety, shorter product life cycles and competitive pressure to reduce costs have resulted in the need for zero inventory systems. However to maintain market share the system must be fast responding which implies that more stock has to be maintained. These conflicting requirements demand efficient, effective and accurate scheduling which is complex in all but the simplest production environments. As a result there is a great need for good scheduling algorithms and heuristics. Scheduling is essentially concerned with solving a Constraint Optimisation Problem (*COP*) and in the context of manufacturing it involves finding a sequential allocation of competing resources that optimises a particular objective function. The deterministic job-shop scheduling problem, hereinafter referred to as Π_J, is the most general of the classical scheduling problems.

Π_J consists of a finite set $\mathcal{J}$ of n jobs $\{\mathcal{J}_i\}_{i=1}^{n}$ to be processed on a finite set $\mathcal{M}$ of m machines $\{\mathcal{M}_k\}_{k=1}^{m}$. Each job $\mathcal{J}_i$ must be processed on every machine and consists of a chain or complex of m_i operations O_{i1}, O_{i2}, …, O_{im_i}, which have to be scheduled in a predetermined given order (precedence constraint). There are N operations in total, $N = \sum_{i=1}^{n} m_i$. O_{ik} is the

operation of job $\mathcal{J}_i$ which has to be processed on machine $\mathcal{M}_k$ for an uninterrupted processing time period τ_{ik} and no operation may be pre-empted.[1] Each job has its own individual flow pattern through the machines which is independent of the other jobs. Each machine can process only one job and each job can be processed by only one machine at a time (capacity constraints). The duration in which all operations for all jobs are completed is referred to as the makespan C_{max}. The only objective considered in this work is to determine starting times for each operation, $t_{ik} \geq 0$, in order to minimise the makespan while satisfying all the precedence and capacity constraints:

$$C^*_{max} = min\left(C_{max}\right) = \min_{feasible\ schedules} \left(max\left(t_{ik} + \tau_{ik}\right) : \forall\ \mathcal{J}_i \in \mathcal{J},\ \mathcal{M}_k \in \mathcal{M}\right).$$

Note the dimensionality of each Π_J instance is specified as $n \times m$ and N is often assumed to be nm provided that $m_i = m$ for each job $\mathcal{J}_i \in \mathcal{J}$ and that each job has to be processed exactly once on each machine. In a more general statement of the job-shop problem, machine repetitions (or machine absence) are allowed in the given order of the job $\mathcal{J}_i \in \mathcal{J}$, and thus m_i may be greater (or smaller) than m. The main focus of this work is given to the case of $m_i = m$ non pre-emptive operations per job $\mathcal{J}_i$, unless otherwise stated.

Although makespan minimisation may not be perceived as a good theoretical objective function, its usage in academic and industrial practice is wide spread. This criterion has much historical significance and was the first objective applied by researchers in the early fifties to the so called easy problems which later were found to be combinatorially exploding. This criterion is simple to deal with from a mathematical point of view and is easy to formulate. Consequently it has been the principal criterion for academic research and is able to capture the fundamental computational difficulty which exists implicitly in determining an optimal schedule. The flexibility and generic nature of this criteria is suggested by Demirkol *et al*. (1997) who indicate that a solution for C_{max} is likely to perform well on average with respect to the criteria of ΣC_i, ΣT_i and L_{max}.

A scheduling problem with the objective to minimise the makespan can be considered analogous to the well known Travelling Salesman Problem (TSP) which is in a real sense a hypothetical problem. However solving either the TSP or Π_J will provide an insight into finding solutions to more practical problems which have non regular performance measures. As the makespan minimisation problem is so well defined, with an abundance of available literature, it is simple to understand and hence it acts as a training problem to the more difficult and less

[1] In this analysis each operation is given a unique number from 1 to (nm). If $\mathcal{M}_k$ is the k^{th} machine required by o_{ik}, with respect to precedence constraints, then its operation number is $n(k-1) + i$ (cf. table 2 and figure 4).

well documented practical problems. Therefore as Π_J is an important model in scheduling therory serving as a proving ground for new algorithmic ideas and providing a starting point for more practically relevant models, it is worth understanding.

An $n \times m$ size Π_J has an upper bound of $(n!)^m$, thus a 20×10 problem may have at most 7.2651×10^{183} possible solutions. Complete enumeration of all these possibilities to identify feasible schedules and the optimal one is not practical. Due to this factorial explosion Π_J is considered to be a member of a large class of intractable numerical problems known as $\mathcal{NP}$-Hard ($\mathcal{NP}$ stands for non deterministic polynomial). Here we digress to explore the computational complexity. If x is the size of the input and y is a constant then a problem with a time requirement of order $O(x^y)$ is said to have a polynomial complexity and is denoted by $\mathcal{P}$. Whereas if the problem complexity is of order $O(y^x)$ then it is considered to be $\mathcal{NP}$-Hard (if $\mathcal{P} \neq \mathcal{NP}$) (Cook 1971; Garey *et al*. 1976). Here an optimal algorithm would require a number of computational steps that grows exponentially with the input. The common belief is that no efficient algorithm can ever possibly be found to solve these inherently hard problems. Even most of the special cases of Π_J are also $\mathcal{NP}$-Hard or strongly $\mathcal{NP}$-Hard which makes this problem one of the most stubborn members of this class (Nakano and Yamada 1991; Lawler *et al*. 1993). This is highlighted by the fact that algorithms can optimally solve other $\mathcal{NP}$ instances such as randomly generated travelling salesman problems, with more than 4000 cities, and set covering problems with tens of thousands of variables. However as yet strategies have not been devised that can guarantee optimal solutions for Π_J instances which are larger than 20×10. The intractability of Π_J is further emphasised by the fact that unlike other $\mathcal{NP}$-Hard problems local optimisation does not lead to a fixation of favourable solution characteristics and even in randomly generated solutions precedence relations are not uniformly distributed (Mattfeld *et al*. 1998).

This work aims to explore the research direction within Π_J examining the origins of the problem; its current standing as well as providing suggestions for areas that deserve investigation. The paper is organised as follows: after detailing the history of the problem, attention is focused on principal techniques used in this field. The best solutions are then given for the 242 benchmark problems available in the literature. The paper then explores parameters that allow an improved comparison to be made between techniques and a computational comparison is provided on some of the best methods available. The paper is then concluded with pointers for future work.

2. THE PHASES OF Π_J HISTORY ~ THE TECHNIQUES APPLIED

Opinion in the Π_J field is mixed about who first proposed the job-shop problem in its current form. Roy and Sussman (1964) were the first to propose the disjunctive graph representation and Balas (1969) was the first to apply an enumerative approach based on the disjunctive graph. Nevertheless there were earlier works in job-shop scheduling. Giffler and Thompson (1960) proposed a priority dispatch rule template, Jackson (1956) generalised Johnsons's flow-shop algorithm (Johnson 1954) to the job-shop and Akers and Friedman (1955) applied a Boolean Algebra approach in order to represent processing sequences. These works dealt with a problem consisting of *n* jobs, *m* machines and precedence relationships. Thus each job in this problem is processed through the machines in a different order. The objective in all the above works is to complete all the tasks as quickly as possible. Not surprisingly these works cite even earlier references. For example Akers and Friedman (1955) cite a working paper from 1951 concerning industrial production (Salvesen and Anderson 1951). Although it is not clear who can be credited with first proposing Π_J, it is widely accepted that the book "Industrial Scheduling" edited by Muth and Thompson (1963), which collated all the major findings at that time, is the basis for most subsequently conducted research.

Figure 1 summarises the main techniques applied to solve Π_J and the category each technique belongs to, ie. whether it is an approximation or optimisation method and whether it is constructive, builds a solution from the problem data, or iterative, modifies a solution by continually reordering the sequence of operations. Also given are the benchmark problems on which these techniques have been applied to. Figures 2 and 3 summarise the main researchers, who have applied these various techniques to solve Π_J, and the most common representation and classification methods. Although the Gantt chart (cf. figure 5), described in Gantt (1919), Clark (1922) and Porter (1968), has traditionally been the most popular method of solution representation Blazewicz *et al*. (1996) indicate that the disjunctive graph model, $\mathcal{G} = \{\mathcal{N}, \mathcal{A}, \mathcal{E}\}$ (Roy and Sussmann 1964) is now more prevalent. We take this opportunity to describe the disjunctive graph representation in the following paragraphs before returning to the discussion on classification methods.

In this node-weighted graph there is a vertex for each operation where $\mathcal{N}$ is the set of nodes representing operations to be processed on the set of machines $\mathcal{M}$. Included within $\mathcal{N}$ are two special (fictitious) nodes ⓪ and ☆ which correspond to the initial and final operations respectively and are known as the source and sink: $\mathcal{N} = \{⓪, 1, 2, \ldots, ☆\}$. The positive weight

of each node j is equivalent to the processing time, τ_j, of the corresponding operation, where $\tau_{⓪} = \tau_{☆} = 0$. The starting and completion times of these nodes represents the start and finishing times of Π_J respectively. ⓪ is connected to the initial operation of each job and similarly the final operation of each job is connected to ☆.

$\mathcal{A}$ is the set of directed conjunctive arcs representing the precedence constraints for each job, such that $(i, j) \in \mathcal{A}$ indicates that operation i is an immediate predecessor of operation j ($i \prec j$) in the chain of operations of the job. Capacity constraints ensure that two operations i and j processed by the same machine cannot be executed simultaneously. Such operations belong to a set $\mathcal{E}$ of uni-directed but orientable edges where each member of $\mathcal{E}$ is associated with a pair of disjunctive arcs, requiring a common machine, such that $[i, j] = \{(i \prec j), (j \prec i)\}$ and $\{i, j \in O\}$. An example of a 4×3 disjunctive graph is provided in figure 4 which represents the problem defined in table 1. The conjunctive arcs, which are members of $\mathcal{A}$, are shown by complete lines while the dotted and dashed lines represent edges in the disjunctive set $\mathcal{E}$. Extensions and limitations of this model are provided by White and Rogers (1990) who indicate that applying this model to industrial situations can be difficult as parallel processing and indefinite cyclical process flows cannot be modelled directly. Nevertheless the same authors (White and Rogers 1990) succeeded in extending the disjunctive graph model to represent regular performance criteria as well as assembly and disassembly operations, due dates, schedule maintenance, ready times, priority jobs, certain sequence-dependent set-up times and material handling delays.

The most popular method of classifying scheduling problems is the four field notation ($A/B/C/D$) of Conway *et al*. (1967): A is the number of jobs, B is the number of machines, C is the flow pattern within the machine shop and D is the performance measure by which the schedule is evaluated. While this descriptive technique is suitable for basic problems, when non basic problems (involving pre-emption, dependent jobs, etc) require classification then the three field notation ($\alpha/\beta/\gamma$) of Graham *et al*. (1979) is more appropriate: α is the flow pattern and the number of machines, β is the constraints on the jobs and γ is the scheduling criteria. MacCarthy and Liu (1993) indicate that the four field technique has been widely used and is familiar to most scheduling researchers. Consequently they propose a combination of the two methods where the C field is modified to take into account non basic models.

It can be safely assumed that the work on Π_J was initiated by Johnson (1954) which is believed to be one of the earliest works in scheduling theory. An optimal algorithm for a two machine flow-shop was developed in this research. Then other efficient methods, solvable in

polynomial time, for the job-shop problems of the sizes $2\times m$ (Akers 1956); $n\times 2$ (where there are no more than 2 operations per job) (Jackson 1956) and $n\times 2$ (where all operations are of unit processing time) (Hefetz and Adiri 1982) have been proposed. In general one of the most productive periods in the domains of Management Science and Operations Research was the 1950s. During this time many problems were solved by the application of crude but effective heuristics which formed the basis of the development of classical scheduling theory.

During the 1960s the emphasis shifted and was directed at finding exact solutions by the application of enumerative algorithms which adopted more elaborate and sophisticated mathematical constructs. Although these strategies are of tremendous theoretical value providing a significant research accomplishment, as Rinnooy Kan (1976 p36) indicates "*a natural way to attack scheduling problems is to formulate them as mathematical programming models*", the results from many of these works are extremely disappointing. The majority of these techniques are unable to achieve feasible solutions to many problems and are therefore of limited practical use. Consequently they are only applied in the calculation of lower bounds.

The main enumerative strategy is Branch and Bound (BB) where a dynamically constructed tree representing the solution space of all feasible schedules is implicitly searched. This technique formulates procedures and rules to allow large portions of the tree to be removed from the search and for many years it was the most popular Π_J technique. Although this method is very suitable for instances with $N < 250$ its excessive computing requirement prohibits its application to problems of interest. In addition their performance is quite sensitive to individual instances and initial upper bound values (Lawler *et al*. 1993). Current research emphasises the construction of improved branching and bounding strategies and the generation of more powerful elimination rules in order to remove large numbers of nodes from consideration at early stages of the search.

During the 1970s and until the mid 1980s the emphasis was placed on justifying the complexity. Numerous works spanning from Cook (1971) to Lawler *et al*. (1982) clearly highlighted the extreme intractability of Π_J demonstrating that only a very few special instances are solvable in polynomial time. As a result Parker (1995) refers to the era before 1970 as BC (Before Complexity). Consequently we designate the era since then as AD (Advanced Difficulty). The AD research soon showed that the problems which were solved as a specific instance in the 1950's as well as many others are $\mathcal{NP}$-Hard problems if they are generalised. Hence not surprisingly Garey and Johnson (1979) cite 320 $\mathcal{NP}$-Hard problems.

Due to the general limitation of exact enumeration techniques, approximation methods became a viable alternative. While such methods forego guarantees of an optimal solution for gains in speed they can be used to solve larger problems. The earliest approximation algorithms are priority dispatch rules (pdrs). These construction techniques assign a priority to all the operations which are available to be sequenced and then choose the operation with the highest priority. They are very easy to implement and have a low computation burden. A plethora of different rules have been created (Panwalkar and Iskander 1977) and the research applied in this domain indicates that the best techniques involve a linear or randomised combination of several priority dispatch rules (Fisher and Thompson 1963; Panwalkar and Iskander 1977; Lawrence 1984). Two of the more novel approaches involve Fuzzy Logic (Grabot and Geneste 1994) and Genetic Local Search (Dorndorf and Pesch 1995). Nevertheless these works highlight: the highly problem dependent nature of pdrs, as in the case of makespan minimisation no single rule shows superiority; their myopic nature in making decisions, as they only consider the current state of the machine and its immediate surroundings and that solution quality degrades as problem dimensionality increases.

