Monograph:

Dimopoulos, C. and Zalzala, A.M.S. (1998) Evolutionary Computation for Manufacturing Optimisation: Recent Developments. Research Report. ACSE Research Report 716 . Department of Automatic Control and Systems Engineering

RERVM COGNOSCERE CAVSAS

Department of
AUTOMATIC
CONTROL
and SYSTEMS
ENGINEERING
UNIVERSITY of SHEFFIELD

EVOLUTIONARY COMPUTATION FOR MANUFACTURING OPTIMISATION: RECENT DEVELOPMENTS

C Dimopoulos and AMS Zalzala

Robotics Research Group,
Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield S1 3JD, United Kingdom

Research Report #716
May 1998

RRG
Tel : +44 (0)114 2225250
Fax : +44 (0)114 2731729
EMail : rrg@sheffield.ac.uk
Robotics Research Group

EVOLUTIONARY COMPUTATION FOR MANUFACTURING OPTIMISATION: RECENT DEVELOPMENTS

C Dimopoulos and AMS Zalzala

Robotics Research Group
Department of Automatic Control and Systems Engineering
The University of Sheffield, UK
Email: rrg@sheffield.ac.uk

1. Introduction

Genetic Algorithms (GAs) were formally introduced by Holland [1] more than twenty years ago. Since then, numerous algorithms based on the concept of Darwinian strife for survival have been developed and applied to a large number of optimisation problems.

The operation of a simple GA is straightforward: Given a certain optimisation problem, an initial population of binary-coded solutions (chromosomes) is generated randomly. The performance of each solution is evaluated and assigned with a 'fitness' value. A new population is then created, by evolving chromosomes selected from the old population. The higher the 'fitness' of an individual solution, the better is its chance to be selected for the new population. Chromosomes exchange or alter their genetic material using specially designed genetic operators. The purpose of this function is the best possible exploration of the solutions' search space. The procedure is then repeated until a desired 'fitness' value is reached, or a predefined number of iterations are completed.

Many researchers, who applied it in a series of optimisation problems, introducing sometimes variations in the representation of solutions, the fitness assignment, selection methods and genetic operators, adopted this simple concept. The term 'evolutionary computation methods' describes all these algorithms which are based on the concept of evolution. The most notable members of this group are simple GAs [1,2], evolution strategies [3], evolutionary programming [4], evolution programs [5], classifier systems [6], and genetic programming [7]. Back [8] gives an excellent review of evolutionary computation methods and highlights some recent developments in the field. For a thorough introduction on GAs and evolutionary computation, the reader should refer to Holland [1], Goldberg [2], Michaelewicz [5], Mitchell [9], and Koza [7].

A large number of combinatorial problems are associated with manufacturing optimisation. Most of them are NP-complete, i.e. there is no polynomial time algorithm that can possibly solve them [10]. Heuristic methods are normally employed for the solution of these problems. A growing number of researchers have adopted the use of smart-heuristic techniques (meta-heuristics) for large combinatorial problems. Evolutionary computation methods are smart heuristics that are able to search large regions of the solution space without being trapped in local optima. Some other well-known smart heuristics are Simulated Annealing (SA) [11], and Tabu Search [12-14].

The aim of this paper is to illustrate recent developments in the field of evolutionary computation for manufacturing optimisation. A wide range of optimisation problems is considered, from the classic job-shop and flow-shop scheduling problems, to assembly line balancing and aggregate production planning. We focus mainly on recent publications, but there are pointers to significant earlier approaches. In this way, the reader who is interested in a particular problem can use this report as a starting point.

The evolutionary computation terminology is not yet standardised; thus the term GA is used interchangeably to describe different techniques. We will follow the same line throughout this paper, unless otherwise stated. The rest of the paper is organised as follows: section 2 examines recent evolutionary computation developments for the job-shop scheduling problem. The same procedure is followed in section 3 for the flowshop scheduling problem, section 4 for the dynamic scheduling problem, section 5 for the process planning problem, section 6 for cellular manufacturing optimisation problems, section 7 for assembly optimisation problems and section 8 for design optimisation problems. Section 9 overviews some recent developments in other manufacturing optimisation areas and section 10 draws conclusions.

2. The Job-Shop Scheduling Problem

2.1 Introduction

A great amount of research work in the field of evolutionary computation is devoted to the solution of the job-shop scheduling problem (JSSP). Davis [15] made the first attempt to solve the problem using GAs more that ten years ago. His paper was "rather instructive that realistic" as Jain et al. [16] argue. Some years later Yamada et al. [17] proposed a more natural representation for the solution of the problem using the completion time of operations. Since then, the number of publications has been growing rapidly, and so has the number of different approaches that are proposed for the solution of the problem.

2.2 Formulation of the problem

The job-shop scheduling problem requires the ordering n jobs to be processed in m machines. Each job involves a number of different machining operations. The following conditions hold for the JSSP:

- Each machine is processing only one job at a time
- The sequence of operations for each job is predefined
- Two operations of the same job cannot be processed at the same time
- Pre-emption is not allowed (an operation cannot be withdrawn from a machine unless it is completed
- Processing times are known in advance
- Transportation time between machines is zero

The objective of the scheduling algorithm is the minimisation of makespan (the time interval from the start of the first operation until the finish of the last operation). The JSSP is also known as the $n/m/G/Cmax$ [18] problem. Bierwirth et al. [19] describe it as a "representative of constrained combinatorial problems". It is a static scheduling problem, since unexpected events are not considered in the formulation of the problem. In the special case of $n=m$, the problem is described as the 'generalised assignment problem'. It has been solved using evolutionary computation methods by Chu et al. [20]. Fang et al. [21] approached the open-shop scheduling problem (the case where the sequence of operations is not predefined) using a combined GA-heuristic approach. The dynamic version of the JSSP will be discussed in a following section.

Cao et al. [22] argue that the classic formulation of the JSSP is unrealistic, since it does not represent the vast majority of manufacturing cases, and does not take into account a number of elements which are important in real-life scheduling, like set-up times, due dates and machine off-line times. Several other authors have criticised academic research for considering scheduling problems that are rarely being used in practice [23,24]. As a result, many researchers in the field of evolutionary computation are now using a variety of criteria for the evaluation of schedules. Minimisation of makespan is still used as an objective in many papers [19,25-27], but the general belief is that the objective of manufacturing optimisation should be the minimisation of production cost. Fang et al [28] propose seven quality criteria for good schedules: maximum tardiness, average tardiness, weighted flow time, weighted lateness, weighted tardiness, weighted number of tardy jobs and weighted earliness plus weighted tardiness. These criteria are in accordance with the Just In Time (JIT) principle of having a product made exactly when it is required. In that way the storage costs (earliness) and the lateness fines (tardiness) are minimised. Similar objectives are used in [16,29,30-32]. Due-dates and ready times of the products are pre-specified in these cases.

2.2 Encoding

The classic binary solution representation of the simple GA has rarely been used for the JSSP. Most researchers adopt a purpose-based representation, which is much more effective. Perhaps the most natural representation for the solution of the problem would be an m-partitioned permutation (where m is the total number of machines), with each partition representing the complete schedule of an independent machine. No decoding is needed to obtain the schedule; thus this type of representation is called 'direct'. This representation is especially popular in sequencing problems, where the solution is not partitioned, so all famous Travelling Salesman Problem (TSP) operators can be easily used. However, in the case of the JSSP, efficient operators and computationally exhaustive repair mechanisms or penalty functions are needed to ensure the feasibility of solutions. Dagli et al. [33,34] are among the few that employ the direct solution representation. They overcome feasibility problems by using legal schedules to initialise the population and an order-based crossover operator to preserve the precedence constraints of the problem (a discussion on genetic operators will follow in the section of flowshop scheduling). Aizpuru et al [35] also prefer the direct representation for their hybrid GA/Tabu Search scheduling method. They employ Giefler & Thompson's (GT) [36] algorithm in order to initialise the population with active legal schedules. However, most researchers in the field of evolutionary computation prefer to use the indirect representation of solutions, where an operation number is not explicitly stated in the chromosome, but only the owing job is defined. The chromosome

$$\{J1, J2, J1, J3, J2, J1, \ldots\}$$

indicates that the first operation of the first job should be scheduled first, followed by the first operation of the second job, the second operation of the first job etc. It is obvious that a schedule builder is needed to transform this solution into a feasible schedule (for a discussion about schedule builders see [37,38]). This representation has recently been used in [25,28], and [19]. Shi et al. [39,40], adopt the same representation with the enhancement of an efficient algorithm which decodes the strings into active schedules.

Herrmann et al. [41] propose a different representation, which uses the classic job-shop dispatching rules. Dispatching rules are common manufacturing practice for scheduling [42-44]. The chromosome in this case takes the following form:

$$\{EDD, SPT, FIFO, \ldots\}$$

where EDD: Earliest Due Date, SPT: Shortest Processing Time, and FIFO: First In First out. Each gene represents a machine, and the value of the gene represents the dispatching rule that this machine uses for the scheduling of operations waiting. Fujimoto et al. [45,46], propose the same representation for the scheduling of a Flexible Manufacturing System (FMS). Each gene in this case corresponds to a decision making point in the plant, and the value of the gene defines the dispatching rule that will be used at this point. Kumar et al. [32] use a circular string of dispatching rules as a scheduling policy, whenever a part is requested for processing. All the traditional operators can be used with the dispatching rules representation, without producing infeasible solutions.

A dynamic data structure called 'hierarchical linked list' is proposed by Niemeyer et al. [47] in order to accommodate variable lengths of jobs and operations in a real manufacturing environment. Kim et al. [48] tackle the problem of unfeasibility by using random-keys [49] representation for the solutions.

A paper that has attracted a lot of attention recently is that of Croce et al. [29]. They used the concept of preference lists for the encoding of the solutions, together with a look-ahead evaluation method for transforming non-delay schedules into active ones; (see Baker [50] for a discussion on schedule types). Preference lists are not actual schedules, but a preferable sequence of operations on each machine. Operations are scheduled according to this sequence unless they violate a precedence constraint. In that case the next operation in the preference list is scheduled. The concept of preference lists is also adopted by Kim et al [51,52], in a genetic reinforcement learning approach for a series of scheduling problems, including the JSSP. This method showed outstanding performance in a series of test problems as we will see in the next paragraph. Kobayashi et al. [53] and Ono et al. [54] achieved good results by encoding the solution in the same preference-list form. They used the traditional Giefler & Thompson (GT) algorithm for the decoding of the solutions into active schedules, and they introduced two purpose-based crossover operators, the subsequent exchange crossover (SSX) and job-order based crossover (JOX) respectively. Park et al. [55,56] report their preference list-based GA with the introduction of a crossover operator called Active Schedule Constructive Crossover (ASCX), which is based on the active schedule generation algorithm [50].

Several other representation schemes have been reported: Liang et al. [57], proposed a sparse matrix solution's representation with purpose-based operators to ensure the feasibility of solutions. Cho et al. [58] introduced a method called Total Operation Order Method (TOOM), where a solution is given in the form of a job operation matrix which defines the absolute order of all operations to be processed. Yamada et al. [59] used a disjunctive-graph representation for the solution of the problem. Following the trend of enhancing the evolutionary process with local search techniques, they introduced a crossover operator called Multi-Step Crossover (MSX), which is in effect a local search optimisation algorithm. Dorndorf et al. [26] employ a GA to guide the search of the powerful 'shifting bottleneck' [60] heuristic with encouraging results. Gohtoh et al. [61] applied a special GA-type with neutral mutations [62] to some standard benchmark problems. Finally, Cao et al. [22] addressed a complex JSSP problem with multiple objectives, by using a multi-string chromosome solution, where each string corresponds to a machine schedule. They introduced a hierarchical evaluation model in order to deal with infeasible solutions. The entire framework is called Hierarchical Evaluation GA (HEGA). An excellent analytical review of the representations that have been used for the solution of the JSSP, can be found in [37].

2.3 Test Problems and Case Studies

In recent years, academic research has attempted to consider real-life scheduling problems, thus using evolutionary computation methods in manufacturing practice. Standard benchmark problems do not attract the attention of people in industry, since practical scheduling problems are far more complex than the famous Fisher & Thompson's [64] MT06, MT10, and MT20, and Lawrence's [65] benchmark problems which are still used in most papers. Table 1 gives a summary of results that have been recently published for the three Fisher & Thompson problems. The best and average (wherever available) results of each method are presented. Table 2 summarises the results published for some of Lawrence's benchmark problems.

However, a considerable number of recently published papers addresses real scheduling cases. Herrmann et al [41] describe the development of a global scheduling system for a semiconductor test area. Niemeyer et al. [47] use GAs for the scheduling of factories of a multinational company. Terano et al. [31] present a combined GA/SA approach for the scheduling problem in plastic injection molding. Gilkinson [30] tackles the scheduling problem of a company that produces laminated paper and foil products. Hamada et al. [66] solve a complex scheduling problem in a steel-making company using a hybrid system based on GAs and expert systems. Shaw et al. [38] and Kumar et al. [32] propose evolutionary computation methods for the solution of scheduling problems in companies that produce ready chill meals and defence products, respectively. Finally, Sakawa et al. [67] consider the scheduling problem of a machining centre using GAs.