By the end of the 1980s the full realisation of this advanced difficulty resulted in the main research focus being diverted away from emphasising the $\mathcal{NP}$-Hard status of Π_J and creating optimisation techniques. Instead attention concentrated on solving this problem by the application of approximation methods. However due to the general deficiencies exhibited by priority dispatch rules there was a growing need for more appropriate techniques which apply a more enriched perspective. Work on such applications was initiated by Fisher and Rinnooy Kan (1988) who emphasise the fundamental properties of generating good heuristic techniques: design, analysis and implementation. The suitablity of these techniques is further highlighted by the Committee On the Next Decade of Operations Research (CONDOR 1988) who indicate that they are extremely promising. While the survey of Rodammer and White (1988) emphasises the flexibility and strength of such heuristic approaches in comparison with optimisation techniques. They indicate innovative heuristic search procedures are legitimate application tools for not just Π_J but more real world scheduling problems. Not surprisingly with this new research emphasis substantial progress was made and in the short period of 1988 to 1991, which we shall refer to as the boom period, some of the most innovating algorithms were formulated.

Glover and Greenberg (1989) indicate the emergence of heuristic strategies to difficult problems, such as Π_J, that are inspired by natural phenomena and intelligent problem solving. Although research in the boom period was mainly focused at approximation methods some effort was still being directed at optimisation methods, however applying principles derived from these new heuristic strategies. For example Carlier and Pinson (1989) prescribed a BB method which proved for the first time the optimal solution to FT 10 (see §3.1), a problem that had been eluding researchers for over 25 years. Even larger instances than FT 10 were solved by other BB methods (Carlier and Pinson 1990; Applegate and Cook 1991). Prior to 1988 the only technique available to solve instances with more than 100 operations was priority dispatch rules (Lawrence 1984).

Fox (1987) and Sadeh (1991) were tackling problems from an industrial perspective using Constraint Satisfaction techniques while neural network (NN) applications (Foo and Takefuji 1988a, b, c; Zhou *et al*. 1990, 1991) provided instantaneous solutions to problems as large as 20×20.

Examples of some of the innovative algorithms formulated during the late 1980s and early 1990s to solve Π_J include Large Step Optimisation (Martin *et al*. 1989); Tabu Search (TS) (Glover 1989, 1990; Taillard 1989); Simulated Annealing (SA) (Matsuo *et al*. 1988; Van Laarhooven *et al*. 1988; Aarts *et al*. 1991); Genetic Algorithms (GAs) (Falkenauer and

Bouffouix 1991; Nakano and Yamada 1991) and Genetic Local Search (GLS) (Moscato 1989; Aarts *et al*. 1991) which are described later in this paper. The main contribution of these works is the notion of local search (Johnson *et al*. 1988; Yannakakis 1990) to Π_J. In local search methods a metastrategy is able to guide a myopic algorithm to optimality by accepting non improving solutions. The works of Van Laarhooven *et al*. (1988) and Matsuo *et al*. (1988), which were derived from Balas (1969), and Grabowski *et al*. (1988) laid the foundations for powerful Π_J neighbourhood structures which use local search techniques that are respectively based on the permutation of critical operations and the notion of blocks.

The boom period was started by the technique that has had the greatest influence on approximation methods and was the first heuristic to solve FT 10, the Shifting Bottleneck Procedure (SBP) (Adams, Balas and Zawack 1988). One of the main reasons this constructive algorithm achieves good results is due to the well developed algorithms and software for the one-machine scheduling problem. SBP is characterised by the following tasks: Subproblem identification, bottleneck selection, subproblem solution, and schedule reoptimisation (Demirkol *et al*. 1997). The actual strategy involves relaxing the problem into *m* one machine problems and solving each subproblem one at a time. Each one machine solution is compared with all the others and the machines are ranked on the basis of their solution. The machine having the largest lower bound is identified as the bottleneck machine. SBP sequences the bottleneck machine first, with the remaining unsequenced machines ignored and the machines already scheduled held fixed. Every time the bottleneck machine is scheduled each previously sequenced machine susceptible to improvement is locally reoptimised by solving the one machine problem again. The one machine problem is iteratively solved using the approach of Carlier (1982) which provides an exact and rapid solution. The main contribution of this approach is the way the one machine relaxation is used to decide the order in which machines should be scheduled. A comprehensive computational analysis of this method is provided by Holtsclaw and Uzsoy (1996) and Demirkol *et al*. (1997) and a thorough review is given by Alvehus (1997).

Nevertheless the fundamental problem with SBP is the difficulty in performing reoptimisation and the the generation of infeasible solutions. Dauzère-Pérès and Lasserre (1993) and Balas *et al*. (1995) note substantial differences in the results of Adams *et al*. (1988) when applying four rather than three reoptimisation cycles. This is because the work of Adams *et al*. (1988) does not take into account delayed precedence constraints (DPCs), which are relations between non independent operations on the same machine. As a result Dauzère-Pérès and Lasserre (1993) propose a heuristic strategy while Dauzère-Pérès (1995) and Balas *et al*.

(1995) utilise an exact scheme to deal with DPCs. The most recent work (Balas and Vazacopoulos 1998) amalgamates guided variable search with SBP producing one of the best Π_J approaches available. Eventhough Balas and Vazacopoulos (1998) suggest several elaborate reoptimisation schemes as yet there is no strategy to indicate how this should be done. In addition no method is available to decide the size of the subproblem or which machine(s) to fix and in order to solve problems where job routings are more structured SBP will need to be modified.

Nevertheless as SBP generated much of the initial excitement in the boom period then not surprisingly this procedure has been incorporated in many other works (Causeau and Laburthe 1995; Yamada and Nakano 1996a; Vaessens 1996; Balas and Vazacopoulos 1998) which have improved the upper and lower bounds of several hard problems. SBP has also been applied to many generalisations of Π_J. For example Morton (1990) extends SBP to project scheduling and applies several different performance criteria. Ovacik and Uzsoy (1992, 1996) apply this technique to a semi conductor testing facility. Ramudhin and Marier (1996) adapt the procedure to assembly, dag and open shop applications while Ivens and Lambrecht (1996) extend SBP to deal with a variety of parameters such as set up time and parallel machines. The most recent generalisation is by Balas *et al*. (1998) who combine their exact one machine approach with their guided local search procedure and apply this to the job shop problem with deadlines.

As indicated earlier the euphoria generated in the boom period resulted in the creation and formalisation of many approximation methods which shall now be described.

The insertion algorithm of Werner and Winkler (1995) consists of two phases. The first phase applies a constructive strategy in which an operation is positioned in the schedule such that it minimises the length of the longest path passing through it. The second phase applies a reinsertion strategy to iteratively improve the initial solution. In both phases beam search (Morton and Pentico 1993) is applied to improve the search. Beam search incorporates heuristics into the search preventing the need to follow every possible selection from a given point thereby trying to keep the combinatorial nature of the problem in check. The search is performed in a breadth wise fashion but without backtracking.

If a beam width of β is applied then at each step the β best possibilities of a given partial schedule are chosen for further consideration. This allows a limited number of partial sequences to be evaluated further. A filter width α can also be introduced such that a partial evaluation is made of all β descendants and then α of the β most promising descendants are evaluated fully to determine the beam node. Once the beam node is selected the process then

continues by selecting the β best descendants for further evaluation and finishes once a complete selection is made. Such an approach is similar to the fan candidate list strategy in tabu search (Glover and Laguna 1997) which also allows a limited number of solutions to be searched in parallel. As in BB strategies best descendants are chosen based on lower bound calculations. A filtered beam search strategy has been superimposed by Sabuncuoglu and Bayiz (1997) on selection decisions made by a set of pdrs and results indicate that their technique provides substantial improvement over the results achieved purely by pdrs.

Constraint Satisfaction techniques are examples of iterative approximation methods which apply many of the rules and strategies used in branch and bound algorithms. They aim at reducing the effective size of the search space by applying constraints that restrict the order in which variables are selected and the sequence in which possible values are assigned to each variable. Although these techniques are concerned with trying to achieve a feasible schedule such that the overall deadline can be met many of these methods have difficulty representing constraints and require excessive backtracking. As a result they commonly converge to dead end states. Despite the ebullience witnessed in the boom period generally poor results have been achieved by these methods while the computational effort required is high (Caseau and Laburthe 1995; Pesch and Tetzlaff 1996; Nuijten and Le Pape 1998). Similar conclusions are made about neural networks, which provide the capability to perform distributed processing using a simple but massively interconnected structure of parallel processing units, and only the work of Sabuncuoglu and Gurgun (1996) has been successfully applied to the benchmark problems.

Other iterative methods of note are problem-space methods which generate many different starting solutions, using fast problem-specific constructive procedures, which are then improved by local search. This class of techniques include problem and heuristic space based search (Storer *et al*. 1992, 1995) and greedy random adaptive search procedure (GRASP) (Resende 1997).

The most popular iterative methods are Threshold Algorithms which choose a new configuration if the difference in cost between the current solution and a neighbour is below a given threshold. Members of this group of algorithms include iterative improvement, simulated annealing and threshold acceptance. Due to the limitations of iterative improvement and the relative infancy of threshold acceptance, simulated annealing is the most popular technique in this category and has been applied extensively to Π_J. In simulated annealing (SA) the thresholds are positive and stochastic. SA is a random oriented local search technique that was introduced as an analogy from statistical physics of the computer simulation of the annealing

process of a hot metal until its minimum energy state is reached. However as SA is a generic technique it is unable to quickly achieve good solutions to Π_J problems. As a result research is currently directed at combining SA with other methods in order to improve results and reduce the required computing time. Amalgamation with critical neighbourhoods and active schedule generation (Yamada and Nakano 1995a, 1996a) and genetic algorithms (Kolonko 1998) has made SA competitive with other methods with respect to solution quality but they still require excessive computational effort.

While SA is based on principles of physical science, Genetic Algorithms (GAs) are search techniques based on an abstract model of natural evolution such that the quality of individuals builds to the highest level compatible with the environment (constraints of the problem). Despite the fact that many elaborate schemes have been proposed GAs are unable to successfully represent Π_J and crossover operators cannot generate feasible schedules without losing their efficiency. In addition many GA methods are unable to converge to an optimal solution. These deficiencies initiated the work on Genetic Local Search (GLS), which is also referred to as population based local search or memetic search (Grefenstette 1987; Moscato 1989; Ulder *et al*. 1990, 1991). GLS is a two level local search technique where a child conceived from the GA operators is used as the initial solution for the subsequent local search. The local search directs the offspring to the nearest locally optimal point which is operated on in the next generation by the traditional genetic recombination operators. The most recent works (Mattfeld 1996, Yamada and Nakano 1996b, c) have been able to overcome representation and crossover difficulties as well as poor solution convergence suffered by traditional GAs.

Large Step Optimisation, as GLS, is an example of a bilevel local search technique. Developed by Martin, Otto and Felten (1989, 1992) it encompasses a dual phase optimisation method consisting of a large optimised transition (large step) and then a local search method (small step). Small steps are most commonly performed by iterative improvement or simulated annealing. Large steps involve the application of problem specific techniques allowing local minima to be transcended even at low temperatures. It is a relatively new technique and has only been applied to Π_J by Lourenço (1993, 1995) and Lourenço and Zwijnenburg (1996). The limited analysis indicates that this technique has a very high computational requirement but provides better results than if only the local search method is applied. A similar type of bilevel strategy has also been applied by Brucker *et al*. (1996a, 1997a), to a wider domain of scheduling results achieving some promising results.

Tabu Search (TS) proposed by Glover (1977, 1986, 1989, 1990) is an iterative approximation approach which has led to several successful formulations for Π_J. TS is a simple technique that intelligently guides a search process away from solutions that (based on available information) appear to duplicate or resemble previously achieved solutions. More elaborate schemes can be applied to intensify the search in areas which have historically been good or diversify the search to unexplored regions of the solution space. The technique of Nowicki and Smutnicki (1996) is currently one of the most powerful TS approaches allowing good solutions to be achieved very quickly.

It is clearly evident that work appears to have gone full circle returning to the heuristics notion of the 1950s however with some novel and elaborate modifications. Currently the most dominant methods are of hybrid construction and transcend poor minima by integrating problem specific local heuristics with general metasolvers. To underline the fact that work continues to follow the same cyclic phases, research has currently returned to complexity analysis and the construction of polynomial time ρ-approximation algorithms that guarantee solutions of ρC^*_{max}. For example, Shmoys *et al.* (1994) have constructed an algorithm with a worst case performance guarantee of $(\prod_{max} + (m\text{-}1)\tau_{max})$, where $\prod_{max} = max_{\mathcal{M}_k} \sum_k \tau_{ik}$. The most recent contribution in this area (Williamson *et al.* 1997) prove that Π_J is difficult to solve even approximately as determining the existence of a schedule with a fixed length of four is hard, ie. $\mathcal{NP}$-Complete, even in the case that all processing times are one and for any $\rho < {}^5/_4$ there does not exist a polynomial time ρ approximation algorithm for Π_J unless $\mathcal{P} = \mathcal{NP}$.

Although the success achieved until now is limited, even with the tremendous excitement generated from the boom period, with the superior technology available and improved understanding of the intractability of Π_J a solution to this problem does not appear to be far away. Encouragement should also be taken from the fact that many hard problems have been insurmountable for decades before finally being solved. A perfect example is Hilbert's 13th problem (Hilbert 1900) which was eventually solved by Kolmogorov (Kolmogorov 1957).

3. BENCHMARK PROBLEMS

In the previous section we have briefly described various techniques and algorithms which are used to solve Π_J. However to find the comparative merits of these techniques and algorithms they need to be tested on the same problems. Hence the birth of "benchmark problems" which provide a common standard on which all Π_J algorithms can be tested and

compared. Hence it will be easy to gauge the strength and power of various algorithms from a statement such as "Algorithm *A* achieved a makespan of *x* in time *y* on benchmark problem *B*." As the benchmark problems are of different dimensions and grades of difficulty it is simple to determine the capabilities and limitations of a given method by testing it on the benchmark problems. Also the test findings may suggest the improvements required and where they should be made.

These benchmark problems are formulated by various authors (Fisher and Thompson 1963 (FT); Lawrence 1984 (LA); Adams *et al*. 1988 (ABZ); Applegate and Cook 1991 (ORB); Storer *et al*. 1992 (SWV); Yamada and Nakano 1992 (YN); Taillard 1993 (TA); Demirkol *et al*. 1998 (DMU)) and are freely available.[2] The processing times for these problems are formulated randomly within a given interval (cf. §3.2) from a generator constructed by the authors. In order to create the various problems the generator is initiated from different seed values. Hence given the seed values and the interval of the processing times these problems can be generated by anyone. This section describes all the known benchmark problems for Π_J, 242 of them (Beasley 1990, 1996), highlighting optimal or, in the case of open problems, best solutions and the first author(s) to achieve these bounds. Attention is initially drawn to FT 10, the most famous benchmark instance.

3.1 FT 10

The benchmark problems which have received the greatest analysis are the instances generated by Fisher and Thompson (Fisher and Thompson 1963 p236): FT 06, 6×6; FT 10, 10×10; FT 20, 20×5. While FT06 and FT20 had been solved optimally by 1975, the solution to FT 10 remaned elusive until 1987. Florian *et al*. (1971) indicate that their implementation of the algorithm of Balas (1969) is able to achieve the optimum solution for FT 06. FT 20 however required 12 years to be solved optimally (M^{c}Mahon and Florian 1975) while FT 10 emphasises the difficulty involved in solving Π_J.

FT 10, as generated by Fisher and Thompson (1963) is given in table 2. Table 3 highlights the historical progress of solutions to this instance emphasising the tremendous computational effort directed at the problem. Even though solutions for FT 10 have gradually improved over the years because of the steady progress made by various algorithms, only after 26 years was it proved that the optimal makespan is 930 (figure 5) (Carlier and Pinson 1989).

[2] The FT, LA, ABZ, ORB, SWV and YN problems are available from anonymous ftp site ftp://mscmga.ms.ic.ac.uk/pub/jobshop1.txt while the TA problems are available from ftp://mscmga.ms.ic.ac.uk/pub/jobshop2.txt. Both sites are at the Operations Research Library of the Management School, Imperial College, London, UK. The DMU problems are available from Professor Reha Uzsoy at his Electronics Manufacturing Research

Perregaard and Clausen (1995) indicate that the success of Carlier and Pinson (1989) is realised by the algorithm's ability to add many directed arcs, to the subproblems, between successive branchings. One of the fundamental reasons FT 10 took so long to solve is the large gap, 15 %, between the one machine lower bound of 808 and the optimal makespan. Pesch and Tetzlaff (1996) also note that there is one critical edge linking O_{13} and O_{66} which if wrongly orientated will not allow optimum to be achieved. The best makespan that can be achieved when $O_{13} \prec O_{66}$, even if all the other edges are orientated correctly, is 937. Pesch and Tetzlaff (1996) also highlight the importance of this edge by showing that if this disjunction is fixed then the algorithm of Brucker, Jurisch and Sievers (1994) is able to solve FT 10 within 448 secs on a PC 386 while if no edges are orientated the algorithm takes 1138 secs. Lawler *et al*. (1993) report that within 6000 CPU seconds when applying a deterministic local search to FT 10 more than 9000 local optima have been generated with a best makespan value of 1006. Thereby further emphasising the difficulty and hardness of this problem.

3.2 Other benchmark problems

When defining benchmark problems the sequence of machines for each job is randomly generated. The processing time for each operation is also generated randomly and is drawn from a discrete uniform distribution (except for the ORB instances) and the objective in each problem is to minimise the makespan.

FT – 3 problems of 3 different sizes due to Fisher and Thompson (1963): 6×6, 10×10, 20×5. (The name FT has been given by Applegate and Cook 1991). The processing times are generated from the interval [1, 10] for FT 06 and the interval [1, 99] for FT 10 & FT 20. In the latter two instances in order to simulate a typical machine shop, lower numbered machines are assigned for earlier operations and higher numbered machines for later operations.

LA – 40 problems of 8 different sizes due to Lawrence (1984): 10×5, 15×5, 20×5, 10×10, 15×10, 20×10, 30×10, 15×15. (Lawrence 1984 respectively named the instances as F1-5, G1-5, H1-5, A1-5, B1-5, C1-5, D1-5, I1-5. However the name LA as given by Applegate and Cook 1991 is the one more commonly used). The processing times are generated from the interval [5, 99].

ABZ – 5 problems of 2 different sizes due to Adams, Balas and Zawack (1988): 10×10, 20×15. (The name ABZ has been given by Applegate and Cook 1991). The processing

---

Group in the Department of Engineering, School of Engineering, Purdue Universitiy, Indiana, USA. The URL is http://gilbreth.ecn.purdue.edu/~uzsoy2/benchmark/problems.html.

times for ABZ 5, ABZ 6 and ABZ 7-9 are generated from the intervals [50, 100], [25, 100] and [11, 40] respectively.

ORB – 10 problems used by Applegate and Cook (1991): 10×10. (They were formulated in Bonn in 1986 and are characterised as specially created "tougher" problems. Consequently some of these instances are referred to by their creators as doomed, deadlier, etc. The name ORB has been given by Applegate and Cook 1991).

SWV – 20 problems of 4 different sizes due to Storer, Wu and Vaccari (1992): 20×10, 20×15, 50×10, 50×10. (The name SWV has been given by Vaessens 1996). The processing times are generated from the interval [1,100]. These problems are considered respectively as: Hard, Hard, Hard, Easy and in these problems the set of machines are divided into k ($1 \leq k \leq m$) equally sized subsets. A machine sequence for a job is generated by passing all machines of one set using uniformly distributed assignments before it turns into a machine of the next set. For the easy set $k = 1$ while for the harder instances $k = 2$ (cf. Fisher and Thompson 1963).

YN – 4 problems due to Yamada and Nakano (1992): 20×20. (The name YN has been given by Vaessens 1996). The processing times are generated from the interval [10, 50].

TD – 80 problems of 8 different sizes due to Taillard (1993): 15×15, 20×15, 20×20, 30×15, 30×20, 50×15, 50×20, 100×20. (The name TD has been given by Balas and Vazacopoulos 1998). The processing times are generated from the interval [1, 99]. Note Taillard has also generated 120 Flow-shop problems and 60 Open-shop problems.

DMU – 80 problems of 8 different sizes due to Demirkol, Mehta and Uzsoy (1998): 20×15, 20×20, 30×15, 30×20, 40×15, 40×20, 50×15, 50×20. (The name DMU has been given by us. Also given in table 11 are the names used by the Demirkol *et al*. 1998) The processing times are generated from the interval [1, 200]. Two sets of problems are constructed. For the first 40 problems ($J//C_{max}$) $k = 1$ while for the second set of 40 problems ($J/2SETS/C_{max}$) $k = 2$, hence the name 2SETS (cf. Fisher and Thompson 1963 and Storer *et al*. 1992). Note Demirkol *et al*. (1998) have also generated 320 Job-shop problems with the aim to minimise maximum lateness (L_{max}) as well as 40 Flow-shop problems ($k = m$) with an objective of C_{max} and 160 Flow-shop problems with an objective of L_{max}.

The authors have run the algorithm of Nowicki and Smutnicki (1996) (NS'96) on the problems of Demirkol *et al*. (1998) and the best upper bound achieved from three runs of (NS'96), using the same parameters as Nowicki and Smutnicki (1996), and Demirkol *et al*. (1997) (DMU'97) is presented. The lower bound achieved by (DMU'97) is compared with the lower bound technique of Taillard (1993) and if, and only if, it is better, credit of the best lower bound is given to (DMU'97). For (DMU'97) the lower bounds are achieved by relaxing the capacity constraints on all but one machine and solving the resulting single machine problem. Upper bounds are the best result achieved from three variations of SBP and five pdrs.

Although not compared here a testsuite of 60 benchmark problems has also been generated by Sadeh (1991). This testsuite consists of six groups of ten problems. Each instance is 10×5 incorporating either one or two bottleneck machines which have been formulated by the spreading of release and due dates and are always processed after the same number of steps.

It should be noted that all of these benchmark problems have only integer processing times with a rather small range. In real production scheduling environments the processing times need not be integers and all be from a given interval. As a result it is felt that benchmark problems have a negative impact in the sense of their true practical usefulness. However the analysis of Amar and Gupta (1986), who evaluated the CPU time and the number of iterations of an exact optimising, heuristic and guaranteed performance algorithm on real life and simulated data indicates that real life scheduling problems are easier to solve than simulated ones, regardless of the type of algorithm used.

Tables 4 through 11 indicate the time to achieve the specified upper bound for the benchmark problems. This is an important distinction and it does not necessarily mean that just because optimality has been attained it has been proved. For example Florian *et al*. (1971) report the experiment performed on FT 06 using the algorithm of Balas (1969) can reach a makespan of 55 after 41 iterations. However even after 200 iterations the algorithm is unable to prove optimality, therefore the experiment is stopped without the global proof being achieved.

4. CRITERIA FOR CLASSIFICATION OF HARD PROBLEMS

It appears peculiar that problems such as SWV 3-4 with only 200 operations, TD 3-9 with 225 operations and ABZ 8-9 with 300 operations are still open while instances such as TD 71-80 with 2000 operations were solved optimally within a relatively short period of time after being generated. Such a peculiarity, note Matsuo *et al*. (1988) and Taillard (1989, 1994), is indicative of a general tendency for Π_J instances to become easier as the ratio of jobs to the

number of machines becomes larger (greater than 4 times). Ramudhin and Marier (1996) observe that when $n > m$ the coefficient of variation of the work load increases making it easier to select the bottleneck processor, thus reducing the possibility of becoming trapped in a local minima. This is why solutions for TD 71-80 can be easily attained where ($n = 5m$) and it is also found that if the number of machines is small the instance is further simplified as is highlighted by both Taillard (1994) and Adams *et al*. (1988). Taillard (1994) is able to provide optimal solutions in polynomial time for problems with 1,000,000 operations as long as no more than 10 machines are used, in other words the ratio of jobs to machines is of the order 100,000 : 1 while Adams *et al*. (1988) solved LA 11-15 (20×5) and LA 31-35 (30×10) using the earliest elegant heuristic method. Further it is worth noting that for many easier problems several local minima equal the global optimum.

However once the size of the problem increases and the instance tends to become square in dimensionality ie. $n \rightarrow m$ it is much harder to solve. Caseau and Laburthe (1995) indicate that for LA 31-35 (30×10) optimality can be easily proved while for LA 21 (15×10) and LA 36-40 (15×15) it requires much effort and YN 1-4 (20×20) cannot be solved. Thus highlighting the difficulties of instances which have a similar number of jobs and machines. In addition as LA 31-35 ($n = 3m$) and TD 61 / 63-66 / 68-70 ($n = 2.5m$) are all solved optimally while no global minimum can be achieved for TD 62 / 67 ($n = 2.5m$) it suggests that an additional criteria for hard problems is that they have more than 10 machines.

Also harder benchmark problems are the ones which satisfy the 2SETS principle, $k = 2$, (Demirkol *et al*. 1998) eg. FT 10; SWV 1-15; DMU 41-80. In their SBP analysis Demirkol *et al*. (1997) note that 2SETS problems require one to two orders of magnitude higher computation than the standard problems. Mattfeld *et al*. (1998) also show that for $k = 1$ problems the relative error of a local optimum and a randomly generated solution are much closer to the best known bounds and the hamming distance between pairs of such solutions are much closer than compared with $k = 2$ problems, while the distribution of precedence constraints for $k = 2$ problems is larger and in these problems more variations of solution characteristics occur thus inducing a more plain landscape. As a result problems with $k = 2$ are considered especially hard.

In summary a problem can be considered hard if it has the following structure: $N \geq 200$ where $n \geq 15$, $m \geq 10$, $n < 2.5m$. The problem is made more intractable when $k = 2$ and in such a situation n need not be less than or equal to $2.5m$ (cf. SWV 11, 12 and 15). Credibility of this summary comes from the fact that all open problems conform to this structure.

The advent of more powerful computers and better techniques have shifted the difficulty barrier to instances of dimensionality 15. Even though the size of the limiting instances has only increased by 5 machines, from FT 10, the difficulty is highlighted by the fact that 15! is 360,360 times larger than 10!. Many methods which adopt unsophisticated techniques are unable to cope with this factorial increase. For example (Brucker *et al*. 1994) require to find the position of every preceding and succeeding operation on each critical block before branching. As a result for many large problems the search is terminated before each position can be checked as the computational requirements exceed the capacity of the machine, resulting in memory overflow and no valid result. Hence FT 20 could not be solved optimally within 3 days (Brucker *et al*. 1994). The poor performance is also attributed to the fact that such methods make poor initial choices in deciding which search path to follow and therefore get trapped in weak local minima.

Mattfeld *et al*. (1998) analyse the structure of the fitness landscape of Π_J with respect to how it appears for an adaptive search heuristic. They indicate that local optima are spread all over the fitness landscape, are located far away from each other, precedence relations are not uniformly distributed and easy instances possess a more mountainous landscape. Their analysis emphasises just how how hard Π_J is, especially in comparison to other combinatorial optimisation problems.