3. The Flowshop Scheduling Problem

3.1 Introduction

The permutation flowshop scheduling, or job sequencing as it is often called, is another manufacturing optimisation problem that attracts particular research interest. Applying evolutionary computation methods to this problem is relatively easy since it can be formulated as a classical Travelling Salesman Problem (TSP) with path representation. This latter problem has been a subject of research from the early days of the evolutionary computation field. As a result, all the efficient operators that have been developed for the TSP, are directly applicable to the flowshop scheduling problem.

PAPERS	*FT 6X6*		*FT 10X10*		*FT 20X5*	
	Best	***Aver.***	***Best***	***Aver.***	***Best.***	***Aver.***
Aizpuru et al. [35]	-	-	930	-	-	-
Cao et al. [22]	-	-	945	953.5	1176	1198.3
Cho et al. [58]	55	-	943	-	-	-
Croce et al. [29]	55	55	946	965.2	1178	1199
Dorndorf et al. [26]	55	-	938	-	1178	-
Gen et al. [25]	55	-	962	-	1175	-
Gohtoh et al. [61]	-	-	930	935.36	1165	1180.34
Kim et al.(1995)[51]	-	-	930	931.57	1165	1165.97
Kim et al.(1996)[52]	-	-	930	930	1165	1165.27
Kobayashi et al. [53]	-	-	930	934.3	1165	1217.4
Ono et al. [54]	-	-	930	931.1	1165	1176.5
Park et al. [56]	-	-	936	949	1178	1185
Shi et al. [39]	-	-	930	946.2	1165	-
Yamada et al. [59]	-	-	930	934.5	1165	1173.7

Table 1: Published results on Fisher & Thompson's benchmark problems. Optimal values: FT 6X6: **55**, FT 10X10: **930**, FT 20X5: **1165**

	Aizpuru et al. [35]		*Cao et al. [22]*		*Kim et al. [51]*		*Ono et al. [54]*		*Park et al. [55]*	
TEST NO.	Best	Aver.	Best	Aver.	Best	Aver.	Best	Aver.	Best	Aver.
LA01			666*	666	666*	666	666*	666		
LA06			926*	926	926*	926	926*	926		
LA11			1222*	1222	1222*	1222	1222*	1222		
LA16			956	980	945*	945.4	979	989		
LA21	1056	-	1061	1083.6	1055	1055.8	1097	1136		
LA22					935	935.47			935	949
LA26			1227	1231.2	1218*	1218	1231	1248		
LA27	1255				1255	1264.9				
LA31			1784*	1784	1784*	1784	1784*	1784		
LA36			1337	1348			1305	1330.4		

Table 2: Published results on Lawrence's benchmark problems. (*) denotes optimal value

3.2 Problem formulation

The permutation flowshop scheduling problem is the problem of ordering n jobs to be processed in m machines. The difference between the job shop and the flowshop scheduling problem is that in the latter case each job undergoes the same machining sequence and each machine processes the jobs in the same sequence. This means that the solution of the problem can be represented as a permutation of all jobs to be processed:

$$\{ J1, J2, J3, \ldots\ldots. Jn \}$$

where n is the total number of jobs. The conditions that were introduced for the JSSP, hold for the flow shop scheduling problem as well. The minimisation of makespan is the objective in the classic formulation of the problem, which is also known as the $n/m/P/Cmax$ problem[18]. In the special case of m=1, the problem is described as the one-machine scheduling problem.

There are still several researchers in the field of evolutionary computation, who adopt the classic formulation of the problem and use the minimisation of makespan as the main objective of their algorithms [68]-[72]. However, in recent years, the majority of researchers considers more complicated formulations of the problem with a number of alternative optimisation criteria included. Murata et al. [73] use their Multi-Objective GA approach for a flowshop scheduling problem, aiming to simultaneously minimise makespan, total tardiness, and total flowtime of the production. Total tardiness is also used as an optimisation criterion by Lam et al. in [74]. Sikora [75] attempts to minimise makespan, holding costs (earliness), and overtime (tardiness), in a flow-line with limited buffer capacity. Sannomiya et al. [76,77], also try to minimise makespan, keeping at the same time the processing rate of each product as constant as possible. Their formulation of the problem considers the existence of a carrier that transfers products between the machines. Lee et al. [78,79] assign earliness and tardiness penalty weights to schedules, for a one-machine scheduling problem. Lee et al. [80] present an interesting formulation of the problem, introducing the concept of a flexible flow line with variable lot sizes. In this case jobs consist of splitable lots and an efficient GA is used to simultaneously optimise the ordering of jobs and the lot sizing. Gonzalez et al. [81] consider the 'no-wait' version of the job sequencing problem, where once the processing of a job has started in the first machine of the production line, there must be no-time delay between the consequent operations of the job at the following machines. A GA enhanced with heuristic methods is used for the solution of the problem. Herrman et al. [82] describe a class-one machine scheduling problem. In this case jobs belong to different classes, with each class having sequence-dependent set-up times. The evolutionary algorithm that they employ for the solution of the problem searches the problem's space instead of the solutions' space and a heuristic accomplishes the task of producing legal schedules. Karabati et al. [83] address the flowshop scheduling problem with controllable processing times, i.e. the problem where the processing time of a part is not fixed, but can assume a number of different values. A GA is employed for the solution of large scale problems of this type. Finally, Ishibuchi et al. [84] propose a fuzzy mathematical formulation of the problem, using the concept of fuzzy due dates. The optimisation criteria are the maximisation of the minimum satisfaction grade and the maximisation of the total satisfaction grade.

3.3 Encoding

The permutation representation in used in most of the papers that we have examined in this section. A permutation is a natural representation for the solution of the problem and there are a lot of efficient and well-tested operators to ensure the feasibility of solutions and to enhance the evolutionary process.

There are however some exceptions to this rule. The most notable is that of Lam et al. [74] who introduce a pigeon-hole coding scheme. This representation allows the use of traditional crossover and mutation operators without producing infeasible solutions. Some slight modifications in the encoding of solutions are also present in [80] and [75], in order to accommodate the simultaneous lot-sizing, which is attempted by these algorithms. Lee et al [78,79] use Parallel Genetic Algorithms (PGAs) with binary representation for one machine scheduling problems. Finally Kebbe et al. [85] adopt the

Vibrational-Potential Method (VPM) for the solution of sequencing problems. VPM is an evolutionary computation method based on the concept of information processing, thus it uses a different representation scheme.

3.4 Operators

There is a long-running debate about the suitability of particular crossover operators for sequencing problems. Michaelewicz [6] argues that the flowshop scheduling problem has certain characteristics that distinguish it from the TSP, thus the suitability of a particular operator for the TSP is not necessarily valid for all sequencing problems. Since most of the crossover operators have been developed for the TSP, it is easy to understand that the selection of a crossover operator for flowshop scheduling is not straightforward.

The perseverance of order, position and adjacency of genes are the main characteristics of an operator for sequencing problems. One of the oldest crossover operators is Order Crossover (OX) introduced by Davis [86]. OX preserves the relative order of jobs from the parent chromosomes. Kutoh et al. [87] use OX for the solution of a one-machine scheduling problem with uncertain processing times. Davis also introduced the uniform order-based crossover operator[88], which is adopted by Drake et al. [71] and by Lee et al. [78] in their attempts to solve job sequencing problems. Uniform order-based crossover preserves absolute position and relative order of jobs from the parent chromosomes. Researchers have proposed many variations of OX for the flowshop scheduling problem, like the one and two-point crossover operators introduced by Murata et al. [89,70]. Sannomiya et al. [76,77], and Fichera et al. [90] propose their versions of OX. Another famous TSP operator is the Partially Mapped Crossover operator (PMX), introduced by Goldberg et al. [91], which preserves elements of absolute order and relative position of jobs from the parent chromosomes. Chen et al. [69,92] use PMX in their attempt to solve a continuous flowshop scheduling problem. An interesting element of their approach is that they employ three powerful heuristics (Job Insertion Method (JIB)[93], Campbell, Dudec and Smith's (CDS) heuristic, and Dannenbring's heuristic) for the initialisation of the population. Shridhar et al. [94,95] also use the PMX operator together with their own DELTA operator (which determines the selection policy of the algorithm), in order to obtain optimal schedules for a flow-line based manufacturing cell. The same operator is present in the algorithm proposed by Braglia et al. [72], who enhance the evolutionary process by using a neighbourhood search algorithm for fine local tuning.

An increased number of researchers adopt the edge-recombination operator [95] and it's enhanced version [96]. It was originally deigned for the TSP and its main characteristic is that it preserves adjacency information from the parent chromosomes. It has recently been used by Sikora [75], Lee et al. [80] and Gonzalez et al. [81] in job sequencing problems.

Asveren et al. [98] propose two new operators for sequencing problems, namely Neighbourhood Relationship operator (NRX) and Meta-Ordering operator (MOX). NRX is in fact a neighbourhood search algorithm, as the MSFX operator proposed by Yamada et al. [99]. MSFX has also been used for the JSSP, as we saw in the previous section. Yaguira et al [100] propose a hybrid system based on GAs and Dynamic Programming (DP) for the solution of one-machine scheduling problems. DP is employed in the crossover phase of the algorithm, leading to a powerful and fast optimisation method.

The diversity of the operators used in all previous research papers is a result of the uncertainty that exists about the superiority of a particular operator. The use of independent test problems by each researcher, makes the comparison very difficult, as we will discuss in the next paragraph. Murata et al. [89,70] compare the performance of seven crossover operators and their results show the superiority of the two-point crossover operator. On the other hand Lee et al. [80] compare the edge-recombination, PMX, CX (Cycle crossover) [101] and OX operators and found that the edge recombination operator produces the best quality solutions.

The mutation operator is often given little importance in research papers, however its contribution to the best possible exploration of the solutions' search space is important for the evolutionary process. The most famous mutation operators for sequencing problems are the 'swap' operator, which simply exchanges the position of two randomly selected genes in the sequence, and the 'shift' operator which shifts the position of a gene some places to the left or to the right. Murata et al. [89] investigated the performance of five mutation operators on sequencing problems, and the best results were given by the 'shift' operator. Their research also proved that the combined effect of the best crossover and mutation operators is not necessarily positive. The results showed that the best performance is given by the combined effect of two mid-performance operators. This conclusion confirms the difficulty of selecting operators for job sequencing problems.

The selection of GA parameters (population size, probability of crossover, probability of mutation etc.) plays a vital role to the performance of the algorithm. Chen et al. [92] propose the introduction of a meta-level GA [102] that optimally controls the values of parameters, while the evolutionary process is in progress. Pakath et al. [103] also present an algorithm for the on-line specification of the parameters in a GA-scheduler.

3.4 Test problems and case studies

It is extremely difficult to compare the performance of different evolutionary algorithms for flowshop scheduling, since most researchers use their own instances of test problems, i.e. problems where the processing times and due dates of the jobs are selected randomly from a uniform distribution. Since these instances are not published in detail, they are rarely used by other researchers. Therefore, a comparison of results taken from that type of problems would not be valid.

Most of the papers referenced in this section use their own problem instances, or test problems not widely available. Exceptions are Reeves [68] and Yamada et al. [99] who present results on standard benchmark problems taken from Tailard [104], Lee et al. [80] and Sikora [75], who consider the scheduling of a manufacturing plant producing Printed Circuit Boards (PCB's) as a case study.

4. The Dynamic Scheduling Problem and Comparisons between different Scheduling Algorithms

4.1 Dynamic scheduling

The cases that we have addressed so far in job shop and flow shop scheduling considered static scheduling problems, i.e. problems where the dynamic nature of the scheduling decision is not examined. However, in practical scheduling, a scheduler often has to react to unexpected events. The main uncertainties encountered in a real manufacturing system, are the following:

- machine breakdowns including uncertain repair times
- increased priority of jobs
- change in due dates
- order cancellations

Whenever an unexpected event happens in a manufacturing plant, a scheduling decision must be made in real-time about the possible reordering of jobs. This process is known as 'rescheduling'. The main objective of rescheduling is "to find immediate solutions to problems resulting from disturbances in the production system" [116].

Until recently, evolutionary computation methods have rarely been used for dynamic scheduling, due to their inability to cope with real-time decision making. They were developed and tested on static scheduling problems that did not require real-time control. However, the last few years, evolutionary algorithms have been used as a part of hybrid dynamic scheduling systems, which exploit their useful characteristics.

Machine learning is one of the methods used in a manufacturing environment to face uncertainties. Chiu et al. [105] propose a learning-based methodology for dynamic scheduling. They divide the scheduling process in a series of ordered scheduling points. An evolutionary algorithm finds which dispatching rules performs better for each of these points, given a set of plant conditions (system status). The chromosome is formed by a series of genes, each one representing a respective scheduling point and taking as a value one of the available dispatching rules. The performance of the algorithm is simulated under different plant conditions, forming a knowledge-base that describes the scheduling rules that are preferable in different cases. A binary decision tree is used to describe the gained knowledge. This method has the advantage of being able to modify its existing knowledge (new system conditions), without having to reconstruct the entire knowledge-base. Aytug et al. [106] present a different machine learning approach for dynamic scheduling, based on classifier systems [3]. In this case, an initial knowledge base is given, and a GA modifies it, using results taken from the simulation of the production line. In that way the system learns to react to certain unexpected events. A hybrid system based on neural networks, GAs, and an inductive learning algorithm called Trace Driven Knowledge Acquisition (TDKA) [107], is used by Jones et al. [108]-[110], to infer knowledge about the scheduling process. A back-propagation neural network selects a number of candidates dispatching rules from a larger set of available rules. The schedules formed by these dispatching rules are used as the initial population of a GA that evolves an optimal schedule. The results taken from the simulation of the schedule help TDKA to create a set of rules that form the knowledge-base. Lee et al. [111] propose a hybrid scheduling framework which consists of an inductive learning system for job releasing in the plant, and a GA-based system for the dispatching of jobs at the machines. The Genetics-Based Machine Learning (GBML) method of Golberg [2], and a GA-based status selection method have also been used by Tamaki et al. [112] and Ikkai et al. [113] respectively, to induce scheduling knowledge from manufacturing systems.