Pesch and Tetzlaff (1996) indicate that hard instances are categorised by a substantial difference between the optimal makespan of the subproblem and the makespan of this subproblem in an optimal schedule of the full problem, ie. a decomposition approach is unable to provide tight lower and upper bounds. These bounds are discussed in detail in the next section. Martin (1996) classifies hard instances into two categories. In the first category the lower bound is very tight or nearly tight to the optimal makespan but achieving the optimal upper bound is very difficult. LA 27 and LA 37 are members of this set. The second group exhibits the opposite characteristic in that obtaining the upper bound is relatively easy however the lower bound is far from the optimal makespan. LA 38 is an instance belonging to this set. It is clearly evident that the attainment of strong bounds is the crux of solving problems as good initial settings can make computation to optimality much faster.

4.1 Upper and Lower Bounds

The lower bound (LB) is the length of the longest path passing through a given partial sequence. As all operations have not been scheduled then the LB is an estimate of the makespan of a solution containing this partial selection and is the minimum value it can have. As more and more operations are sequenced this value tends towards C^*_{max} from below. The

strongest LBs are currently attained by Martin (1996) and Vaessens (1996) as highlighted in tables 5 through 10. Martin (1996) applies a technique called shaving where each operation is allocated a time window in which it can be processed and based on various rules and selections one or more time units is attempted to be removed (shaved) off the current makespan, by reducing the time windows of various operations. While Vaessens (1996) combines optimised versions of "Shuffle" and "Bottle" (Applegate and Cook 1991) with an initial solution derived from one of several techniques (Balas and Vazacopoulos 1998; Nowicki and Smutnicki 1996; Yamada and Nakano 1996a; etc). Although the values for the bounds are very good, both methods are computationally expensive. Therefore the main research challenge is to construct techniques that provide tighter lower bounds but within a reasonable amount of computing time (Applegate and Cook 1991).

The upper bound (UB) is the makespan of a full sequence and is the maximum value it can have. Vaessens *et al*. (1996) indicate that in general for most scheduling instances it is easier to obtain a good upper bound and in many algorithms it is the best solution achieved so far where any schedule with a makespan larger than this is disregarded. As improvements are continually made this value tends towards C^*_{max} from above and at optimum LB = UB. By deriving good bounds many solutions in the search domain can quickly be discarded and hence the search space can be reduced.

In the case of upper bounds the computational analysis (see §6) indicates that of the approximation techniques the approaches of Balas and Vazacopoulos (1998); Nowicki and Smutnicki (1996) and Yamada and Nakano (1996b, c) are the most effective. Balas and Vazacopoulos (1998) present a variable depth guided local search procedure (GLS), embedded within SBP, that applies an interchange scheme based on a local neighbourhood structure that reverses more than one disjunction at a time. Nowicki and Smutnicki (1996) apply a tabu search approach with critical block neighbourhoods which is combined with an elite solution recovery strategy and Yamada and Nakano (1996b, c) apply a genetic local search technique. Two parent schedules are generated as far apart as possible with respect to their disjunctive graph distance. A biased stochastic selection procedure then selects children which are closer to the second parent with respect to their disjunctive graph distance.

The results of these three methods on the benchmark instances suggest that Balas and Vazacopoulos (1998) have difficulty in dealing with the second category of problems, as defined by Martin (1996), because their method requires substantially more time to solve LA 38 in comparison to LA 27 and LA 37. Although Yamada and Nakano (1996b, c) achieve results for LA 38 faster than LA 27 they are unable to optimally solve LA 38 which indicates

that their method also has greater difficulty with the second category of problems. While Nowicki and Smutnicki (1996) find the first category of problems considerably harder as their method is unable to solve either LA 27 or LA 37 optimally. Therefore an effective technique should take aspects from Nowicki and Smutnicki (1996) and combine them with attributes from Yamada and Nakano (1996b, c) and Balas and Vazacopoulos (1998) such that both problem classes can be solved equally well within comparable times. We suggest a good strategy would consist of critical block neighbourhoods and a variable depth search combined with an elite recovery solution strategy. To allow for a diverse exploration of the solution space, search paths should be created between two elite schedules where the sequences generated could be biased towards one of the elite schedules.

5. COMPARISONS

Cherkassky *et al*. (1996) highlight some of the general difficulties encountered in comparing heuristic techniques. They suggest that it is more accurate to discuss comparisons between software implementations which is very much true for the comparisons of Π_J techniques as well. However when making such comparisons it is well known that some machines are obviously more powerful than others and can therefore achieve solutions faster. As a result the Computer-Independent Central Processing Unit (CI-CPU) time has been formulated (Vaessens 1995), CI-CPU = TF×CPU, which takes into account the capabilities of the machine used. TF is the transformation factor and is based on our interpretation of the performance comparisons made by Dongarra (1998). Some typical TF values are highlighted in table 12. Nevertheless these factors have to be interpreted with a great deal of care. For fair comparisons Cherkassky *et al*. (1996) also suggest:

- Careful specification of the goals, methodology and design of experiments
- The creation of fully or semi automatic methods so that they can be easily applied by non expert users. The only true way to remove the power of the method from the expertise of the creator is to ensure, when dealing with a problem, that no parameters require modification (fully automatic) or only a few parameters require adjustment by the user (semi-automatic).

The main issue is whether methods can be made semi-automatic without compromising their performance. For techniques such as GAs, TS, SA etc this is very difficult. Due to their structure many parameters need to be optimised and Johnson *et al*. (1988) indicate that the selection of these parameters is inherent to these techniques. Therefore if (semi-) automatic methods are to be encouraged then a generic set of parameters must be created. Cherkassky *et*

al. (1996) note that (semi-) automatism is achieved by relying on (compute-intensive) internal optimisation rather than user expertise to choose proper parameter settings.

In many situations comparisons become invalid as they are not performed from a balanced perspective. This is due to the fact that when a comparative study is given the approach constructed by the author(s) is presented with the optimal choice of parameters, etc. however, in many cases, the other approaches included in the study are not fully optimised. Hence equivalence of testing is not achieved. One such example is seen in the implementation of the SA technique of Van Laarhooven *et al*. (1992) by Hara (1995) in which he is unable to solve FT 10 & FT 20 due to poor parameterisation. However Van Laarhooven *et al*. (1992) themselves achieved optimality for both these instances. Therefore in order to avoid this problem, unless successful parametric optimisation can be achieved, the values and times quoted should be taken directly from the article itself. Another important attribute of fair comparisons is the ability to provide insight into the performance of the algorithm and from this insight further improvements can be made. For example many runs on a problem instance might be required to reveal interesting insights that (sometimes) contradict the findings from smaller studies.

Such concerns have also been voiced by Barr *et al*. (1995) who focus on the issues involved in designing computational experiments to test heuristics. Five key areas are identified and appropriate guidelines and concerns are given for each.

- Setting experimentation goals
- Choosing performance measures and factors
- Designing and executing the experiment
- Analysing the results and drawing conclusions
- Reporting the computational experiments

It is concluded that rigorous reporting of experimental studies is still in its infancy and much progress is needed to achieve the level of quality observed in more established experimental sciences. In addition the need for larger more accessible archives of benchmark problems, test beds and codes exist.

Hooker (1995) believes that current experimental studies of heuristics are inherently flawed. The two main reasons for this are, the empahsis on competition is fundamentally anti-intellectual and does not allow the sort of insight which in the long term provides better algorithms and competition diverts time and resources from productive investigation. To counteract this Hooker suggests a more scientific framework should be applied which involves carrying out more controlled experiments with emphasis on what to measure and how it

should be measured. He indicates the biggest area of concern are benchmark problems where it is commonly the case that problems begin to design the algorithms (note the case for FT10). Although Reeves (1993) indicates that heuristics can be measured by either analytical testing, empirical testing or statistical inference he concludes that the evaluation of heuristic performance lacks a unifying principle and developement has been piecemeal. In general the choice of a heuristic for a given problem instance is not clear and depends on several factors.

While we attempt to incorporate many of the suggestions presented it is important to note that care should be taken when interpreting these recommendations as they are not completely applicable to Π_J. For example the work of Cherkassky *et al*. (1996) is directed at the domain of sample comparisons using neural networks. Nevertheless these work are all of considerable interest and very appropriate for all optimisation fields not just this one.

6. COMPUTATIONAL COMPARISON

Applegate and Cook (1991) denote the seven problems which they could not solve: LA (21, 27, 29, 38); ABZ (7, 8, 9) as computational challenges as they are much harder than FT 10 and until recently their optimal solutions were unknown even though every algorithm had been tried on them. Of these seven instances ABZ 8 and ABZ 9 are still open. Vaessens *et al*. (1996) also indicate that LA (24, 25, 40) are challenging. As a result Vaessens *et al*. (1996) include these seven challenging instances of Lawrence as well as the remaining 15×15 problems LA (36, 37, 39), two smaller instances LA (2, 19) and FT 10 when comparing the performance of several methods. Balas and Vazacopoulos (1998) have also applied these 13 instances in their computational study of several techniques. Consequently as these problems provide a suitable comparative test bed tables 13 through 15 present, as far as they are available, the performance of several techniques on these instances.

Boyd and Burlingame (1996) also note that the instances quoted by Applegate and Cook (1991) are orders of magnitude harder than those which have been already solved. Consequently Boyd and Burlingame (1996) believe that solving such instances within 24 hours will require an algorithm based on a fundamentally new mathematical approach. This further highlights the deficiencies of exact enumerative methods as they require excessive computing effort. On the contrary Vaessens *et al*. (1996) believe that a good approximation approach should be able to solve all 13 instances within 100 CI-CPU secs with a total MRE of no more than 2 %.

As only limited computational results are available, ie. solutions do not exist for all thirteen instances, mean values are used and comparisons are made with the optimum solution just purely for those instances attempted. The study indicates that hybrid methods involving

TS, SBP, GLS and SA techniques are most dominant as these methods obtain solutions that are uniformly as good as or better than the best previously known solutions even for the majority of available benchmark instances. Especially good results with regards to solution quality and time are achieved by the SB-RGLS*k* technique (Balas and Vazacopoulos 1998). This method is able to attain an average MRE of 0.05 % in 999 CI-CPU secs, solving 11 out of the 13 problems optimally. The TS algorithm of Nowicki and Smutnicki (1996) is also able to provide good results solving 7 out of the 13 problems optimally. Although it has a slightly higher MRE, 0.20 %, than the SBP method it is considerably faster (nearly 10 times). Note the time presented for their algorithm does not include the initial solution. As the initial solution has a complexity of $O(n^3m^5)$ Aarts (1996) indicates that this is somewhat misleading, to put it midly.

The other TS strategies (Dell'Amico and Trubian 1993; Taillard 1994; Barnes and Chambers 1995), although not presented here, are all integrated with several other methods and are able to provide solutions of high quality within reasonable computing times. In comparison the remaining SBP techniques (Adams *et al.* 1988; Dauzère- Pérès and Lasserre 1993; Balas *et al.* 1995) are not competitive as they are not of hybrid formulation. Such conclusions are also made by Ovacik and Uzsoy (1992, 1996) and Holtsclaw and Uzsoy (1996) who show that the performance of SBP clearly improves when incorporated with local search. Without local search SBP is concluded to be no better than pdrs, while requiring much more computing time.

Although SA is not a powerful Π_J technique, hybrid SA approaches are providing strong competition to other methods and the work of Yamada and Nakano (1995a, 1996a) is able to solve 7 of the 10 instances attempted optimally with an average MRE and time of 0.08 % and 132,961 CI-CPU secs respectively. Pure GA implementations are also very poor. The GA method of Bierwirth (1995) which applies a generalised permutation representation is extremely weak only achieving an average MRE of 3.10 % in 11,445 CI-CPU secs. In addition no instance is solved optimally. However the GLS approach of Yamada and Nakano (1996b, c) successfully combining a simple crossover operator into local search has allowed significant advances to be made. Their technique optimally solves 6 out of the 8 instances attempted and achieves an average MRE of 0.17 % within 161,977 CI-CPU secs.

Other approximation techniques analysed such as priority dispatch rules, threshold accepting and iterative improvement have produced extremely poor results while some of the relatively newer approaches such as problem space methods, large step optimisation and neural networks have the potential to be promising techniques. However further analysis is required especially concerning the testing and tuning of parameter values and the application

of more problem specific information in order to make these methods more suitable for larger and harder problems. In the case of neural networks several authors conclude it is too early to make an assessment of their application to Π_J. Currently results have only been provided by Sabuncuoglu and Gurgun (1996) which are solely applied to small problems. In addition their method requires excessive effort even for these small instances where many other approaches can solve similar problems in secs on personal computers.

Although branch and bound approaches are able to produce good solutions, the test bed of problems used in the analysis is just at the limit of the problems which can be solved by these methods. Hence for instances of slightly higher dimensionality the BB algorithm is unable to cope with the size of the search tree generated and invariably memory overflow occurs. Even on this test set Perregarrd and Clausen (1995) are unable to solve LA27 and 29 due to these limitations. Martin (1996) is able to cope better with this test set and is able to solve all 13 instances, the only method to do so, in a less average time than Perregaard and Clausen (1995). Thereby suggesting that a time orientation representation merits more consideration. Neverthless for instances of greater dimensionality Martin (1996) invariably suffers with the same problems. In order to avoid such difficulties we suggest incorporating heuristics into the BB methods (cf. Nuijten and Le Pape (1998) and Thomsen (1997). These methods are able to achieve an average MRE of 0.12 and 0.05 % in 29,335 and 7,650 CI-CPU secs respectively).

In general it is evident that the best results (with respect to time and solution quality) are achieved from hybrid approximation algorithms, as they combine several methods.

7. HYBRID TECHNIQUES USING LOCAL SEARCH

From a general perspective, the solution to Π_J can be considered as a collection of local decisions concerning which operation to schedule next. Dorndorf and Pesch (1995) suggest that a framework should be constructed which navigates these local decisions through the search domain in order to determine a high quality global solution in a reasonable amount of time. In the framework local decisions made by myopic problem specific heuristics are guided beyond local optimality by an underlying metastrategy. Such hybrid methods are known as meta-heuristics or iterated local search algorithms. They cover a broad spectrum of techniques varying from those which are rather simple, such as the iterative improvement algorithm of Aarts *et al.* (1994), to others which are extremely elaborate, such as the variable depth guided local search approach of Balas and Vazacopoulos (1998). Vaessens *et al.* (1996) show that local search methods currently dominate all other techniques and Glover and Greenberg (1989) attribute their success to the fact that they are carefully tailored to take advantage of domain specific requirements.