Fang et al [114] adopted a different rescheduling strategy based of the rolling horizon optimisation method. Scheduling is performed periodically on a predefined number of jobs that form the 'job-window'. Rescheduling is initiated either by the elapse of a job-window or by the occurrence of an unexpected event. A GA evolves an optimal schedule for each planning horizon, considering the status of the system. The same concept of job-windows is adopted by Cartwright et al. [115] who employ a GA to dynamically control the scheduling of a chemical flowshop.

Finally Jain et al. [116] propose a steady-state GA for the scheduling of an FMS system. Specially designed algorithms deal with unexpected events like machine breakdowns and order cancellations. A series of test cases indicate the validity of the method for scheduling and rescheduling purposes.

4.2 Comparisons

Evolutionary computation is not the only non-analytical optimisation method that has been proposed for scheduling problems. Iterative improvement techniques, random search techniques, Simulated Annealing, Tabu Search and hybrid techniques are some well known scheduling optimisation methods. Tsang [117] gives an overview of Operations Research and Artificial Intelligence methods that have been used in scheduling problems. Many researchers have attempted to compare the performance of these optimisation methods, and results were recently published in a series of papers:

Dorn et al. [118] compare the performance of iterative deepening [119], random search, Tabu Search and GAs on the scheduling of a steel manufacturing plant in Austria. Iterative Deepening and Tabu Search produced the best results for this particular case study. The same techniques with the addition of a hybrid GA-local search method and Simulated Annealing were tested on a one-machine scheduling problem by Yaguira et al. [120]. Their conclusion is that while local search techniques are computationally efficient and produce moderate solutions, Simulated Annealing and Genetic Local Search perform much better but introduce a significant computational overhead. The one-machine scheduling problem is also used for the comparison of local search, Simulated Annealing, Tabu Search and GAs by McMahon et al. [121]. Simulated Annealing gives the best performance both in numerical and computational results.

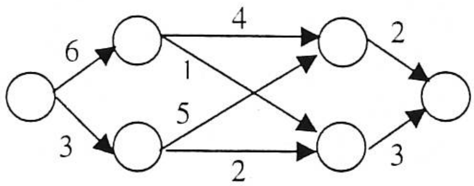
In all previous comparisons, specific representations and genetic operators were used in the evolutionary approaches, so we cannot generalise the conclusions made from the results. On the other hand, there are indications that the evolutionary process is greatly enhanced when it is hybridised with local search techniques. Additional evidence are given by Glass et al. [122] who compare the performance of multi-start descent, threshold accepting, Simulated Annealing, Tabu Search and GAs in a number of flowshop scheduling problems. While the performance of GAs is poor initially, it is greatly improved when the algorithm is hybridised with a local search method. The same results are reported by Ishibuchi et al. [84] for their fuzzy flowshop scheduling problem that we discussed in the previous section. They compare the performance of multi-start descent, Simulated Annealing, Tabu Search and GAs, and while Tabu Search outperforms all other individual optimisation methods, a hybrid multi-start descent-GAs system performs equally well.

5. Process Planning

Process planning is one of the most complex manufacturing phases. It comprises of a series of tasks that are heavily dependent on the type of product that is to be processed. Process planning takes as input the design characteristics of the product (CAD files) and gives as output its complete production plan. This plan should determine the machining processes needed, the tools that are going to be used, and the sequencing of operations. If more than one plans are available, then an optimal plan should be selected. Process planning can be more or less elaborate, according to the processing requirements of a particular part. Horvath et al. [123] list some elements of the process that should be determined by a process plan.

Process planning is the link between the design and the manufacturing phase of a product. The design phase is highly automated nowadays with the introduction of state-of-the-art Computer Aided Design (CAD) programs. However, Computer Aided Process Planning (CAPP) programs are not so highly developed and the research interest in the field is growing. An excellent review of the methods that have been used so far for CAPP can be found in [123].

Evolutionary computation methods have recently been used for handling independent tasks of process planning, especially for the optimal selection problem, which is the task of selecting an optimal process plan among a population of alternative plans. The problem is usually modelled with the help of flow-networks, i.e. a construction of arcs and nodes which determines alternative sequences of machining for a given product:



Each stage of this graph represents a machining operation and the nodes denote the number of alternative machines that are capable of performing this operation. The weighted arcs define the cost of following a particular machining sequence.

Awadh et al. [124] use an evolutionary algorithm for the solution of this problem. Each stage is represented with a binary-coded matrix, where the occurrence of a bit with positive value denotes the presence of a connection between the corresponding nodes of the matrix. This representation can sometimes lead to the existence of more than one processing plans for a single chromosome solution. An efficient algorithm called 'Path Modifier' ensures that there is a '1 to 1' relationship between the genotype and the phenotype of each solution. The objective of their approach is the minimisation of the overall cost. Zhou et al. [125] argue that there are fast and efficient algorithms like the shortest path method and dynamic programming which can give good solutions for single-objective process planning problems like the previous one. Evolutionary computation methods are ideal for the multi-objective version of the problem, which cannot be easily expressed as a shortest path or dynamic programming problem. They constructed a GA that uses the same network flow model but has an efficient integer solution representation that does not require the existence of additional operators like the 'Path Modifier'. Multi-objective optimisation is accomplished with the use of the Pareto-optimality approach[2].

Horvath et al. [123] describe a complete process planning procedure, from the input of part specifications in the form of CAD files, to the optimisation of the constructed process plan. They use an object-oriented approach in the form of 'features'. 'Features' are objects that define specific operations and contain all the relative functional, geometrical and technological data. Knowledge-based reasoning is used for the generation of plans, which are then optimised with the help of a genetic algorithm.

Concurrent Engineering has received a lot of attention lately, as a modern approach to manufacturing optimisation. It is a manufacturing philosophy where the design and the related manufacturing processes of a product are integrated into one procedure (the reader should refer to Singh [126] for an overview of concurrent engineering). Process planning and scheduling are two manufacturing processes closely related. One of the aspects of concurrent engineering is the integrated process planning and scheduling of a product. Husbands et al. [127], and McIlhaga et al. [128] propose a GA-based method for the simultaneous determination of planning and scheduling in a vehicle manufacturing company. They use a Distributed Genetic Algorithms (DGA) approach with a diploid chromosome representation, which defines both the sequencing of operations and the use of alternative machines. A number of different optimisation objectives drive the algorithm, like the minimisation of makespan, flowtime and tardiness.

Bowden et al. [129] has created a hybrid system called GUARDS, based on unsupervised machine learning and GAs, in order to optimise the control of a manufacturing process. The system learns to select the optimum process plan according to the status of the plant. GUARDS is an extension of the well known SAMUEL system, introduced by Greffenstette [130].

Hayashi et al. [131] introduce an interesting method for the evaluation of future plans, in a manufacturing plant with uncertain parameter values. A binary-coded GA is used to efficiently create scenario which are used for the evaluation task. The solution is represented by a string of all plant's parameters, and the objective of the algorithm is defined according to user's preferences.

Operation sequencing is an important task of process planning. The planner must determine the machining sequence of parts, independently of the resources that are available. A feasible process plan must take in account all the precedence constraints for the machining of features. These constraints are normally given in the form of a precedence graph. Usher et al. [132] propose an evolutionary computation approach for the solution of this problem. The number of genes in the solution are equal to the number of features that must be machined. There is a special decoding procedure based on the feature precedence graph, which transforms any string into a feasible sequence of machining operations. This representation was first proposed by Yip-Hoi et al. [133]. The quality of solutions is determined by the total number of set-ups, the continuity of motion and the loose precedence. Takatori et al. [134] adopt a TSP representation for the solution of the problem, and use a repair mechanism to cope up with solutions that violates the constraints. The objectives of their algorithm are the minimisation of the total change cost, the machining cost and the non-machining cost.

Kamhawi et al. [135] developed an elaborate feature sequencing system based on GAs. The representation scheme is the same as in [134], but the evaluation of solutions is based on rules and constraints about safety, quality, and minimisation of tool changes and tool travel. The user assigns a weight to each of these objectives, according to his preferences.

Norman et al. [136] discuss the problem of operation sequencing and tool allocation in Parallel Machine Tools (PMTs).A PMT is a machine capable of processing more than one part at a time, since it contains multiple spindles. A random-keys coded GA is proposed for the solution of the problem. The tool allocation task is dealt with the introduction of an integer part to the value of the genes. This part defines the machining unit (MU) which is responsible for a particular operation. The decimal part of the value determines the sequence of operations. The authors also propose the enhancement of the algorithm with a heuristic method, presenting results that justify their decision. Yip-Hoi et al. [133] tackle the same problem using an efficient solution representation based on feature precedence graphs, as was discussed earlier. The objective of their algorithm is the minimisation of the part's total processing time.

6. Optimisation Problems in Cellular Manufacturing

6.1 Introduction

Cellular Manufacturing is the application of Group Technology (GT) in manufacturing systems. GT was first introduced in the former USSR by Mitrofanov [137], and was popularised in the west by Burbidge [138], who introduced Production Flow Analysis (PFA) the first scientific method for creating manufacturing cells. Cellular manufacturing is a manufacturing philosophy that attempts to convert a manufacturing system into a number of cells. Each cell manufactures products with similar processing characteristics. Ideally all the processing operations of a part should be completed within a cell. However, in realistic cases, intercell movements of parts are always present. Cellular manufacturing offers certain advantages to mid-variety, mid-volume production lines like the reduction of set-up and transfer costs, the minimisation of inventory, improved quality and significant savings in search space.

A vast bibliography exists on the subject of cellular manufacturing. A good introduction is given by Burbidge [139], but a more realistic view of up-to-date developments can be found in [140]. A considerable number of industries has adopted the concept of cellular manufacturing as Wemmerlov and Hyer illustrate in a series of papers [141]-[144].

There are three main phases in the design of a manufacturing cell: (i) the grouping of machines into cells, better known as the cell-formation problem, (ii) the layout of cells in the plant and (iii) the layout of machines within the cells. The implementation of each of these stages leads to difficult optimisation problems, where traditional optimisation methods are incapable of finding optimal solutions in reasonable time. In the following paragraphs we will examine some evolutionary methods that have recently been used to tackle optimisation problems associated with cellular manufacturing.

6.2 Formation of manufacturing cells

The formation of manufacturing cells is an optimisation problem that has been extensively researched the last twenty years. A considerable number of alternative methods have been proposed for the solution of the problem. Singh [145], Offodile [146] and Morad [147] give comprehensive reviews of the problem, and attempt to taxonomies all these methods into certain categories. Analytical review of the methods are beyond the scope of this report. However, it is important to reference the most significant of them. As already mentioned, Burbidge [138], the pioneer researcher in cellular manufacturing, introduced the first method of designing manufacturing cells, namely Production Flow Analysis. The method aims to create manufacturing cells by a series of manual manipulations on the rows and the columns of the machine-component matrix. The majority of modern methods, like Rank-Order Clustering (ROC) [148] and Direct Clustering Algorithm (DCA) [149] is based on the same matrix. Another famous cell-formation method is Single Linkage Cluster Analysis (SLCA), introduced by McAuley [150], which is based on similarity coefficients between the machines. Coding and classification methods [151], graph-partitioning [152], mathematical programming [153], neural networks [154], and fuzzy logic [155], are some other methods that have been proposed for the solution of the cell-formation problem.

Unlike scheduling, the cell-formation problem had not been a subject of evolutionary computation research until very recently. Venugopal et al. [156] were the first researchers to approach the cell-formation problem using GAs. Their objective was the minimisation the intercell traffic and the balancing of load in the cells. They employ a different population of solutions for each of the objectives. The solution representation is simple and efficient. Each machine in the plant corresponds to a gene in the chromosome. The value of the gene defines the cell where the respective machine belongs. The total number of cells in the plant is predetermined, but the formulation of the problem considers the processing time of parts, which is a serious improvement in comparison with the traditional cell-formation methods. Gupta et al. [157,158] enhanced this formulation by considering the intracell moves of the parts and the intracell layout. Special care was also taken to ensure that no cell remains empty during the evolutionary process. Billo et al. [159] adopted a direct solution representation, based on a two-part chromosome. The first part is a permutation of all parts to be processed, while the second part denotes the cut-off points of the first part. Each segment between cut-off points denotes a part-family. The objectives of their algorithm is the maximisation of machines similarity within the cells and the minimisation of the total number of cells. The advantage of this method is that the total number of cells is not predefined, but the structure of the chromosomes is quite complex and computationally exhaustive. However, the algorithm performs well on a series of test problems, including some ill-structured machine-component matrices. Joines et al. [160] introduce a new efficient integer programming formulation of the problem, which reduces the search space significantly. An evolutionary algorithm is then employed for the solution of the problem, with the variables of the mathematical formulation coded into the chromosome. Only the upper bound of the total number of cells needs to be specified. The objective of the algorithm is the minimisation of exceptional elements and voids (zero's in the diagonal) in the machine-component matrix. The validity of the method is depicted by results in test problems taken from the literature. Su et al. [161] use the classic Venugopal's solution representation, but their chromosome also accommodates the existence of multiple machines of the same type. Morad et al. [162] propose the simultaneous optimisation of several objectives, using a weighted-sum approach. Pierreval et al. [163] adopt the classic representation scheme, with binary-coded genes. Suer [164] presents a preliminary discussion on the design of part families using evolutionary programming [5]. Dimopoulos & Zalzala [165] propose an evolutionary algorithm for the cell-formation problem of a pharmaceutical company. The representation of the solution and the operators are purpose-based. Different multi-objective optimisation methods are compared on the solution of the problem.