Π_J meta-heuristics are based on the neighbourhood strategies developed by Grabowski *et al*. (1986), Matsuo *et al*. (1988), Van Laarhooven *et al*. (1988) and Nowicki and Smutnicki (1996). Each neighbourhood structure permutes critical operations which require processing on the same machine. In essence these strategies differ in which operations in the neighbourhood are swapped and how this exchange is performed. The strength of meta-heuristics comes from the fact that they allow perturbations to non improving schedules and hence are able to transcend local optima. Not surprisingly such methods are considered to be far superior to current BB and AI methodologies (Glover 1990; Nowicki and Smutnicki 1996). The prominence of iterated local search techniques is clearly brought out by the works of Laguna *et al*. (1991, 1993) in which a TS technique is compared with a BB approach. While solution quality is comparable, in some instances the TS technique can achieve results 80 times faster even though the BB approach is implemented on a computer that is twice as powerful. This time discrepancy is further emphasised as the system of Vaessens (1996) requires 219,570 CI-CPU secs to solve LA 21 optimally to a makespan of 1046 while Nowicki and Smutnicki require 10.08 CI-CPU secs to achieve a makespan of 1047. Hence it is evident that the computing requirements for these hybrid local search methods are substantially lower than BB methods.

Dorndorf and Pesch (1995) state that the outcome of an approach depends heavily on the underlying metastrategy. GLS and SA choose moves from a probabilistic learning scheme derived from natural selection and physical science respectively while SBP and TS adopt a deterministic learning method which embraces the aggressive orientation of selecting the best move within the limits of current restrictions (tabu moves for TS and evaluations and guideposts for the variable depth SBP). Hence the computational study of Vaessens *et al*. (1996) as well as the results documented here suggest TS and SBP provide better strategies to navigate the myopic heuristic through the search domain, thereby generating improved and in general faster solutions. Another reason for the superiority of TS and SBP, implies Glover (1995), is that a bad strategic decision is able to provide more information than a good random choice (such selections are made in SA and GAs). If one uses a strategic decision process rather than a random selection process to arrive at the solution then all the steps taken are known. Hence from a poor strategic decision one can determine the steps at which deficiencies occur and appropriate actions can be taken. However as a random decision provides no indication of the procedure applied to arrive at a particular solution such modifications cannot be made. Although BB and AI methods also use a strategic decision process Glover (1995) believes that,

unlike TS which permits responsive exploration, they are too rigid and are therefore unable to provide the required flexibility.

7.1 Limitations of local search

The primary deficiency of local search methods is that they are highly problem dependent requiring multiple runs of the algorithm, in order to obtain meaningful results, and several parameters have to be carefully selected to achieve good solutions. Johnson *et al*. (1989) suggest that applying these techniques is more of an art as many choices have to be made concerning parameter values which are not trivial and are quite often very poorly made by trial and error. In many situations determination of these values cannot even be realised after several experiments as successful implementation requires much experience.

Many local search methods are also dependent on the initial solutions (Applegate and Cook 1991; Lourenço 1993, 1995; Nowicki and Smutnicki 1996). Poor initialisation can lead to weak final schedules or excessive computing times. In many methods the termination criterion is the number of iterations and as a result these techniques do not stop when optimality is achieved. Hence a futile search of the domain space continues. The suggestion here is a termination criterion should be bilevel: LB and iteration based such that when either one is achieved the algorithm stops.

8. FUTURE WORK

Although a classification of hard and easy problems has been provided in this paper, scope for future work exists in understanding the reason a 20×10 instance is hard while a 30×10 instance is easy. Also problems of dimensionality $m > n$ and $m >> n$ need to be investigated because as yet there has been little analysis on instances of this structure.

Local search techniques have made considerable progress in the last 20 years and have quickly reached a certain degree of maturity. Even though extensive computational analysis of these techniques has been performed their overall powers and limitations are less than completely known. There is also no formal method to suggest effective ways of combining these techniques. For example Yamada and Nakano (1995a, 1996a) believe that the most appropriate location for SBP in their SA algorithm is during the acceptance / rejection phase however other researchers might decide that an alternative arrangement is more suitable. Hence there is a need for looking into how to assemble the components together and also conversely how to dismantle a hybrid model and recombine it in such a way that it becomes a new and more powerful search tool.

Due to the combinatorial nature of Π_J problems current methods have extreme difficulty in effectively searching the solution space for instances of dimensonality greater than

15. To improve existing techniques parallel approaches can be applied to simultaneously perform multiple searches of the neighbourhood. Unfortunately the most recent works applying such approaches have met with limited success (Taillard 1994; Perregaard and Clausen 1995; Boyd and Burlingame 1996). Hence techniques which are more naturally suited to such formulations, such as neural networks should be tried.

We consider that AI methods such as neural networks have yet to show their true potential. Current NN approaches to Π_J (Sabuncuoglu and Gurgun 1996) are only suitable for small instances and require excessive computing time. Nevertheless if a suitable technique is to be created it will be of hybrid, possibly NN based, construction. While the system developed should be robust emphasis will also be placed on flexibility. By exploiting its inherent parallel distributed processing capability the NN will quickly prune the vast search space and permit an embedded metastrategy to find the optimal schedule in the reduced solution domain.

Even though the above said goals are realisable in the long term there is much work which can be performed in the short term. One area that merits exploration is the creation of local search approaches that consist of three or more search levels. Another area of interest will be combining local search and exact enumerative techniques. This will provide suitable integration of optimisation and approximation methods such that convergence to strong LBs and proof of optimality are achieved from the exact part of the algorithm using a strong UB attained by the heuristic part of the algorithm. Results obtained by Caseau and Laburthe (1995), Thomsen (1997) and Nuijten and Le Pape (1998) indicate their potential. Glover (1994) also suggests the tremendous scope in using both LBs and UBs in order to find good solutions and he refers to this process as “ghost imaging”.

It is important to realise that strict adherence to the constraints and boundaries of a problem can prevent the investigation of surrogate approaches which do not precisely correspond to these rules. This is clearly highlighted by the poor performance of many approaches, especially those of an exact nature where extremely tight constraints are specified. However within the domain of TS the technique of strategic oscillation encourages passing through non feasible solutions in order to converge to better minima. The advantage of such an approach is that it executes moves that are less complex than might otherwise be. Moving outside the feasible boundary and returning from different directions also uncovers opportunities for improvement that are not readily attainable when a search is more narrowly confined.

9. CONCLUSIONS

Up to the boom period it is evident that research within Π_J has followed distinct phases. Originally much of the work was focused on the creation of heuristics, then the emphasis shifted to exact algorithms and complexity analysis before returning to the study of heuristics however from an enriched perspective. While some attention has returned to the issue of complexity and the derivation of efficient algorithms much of the focus has remained with these elegant heuristics. Hence current research is progressing in both aspects.

This paper has presented a concise overview of job shop scheduling techniques over the last 40 years and has not only led to an improvement in the understanding of Π_J, highlighting the main people, techniques and benchmark instances but it also has detailed the structure of difficult problems. In addition pointers for future work and directions for new applications and research are given where it is suggested that hybrid systems which incorporate metastrategies of the form of TS, SBP as the driving mechanism for NNs and other such AI paradigms, in the development of new applications and improved methodologies, are most suitable. Such models would therefore be appropriate not just for Π_J but also to solve production scheduling, network planning and other complex combinatorial problems. Although considerable progress has been made in recent years where FT 10 can now be solved in seconds there are still instances of just slightly larger dimensionality which are still open, emphasising the shear intractability of this problem. As a result within the domain of constraint optimisation problems, Π_J remains a prime challenge and one which will stimulate interest well into the next century.

REFERENCES

Aarts, E. H. L., Van Laarhooven, P. J. M., and Ulder, N. L. J., (1991), "Local search based algorithms for job-shop scheduling", Working Paper, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands.

Aarts, E. H. L., Van Laarhoven, P. J. M., Lenstra, J. K., and Ulder, N. L. J., (1994), "A computational study of local search algorithms for job-shop scheduling", *ORSA Journal on Computing* 6/2, Spring, 118-125.

Aarts, B. J. M. (1996) A Parallel Local Search Algorithm for the Job Shop Scheduling Problem, *Masters Thesis*, Department of Mathematics & Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands, April.

Aarts, E. H. L. and Lenstra, J. K. (eds) (1997) *Local Search in Combinatorial Optimization*, Wiley, Chichester.

Adams, J., Balas, E., and Zawack, D., (1988), "The shifting bottleneck procedure for job-shop scheduling", *Management Science* 34/3, March, 391-401.

Akers, S. B. Jr., and Friedman, J., (1955), "A Non Numerical Approach to Scheduling Problems", *Operations Research* 3, 429-442.

Akers, S. B. Jr, (1956), "A graphical approach to production scheduling problems", *Operations Research* 4, 244-245.

Alexander, S. M., (1987), "Expert system for the selection of scheduling rules in a job-shop", *Computers and Industrial Engineering* 12/3, 167-171.

Alvehus, M. (1997) The Shifting Bottleneck Procedure a Survey, Graduate Course Report, Linköping University, Linköping, Sweden, March.

Amar, A. D., and Gupta, J. N. D., (1986), "Simulated versus real life data in testing the efficiency of scheduling algorithms", *IIE Transactions* 18, March, 16-25.

Applegate, D., and Cook, W., (1991), "A computational study of the job-shop scheduling problem", *ORSA Journal on Computing* 3/2, Spring, 149-156.

Ashour, S., (1967), "A decomposition approach for the machine scheduling problem", *International Journal of Production Research* 6/2, 109-122.

Ashour, S., and Hiremath, S. R., (1973), "A branch-and-bound approach to the job-shop scheduling problem", *International Journal of Production Research* 11/1, 47-58.

Ashour, S., and Parker, R. G., (1973), "An out-of-kilter approach for machine sequencing problems", *Lect. Notes Econ. Math. Syst.* 86, 206-223.

Ashour, S., Moore, T. E., and Chin, K. -Y., (1974), "An implicit enumeration algorithm for nonpreemptive shop scheduling problem" *AIIE Transactions* 6/1, 62-72.

Balas, E., (1965), “An additive algorithm for solving linear programs with zero-one variables”, *Operations Research* 13, 517-546.

Balas, E., (1969), “Machine scheduling via disjunctive graphs: An implicit enumeration algorithm”, *Operations Research* 17, 941-957.

Balas, E., (1979), “Disjunctive programming”, in: P. L. Hammer, E. L. Johnson, and B. Korte, (eds.), *Discrete Optimisation II,* North Holland, Amsterdam, The Netherlands, 3-52.

Balas, E., (1985), “On the facial structure of scheduling polyhedra”, *Mathematical Programming Study* 24, 179-218.

Balas, E., Lenstra, J. K., and Vazacopoulos, A., (1995), “The one-machine problem with delayed precedence constraints and its use in job shop scheduling”, *Management Science* 41/1, Jan, 94-109.

Balas, E., Lancia, G., Serafini, P., and Vazacopoulos, A., (1998), “Job-shop scheduling with deadlines”, *Journal of Combinatorial Optimization* 1/4, 329-353.

Balas, E., and Vazacopoulos, A., (1998), “Guided local search with shifting bottleneck for job-shop scheduling”, *Management Science* 44/2, Feb, 262-275.

Baptiste, P., and Le Pape, C., (1995), “A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling,” *IJCAI’14 Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada.

Baptiste, P., Le Pape, C., and Nuijten, W. P. M., (1995), “Constraint-based optimization and approximation for job-shop scheduling,” *Proceedings of the AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems*, IJCAI-95, Montreal, Canada.

Barker, J. R., and McMahon, G. B., (1985), “Scheduling the general job-shop”, *Management Science* 31/5, May, 594-598.

Barnes, J. W., and Chambers, J. B., (1995), “Solving the job shop scheduling problem using tabu search”, *IIE Transactions* 27, 257-263.

Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C. and Stewart, W. R. (Jr) (1995) Designing and Reporting on Computational Experiments with Heuristic Methods, *Journal of Heuristics*, Fall, **1**(1), 9-32.

Bartusch, M., Möhring, R. H., and Radermacher, F. J., (1988), “Scheduling project networks with resource constraints and time windows”, *Annals of Oper. Res.* 16, 201-240.

Beasley, J. E., (1990), “OR-Library: distributing test problems by electronic mail”, *Journal of the Operational Research Society* 41/11, 1069-1072.

Beasley, J. E., (1996), "Obtaining test problems via internet", *Journal of Global Optimization* 8/4, 429-433.

Biegel, J. E., and Wink, L. J., (1989), "Expert systems can do job-shop scheduling: an exploration and a proposal", *Computers and Industrial Engineering* 17/1-4, 347-352.

Bierwirth, C., (1995), "A generalized permutation approach to job-shop scheduling with genetic algorithms", *OR Spektrum*, 17/2-3, 87-92.

Bierwirth, C., Mattfeld, D. C., and Kopfer, H., (1996), "On permutation representations for scheduling problems", in H. M. Voigt, *et al.*, (eds.), *PPSN'IV Parallel Problem Solving from Nature*, Springer-Verlag, Berlin, Heidelberg, Germany, 310-318.

Blackstone, J. H. Jr, Phillips, D. T., and Hogg, G. L., (1982), "A state of the art survey of dispatching rules for manufacturing job-shop operations", *International Journal of Production Research* 20, Jan-Feb, 27-45.

Blazewicz, J., Dror, M., and Weglarz, J., (1991), "Mathematical programming formulations for machine scheduling: a survey", *European Journal of Operational Research* 51/3, 15th April, 283-300.

Blazewicz, J., Domschke, W., and Pesch, E., (1996), "The job-shop scheduling problem: Conventional and new solution techniques", *European Journal of Operational Research* 93/1, 23rd August, 1-33.

Bowman, E. H., (1959), "The schedule-sequencing problem", *Operations Research* 7, 621-624.

Boyd, E. A., and Burlingame, R., (1996), "A parallel algorithm for solving difficult job-shop scheduling problems", Operations Research Working Paper, Department of Industrial Engineering, Texas A&M University, College Station, Texas 77843-3131, USA.

Bratley, P., Florian, M., and Robillard, P., (1973), "On sequencing with earliest starts and due dates with application to computing bounds for the ($n/m/G/F_{max}$) problem", *Naval Res. Logist. Quart*. 20, 57-67.

Brooks, G. H., and White, C. R., (1965), "An algorithm for finding optimal or near optimal solutions to the production scheduling problem", *Journal of Industrial Engineering* 16/1, January, 34-40.

Brucker, P., (1988), "An efficient algorithm for the job-shop problem with two jobs", *Computing* 40, 353-359.

Brucker, P., Jurisch, B., and Sievers, B., (1994), "A branch and bound algorithm for the job-shop scheduling problem", *Discrete Applied Mathematics* 49, 109-127.

Brucker, P., (1994), "A polynomial algorithm for the two machine job-shop scheduling problem with a fixed number of jobs", *OR Spektrum* 16, 5-7.

Brucker, P., Hurink, J., and Werner, F., (1996a), "Improving local search heuristics for some scheduling problems part I", *Discrete Applied Mathematics* 65/1-3, 7th March, 97-122.

Brucker, P., Kravchenko, S. A., and Sotskov, Y. N., (1996b), "Preemptive scheduling of the job-shop problems with the fixed number of jobs", Osnabrück. Schrift. Math. Reihe P: Preprints, Heft 184, 47 p.

Brucker, P., Hurink, J., and Werner, F., (1997a), "Improving local search heuristics for some scheduling problems part II", *Discrete Applied Mathematics* 72/1-2, 10th January, 47-69.

Brucker, P., Kravchenko, S. A., and Sotskov, Y. N., (1997b), "On the complexity of two machine job shop scheduling with regular objective functions", *OR-Spektrum* 19, 5-10.

Carlier, J., (1982), "The one-machine sequencing problem", *European Journal of Operational Research* 11, 42-47.

Carlier, J., and Pinson, E., (1989), "An algorithm for solving the job shop problem", *Management Science* 35/2, February, 164-176.

Carlier, J., and Pinson, E., (1990), "A practical use of Jackson's pre-emptive schedule for solving the job-shop problem", *Annals of Operations Research* 26, 269-287.

Carlier, J., and Pinson, E., (1994), "Adjustment of heads and tails for the job-shop problem", *European Journal of Operational Research* 78/2, October 27, 146-161.

Caseau, Y., and Laburthe, F., (1994), "Improved CLP scheduling with task intervals", in: P. Van Hentenryck, (ed.), *ICLP'94 Proceedings of the Eleventh International Conference on Logic Programming*, MIT Press.

Caseau, Y., and Laburthe, F., (1995), "Disjunctive scheduling with task intervals", LIENS Technical Report n° 95-25, Laboratoire d'Informatique de l'École Normale Supérieure Département de Mathématiques et d'Informatique, 45 rue d'Ulm, 75230 Paris, France.

Cedimoglu, I. H., (1993), "Neural networks in shop floor scheduling", *Ph.D. Thesis*, School of Industrial and Manufacturing Science, Cranfield University, U.K.

Chang, Y. L, Sueyoshi, T., and Sullivan, R. S., (1996), "Ranking dispatching rules by data envelopment analysis in a job-shop environment", *IIE Transactions*, 28/8, 631-642.

Charalambous, O., and Hindi, K. S., (1991), "Review of artificial intelligence based job-shop scheduling systems", *Inf. Decis. Technol.* 17/3, July, 189-202.

Charlton, J. M., and Death, C. C., (1970a), "A generalized machine scheduling algorithm", *Oper. Res. Quart.* 21/1, 127-134.

Charlton, J. M., and Death, C. C., (1970b), "A method of solution for general machine scheduling problems", *Operations Research* 18/4, 689-707.

Cheng, R., Gen, M., and Tsujimura, Y., (1996), "A tutorial survey of job-shop scheduling problems using genetic algorithms-I. Representation," *Computers & Industrial Engineering* 30/4, 983-997.

Cheng, C-C., and Smith, S. F., (1997), "Applying constraint satisfaction techniques to job shop scheduling", *Annals of Operations Research* 70, 327-357.

Cherkassky, V., Gehring, D., and Mulier, F., (1996), "Comparison of adaptive methods for function estimation from samples", *IEEE Transactions on Neural Networks* 7/4, July, 969-984.

Chu, C., Portmann, M. C., and Proth, J. M., (1992), "A splitting-up approach to simplify job-shop scheduling problems", *International Journal of Production Research* 30/4, 859-870.

Clark, W., (1922), *The Gantt Chart: A Working Tool of Management 3rd Edition*, The Ronald Press, Pitman, New York.

Colorni, A., Dorigo, M., Maniezzo, V., and Trubian, M., (1994), "Ant-system for job-shop scheduling", *Belgian Journal of Operations Research, Statistics and Computer Science* 34/1, 39-54.

CONDOR, Committee On the Next Decade of Operations Research (1988) Operations Research: The Next Decade, *Operations Research*, **36**(4), 619-637.

Conway, R. W., Maxwell, W. L., and Miller, L. W., (1967) *Theory of Scheduling*, Addison-Wesley, Reading Massachusetts.

Cook, S. A., (1971), "The complexity of theorem proving procedures", *Proceedings of the Third Annual ACM Symposium on the Theory of Computing*, Association of Computing Machinery, New York, 151-158.

Crowston, W. B., Glover, F., Thompson, G. L., and Trawick, J. D., (1963), "Probabilistic and parametric learning combinations of local job-shop scheduling rules", ONR Research Memorandum No. 117, Carnegie Institute of Technology, Pittsburgh, PA, USA.

Dagli, C. H., Lammers, S., and Vellanki, M., (1991), "Intelligent scheduling in manufacturing using neural networks", *Journal of Neural Network Computing Technology Design and Applications*, 2/4, 4-10.

Dagli, C. H., and Sittisathanchai, S., (1995), "Genetic neuro-scheduler - a new approach for job-shop scheduling," *International Journal of Production Economics* 41/1-3, 135-145.

Dauzère- Pérès, S., and Lasserre, J. B., (1993), "A modified shifting bottleneck procedure for job-shop scheduling", *International Journal of Production Research* 31/4, April, 923-932.

Dauzère-Pérès, S., (1995), "A procedure for the one machine sequencing problem with dependent jobs", *European Journal of Operational Research* 81, 579-589.

Davidor, Y., Yamada, T., and Nakano, R., (1993), "The ecological framework II: improving GA performance at virtually zero cost", *ICGA'5 5th International Conference on Genetic Algorithms*, 171-176.

Davis, L., (1985), "Job-shop scheduling with genetic algorithm", in: J. J. Grefenstette (ed.), *Proceedings of the 1st International Conference on Genetic Algorithms and their Applications*, Lawrence Erlbaum, Pittsburgh, PA, USA, 136-140.

Della Croce, F., Tadei, R., and Rolando, R., (1994), "Solving a real world project scheduling problem with a genetic approach", *Belgian Journal of Operations Research, Statistics and Computer Science* 33/1-2, 65-78.

Della Croce, F., Tadei, R., and Volta, G., (1995), "A genetic algorithm for the job shop problem", *Computers and Operations Research* 22/1, January, 15-24.

Dell'Amico, M., and Trubian, M., (1993), "Applying tabu search to the job-shop scheduling problem", *Annals of Operations Research* 41, 231-252.

Demirkol, E., Mehta, S. and Uzsoy, R. (1997) A Computational Study of Shifting Bottleneck Procedures for Shop Scheduling Problems, *Journal of Heuristics*, Winter, 3/2, 111-137.

Demirkol, E., Mehta, S. and Uzsoy, R., (1998), "Benchmarks for shop scheduling problems", *European Journal of Operational Research* 109/1, Aug 16, 137-141.

Dongarra, J. J., (1998), "Performance of various computers using standard linear equations software", Technical Report CS - 89 - 85, Computer Science Department, University of Tennessee, Knoxville, TN 37996-1301 and Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN 37831, Tennessee, USA, September 25.

Dorndorf, U., and Pesch, E., (1995), "Evolution based learning in a job-shop scheduling environment", *Computers and Operations Research* 22/1, January, 25-40.

Dyer, M. E., and Wolsey, L. A., (1990), "Formulating the single machine sequencing problem with release dates as a mixed integer program", *Discrete Applied Mathematics* 26, 255-270.

Erschler, J., Roubellat, F., and Vernhes, J. P., (1976), "Finding some essential characteristics of the feasible solutions for a scheduling problem", *Operations Research*, 24/4, Jul-Aug, (Technical Note), 774-783.

Evans, J. R., (1987), "Structural analysis of local heuristics in combinatorial optimization", *Computers and Operations Research* 14/6, 465-477.

Falkenauer, E., and Bouffouix, S., (1991), "A genetic algorithm for the job-shop", *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacremento, California, USA.

Fang, H. L., Ross, P., and Corne, D., (1993), "A promising genetic algorithm approach to job-shop scheduling, rescheduling and open-shop scheduling problems", *ICGA'5 5th International Conference on Genetic Algorithms*, 375-382.

Fisher, H., and Thompson, G. L., (1963), "Probabilistic learning combinations of local job-shop scheduling rules", in: J. F. Muth and G. L. Thompson, (eds.), *Industrial Scheduling*, Prentice Hall, Englewood Cliffs, New Jersey, 225-251.

Fisher, M. L., (1973a), "Optimal solution of scheduling problems using Lagrange multipliers: part I", *Operations Research* 21, 1114-1127.

Fisher, M. L., (1973b), "Optimal solution of scheduling problems using Lagrange multipliers: part II", *Symposium on the Theory of Scheduling and its Applications*, Springer, Berlin.

Fisher, M. L., Northup, W. D., and Shapiro, J. F., (1975), "Using duality to solve discrete optimization problems: Theory and computational experience", *Math. Programming Stu.* 3, 56-94.

Fisher, M. L., (1976), "A dual algorithm for the one-machine scheduling problem", *Math. Programming* 11, 229-251.

Fisher, M. L., Lageweg, B. J., Lenstra, J. K., and Rinnooy Kan, A. H. G., (1983), "Surrogate duality relaxation for job-shop scheduling", *Discrete Applied Mathematics* 5/1, 65-75.

Fisher, M. L., and Rinnooy Kan, A. H. G., (eds) (1988), "The design, analysis and implementation of heuristics", *Management Science* 34/3, Special Issue, March, 263-401.

Florian, M., Trépant, P., and McMahon, G., (1971), "An implicit enumeration algorithm for the machine sequencing problem", *Management Science Application Series* 17/12, August, B782-B792.

Foo, S. Y., and Takefuji, Y., (1988a), "Stochastic neural networks for solving job-shop

scheduling: Part 1. Problem representation", in: B. Kosko (ed.), *IEEE International Conference on Neural Networks*, San Diego, California, USA, 275-282.

Foo, S. Y., and Takefuji, Y., (1988b), "Stochastic neural networks for solving job-shop scheduling: Part 2. Architecture and simulations", in: B. Kosko (ed.), *IEEE International Conference on Neural Networks*, San Diego, California, USA, 283-290.

Foo, S. Y., and Takefuji, Y., (1988c), "Integer linear programming neural networks for job-shop scheduling", in: B. Kosko (ed.), *IEEE International Conference on Neural Networks*, San Diego, California, USA, 341-348.

Foo, S. Y. , Takefuji, Y., and Szu, H., (1994), "Job-shop scheduling based on modified Tank-Hopfield linear programming networks", *Eng. Appl. Artif. Intell.* 7/3, June, 321-327.

Foo, S. Y., Takefuji, Y., and Szu, H., (1995), "Scaling properties of neural networks for job-shop scheduling", *Neurocomputing* 8/1, 79-91.

Fox, M. S., (1987), *Constraint - Directed Search: A Case Study of Job-Shop Scheduling*, *Research Notes in Artificial Intelligence*, Pitman Publishing, London.

Gantt, H. L., (1919), "Efficiency and democracy", *Trans. Amer. Soc. Mech. Engin.* 40, 799-808.

Garey, M. R., Johnson, D. S., and Sethi, R., (1976), "The complexity of flow shop and job-shop scheduling", *Mathematics of Operations Research* 1/2, May, 117-129.

Garey, M. R., and Johnson, D. S., (1979), *Computers and Intertractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, California.

Gere, W. S. Jr., (1966), "Heuristics in job-shop scheduling", *Management Science* 13, 167-190.

Giffler, B., and Thompson, G. L., (1960), "Algorithms for solving production scheduling problems", *Operations Research*, **8**(4), 487-503.

Glover, F., (1968), "Surrogate constraints", *Operations Research* 16, 741-749.

Glover, F., (1975), "Surrogate constraint duality in mathematical programming", *Operations Research* 23, 434-451.

Glover, F., (1977), "Heuristics for integer programming using surrogate constraints", *Decision Sciences* 8/1, 156-166.

Glover, F., (1986), "Future paths for integer programming and links to artificial intelligence", *Computers and Operations Research* 13/5, 533-549.

Glover, F., and Greenberg, H. J., (1989), "New approaches for heuristic search: a bilateral linkage with artificial intelligence", *European Journal of Operations Research* 39/2, 119-130.

Glover, F., (1989), “Tabu search - Part I”, *ORSA Journal on Computing* 1/3, Summer, 190-206.

Glover, F., (1990), “Tabu search - Part II”, *ORSA Journal on Computing* 2/1, Winter, 4-32.

Glover, F. (1994) Optimization by Ghost Image Processes in Neural Networks, *Computers and Operations Research*, **21**(8), 801-822.

Glover, F., (1995), “Tabu search fundamentals and uses”, Working Paper, College of Business and Administration and Graduate School of Business Administration, University of Colorado, Boulder, Colorado, USA, June.

Gonzalez, T., and Sahni, S., (1978), “Flowshop and jobshop schedules: Complexity and approximation”, *Operations Research* 20, 36-52.