Since the formation of cells is a difficult optimisation problem, some researchers have proposed ways of enhancing the evolutionary process. Paris et al. [166] use Distributed Evolutionary Algorithms, attempting to increase the speed of the process in comparison with the methods used so far, which are in their own words "notoriously slow". Hwang et al. [167] formulate the problem using a quadratic assignment mathematical model. The representation of solutions is a permutation of all machines in the plant-each one uniquely identified by a number. A greedy heuristic is then employed to assign machines to cells. The authors use a number of comparative measures to evaluate the performance of the method in various test problems. Finally, Zhao et al. [168] present a fuzzy clustering method for the solution of the problem, which takes in account the uncertainty and imprecision that usually exists in the problem data. Fuzzy clustering is implemented using a GA that employs fuzzy c-partitions as individual chromosomes. This method is a typical example of hybrid systems that exploit the positive characteristics of individual algorithms and result in robust optimisation methods.

6.3 Cell layout and machine layout optimisation methods

Once the configuration of cells has been determined, the designer must define the layout of machines inside the cells, and the layout of cells in the plant area. These optimisation problems belong to the general category of the Facility Layout Problem (FLP). The FLP is a well-known combinatorial problem. It has been formulated as a quadratic set covering problem, linear integer programming problem, mixed integer programming problem and graph theoretic problem [169]. However, the Quadratic Assignment Problem (QAP) formulation is the most popular in the literature and since QAP is known to be NP-complete for most problem instances, efficient algorithms must be used for the solution of the problem.

Several researchers have used evolutionary algorithms to tackle FLP problems in manufacturing. Early approaches can be found in the survey given by Mavridou et al. [170]. Recently Tate et al. [171] attempted to solve the FLP using GAs. They adopt the QAP formulation of the problem with the objective of minimising the sum of products of total material flow and rectilinear distances between the departments. They propose a flexible-bay layout structure that accommodates unequal sizes for the departments. The plant is divided into a number of bays by end-to-end slices in one direction, and then the bays are split into departments by perpendicular slices. A permutation representation of the solution is used, which determines both the allocation of departments in the layout and the place of bay-divisions. Norman et al. [172] enhanced this representation by using a random-keys GA-thus avoiding feasibility constraints - and by incorporating uncertainty in the mathematical formulation of the problem. Material handling costs are expressed using expected values and standard deviations for the product volume over time. Suresh et al. [173] adopts the permutation representation for the solution but uses a much simpler grid-structure for the layout. Kazerooni et al. [174] propose an integrated approach for the design of manufacturing cells, which incorporates steps for the simultaneous determination of cell and machine layouts.

Banerjee et al. [175] model the problem using a mixed-integer programming formulation. They propose a complex but powerful representation based on nodes and edges. Nodes correspond to input-output cell stations and edges correspond to material flows between the stations. The layout structure is continuous, thus much more flexible than the grid and bay structures which restrict the shape of cells. Genetic search is employed as a part of the overall algorithm, aiming to transform the problem into a series of iterative linear programming problems. The robustness of the method is illustrated in a number of test cases taken from the literature, where it is shown to outperform traditional methods.

Conway et al. [176] consider an interesting version of the FLP, the dynamic FLP. In this case, the facility layout changes with time, and the algorithm must find the best allocation of facilities over an entire planning horizon. The authors propose a GA for the solution of the problem. The layout is represented by a multi-part chromosome, with each part corresponding to a planning period. The position of a gene is a fixed place in the layout, and the value of the gene denotes the facility that occupies this place for a particular period. The objective of the algorithm is the minimisation of layout rearrangements costs and materials flow costs over the entire planning horizon.

The papers that we have reviewed so far in this paragraph, introduce methods that normally apply to the cell layout problem. The machine layout problem is a special type of FLP and it is usually addressed individually, because various assumptions that are made for the FLP are not valid for this problem. Bazargan et al. [177] discuss some of these assumptions, such as the equal-sized areas and the a-priori knowledge of facilities locations. However, elaborate continuous plane FLP methods like [175] can easily be applied to the machine layout problem.

Manufacturing practice usually restricts the search for an optimal intra-cell layout to a small number of fixed configurations, like the single-row layout, the multi-row layout, the semi-circled layout and the loop layout. Braglia et al. [179] use a GA in order to find the machine layout in a pre-fixed single-row structure. The objective of the algorithm is the minimisation of the distance travelled by the material handling device of the cell. The solution is represented by a permutation of all machines in the row. This method performs well on large problem instances, in comparison with heuristic approaches. In a similar paper Braglia et al. [179] use the minimisation of jobs backtracking as the objective of the algorithm. They also present an interesting hybrid method for the solution of the same problem[180], where a GA is employed for the optimisation of simulated annealing parameters. Cheng et al. [181] address the loop machine layout problem using two different objectives: the minimisation of the total number of reloads for all products (minsum problem) and the minimisation of the maximum number of reloads for all products (minmax problem). The layout is considered to be unidirectional and there is a single loading-unloading station. The solution is once again represented by a permutation, and the PMX operator is used for crossover purposes. Gen et al. [182] introduce a hybrid fuzzy-GA approach for the solution of complex multi-row machine layout problems. The objective of the algorithm is the minimisation of travel cost between the machines, and the solution is represented by a multi-part chromosome, which contains information about the total number of rows, the permutation of machines in each row and the clearances between the machines. Fuzzy sets are used for the representation of the uncertainty that exists in the value of clearances.

Finally, we should note that Bolte et al. [183] have addressed the QAP problem using Simulated Annealing. The connection of this paper with evolutionary computation is that genetic programming is employed for the optimisation of the annealing schedule. The system finds good solutions while maintaining acceptable run times. This is one of the few examples where genetic programming has been used for a problem connected with manufacturing optimisation.

7. Optimisation of Assembly Lines

Assembly lines are widespread in manufacturing plants. A number of optimisation problems are associated with assembly lines, like the assembly sequence planning problem, the sequencing of mixed model assembly lines and the assembly line balancing problem.

A variety of evolutionary computation methods have recently been proposed for the solution of assembly line optimisation problems. The Assembly Sequence Planning Problem (ASSP) is tackled by Sebaaly et al. in a number of papers [184]-187]. It is the problem of finding an optimal sequence of assembling a product that consists of N parts, given its design characteristics. An assembly sequence is feasible if it does not violate the assembly rules and constraints that are defined by the designer. The authors propose a GA approach for the solution of the problem, where an individual chromosome is a randomly constructed sequence of parts. An efficient mapping procedure is introduced, which transforms any random assembly sequence into a feasible one. Gropetti et al. [188] analyse the assembly planning procedure and use a GA in order to obtain a clear contact relational graph.

It is often the case that several products with similar characteristics (models) are assembled in a single line. This type of line is called mixed-model assembly line. The sequencing of models in mixed-model assembly lines is an important task, especially if we wish to apply the JIT principle in the production line. There are a number of objectives associated with this task [189], like the minimisation of line's length, the minimisation of total utility work, and the minimisation of the variability of parts' consumption (vpc). This latter objective is critical in JIT systems. Leu et al. [190] address the problem of sequencing in a mixed-model assembly line with the objective of minimising vpc in a JIT production system. A GA is used for the solution of the problem, with each chromosome representing a sequence of models to be assembled. The sequence is cyclic, and the number of individual models in each sequence is fixed. The method performs better on some test problems than the traditional Toyota's Goal Chasing Algorithm (GCA) [191], which is often used in JIT production systems. Kim et al. [189] adopt the same representation for the sequencing of a mixed model assembly line, where the objective is the minimisation of the total length of the line.

Another well-known optimisation problem of assembly lines, is the assembly line balancing problem. Given n workstations and m parts to be assembled, the assignment of parts to workstations should be defined according to certain optimisation criteria. Two versions of the problem are usually considered: the first version aims to minimise the total number of workstations in the plant given a fixed cycle time, while the second version aims to minimise the cycle time, given a fixed number of workstations. Secondary objectives like the minimisation of balance delay and the minimisation of probability of line stoppage are also considered. Suresh et al. [192] give an excellent literature review of the assembly line balancing problem. They also propose a GA approach for the solution of a similar problem where the objective is mainly the minimisation of the smoothness index of balance delay. The solution is represented by a list of sets with length equal to the total number of workstations. Each set contains one or more jobs. All the initial solutions are feasible and special operators are used to ensure the feasibility of solutions. The authors also present an alternative version of the algorithm, where a number of infeasible solutions is allowed in the population. This particular version works well on large problem instances. Rubinovitz et al. [193] also use a GA for an assembly line balancing problem. Their solution representation is a permutation of all parts, divided into a number of sections equal to the total number of workstations. Initially, random sequences are constructed, and then special mechanisms are employed to reorder the sequences according to the precedence constraints and to divide them in an appropriate number of sections. Tsujimura et al. [194] present an interesting method for solving the assembly line balancing problem based on GAs and fuzzy logic, aiming to minimise the balance delay. The solution representation is a classic permutation, which takes in account all precedence constraints. The processing time of each job is not deterministic, but is defined by a fuzzy set. The allocation of jobs to workstations is accomplished using the GA sequence, fuzzy sets, and a the following standard predefined maximum completion time. Starting with the first job of the sequence, the fuzzy sets of processing times are added, until the upper limit of the sum of fuzzy sets becomes bigger than the predefined maximum completion time. The set of jobs that comprises the sum is assigned to the first workstation, and the procedure starts again from the next job after this set in the sequence. Special mechanisms and operators ensure the feasibility of solutions.

A number of secondary optimisation problems in assembly lines have also been the subject of evolutionary computation research. Among them, the optimisation of buffer sizes between workstations in an assembly system [195], the scheduling of multi-level assemblies [196] and the scheduling of flexible assembly systems [197]. Finally Watanabe et al. [198] have used a GA in order to solve the generalised line balancing problem.

8. Design Optimisation Problems

Design is a complicated and time consuming phase in the development of a product. Although most of the times design is not directly addressed as a manufacturing optimisation problem, it constitutes one of its most critical aspects, since it constraints irretrievably the manufacturing process. Every design must be faultless and properly optimised, otherwise the result will be huge redesign costs. Enormous effort has been devoted to the development of efficient CAD systems in order to simplify and speed up the design process. Evolutionary computation methods have been applied successfully to complex design optimisation problems. In the following paragraphs we will review some of the recent papers in this field.

Traditionally, the design process starts with the creation of a mathematical model for the product that is to be manufactured. The model is then implemented as a computer program, allowing the designer to explore the effects of altering the values of the parameters. This optimisation process is usually implemented on a 'trial and error' basis. The incorporation of a GA in the heart of the design process, enhances and automates the procedure of parameter optimisation [199].

Cao et al. [200] use this method in a mechanical design optimisation problem: a number of design variables need to be optimised, subject to certain constraints. Continuous, binary, integer and discrete variables are included in the mathematical model, a condition that makes the optimisation procedure even harder. The authors adopt an evolutionary programming [5] approach for the solution of the problem. The solution is a string of all design variables, initialised within the constraints, while a special mutation procedure is used for each type of variable. Two design problems are used to illustrate the method, the design of a gear train and the design of a pressure vessel. The algorithm performs equally well or better in comparison with other optimisation methods like the branch & bound algorithm and simulated annealing. Rasheed et al. [201] propose a GA for the solution of a similar parameter optimisation problem which involves only continuous variables. The solution is a string of all parameters that need to be optimised, initialised within their constraints. Feasibility problems are accommodated using a penalty function. The evolutionary process is enhanced with the introduction of two crossover operators namely line crossover and guided crossover, which produce an offspring on the line connecting the parent chromosomes, considering the solutions' search space. The algorithm was tested on two complex design optimisation problems, the design of a supersonic transport aircraft and the design of a supersonic missile inlet. The method performs much better on these problems than a classic binary-coded GA and a sequential quadratic programming method. Coello et al. [202] give a nice extension to the use of GAs for parameter optimisation, by incorporating the weighted-sum multi-objective optimisation method to the evolutionary process. It is a realistic extension since conflicting objectives always exist during the design phase of a product. The method is tested on the design of an I-beam and a machine tool spindle, considering multiple conflicting objectives.

It is often the case that design optimisation problems are quiet complicated, involving a series of tasks that are highly related to each other. In these circumstances, the problem is normally divided into a series of sub-problems, each one comprised by a certain number of tasks. The optimal decomposition of multi-disciplinary optimisation problems and the optimal ordering of tasks within each sub-problem are considered by Altus et al. [203] using an evolutionary algorithm. The objective is the minimisation of the total length of feedback lines between the tasks. The representation of the solution is a permutation of all tasks involved and a break character is used to divide the string and thus the problem into a series of sub-problems. The system is called AGENDA (A GENetic algorithm for Decomposition Analysis) and it performs well on some decomposition problems taken from the literature.

Thornton et al. [204] developed an integrated software tool called CADET (Computer Aided Design Embodiment Tool) which supports the embodiment phase of the design process, i.e. the creation of a geometrical model of the product, according to the designer's specifications. CADET takes these specifications as input and finds a geometrical model that satisfies the constraints. GAs are proposed as an option in the constraint satisfaction part of the system.

Carlson [205] uses a GA to optimally select components for catalogue design processes. In catalogue design, a system is constructed from off-the-shelf components. The GA solution is a string comprised of all types of components used in the design. The value of a gene determines the component selected out of all possible components available for this type. A penalty function handles the violation of constraints. The applicability of the algorithm is demonstrated using the design of a hydraulic system and the design of a thermal fluid system.

Finally Iannuzzi et al. [206] address the problem of optimally allocating tolerances on product dimensions, in order to minimise the total production costs. The authors tackle the problem using a GA-based method, which shows satisfactory results on a series of test problems.

9. Other Manufacturing Optimisation Problems

In the previous sections we reviewed recent papers in the field of evolutionary computation for some standard manufacturing optimisation problems. However, these are not the only optimisation problems associated with manufacturing. The purpose of this section is to illustrate some recent evolutionary computation developments in various manufacturing areas.

The efficient auto-tuning of PID controllers is a significant optimisation problem in the field of process manufacturing. Despite the fact that the problem has been well-researched by control-engineers, the traditional Ziegler & Nichols tuning rules [207] are still being used in practice. Evolutionary computation methods provide the means of efficient tuning, since a solution representation based on PID parameters can be easily constructed. This potential has been recognised by a considerable number of researchers who have used evolutionary algorithms to tune PID controllers. Jones et al. [208] built a GA-based system which initially identifies the process model and then uses this model to tune off-line the parameters of the controller. The same authors [209] propose an evolutionary technique for the design of robust S.I.S.O. Smith Predictor PID controllers. Jones et al. [210] tuned the parameters of a digital PID controller using an evolutionary algorithm. A co-evolutionary model is proposed by Jones et al. [211] for the design of robust PID controllers. Krohling [212] presents a GA which optimises a PID controller for disturbance rejection. Vlachos et al. [213] extend the concept of genetic tuning to PI controllers for multivariable processes. The parameters of all controllers are simultaneously coded in one solution representation. Salami et al. [214,215] introduce a hardware-implemented GA (GA processor) which tunes the parameter of a PID controller. The authors also present encouraging results taken from experiments with multiple GA processors.

Qi et al. [216] use an evolutionary algorithm to optimally tune the parameters of a fuzzy logic controller (FLC) which has been designed for high order processes. Kim et al. [217] address the same problem using Hierarchical Distributed GAs (HDGAs). HDGAs are multilevel hierarchical systems that are constructed from local hybrid GAs-expert system structures. These structures are organised in levels, and in each level the problem is solved on a different degree of abstraction. The advantage of the systems is its ability to dynamically change its structure in order to explore promising regions of the search space. Tarng et al. [218] employ a binary-coded GA for the design of an optimal fuzzy logic controller which is used in turning operations.

The identification of process models is essential for the optimal control of manufacturing systems. Polheim et al. [219] use genetic programming in order to identify the model of a manufacturing process. Common control engineering tools, like transfer function blocks are used for the creation of trees (programmes). In this way, the algorithm provides structured process models, giving the control engineer a useful insight of systems' internal configuration. Test problems validate the performance of the method and especially its ability to generalise. McCay et al. [220] also employ genetic programming for system identification, constructing the trees with common mathematical functions. There are also a number of researchers who address the problem of system identification using GAs. Among them Reeves et al. [221] propose an interesting method where the solution is coded in terms of the radii and angles of poles and zeros of the transfer function. The values of these variables are constrained within the stability regions, thus the final solution is guaranteed to be stable.

Xia et al. [222] tackle the problem of optimal design and synthesis of chemical batch plants using a stochastic optimiser called EASY, which is a combination of evolutionary algorithms and simulated annealing. The batch scheduling problem is also addressed by Morad & Zalzala [162] with the help of an evolutionary algorithm.

All the equipment in the plant is interconnected with various kind of pipes. The plant pipe-route optimisation problem aims to find the minimum length of pipe interconnections in the plant, which satisfies all the requirements. Kim et al. [223] employ a GA with sterner points representation for the solution of the problem. The method works well in a series of test problems.

In a pull (JIT) production system, the demand must always be satisfied without the help of excessive stocks. The total number of kanbans in the plant and the corresponding production trigger values should be optimally defined in order to achieve this objective. Bowden et al. [224] address this problem using an evolutionary algorithm seeded with the optimal solution of the Toyota equation [191].

The failure of machines in the plant is inevitable. Shop-floor engineers aim to diagnose the failure of a machine as quickly as possible. They normally use a number of symptom parameters that are sensitive to changes of specific signals from the plant. Chen et al. [225] describe some of these parameters and propose an evolutionary approach for the determination of an optimal sequence of symptom parameters. Their method resembles genetic programming, in terms of the tree structures that are used as individual chromosomes. Petrovic et al. [226] present a hybrid method for machine-noise diagnosis, which is based on neural networks, expert systems and GAs. Guzman et al. [227] developed a hybrid Bayesian networks - GAs system that performs on-line monitoring and failure diagnosis, based on data taken from the plant.

Maintenance scheduling is another important process in the shop-floor, since the disruption of the production process must be as little as possible, but on the same time the machines must work without failures for the longest time possible. Kim et al. [228] propose an interesting hybrid of GAs and simulated annealing for optimal maintenance scheduling. The acceptance probability of simulated annealing is used for the survival of the less fit offspring in the population.

Quality control is an important aspect of modern manufacturing. The optimal allocation of inspection stations in the plant ensures that products are manufactured according to the quality criteria set by the management team. Viswanadham et al. [229] address this problem in a multi-stage manufacturing system and employ an evolutionary algorithm to optimally locate inspection stations. The solution is binary coded, with each gene representing a manufacturing stage. The presence of a station at a particular stage is denoted by a positive value. Patro et al. [230] designed a system which performs statistical processing control using neural networks and evolutionary computation. The neural network identifies the process model, and the evolutionary algorithm adjusts the control parameters in order to obtain the desired quality performance. Lu et al. [231] present a GA-based system which optimises the motion of a co-ordinate measuring machine used in inspection systems. A permutation representation is used for the solution of the problem, with each gene corresponding to a testing point that the measuring machine must visit. The algorithm finds the optimal sequence of visiting points that minimises the total length of the inspection path.

We will now discuss some advanced manufacturing optimisation problems, which have been the subject of evolutionary computation research. Mak et al. [232] consider the problem of designing an optimal integrated production-inventory-distribution system, aiming to minimise the overall costs, which include inventory holding costs, delivery costs, manufacturing costs and shortage costs. An evolutionary algorithm is used for the solution of the problem. Integer programming formulation is adopted and the solution is represented using the variables of the model. A similar problem is addressed by Disney et al. [233] about the control of a production and inventory system. Transfer functions are used for the modelling of the problem, illustrated in the form of block diagrams. The solution of the problem is represented by the variables of the transfer function, and a fitness measure is designed based on stock reduction, production robustness and inventory recovery. A difficult decision that the marketing team often has to face is the location of inventory centres for the accommodation of department stores, and the allocation of an inventory centre to each of these stores. This difficult location-allocation problem is formulated as a non-linear mixed-integer programming problem and solved by Gong et al. [234] using an evolutionary approach for the location task, and a Langrangian relaxation method for the allocation task. Aggregate production planning is a high level decision making procedure which takes as input product capacities and forecast demands, and produces aggregate production plans. Stockton et al. [235] address this problem using a binary-coded GA. The algorithm determines the amount of resources needed each month in order to meet the demand. The resources are expressed in the form of overtime, subcontracts and stock. Wang et al. [236] formulates the same problem using a fuzzy linear programming model. They employ Zimmerman's tolerance approach [63] to transform the problem into a linear programming model. The variables of the model form the chromosome of an evolutionary algorithm that is used for the solution of the problem. Feng et al. [237] address the problem of joint marketing/production decision making aiming to maximise the net profit of a company. The decision problem consists of the promotion problem for the marketing department and the production problem for the manufacturing department. Each problem is formulated mathematically and a respective number of GAs are employed to find optimal solutions. The decision variables of the mathematical models are used for the representation of solutions. Garavelli et al. [238] consider the production planning problem of a multinational company that owns manufacturing plants all over the world. Parameters like local market demands and independent capacities must be taken in account in the formulation of the problem. A GA defines which plants will be activated for production and the timing of activation.

The dynamic lot-sizing problem in a multi-stage, multi-item production system is described by Jinxing [239]. He proposes an evolutionary programming approach with binary representation for the solution of the problem. The objective is the minimisation of set-up, production and inventory costs.

Rao et al. [240] developed a new entropy measure which determines the optimal timing for reconfiguration in a manufacturing plant, and presented a genetic framework for the design of manufacturing systems. Kubota et al. [241] describe an application of their Virus Evolutionary GA (VEGA) approach to the self-organisation of a cellular manufacturing system. The same problem was tackled by Kawauchi et al. [242] with the help of a conventional evolutionary algorithm. Zhao et al. [243] address the problem of robot selection and workstation assignment in a Computer Integrated Manufacturing (CIM) system. A bin-packing formulation of the problem is proposed and a GA is employed for the solution of the problem. The solution is represented by a diploid chromosome that accommodates both parts of the problem.

Finally, evolutionary computation applications are also reported for the problems of vehicle distribution scheduling [244], warehouse scheduling [245], sequencing optimisation in automotive manufacturing industries [246], optimal control of spinning processes [247], design of flexible electronic assembly systems [248], optimisation of textile processes [249] and optimisation of area loss in flat glass cutting [250].

10. Conclusions

It is now obvious that the use of evolutionary computation methods for manufacturing optimisation is expanding. The number of papers published is increasing rapidly, and research covers a wide range of manufacturing problems. The amount of work itself indicates that evolutionary computation methods have established themselves as a powerful optimisation technique in the manufacturing field, despite the fact that they are not based on stable mathematical foundations.

Evolutionary computation research has been criticised for the consideration of artificial test problems that are much simpler than real-life manufacturing cases. Our study shows that researchers have reacted to this criticism by considering realistic problems taken from manufacturing plants. This move has also been triggered by the low response of evolutionary computation in manufacturing practice. It is encouraging to report recent projects where companies have adopted evolutionary computation methods in their plants. The gap between academic research and manufacturing practice is not a problem restricted to the field of evolutionary computation. However, in our case, there are a number of additional reasons that make the approach harder:

- The terminology of evolutionary computation is vague for the manufacturing engineer. Despite the fact that the driving logic of evolutionary algorithms is amazingly simple and efficient, the terminology inherited from genetics predisposes manufacturing engineers to think the opposite.
- Evolutionary computation is a relatively new technique, still in development. Mathematical foundations exist only for the simple GA case, and there are no universally accepted methods for the determination of technical parameters like population size, probability of applying operators etc. There is no guarantee that an algorithm will converge to an optimal or near-optimal solution.
- There is no standard evolutionary computation toolkit that can be used easily by manufacturing people who are not familiar with evolutionary concepts.

On the other hand evolutionary computation methods offer for the first time solutions which combine computational efficiency and good performance. This significant feature will certainly continue to attract the interest of engineers.

Researchers in the field of evolutionary computation are trying to enhance the robustness of their algorithms by hybridising them with other optimisation methods like local search techniques, simulated annealing, tabu search, neural networks and expert systems. The number of papers introducing hybrid systems is growing, an indication that there is a trend toward this direction.

This reports highlights the lack of genetic programming applications in the manufacturing field. The authors are currently investigating the potential use of genetic programming in a series of manufacturing optimisation problems.

11. References

[1] J.H Holland, *Adaptation in Natural and Artificial Systems*.Ann Arbor, Univ.of Michigan Press, MI, 1975.

[2] D.E.Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[3] L.B.Booker, D.E.Goldberg and J.H.Holland, "Classifier systems and genetic algorithms", in *Machine Learning: Paradigms and methods*, pp.235-282, J.G. Carbonnel (ed.), MIT Press/Elsevier, MA, 1989.

[4] I. Rechenberg, Evolutionsstrategie: *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, Germany, 1973.

[5] L.J.Fogel, A.J.Owens and M.J.Walsch, *Artificial Intelligence through Simulated Evolution*. Wiley, New York, 1996.

[6] Z.Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, 1992.

[7] J.R.Koza, Genetic Programming: *On the programming of Computers by Means of Natural*

Selection. MIT Press, Cambridge, 1992.

[8] T.Bäck,U.Hammel and H.P. Schwefel, "Evolutionary computation: comments on the history and current state", *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, pp.3-17, 1997.

[9] M.Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.

[10] M.Garey and D.Johnson, Computers and Intractability: *A Guide to the Theory of NP-Completness*. W.H.Freeman, San Francisco, 1979.

[11] S.Kirkpatrick, C.D.Gelatt Jr. and M.P.Vecchi, "Optimisation by Simulated Annealing", *Science*, vol.220, pp.671-679, 1985.

[12] F.Glover, "Tabu Search - Part I", *ORSA Journal on Computing*, vol.1, no.3, pp.190-206, 1989.