Grabot, B., and Geneste, L., (1994), “Dispatching rules in scheduling: A fuzzy approach”, *International Journal of Production Research* 32/4, April, 903-915.

Grabowski, J., Nowicki, E., and Zdrzalka, S., (1986), “A block approach for single machine scheduling with release dates and due dates”, *European Journal of Operational Research* 26, 278-285.

Grabowski J., Nowicki, E., and Smutnicki, C., (1988), “Block algorithm for scheduling operations in a job-shop system”, *Przeglad Statystyczny* XXXV/1, 67-80, *in Polish*.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G., (1979) “Optimisation and approximation in deterministic sequencing and scheduling: a survey”, *Annals of Discrete Mathematics* 5, 287-326.

Greenberg, H., (1968), “A branch and bound solution to the general scheduling problem”, *Operations Research* 16/2, March, 353-361.

Grefenstette, J. J., (1987), “Incorporating problem specific knowledge into genetic algorithms”, in: L. Davis (ed.), *Genetic Algorithms and Simulated Annealing*, Pitman, London, 42-60.

Hara, H., (1995), “Job-shop scheduling by minimal conflict search”, *Japanese Artificial Intelligence Society* 10/1, January, 88-93.

Hardgrave, W. H., and Nemhauser, G. L., (1963), “A geometric model and graphical algorithm for a sequencing problem”, *Operations Research* 11, 889-900.

Harvey, W. D., (1995), “Nonsystematic backtracking search”, *Ph.D. Thesis*, Department of Computer Science, Stanford University, California, USA.

Harvey, W. D., and Ginsberg, M. L., (1995), “Limited discrepancy search”, *IJCAI’95 International Joint Conference on Artificial Intelligence*, Montréal, Québec, Canada.

Haupt, R., (1989), “A survey of priority rule based scheduling”, *OR Spektrum* 11, 3-16.

Hefetz, N., and Adiri, I., (1982), “An efficient optimal algorithm for the two-machines unit-time job-shop schedule-length problem”, *Mathematics of Operations Research* 7, 354-360.

Hilbert, D., (1900), “Mathematische probleme”, Nachrichten der Akademie der Wissenschaften Göttingen, 290-329.

Hoitomt, D. J., Luh, P. B., and Pattipati, K. R., (1993), “Practical approach to job-shop scheduling problems”, *IEEE Trans. Rob. Autom.* 9/1, Feb, 1-13.

Holtsclaw, H. H., and Uzsoy, R., (1996), “Machine criticality measures and subproblem solution procedures in shifting bottleneck methods: a computational study,” *Journal of the Operational Research Society* 47, 666-677.

Hoogeveen, J. A., and Van De Velde, S. L., (1995), “Stronger Lagrangian bounds by the use of slack variables: Applications to machine scheduling problems”, *Mathematical Programming* 70/2, 30th October, 173-190.

Hooker, J. N., (1995), “Testing heuristics: We have it all wrong”, *Journal of Heuristics*, Fall, 1/1, 33-42.

Ivens, P., and Lambrecht, M., (1996), “Extending the shifting bottleneck procedure to real-life applications”, *European Journal of Operational Research* 90, 252-268.

Jackson, J. R., (1955), “Scheduling a production line to minimise maximum tardiness”, Research Report 43, Management Science Research Projects, University of California, Los Angeles, USA.

Jackson, J. R., (1956), “An extension of Johnson’s result on job lot scheduling”, *Nav. Res. Logist. Quart*. 3/3, 201-203.

Jain, A. S. and Meeran, S. (1998a) “An improved search template for job-shop scheduling”, *INFORMS Spring Meeting*, Montreal, Quebec, Canada, April 26-29.

Jain, A. S., and Meeran, S., (1998b), “Job-shop scheduling using neural networks”, *International Journal of Production Research* 36/5, May, 1249-1272.

Jain, A. S., (1998) “A multi-level hybrid framework for the deterministic job-shop scheduling problem”, *Ph. D. Thesis*, University Of Dundee, Dundee, Scotland, UK, October.

Jain, A. S., Rangaswamy, B., and Meeran, S., (1998a), “New and “stronger” job-shop neighbourhoods: Are they as good as they seem ?”, *Submitted to the Journal of Heuristics*.

Jain, A. S., Rangaswamy, B. and Meeran, S., (1998b), “Is solution quality affected when estimating moves in deterministic job-shop scheduling neighbourhoods ?”, *Submitted to the Journal of Scheduling*.

Jeremiah, B., Lalchandani, A., and Schrage, L., (1964), “Heuristic rules toward optimal scheduling”, Research Report, Department of Industrial Engineering, Cornell University.

Johnson, D. S., Papadimitriou, C. H. and Yannakakis, M. (1988) “How easy is local search?”, *Journal of Computer and System Sciences* 37/1, August, 79-100.

Johnson, D. S., Aragon, C. R., M^{c}Geoch, L. A., and Schevon, C., (1989), "Optimization by simulated annealing: An experimental evaluation; Part I, graph partitioning", *Operations Research* 37/6, Nov-Dec, 865-892.

Johnson, S. M., (1954), "Optimal two- and three-stage production schedules with set-up times included", *Nav. Res. Logist. Quart*. 1, 61-68.

Kanzedal, S. A., (1983), "A decomposition approach to solve large scale scheduling problems", *Avtomat. i Telemekh.* 10, 144-151 (in Russian).

Kim, S. Y., Lee, Y. H., and Agnihotri, D., (1995), "A hybrid approach for sequencing jobs using heuristic rules and neural networks", *Production Planning and Control* 6/5, 445-454.

Kobayashi, S., Ono, I., and Yamamura, M., (1995), "An efficient genetic algorithm for job shop scheduling problems", *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Francisco, CA, 506-511.

Kolmogorov, A. N., (1957), "On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition", *Doklady Akademiia Nauk SSR* 14/5, 953-956 (in Russian). (American Mathematical Society Translation 28 (1963) 55-59).

Kolonko, M., (1998), "Some new results on simulated annealing applied to the job shop scheduling problem", *to appear in the European Journal of Operational Research*.

Kravchenko, S. A., and Sotskov, Y. N., (1996), "Optimal makespan schedule for three jobs on two machines", *Z. Oper. Res.* 43/2, 233-238.

Kravchenko, S. A., (1998), "A polynomial algorithm for a two-machine no-wait job-shop scheduling problem", *The European Journal of Operational Research*, 106/1, 101-107.

Krüger, K., Shakhlevich, N. V., Sotskov, Y. N., and Werner, F., (1995), "A heuristic decomposition algorithm for scheduling problems on mixed graphs", *Journal of the Operational Research Society* 46, 1481-1497.

Kubiak, W., and Timkovsky, V. G., (1997), "Total completion time minimization in two-machine job shops with unit time operations," *to appear in the European Journal of Operational Research*.

Kusiak, A., and Chen, M., (1988), "Expert systems for planning and scheduling manufacturing systems", *European Journal of Operational Research* 34, 113-130.

Lageweg, B. J., Lenstra, K., and Rinnooy Kan, A. H. G., (1977), "Job-shop scheduling by implicit enumeration", *Management Science*, 24/4, December, 441-450.

Lageweg, B. J., (1982), "Combinatorial planning models", *PhD. Thesis*, Mathematisch Centrum, Amsterdam, The Netherlands.

Lageweg, B. J., (1984), Private Communication with Peter J. M. Van Laarhooven; Emile H. L. Aarts and Jan Karel Lenstra discussing the achievement of a makespan of 930 for FT 10.

Laguna, M., Barnes, J. W., and Glover, F. W., (1991), "Tabu search methods for a single machine scheduling problem", *Journal of Intelligent Manufacturing* 2, 63-74.

Laguna, M., Barnes, J. W., and Glover, F. W., (1993), "Intelligent scheduling with tabu search: an application to jobs with linear delay penalties and sequence-dependent setup costs and times", *Journal of Applied Intelligence* 3, 159-172.

Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G., (1982), "Recent developments in deterministic sequencing and scheduling: A survey", in: M. A. H. Dempster, *et al*. (eds.), *Deterministic and Stochastic Scheduling*, Reidel, Dordrecht, 35-73.

Lawler, E. L., (1983), "Recent results in the theory of machine scheduling", in: A. Bachem, M. Grötschel and B. Korte (eds.), *Mathematical Programming: The State of the Art*, Springer-Verlag, Berlin, 202-234.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B., (1993), "Sequencing and scheduling: algorithms and complexity", *Handbook in Operations Research and Management Science* 4: *Logistics of Production and Inventory*.

Lawrence, S., (1984), "Supplement to resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques", Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA, October.

Lenstra, J. K., (1976), "Sequencing by enumerative methods", *PhD. Thesis*, Mathematisch Centrum Tract 69, Amsterdam, The Netherlands.

Lenstra, J. K., Rinnooy Kan, A. H. G., and Brucker, P., (1977), "Complexity of machine scheduling problems", *Ann. Discrete. Math.* 7, 343-362.

Lenstra, J. K., and Rinnooy Kan, A. H. G., (1979), "Computational complexity of discrete optimization problems", *Annals of Discrete Mathematics* 4, 121-140.

Lo, Z-P., and Bavarian, B., (1993), "Multiple job-shop scheduling with artificial neural networks", *Comput. Electr. Eng.* 19/2, March, 87-101.

Lourenço, H. R. D., (1993), "A computational study of the job-shop and the flow-shop scheduling problems", *PhD. Thesis* TR - 1060, School of Operations Research & Industrial Engineering, Cornell University, Ithaca, New York 14853-3801, July.

Lourenço, H. R. D., (1995), "Job-shop scheduling: computational study of local search and large-step optimization methods", *European Journal of Operational Research* 83, 8th June, 347-364.

Lourenço, H. R. D., and Zwijnenburg, M,. (1996), "Combining the large-step optimization with tabu-search: application to the job-shop scheduling problem", in: I. H. Osman and J. P. Kelly (eds.), *Meta-heuristics: Theory and Applications*, Kluwer, Boston, 219-236.

MacCarthy, B. L., and Liu, J., (1993), "Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling", *International Journal of Production Research* 31/1, 59-79.

Martin, O., Otto, S. W., and Felten, E. W., (1989), "Large-step Markov chains for traveling salesman problem", *Complex Systems* 5, 299-326.

Martin, O., Otto, S. W., and Felten, E. W., (1992), "Large-step Markov chains for TSP incorporating local search heuristics", *Operations Research Letters* 11, 219-224.

Martin, P. D. (1996) A Time-Oriented Approach to Computing Optimal Schedules for the Job-Shop Scheduling Problem, *PhD. Thesis*, School of Operations Research & Industrial Engineering, Cornell University, Ithaca, New York 14853-3801, August.

Matsuo, H., Suh, C. J., and Sullivan, R. S., (1988), "A controlled search simulated annealing method for the general job-shop scheduling problem", Working Paper #03-04-88, Graduate School of Business, The University of Texas at Austin, Austin, Texas, USA.

Mattfeld, D. C., Kopfer, H., and Bierwirth, C., (1994), "Control of parallel population dynamics by social-like behaviour of GA-individuals", *PPSN'3 Proceedings of the Third International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, 15-25.

Mattfeld, D. C., (1996), "Evolutionary Search and the Job Shop: Investigations on Genetic Algorithms for Production Scheduling", Physica-Verlag, Heidelberg, Germany.

Mattfeld, D. C., Bierwirth, C., and Kopfer, H., (1998), "A Search Space Analysis of the Job Shop Scheduling Problem", *to appear in Annals of Operations Research*.

M^{c}Mahon, G. B., and Florian, M., (1975), "On scheduling with ready times and due dates to minimize maximum lateness", *Operations Research* 23/3, May-June, 475-482.

Moore, L. J., (1968), "An n-job, one-machine sequence algorithm for minimizing the number of late jobs", *Management Science* 15, 102-109.

Morton, T. E., (1990), "Shifting bottleneck methods in job shop and project scheduling: tutorial and research directions", Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA.

Morton, T. E., and Pentico, D. W., (1993), "*Heuristic Scheduling Systems*", Wiley Series in Engineering and Technology Management, Wiley, New York.

Moscato, P., (1989), "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms", *C3P Report 826*, Caltech Concurrent Computation Program, Caltech, California, USA.

Nabeshima, I., (1971), "General scheduling algorithms with applications to parallel scheduling and multiprogramming scheduling", *J. Oper. Res. Soc. Japan* 14/2, 72-99.

Nakano, R., and Yamada, T., (1991), "Conventional genetic algorithm for job-shop problems", in: M. K. Kenneth and L. B. Booker (eds.), *Proceedings of the 4th International Conference on Genetic Algorithms and their Applications*, San Diego, California, USA, 474-479.

Norman, B., and Bean, J., (1995), "Random keys genetic algorithm for job-shop scheduling: Unabridged Version", Technical Report, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, USA.

Nowicki, E., and Smutnicki, C., (1993), "A fast taboo search algorithm for the job-shop problem", Preprinty nr 8/93, Institute of Engineering Cybernetics, Technical University of Wroclaw, Wroclaw, Poland.

Nowicki, E., and Smutnicki, C., (1996), "A fast taboo search algorithm for the job-shop problem", *Management Science* 42/6, June, 797-813.

Nuijten, W. P. M., and Aarts, E. H. L., (1994), "Constraint satisfaction for multiple capacitated job shop scheduling", in: A. G. Cohn (ed.), *ECAI'94 Proceedings of the 11th International Conference on Artificial Intelligence*, 635-639.

Nuijten, W. P. M., and Aarts, E. H. L., (1996), "A computational study of constraint satisfaction for multiple capacitated job shop scheduling", *European Journal of Operations Research* 90/2, 19th April, 269-284.

Nuijten, W. P. M., and Le Pape, C., (1998), "Constraint-Based job shop scheduling with ILOG SCHEDULER", *Journal of Heuristics*, March, 3/4, 271-286.

Ovacik, I. M. and Uzsoy, R., (1992), "A shifting bottleneck algorithm for scheduling semiconductor testing operations", *Journal of Electronics Manufacturing*, vol 2, 119-134.