[13] F.Glover, "Tabu Search - Part II", *ORSA Journal on Computing*, vol.2, no.1, pp.4-32, 1990.

[14] F.Glover, "Tabu Search: a tutorial", *Interfaces*, vol.20, no.3, pp. , 1990.

[15] L.Davis, "Job shop scheduling with genetic algorithms", in *Proc. of the 1st Int. Conf. on Genetic Algorithms and their Applications*, pp.136-140, J.J.Grefenstette (ed.), Lawrence Erlbaum, Hillsdale, NJ, 1985.

[16] A.K.Jain and H.A.Elmaraghy, "Single process plan scheduling with genetic algorithms", *Production Planning & Control*, vol.8, no.4, pp.363-376, 1997.

[17] T.Yamada and R.Nakano, "A genetic algorithm applicable to large scale job shop problems", in *Proc.of the 2nd Int.Conf.on PPS from Nature*, pp.281-290, Elsevier Science Publishers, North Holland, 1992.

[18] S.French, *Sequencing and Scheduling: An Introduction to the Mathematics of Job - Shop*. Wiley, New York, 1892

[19] C.Bierwirth, D.C.Mattfeld and H.Kopfer, "On permutation representations for scheduling problems", in *Proc.of the 4th Int.Conf.on PPS from Nature*, pp.310-318, Springer-Verlag, Berlin, Germany, 1996.

[20] P.C.Chu and J.E.Beasley, "A genetic algorithm for the generalised assignment problem", *Computers & Operations Research*, vol.24, no.1, pp17-23, 1997.

[21] H.Fang, P.Ross and D.Corne, "A promising hybrid GA/heuristic approach for open-shop scheduling problems", in *ECAI'94: 11th European Conf. on Artificial Intelligence. Proceedings*, pp.590-594, Chicester, UK, 1994.

[22] H.Cao, H.Xi, Y.Luo and S.Yang, "GA with hierarchcal evaluation: a framework for solving complex machine scheduling problems in manufacturing", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.326-331, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[23] S.S.Panwalker, R.A.Dudec and M.L.Smith, "Sequencing research and the industrial scheduling problem", in *Symposium of the Theory of Scheduling and its Applications*, pp.29-38, S.E.Elmaghraby (ed.), Springer-Verlag, Berlin, 1973.

[24] V.Parunak and W.Fulkerson, "GA's and production scheduling", *Genetic Algorithms Digest(electronic version)*, vol.8,no.8, available from ftp://ftp.aic.nrl.navy.mil/pub/galist/digests/v8n8, 1994.

[25] M.Gen, Y.Tsujimura and E.Kubota, "Solving job shop scheduling problems by genetic algorithms" in *Proc.of the 1994 IEEE Int.Conf. on Systems, Man & Cybernetics*, pp.1577-1582, IEEE, New York, NY, USA,1994.

[26] U.Dorndorf and E.Pesch, "Evolution-based learning in a job-shop scheduling environment", *Computers & Operations Research*, vol.22, no.1, pp.177-181, 1995.

[27] K.Mesghouni, S.Hamadi and P.Borne, "Production job-shop scheduling using Genetic Algorithms", in *Proc.of the 1996 IEEE Int.Conf. on Systems, Man & Cybernetics: Part 2*, pp.1519-1524, IEEE, Piscataway, USA, 1996.

[28] H.L.Fang, D.Corne and P.Ross, "A Genetic Algorithm for job-shop problems with various schedule quality criteria", in *Evolutionary Computing. AISB Workshop. Selected Papers*, pp.39-49, Springer-Verlag, Berlin, Germany, 1996.

[29] F.D.Croce, R.Tadei and G.Volta, "A Genetic Algorithm for the job-shop problem", Computers & Operations Research, vol.22, no.1, pp.15-24, 1995.

[30] J.C.Gilkinson, L.C.Rabelo and B.O.Bush, "A real-world scheduling problem using Genetic Algorithms", *Computers & Industrial Engineering*, vol.29, no.1-4, pp.177-181,1995.

[31] K.Terano, Y.Yao, K.Okamoto, Y.Hashimoto, I. Nishikawa, T.Watanabe and H.Tokumuru, "Application of simulated annealing method and genetic algorithm to scheduling problems in plastic injection molding", in *Proc.of the Japan/USA Symposium on Flexible Automation*, pp.1225-1228, ASME, 1996.

[32] N.S.H. Kumar and G.Srinivasan, "A genetic algorithm for job-shop scheduling - a case study, *Computers in Industry*, vol.31, no.2, pp.155-160, 1996.

[33] C.H.Dagli and S.Sittisathancai, "Genetic neuro-scheduler: a new approach for job shop scheduling, *Int.J. of Production Economics*, vol.41, no.1-3, pp.135-145, 1995.

[34] S.Sittisathancai, C.H.Dagli and H.C.Lee, "A genetic scheduler for job-shops", in *ANNIE'94: Artificial Neural Networks in Engineering. Proceedings*, pp.351-356, ASME Press ,1994.

[35] J.R.Z. Aizpuru and J.A.Usunariz, "GA/TS: A hybrid approach for job-shop scheduling in a production system" in *Proc.of the 7th Potuguese Conf.on Artificial Intelligence - EPIA'95*, pp.153-164, Springer-Verlag, Berlin, Germany, 1995.

[36] B.Gieffler amd G.L.Thompson, "Algorithms for solving production scheduling problems", *Operations Research*, vol.8, pp.487-503, 1969.

[37] R.Cheng, M.Gen and Y.Tsujimura, "A tutorial survey of job-shop scheduling problems", *Computers & Industrial Engineering*, vol.30, no.4, pp.983-997,1996.

[38] K.J.Shaw and P.J.Flemming, "Including real-life preferences in genetic algorithms to improve optimisation of production schedules" in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.239-244, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[39] G.Shi, "A genetic algorithm applied to a classic job-shop scheduling problem", *Int.J. of Systems Science*, vol.28, no.1, pp.25-32, 1997.

[40] G.Shi, "A new encoding scheme for soving job-shop problems by genetic algorithm", in *Proc.of the 35th Conf.on Decision and Control*, pp.4395-4400, IEEE, 1996.

[41] J.W.Herrmann, C.Y.Lee and J.Hinchman, "Global job-shop scheduling with a genetic algorithm", *Production & Operations Management*, vol.4, no.1, pp.30-45, 1995.

[42] R.Haupt, "A survey of priority rule-based scheduling", *OR Spektrum*, vol.11, no.1, pp.3-16, 1989.

[43] J.H.Blackstone, D.T.Philips and C.L.Hogg, "A state of the art survey of dispatching rules for manufacturing job shop operations", *Int.J.of Production Research*, vol.20, pp.27-45, 1982.

[44] K.Caskey and R.L.Storch, "Heterogeneous dipatching rules in job and flow shops", *Production Planning & Control*, vol.7, no.4, pp.351-361, 1996.

[45] H.Fujimoto, C.Lian-yi,Y.Tanigawa and K.Iwahashi, "Application of genetic algorithm and simulation to dispatching rule-based FMS", in *Proc.of the 1995 IEEE Int.Conf.on Robotics & Automation*, pp.190-195, IEEE, New York, NY, USA, 1995.

[46] H.Fujimoto, C.Lian-yi,Y.Tanigawa and K.Iwahashi, "FMS scheduling by hybrid approaches using genetic algorithms and simulation", in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.442-447, IEE Conf.Publ. no.414, IEE, London, England, 1995.

[47] G.Niemeyer and P.Shiroma, "Production scheduling with genetic algorithms and simulation", in *Proc.of the 4th Int.Conf.on PPS from Nature*, pp.930-939, Springer-Verlag, Berlin, Germany, 1996.

[48] J.-U.Kim and Y.-D.Kim, "Simulated annealing and genetic algorithms for scheduling problems with multi-level product structure", *Computers & Operations Research*, vol.23, no.9, pp.857-868, 1996

[49] J.C.Bean, "Genetic algorithms and random-keys for sequencing and optimisation", *ORSA Journal on Computing*, vol.6, no.2, pp.154-160, 1994.

[50] K.Baker, *Introduction to Sequencing and Scheduling*, Whiley, New York, 1974.

[51] G.H.Kim and C.S.G.Lee, "Genetic reinforcement learning approach to the machine scheduling problem", in *Proc.of the 1995 IEEE Int.Conf.on Robotics & Automation*, pp.196-201, IEEE, New York, NY, USA, 1995.

[52] G.H.Kim and C.S.G.Lee, "Genetic reinforcement learning for scheduling heterogeneous machines", in *Proc.of the 1996 IEEE Int.Conf.on Robotics & Automation*, pp.2798-2803, IEEE, New York, NY, USA,1996.

[53] S.Kobayashi,I.Ono and M.Yamamura, "An efficient genetic algorithm for job-shop scheduling problems", in *Proc. of the 6tht Int.Conf.on Genetic Algorithms and their Applications*, pp.506-511, Morgan Kaufman Publishers, San Francisco, California, 1995.

[54] I.Ono, M.Yamamura and S.Kobayashi, "A genetic algorithm for job-shop scheduling problems using job-based order crossover", in *Proc.of the 1996 IEEE Int.Conf. on Evolutionary Computation*, pp.2798-2803, IEEE, New York, NY, USA,1996.

[55] L.-J. Park and C.H.Park, "Genetic algorithms for job-shop scheduling problems based on two representational schemes", *Electronics Letters*, vol.31, no.3, pp.205-207, 1995.

[56] L.-J. Park and C.H.Park, "Application of genetic algorithms to job-shop scheduling problems with active-schedule constructive crossover", in *Proc.of the 1995 IEEE Int.Conf. on Systems, Man & Cybernetics*, pp.530-535, IEEE, New York, NY, USA, 1995.

[57] S.J.T.Liang and M.Mannion, "Scheduling of a flexible assemby system using genetic algorithms", in *Proc.of the 11th Int.Conf.on Computer-Aided Production Engineering*, pp.173-178, Mech.Eng.Publications, Bury St.Edmonds, UK, 1995.

[58] B.J Cho, S.C.Hong and S.Okoma, "Job-shop scheduling using genetic algorithms", *in Critical Technology: Proc. of the 3rd World Congress on Expert Systems*", pp.351-358, Cognizant Com.Corp., New York, 1996.

[59] T.Yamada and R.Nakano, "A genetic algorithm with multi-step crossover for job-shop scheduling problems", in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.146-151, IEE Conf.Publ. no.414, IEE, London, England, 1995.

[60] J.Adams, E.Balos and D.Zawack, "The shifting bottleneck procedure for job-shop scheduling", *Management Science*, vol.34, pp.391-401, 1988.

[61] T.Gohtoh, K.Ohkura, K.Ueda, "An application of genetic algorithm woth neutral mutations to job-shop scheduling problems", *Proc.of the Int.Conf. on Advance in Production Systems, APMS'96*, pp.563-568, Kyoto Univ., Kyoto, Japan, 1996.

[62] K.Ohkura and K.Ueda, "Genetic algorithm with neutral mutation for massively multimodal

fuction optimisation", in *Proc.of the 1995 IEEE Int.Conf. on Evolutionary Computation*, pp.361-364, IEEE, New York, NY, USA, 1995.

[63] H.-J.Zimmermann, "Description and optimization of fuzzy system", Int.J.of General Systems, vol.2, pp.209-216, 1976.

[64] H.Fischer and G.L.Thompson, "Probabilistic learning combinations of local job-shop scheduling rules" in *Industrial Scheduling*, pp.225-251, J.F.Muth and G.L.Thompson (eds), Prentice Hall, Englewood Cliffs, NJ, 1963.

[65] S.Lawrence, *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques*, GSIA, Carnegie Mellon University, 1984.

[66] K.Hamada, T.Baba, K.Sato, and M.Yufu, "Hybridising a genetuc algorithm with rule-based reasoning for production planning", *IEEE Expert*, vol.10, no.5, pp.60-67, 1995.

[67] M.Sakawa, K.Kosuke and T.Mori, "Flexible scheduling in a machining center through genetic algorithms", *Computers & Industrial Engineering*, vol.30, no.4, pp.931-940, 1996.

[68] C.R.Reeves, "A genetic algorithm for flowshop scheduling", *Computers & Operations Research*, vol.22, no.1, pp.5-13, 1995.

[69] C.-L.Chen, R.V.Neppali and N.Aljaber, "Genetic algorithms applied to the continuous flow-shop problem", *Computers & Industrial Engineering*, vol.30, no.4, pp.919-929, 1996.

[70] T.Murata, H.Ishibuchi and H.Tanaka, "Genetic algorithms for flowshop scheduling problems", *Computers & Industrial Engineering*, vol.30, no.4, pp.1061-1071, 1996.

[71] P.R.Drake and I.A. Choudry, "From apes to schedules", *Manufacturing Engineer*, vol.76, no.1, pp.35-45, 1997.

[72] M.Braglia and E,Gentili, "An improved genetic algorithm for flowshop scheduling problems" in *Proc.of the 10th ISPE/IFAC Int.Conf. on CAD/CAM, Robotics and Factories of the Future*, pp.137-142, OCRI, Ontario, Canada, 1994.

[73] T.Murata, H.Ishibuchi and H.Tanaka, "Multi-objective genetic algorithm and its application to flow-shop scheduling", *Computers & Industrial Engineering*, vol.30, no.4, pp.957-968, 1996.

[74] S.S.Lam, K.W.C.Tang and X.Cai, "Genetic algorithm with pigeon-hole coding scheme for solving sequencing problems", *Applied Artificial Intelligence*, vol.10, no.3, pp.239-256, 1996.

[75] R.Sikora, "A genetic algorithm for integrating lot-sizing and sequencing in scheduling a capacitated flow-line", *Computers & Industrial Engineering*, vol.30, no.4, pp.969-981, 1996.

[76] N.Sannomiya and H.Iima, "Application of a genetic algorithm to scheduling problems in manufacturing processes", *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.523-528, IEEE, New York, NY, USA, 1996.

[77] N.Sannomiya and H.Iima, "Genetic algorithm approach to a modified flowshop scheduling problem", in *Proc.of the Japan/USA Symposium on Flexible Automation*, pp.1229-1234, ASME, 1996.

[78] C.Y.Lee and J.Y.Choi, "A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights", *Computers & Operations Research*, vol.22, no.8, pp.857-869, 1995.

[79] C.Y.Lee, and J.Y.Choi, "Parallel genetic algorithms for the earliness-tardiness job scheduling problem with general penalty weights", *Computers & Industrial Engineering*, vol.28, no.2, pp.231-243, 1995.

[80] I.Lee, R.Sikora and M.J.Shaw, "A genetic algorithm-based approach to flexible flow-line scheduling with variable lot sizes", in *IEEE Transactions on Systems, Man & Cybernetics - Part B: Cybernetics*, vol.27, no.1, pp.36-54, 1997.

[81] B.Gonzalez, M.Torres amd J.A.Moreno, "A hybrid genetic algorithm approach for the 'no-wait' flowshop scheduling problem" in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.59-64, IEE Conf.Publ. no.414, IEE, London, England, 1995.

[82] J.W.Herrmann and C.-Y.Lee, "Solving a class scheduling problem with genetic algorithms", *ORSA Journal on Computing*, vol.7, no.4, pp.443-452, 1995.

[83] S.Karabati and P.Kouvelis, "Flow line scheduling problem with controllable processing times", *IIE Transactions*, vol.29, no.1, pp.1-14, 1997.

[84] H.Ishibuchi, N.Yamamoto, T.Murata and H.Tanaka, "Genetic algorithms and neighborhood search algorithms for fuzzy flowshop scheduling problems", *Fuzzy Sets and Systems*, vol 67, no.1, pp.81-100, 1994.

[85] I.Kebbe, H.Yokoi, K.Suzuki and Y.Kakazu, "Vibrational Potentional methodfor large scale scheduling problems", *Proc.of the Int.Conf. on Advance in Production Systems, APMS'96*, pp.585-590, Kyoto Univ., Kyoto, Japan, 1996.

[86] L.Davis, "Applying adaptive algorithms to epistatic domains" in *Proc.of the 9th Int.Joint Conf. on Artificial Intelligence*, pp.162-164, 1985.

[87] S.Katoh,Y.Mutsunori and T.Ibaraki, "Application of genetic algorithms to single machine scheduling problems under changing environment", in *Proc.of the Int.Conf. on Advance in Production Systems, APMS'96*, pp.543-546, Kyoto Univ., Kyoto, Japan, 1996.

[88] L.Davis, *Handbook of Genetic Algorithms*, Van Nostrad Reinhold, Princeton, NJ, 1991.

[89] T.Murata, H.Ishibuchi, "Positive and negative combination effects of crossover and mutation operators in sequencing problems", in *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.170-175, IEEE, New York, NY, USA,1996.

[90] S.Fichera, V.Grasso, A.Lombardo and E.Lo Valvo, "Genetic algorithms efficiency in flowshop scheduling", in *Proc. of the 10th Int.Conf.on Applications of Artificial Intelligence in Engineering-AIENG'95*, pp.261-270, Comput.Mech, Southampton, UK, 1995.

[91] D.E.Goldberg and R.Lingle, "Alleles, loci and the TSP", in *Proc. of the 1st Int. Conf. on Genetic Algorithms and their Applications*, pp.254-159 , J.J.Grefenstette (ed.), Lawrence Erlbaum, Hillsdale, NJ, 1985.

[92] C.-L.Chen, V.S.Vempati and N.Aljaber, "An application of genetic algorithms for flow shop problems", *European Journal of Operational Research*, vol.80, no.2, pp.389-396, 1995.

[93] C.Rajendran and D.Chaudhuri, "Heuristic algorithms for continuous flow-shop problems", *Naval Research Logistics*, vol.37, pp.695-705, 1990.

[94] J.Sridhar and C.Rajendran, "A genetic algorithm for family and job scheduling in a flo-line based manufacturing cell", *Computers & Industrial Engineering*, vol.27, no.1-4, pp.469-472, 1994.

[95] J.Sridhar and C.Rajendran, " Scheduling in flowshop and cellular manufacturing systems with multiple objectives - a genetic algorithm approach", *Production Planning and Control*, vol.7, no.4, pp.374-382, 1996.

[96] D.Whitley, T.Starkweather and D.Fuquay, "Scheduling problems and traveling salesmen: the genetic edge recombination operator", in *Proc. of the 3rd Int. Conf. on Genetic Algorithms and their Applications*, pp.133-140, J.Shaffer (ed.), Morgan Kaufman Publishers, Los Altos, CA, 1989.

[97] T.Starkweather, S.McDaniel, K.Mathias, D.Whitley, and C.Whitley, "A comparison of genetic sequencing operators", in *Proc. of the 4th Int. Conf. on Genetic Algorithms and their Applications*, pp.69-76 , R.Belew and L.Booker (eds), Morgan Kaufman Publishers, Los Altos, CA, 1991.

[98] T.Asveren and P.Molitor, " New crossover methods for sequencing problems" in *Proc.of the 4th Int.Conf.on PPS from Nature*, pp.290-299, Springer-Verlag, Berlin, Germany, 1996.

[99] T.Yamada and C.R.Reeves, "Pemutation flowshop scheduling by genetic local search" in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.232-238, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[100] M.Yagiura amd T.Ibaraki, "The use of dynamic programming in genetic algorithms for permutation problems", *European Journal of Operational research*, vol.92, no.2, pp.387-401, 1996.

[101] I.M.Oliver, D.J.Smith and J.R.Holland, "A study of permutation crossover operators of the Travelling Salesman Problem", in *Proc. of the 2nd Int. Conf. on Genetic Algorithms and their Applications*, pp.224-230, Lawrence Erlbaum, Hillsdale, NJ, 1987.

[102] J.J.Grefenstette, "A system for learning control strategies with genetic algorithms, in *Proc. of the 3rd Int. Conf. on Genetic Algorithms and their Applications*, pp.160-168, J.Shaffer (ed.), Morgan Kaufman Publishers, Los Altos, CA, 1989.

[103] R.Pakath and J.S.Zaveri, "Specifying critical inputs in a genetic algorithm-driven decision support system", *Decision Sciences*, vol.26, no.6, pp.749-779, 1995.

[104] E.Tailard, "Benchmarks for basic scheduling problems", *European Journal of Operations Research*, vol.64, pp.278-285, 1993.

[105] C.Chiu and Y.Yih, "A learning-based methodology for dynamic scheduling in distributed manufacturing systems", *Int.J.of Production Research*, vol.33, no.11, pp.3217-3232, 1995.

[106] H.Aytug, G.H.Koehler and J.L.Snowdon, "Genetic learning of dynamic scheduling within a simulation environment", *Computers & Operations Research*, vol.21, no.8, pp.909-925, 1994.

[107] Y.Yih, "Trace-Driven Knowledge Acquisition (TDKA) for rule-based real-time scheduling systems", *Journal of Intelligent Manufacturing*, vol.1, pp.217-230, 1990.

[108] A.Jones, L.Rabelo and Y.Yih, "A hybrid approach for real-time sequencing and scheduling", *Int.J.of Computer Integrated Manufacturing*, vol.8, no.2, pp.145-154, 1995.

[109] L.C.Rabelo, A.Jones and Y.Yih, "Neural Networks, Simulation, Genetic Algorithms and Machine Learning for manufacturing scheduling", in *WCCN'95. World Congress on Neural Networks*, pp.130-135, Lawrence Erlbaum, Mahwah, NJ, USA, 1995

[110] A.Jones, F.Riddick and L.Rabelo "Development of a predictive-reactive scheduler using genetic algorithms and simulation-based scheduling software", in *Advanced Manufacturing Processes Systems and Technologies - AMPST'96*, pp.589-598, Mech.Eng.Publications, Bury St.Edmunds, UK, 1996.

[111] C.-Y.Lee, S.Piramuthu and Y.-K.Tsai, "Job-shop scheduling with a genetic algorithm and machine learning", *Int.J.of Production Research*, vol.35, no.4, pp.1171-1191, 1997.

[112] H.Tamaki, M.Ochi and M.Araki, "Application of genetics-based machine learning to production scheduling" in *Japan/USA Symposium on Flexible Automation*, pp.1221-1224, ASME, 1996

[113] Y.Ikkai, M.Inoue, T.Ohkawa and N.Komoda, "A learning method of scheduling knowledge by genetic algorithms", in *Proc.of the 1995 IEEE Symposium on Emerging Technologies and Factory Automation*, pp.641-648, IEEE, Los Alamitos, CA, USA,1995.

[114] J.Fang and Y.Xi, "A rolling horizon job shop rescheduling strategy in the dynamic environment", *Int.J.of Advanced Manufacturing Technology*, vol.13, no.3, pp.227-232, 1997.

[115] H.M.Cartwright and A.LTuson, "Genetic algorithms and flowshop scheduling: towards the development of a real-time process control system" in *Evolutionary Computing. AISB Workshop. Selected Papers*, pp.277-290, Springer-Verlag, Berlin, Germany, 1994.

[116] A.K.Jain and H.A.Elmaraghy, "Production scheduling/rescheduling in flexible manufacturing systems", *Int.J.of Production Research*, vol.35, no.1, pp.281-309, 1997.

[117] E.P.K.Tsang, "Scheduling techniques - a comparative study", *BT Technology Journal*, vol.13, no.1, pp.16-29, 1995.

[118] J.Dorn, M.Girsch, G.Skele and W.Slany, "Comparison of iterative improvement techniques for schedule optimisation", *European Journal of Operational Research*, vol.94, no.2, pp.349-361, 1996.

[119] R.E.Korf, "Depth-first iterative deepening", *Artificial Intelligence*, vol.27, no.1, pp.97-109, 1987.

[120] M.Yagiura and T.Ibaraki, "Metaheuristics as simple and robust optimisation tools", *Proc.of the 1996 IEEE Conf.on Evolutionary Computation*, pp.541-546, IEEE, New York, NY, USA,1996.

[121] G.McMahon and D.Hadinoto, "Comparison of heuristic search algorithms for single-machine scheduling problems", *Proc.of the AI'93 and AI'94 Workshops on Evolutionary Computation*, pp.293-303, Springer-Verlag, Berlin, Germany, 1995.

[122] C.A.Glass and C.N.Potts, "A comparison of local search methods for flowshop scheduling" *Annals of Operations Research*, vol.63, pp.489-509, 1996.

[123] M.Horvath, A.Markus and C.Vancza, "Process planning with genetic algorithms on results of knowledge-based reasoning", *Int.J.of Computer Integrated Manufacturing*, vol.9, no.2, pp.145-166, 1996.

[124] B.Awadh, N.Sepehri and O.Hawaleshka, "A computer-aided process planning model based on genetic algorithms", *Computers & Operations Research*, vol.22, no.8, pp.841-856, 1995.

[125] G.Zhou and M.Gen, "Evolutionary computation of multi-criteria production process planning problem", in *Proc.of the 1997 IEEE Conf.on Evolutionary Computation*, pp.419-424, IEEE, 1997.

[126] N.Singh, *Systems Approach to Computer-Integrated Design and Manufacturing*, Wiley, Chchester, New York, 1996.

[127] P.Husbands, "Distributed co-evolutionary genetic algorithms for multi-criteria and multi-constrainedoptimisation", in *Evolutionary Computing. AISB Workshop. Selected Papers*, pp.150-165, Springer-Verlag, Berlin, Germany, 1994.

[128] M.McIlhagga,P.Husbands and R.Ives, "A comparison of optimisation techniques for integrated manufacturing planning and scheduling" in *Proc.of the 4th Int.Conf.on PPS from Nature*, pp.604-613, Springer-Verlag, Berlin, Germany, 1996.

[129] R.Bowden and S.F.Bullington, "Development of manufacturing control strategies using unsupervised machine learning", *IIE Transactions*, vol.28, no.4, pp.319-331, 1996.

[130] J.J.Grefenstette, "Strategy acquisition with genetic algorithms" in *Handbook of Genetic Algorithms*, L.Davis (ed.), Van Nostrad Reinhold, New York, 1991.

[131] Y.Hayashi, H.Kim and K.Nava, "Scenario creation method by genetic algorithms for evaluating future plans", in *Proc.of the 1996 IEEE Conf.on Evolutionary Computation*, pp.880-885, IEEE, New York, NY, USA, 1996.

[132] J.M.Usher and R.Bowden, "The application of genetic algorithms to operation sequencing for use in computer-aided process planning", *Computers & Industrial Engineering*, vol.30, no.4, pp.999-1013, 1996.