Ovacik, I. M. and Uzsoy, R., (1996), "Decomposition methods for scheduling semiconductor testing facilities", *International Journal of Flexible Manufacturing Systems*, vol 8, 357-388.

Panwalkar, S. S., and Iskander, W., (1977), "A survey of scheduling rules", *Operations Research* 25/1, Jan-Feb, 45-61.

Parker, R. G., (1995), *Deterministic Scheduling*, Chapman and Hall, London.

Perregaard, M., and Clausen, J., (1995), "Parallel branch-and-bound methods for the job-shop scheduling problem", Working Paper, University of Copenhagen, Copenhagen, Denmark.

Pesch, E. (1993) Machine Learning by Schedule Decomposition, Working Paper, Faculty of Economics and Business Administration, University of Limburg, Maastricht, The Netherlands.

Pesch, E., and Tetzlaff, U. A. W., (1996), "Constraint propagation based scheduling of job shops", *INFORMS Journal on Computing* 8/2, Spring, 144-157.

Pinson, E., (1995), "The job-shop scheduling problem: A concise survey and some recent developments", in: P. Chrétienne, E. G. Coffman Jr, J. K. Lenstra and Z. Liu (eds.), *Scheduling Theory and its Applications*, John Wiley & Sons, 277-293.

Porter, D. B., (1968), "The Gantt chart as applied to production scheduling and control", *Naval Research Logistics Quarterly* 15, 311-317.

Radermacher, F. J., (1985), "Scheduling of project networks", *Annals of Operations Research* 4/6, 227-252.

Ramudhin, A., and Marier, P., (1996), "The generalised shifting bottleneck procedure", *European Journal of Operational Research* 93/1, 34-48.

Reeves, C. R., (1993), "Evaluation of heuristic performance", in: C. R. Reeves, (ed.), *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Osney Mead, Oxford, England, 304-315.

Resende, M. G. C., (1997) "A GRASP for job shop scheduling", *INFORMS Spring Meeting*, San Diego, California, USA, May.

Rinnooy Kan, A. H. G., (1976), "Machine scheduling problems: Classification, complexity and computations", in: H. E. Stenfert Kroese and B. V. Leiden (eds.), Martinus Nijhoff, The Hague, The Netherlands.

Rodammer, F. A., and White, K. P. Jr, (1988), "A recent survey of production scheduling", *IEEE Transactions of Systems, Man and Cybernetics* 18/6, Nov-Dec, 841-851.

Roy, B., and Sussmann, B., (1964), "Les problèmes d'ordonnancement avec contraintes disjonctives", Note D.S. no. 9 bis, SEMA, Paris, France, Décembre.

Sabuncuoglu, I., and Gurgun, B., (1996), "A neural network model for scheduling problems", *European Journal of Operations Research* 93/2, September, 288-299.

Sabuncuoglu, I., and Bayiz, M., (1997), "A beam search based algorithm for the job shop scheduling problem", Research Report IEOR-9705, Dept. of Industrial Engineering, Bilkent University, Turkey.

Sadeh, N., (1991), “Look-ahead techniques for micro-opportunistic job shop scheduling”, *Ph.D. Thesis*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA, March.

Sadeh, N., Sycara, K., and Xiong, Y. L., (1995), “Backtracking techniques for the job-shop scheduling constraint satisfaction problem,” *Artificial Intelligence* 76/1-2, 455-480.

Sadeh, N., and Fox, M. S., (1996), “Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem,” *Artificial Intelligence* 86/1, 1-41.

Sadeh, N., and Nakakuki, Y., (1996), “Focused simulated annealing search - an application to job-shop scheduling,” *Annals of Operations Research* 63, 77-103.

Sadeh, N., Nakakuki, Y., and Thangiah, S. R., (1997), “Learning to recognise (un)promising simulated annealing runs: Efficient search procedures for job-shop scheduling and vehicle routing,” *Annals of Operations Research* 75, 189-208.

Salvesen, M. E., and Anderson, S. L., (1951), “ Some notes on the problem of programming industrial production”, George Washington University Logistics papers.

Satake, T., Morikawa, K. and Nakamura, N. (1994) Neural Network Approach for Minimizing the Makespan of the General Job-Shop, *International Journal of Production Economics*, Jan, **33**(1-3), 67-74.

Schrage, L., (1970), “Solving resource-constrained network problems by implicit enumeration: non pre-emptive case”, *Operations Research* 18, 263-278.

Shakhlevich, N. V., Sotskov, Y. N., and Werner, F., (1996), “An adaptive approach in production scheduling based on the mixed graph model”, *IEE Proc.-Control Theory Appl*. 143/1, 9-16.

Shi, G., (1997), “A genetic algorithm applied to a classic job-shop scheduling problem”, *International Journal of Systems Science* 28/1, 25-32.

Shmoys, D. B., Stein, C., and Wein, J., (1994), “Improved approximation algorithms for shop scheduling problems”, *SIAM J. Comput.* 23/3, June, 617-632.

Sim, S. K., Yeo, K. T., and Lee, W. H., (1994), “An expert neural network system for dynamic job-shop scheduling”, *International Journal of Production Research* 32/8, August, 1759-1773.

Smith, W. E., (1956), “Various optimizers for single stage production”, *Naval Research Logistics Quarterly* 3, 59-66.

Sotskov, Y. N., (1985), "Optimal scheduling two jobs with regular criterion", in: *Design Processes Automating*, Institute of Engineering Cybernetics of the Academy of Sciences of Belarus, Minsk, Belarus, 86-95 (in Russian).

Sotskov, Y. N., (1991), "The complexity of shop-scheduling problems with two or three jobs", *European Journal of Operational Research* 53, 326-336.

Sotskov, Y. N., and Shakhlevich, N. V., (1995), "$\mathcal{NP}$-hardness of shop scheduling problems with three jobs", *Discrete Applied Mathematics* 59/3, 26th May, 237-266.

Sotskov, Y. N., (1996) "Software for production scheduling based on the mixed (multi)graph approach", *Computing and Control Engineering Journal* 7/5, October, 240-246.

Sotskov, Y. N., (1997), "Mixed multi-graph approach to scheduling jobs on machines of different types", *Optimization*, 1-36.

Storer, R. H., Wu, S. D., and Vaccari, R., (1992), "New search spaces for sequencing problems with applications to job-shop scheduling", *Management Science* 38/10, October, 1495-1509.

Storer, R. H., Wu, S. D., and Vaccari, R., (1995), "Problem and heuristic space search strategies for job shop scheduling", *ORSA Journal on Computing*, 7/4, Fall, 453-467.

Sun, D. K., Batta, R., and Lin, L., (1995), "Effective job-shop scheduling through active chain manipulation", *Computers & Operations Research* 22/2, 159-172.

Sussmann, B., (1972), "Scheduling problems with interval disjunctions", *Z. Oper. Res*. 16, 165-178.

Szwarc, W., (1960), "Solution of the Akers-Friedman scheduling problem", *Operations Research* 8, 782-788.

Taillard, É., (1989), "Parallel taboo search technique for the job-shop scheduling problem", Internal Research Report ORWP89/11, Départment de Mathématiques (DMA), École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland, July.

Taillard, É., (1993), "Benchmarks for basic scheduling problems", *European Journal of Operational Research* 64/2, 278-285.

Taillard, É., (1994), "Parallel taboo search techniques for the job-shop scheduling problem", *ORSA Journal on Computing* 16/2, 108-117.

Tamaki, H., and Nishikawa, Y., (1992), "A paralleled genetic algorithm based on a neighbourhood model and its application to the jobshop scheduling", in: R. Männer, and B. Manderick, (eds), *PPSN'2 Proceedings of the 2nd International Workshop on Parallel Problem Solving from Nature*, Brussels, Belgium, 573-582.

Thomsen, S., (1997), "Metaheuristics combined with branch & bound", *Technical Report*, Copenhagen Business School, Copenhagen, Denmark. (in Danish).

Timkovsky, V. G., (1995), "The complexity of unit time job shop scheduling problems", Working Paper, Department of Computer Science and Systems, M^{c}Master University, Hamilton, Ontario, Canada.

Ulder, N. L. J., Aarts, E. H. L., Bandelt, H. -J., Van Laarhooven, P. J. M., and Pesch, E., (1990), "Improving TSP exchange heuristics by population genetics", *PPSN'1 First International Workshop on Parallel Problem solving from Nature*, Dortmund Germany.

Ulder, N. L. J., Aarts, E. H. L., Bandelt, H.-J., Van Laarhooven, P. J. M., and Pesch, E., (1991), "Genetic local search algorithm for the travelling salesman problem", *Lecture Notes in Computer Science* 496, 109-116.

Vaessens, R. J. M., (1995), "Generalised job-shop scheduling: Complexity and local search", *PhD. Thesis*, Department of Mathematics & Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands.

Vaessens, R. J. M., Aarts, E. H. L., and Lenstra, J. K., (1995), "A local search template", Revised Version, Memorandum Computer Science and Operations Research (COSOR), Department of Mathematics & Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands, January.

Vaessens, R. J. M., (1996), Operations research library of problems, Management School, Imperial College London, Anonymous FTP site at ftp://mscmga.ms.ic.ac.uk/pub/ jobshop1.txt.

Vaessens, R. J. M., Aarts, E. H. L., and Lenstra, J. K., (1996), "Job-shop scheduling by local search", *INFORMS Journal on Computing*, 8, 302-317.

Van den Akker, J. M., (1994), "LP-based solution methods for single machine scheduling problems", *PhD. Thesis*, Eindhoven University of Technology, Eindhoven, The Netherlands.

Van De Velde, S., (1991), "Machine scheduling and Lagrangian relaxation", *PhD. Thesis*, CWI Amsterdam, The Netherlands.

Van Hulle, M. M., (1991), "A goal programming network for mixed integer linear programming: A case study for the job-shop scheduling problem", *International Journal of Neural Systems*, 2/3, 201-209.

Van Laarhoven, P. J. M., Aarts, E. H. L., and Lenstra, J. K., (1988), "Job shop scheduling by simulated annealing", Report OS-R8809, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands.

Van Laarhoven, P. J. M., and Aarts, E. H. L., (1989), *Simulated Annealing: Theory and Applications, Mathematics and its Applications Series*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Van Laarhooven, P. J. M., Aarts, E. H. L., and Lenstra, J. K., (1992), "Job shop scheduling by simulated annealing", *Operations Research* 40/1, Jan-Feb, 113-125.

Wagner, H. M., (1959), "An integer programming model for machine scheduling", *Naval Research Logistics Quarterly* 6, 131-140.

Wang, W., and Brunn, P., (1995), "Production scheduling and neural networks", in: U. Derigs, A. Bachem and A. Drexl (eds.), *Operations Research Proceedings 1994*, Springer-Verlag, Berlin, 173-178.

Watanabe, T., Tokumaru, H., and Hashimoto, Y., (1993), "Job-shop scheduling using neural networks", *Control Eng. Prac*., Dec, 1/6, 957-961.

Wennink, M., (1995), Operations research library of problems, Management School, Imperial College London, Anonymous FTP site at ftp://mscmga.ms.ic.ac.uk/pub/jobshop1.txt.

Werner, F., and Winkler, A., (1995), Insertion techniques for the heuristic solution of the job-shop problem", *Discrete Applied Mathematics*, 58/2, 191-211.

Willems, T. M. and Rooda, J. E. (1994) Neural Networks for Job-Shop Scheduling, *Control Eng. Practice*, Feb, **2**(1), 31-39.

Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., Sevast'janov, S. V., and Shmoys, D. B., (1997), "Short shop schedules", *Operations Research*, 45/2, March-April, 288-294.

White, K. P., and Rogers, R. V., (1990), "Job-shop scheduling: limits of the binary disjunctive formulation", *International Journal of Production Research* 28/12, 2187-2200.

Yamada, T., and Nakano, R., (1992), "A genetic algorithm applicable to large-scale job-shop problems", in: R. Männer and B. Manderick (eds.), *PPSN'2 Proceedings of the 2nd International Workshop on Parallel Problem Solving from Nature*, Brussels, Belgium, 281-290.

Yamada, T., Rosen, B. E., and Nakano, R., (1994), "A simulated annealing approach to job-shop scheduling using critical block transition operators", *IEEE ICNN'94 International Conference on Neural Networks*, Orlando, Florida, USA, 4687-4692.

Yamada, T., and Nakano, R., (1995a), "Job-shop scheduling by simulated annealing combined with deterministic local search", *MIC'95 Metaheuristics International Conference*, Hilton, Breckenridge, Colorado, USA, 344-349.

Yamada, T., and Nakano, R., (1995b), “A genetic algorithm with multi-step crossover for job-shop scheduling problems”, *GALESIA’95 Proceedings of the Int. Conf. on GAs in Eng. Sys.*, 146-151.

Yamada, T., and Nakano, R., (1996a), “Job-shop scheduling by simulated annealing combined with deterministic local search”, *Meta-heuristics: Theory and Applications*, Kluwer Academic Publishers, MA, USA, 237-248.

Yamada, T., and Nakano, R., (1996b), “Scheduling by genetic local search with multi-step crossover”, *PPSN’IV Fourth International Conference on Parallel Problem Solving from Nature*, Berlin, Germany, September 22-26, 960-969.

Yamada, T., and Nakano, R., (1996c), “A fusion of crossover and local search”, *ICIT’96 IEEE International Conference on Industrial Technology,* Shanghai, China, December 2-6, 426-430.

Yannakakis, M., (1990), “The analysis of local search problems and their heuristics”, *Lecture Notes in Computer Science* 415, Springer, Berlin, 298-311.

Zhou, D. N., Cherkassky, V., Baldwin, T. R., and Hong, D. W., (1990), “Scaling neural networks for job-shop scheduling”, *IJCNN’90 International Joint Conference on Neural Networks*, San Diego, CA, 889-894.

Zhou, D. N., Cherkassky, V., Baldwin, T. R., and Olson D. E., (1991), “A neural network approach to job-shop scheduling”, *IEEE Transactions on Neural Network* 2/1, 175-179.