[133] D.Yip-Hoi and P.Dutta, "A genetic algorithm application for sequencing operations in process planning for parallel machining", *IIE Transactions*, vol.28, no.1, pp.55-68, 1996.

[134] N.Takatori, M.Minagawa and Y.Kakazu, "A GA-based approach to a process palnning problem with geometric constraints", in *ANNIE'94: Artificial Neural Networks in Engineering. Proceedings*, pp.369-374, ASME Press,1994.

[135] H.N.Kanhawi, S.R.Leclair and C.LPhilip, " Feature seequencing in the rapid design system

using a genetic algorithm", *Journal of Intelligent Manufacturing*, vol.7, no.1, pp.55-67, 1996.

[136] B.ANorman and G.C.Bean, "Operation sequencing and rool assignment for multiple spindle CNC machines", in *Proc.of the 1997 IEEE Conf.on Evolutionary Computation*, pp.425-429, IEEE, New York, NY, USA, 1997.

[137] S.P.Mitrovanov, *The Scientific Principles of Group Technology*, 1966 (english translation).

[138] J.L.Burbidge, "Production flow analysis", *Production Engineer*, vol.42, pp.472- , 1963.

[139] J.L.Burbidge, *The Introduction of Group Technology*, John Wiley and Sons, New York, USA, 1975.

[140] J.A.Brandon, *Cellular Manufacturing: Integrating Technology and Management*, Research Studies Preess, Somerset, England, 1996.

[141] N.L.Hyer and U.Wemmerlov, "Group technology and productivity", *Harvard Business Review*, vol.4, pp.140-149, 1984

[142] N.L.Hyer, "Case studies in manufacturing cells: implications of research", in *Proc.of the Decision Sciences Institute*, pp.1407-1409, 1991.

[143] N.L.Hyer and U.Wemmerlov, "Group technology in U.S. manufacturing industry: a survey of current practices", *Int.J.of Production Research*, vol.27, no.8, pp.1287-1304, 1989.

[144] U.Wemmerlov and N.L.Hyer, "Cellular manufacturing in the U.S. industry: a survey of users", *Int.J.of Production Research*, vol.27, no.9, pp.1511-1530, 1989.

[145] N.Singh, "Design of cellular manufacturing systems: an invited review", *European Journal of Operational Research*, vol.69, pp.284-291, 1993.

[146] O.Offodile, A.Mehrez and J.Grznar, "Cellular manufacturing: a taxonomic review framework", *Journal of manufacturing systems*, vol.13, no.3, pp.196-220, 1994.

[147] N.Morad, *Optimisation of Cellular Manufacturing Systems using Genetic Algorithms*, Ph.D Thesis, University of Sheffield, 1997.

[148] J.R.King, "Machine-component grouping in production flow analysis: an approach using rank-order clustering algorithm", *Int.J.of Production Research*, vol.18, no.1, pp.213-232, 1980.

[149] H.A.Chan and D.A.Milner, "Direct Clustering Algorithm for group formation in cellular manufacturing", *Journal of Manufacturing Systems*, vol.1, pp.65-72, 1982.

[150] J.McAuley, "Machine grouping for efficient production", *Production Engineer*, vol.51, no.2, pp.53-57, 1972.

[151] M.V.Tatikonda and U,Wemmerlov, "Adoption and implementation of group technology classification and coding systems: insights from seven case studies", *Int.J.of Production Research*, vol.30, pp.2087-2110, 1992.

[152] R.Rajagopalan and J.L.Balta, "Design of cellular production systems: a graph-partitioning theoretic approach", *Int.J.of Production Research*, vol.13, no.6, pp.567-579, 1975.

[153] F.Boctor, "A linear formulation of the machine-part cell formation problem", *Int.J.of Production Research*, vol.29, no.2, pp.343-356, 1991.

[154] S.Kaparthi and N.C.Suresh, "Machine-component cell formation in group teechnology: a neural network approach", *Int.J.of Production Research*, vol.26, no.6, pp.1353-1367, 1992.

[155] C.-H.Chu and J.C.Hayya, "A fuzzy-clustering approach to manufacturing cell formation", *Int.J.of Production Research*, vol.29, no.7, pp.1475-1487, 1991.

[156] V.Venugopal and T.T.Narendran, "A genetic algorithm approach to the machine-component grouping problem with multiple objectives", *Computers & Industrial Engineering*, vol.22, no.4, pp.269-480, 1992.

[157] Y.Gupta, M.Gupta, A.Kumar and C.Sundaram, "A genetic algorithm-based approach to cell-composition and layout design problems", *Int.J.of Production Research*, vol.34, no.2 pp.447-482, 1996.

[158] Y.Gupta, M.Gupta, A.Kumar and C.Sundaram,, "Minimising total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach", *Int.J.of Computer Integrated Manufacturing*, vol.8, no.2, pp.92-101, 1995.

[159] R.E.Billo,B.Bidanda and D.Tate, "A genetic clusteer algorithm for thee machine-component grouping problem", *Journal of Intelligent Manufacturing*, vol.7, no.3, pp.229-243, 1996

[160] J.A.Joines, C.T.Culbreth and R.E.King, "Manufacturing cell design: an integer programming model employing genetic algorithms", *IIE Transactions*, vol.28,no.1, pp.69-85, 1996.

[161] C.-T. Su and C.-M.Hsu, "A two-phased genetic algorithm for the cell formation problem", *Int.J.of Industrial Engineering*, vol.3, no.2, pp.114-125, 1996.

[162] N.Morad and A.M.S.Zalzala, "A genetic-based approach to the formation of manufacturing cells and batch scheduling", in *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.485-490, IEEE, New York, NY, USA, 1996.

[163] H.Pierreval and M.-F.Plaquin, "A genetic algorithm approach to group machines into manufacturing cells", in *Proc of the 4th Int.Conf on CIM and Automation Technology*, pp.267-271, IEEE Press, Los Alamitos, CA, USA, 1994.

[164] G.A.SÜer, "Evolutionary programming for designing manufacturing cells", in *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.379-384, IEEE, New York, NY, USA, 1996.

[165] C.Dimopoulos and A.M.S.Zalzala, "Optimisation of cell configuration and comparisons using evolutionary computation approaches", submitted and accepted in the *1998 IEEE Int.Conference on Evolutionary Computation, part of the 1998 IEEE World Congress on Computational Intelligence.*

[166] J.L.Paris and H.Pierreval, "Manufacturing cell formation using distributed evolutionary algorithms", in *Proc.of the 12th Int.Conf.on CAD/CAM, Robotics and Factories of the Future*, pp.369-374, Middlesex Univ.Press, London, UK, 1996.

[167] H.Hwang and J.-U.Sun, "A genetic algorithm-based heuristic for the GT cell formation problem", *Computers & Industrial Engineering*, vol.30, no.4, pp.941-955, 1996.

[168] L.Zhao, Y.Tsujimura and M.Gen, "Genetic algorithm for fuzzy clustering", in *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.716-719, IEEE, New York, NY, USA, 1996.

[169] A.Kusiak and S.S.Heragu, "The facility layout problem", *European Journal of Operational Research*, vol.29, pp.229-251, 1987.

[170] T.D.Mavridou and P.M.Pardalos, "Simulated annealing and genetic algorithms for the facility layout problem: a survey", *Computational Optimisation and its Applications*, vol.7, no.1, pp.111-126,1997.

[171] D.M.Tate and A.E.Smith, "Unequal-area facility layout by genetic seearch", *IIE Transactions*, vol.7, no.4, pp.465-472, 1995.

[172] B.A.Norman and A.E.Smith, "Random-keys genetic algorithm with adaptive penalty function for optimization of constraineed facility layout problems", in *Proc.of the 1997 IEEE Int.Conf.on Evolutionary Computation*, pp.407-411, IEEE, 1997.

[173] G.Suresh, V.V.Vivod and S.Sahu, "A genetic algorithm for facility layout", *Int.J.of Production Research*, vol.33, no.12, pp.3411-3423, 1995.

[174] M.Kazerooni, L.H.S.Luong, K.Abhary, F.T.S.Chan and F.Pun, "An integrated method for cell layout problem using genetic algorithms", in *Proc.of the 12th Int.Conf.on CAD/CAM, Robotics and Factories of the Future*, pp.752-762, Middlesex Univ.Press, London, UK, 1996.

[175] P.Banerjee, Y.Zhou and B.Montreuil, "Genetically assisted optimisation of cell layout and material flow path skeleton", *IIE Transactions*, vol.29, no.4, pp.277-291, 1997.

[176] D.G.Conway and M.A.Venkataramanan, "Genetic search and the dynamic facility layout problem", *Computers & Operations Research*, vol.21, no.8, pp.955-960, 1994.

[177] M.Bazargan-Lari and H.Kaebernick, "An approach to the machine layout problem in a cellular manufacturing environment", *Production Planning & Control*, vol.8, no.1, pp.41-55, 1997.

[178] M.Braglia and A.Sternieri, "A genetic algorithm for facility layout optimisation in a flowline cellular manufacturing system", in *Proc.of the Int.ICSC Symposia on Intelligent Industrial Automation and Soft Computing*, pp.A21-A26, Int.Comp.Science Conventions, Millet, Canada, 1996.

[179] M.Braglia and L.Zavonella, "Backtracking of jobs and single row machine layout problems in flexible cellular manufacturing system", in *Advanced Manufacturing Processes Systems and Technologies - AMPST'96*, pp.17-25, Mech.Eng.Publications, Bury St.Edmunds, UK, 1996.

[180] M.Braglia, "Optimisation of a simulated anneealing-based heuristic for a single row machine layout problem by geneetic algorithm", *Int.Transactions on Operational Research*, vol3, no.1, pp.37-49, 1996.

[181] R.Cheng, M.Gen and T.Tosawa, "Genetic algorithms for designing loop layout manufacturing systems", *Computers & Industrial Engineering*, vol.31, no.3/4, pp.587-591, 1996.

[182] M.Gen, K.Ida and C.Cheng, "Multirow machine layout problems in fuzzy environment using genetic algorithms", *Computers & Industrial Engineering*, vol.29, no.1-4, pp.519-523, 1995.

[183] A.Bolte and U.W.Thoneman, "Optimisimg simulated annealing schedules with genetic algorithms", *European Journal of Operational Research*, vol.92, no.2, pp.402-416, 1996.

[184] M.F. Sebaaly and H.Fujimoto, "A new approach for constrained GA-search: assembly sequence planning - a case study", in *Proc.of the Int.ICSC Symposia on Intelligent Industrial Automation and Soft Computing*, pp.B138-B144, Int.Comp.Science Conventions, Millet, Canada, 1996.

[185] M.F. Sebaaly and H.Fujimoto, "Integrated planning and scheduling for job-shop assembly based on geenetic algorithms", in *Proc.of the Int.Conf. on Advance in Production Systems, APMS'96*, pp.557-562, Kyoto Univ., Kyoto, Japan, 1996.

[186] M.F. Sebaaly and H.Fujimoto, "A genetic planner for assembly automation", in *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.401-406, IEEE, New York, NY, USA, 1996.

[187] M.F. Sebaaly and H.Fujimoto, "Assembly sequence planning by GA search: a novel approach", in *Proc.of the Japan/USA Symposim on Flexible Automation*, pp.1235-1240, ASME, 1996.

[188] R.Groppeti and R.Muscia, "Genetic algorithms for optimal assembly assembly planning", in *Proc.of the 1st World Congress on Intelligent Manufacturing Processes and Systems*, pp.319-333, Univ.of Puerto Rico, Sa Juan, Puerto Rico, 1995.

[189] Y.K.Kim, C.J.Hyun and Y.Kim, " Sequencing in mixed-model assembly lines: a genetic algorithm approach", *Computers & Operations Research*, vol.23, no.12, pp.1131-1145, 1996.

[190] Y.-Y.Leu, L.A.Matheson and L.P.Rees, " Sequencing mixed-model assembly lines with genetic algorithms", *Computers & Industrial Engineering*, vol.30, no.4, pp.1027-1036, 1996.

[191] Y.Monden, *Toyota Production System*, Industrial Engineering and Management Press, Norcross, CA, 1993.

[192] G.Suresh, V.V.Vivod, and S.Sahu, "A genetic algorithm for assembly line balancing", *Production Planning & Control*, vol.7, no.1, pp.38-46, 1996.

[193] J.Rubinovitz, G.Levitin, "Genetic algorithm for assembly line balancing", *Int.J.of Production Economics*, vol.41, no.1-3, pp.343-354, 1995.

[194] Y.Tsujimura, M.Gen and E.Kubota, "Solving fuzzy assembly-line balancing problems with genetic algorithms", *Computers & Industrial Engineering*, vol.29, no.1-4, pp.543-547, 1995.

[195] A.A.Bulgak, P.D.Diwan and B.Inozu, "Buffer size optimisation in asynchronous assembly systems using genetic algorithms", *Computers & Industrial Engineering*, vol.28, no.2, pp.309-322, 1995.

[196] A.Roach and R.Nagi, "A hybrid GA-SA algorithm for just-in-time scheduling of multi-level assemblies", *Computers & Industrial Engineering*, vol.30, no.4, pp.1047-1060, 1996.

[197] G.List, "Heuristic methods in a flexible assembly system", in *Intelligent Manufacturing Systems 1994-IMS '94*, pp.251-254, Pergamon, Oxford, UK, 1994.

[198] T.Watanabe, Y.Hashimoto, I.Nishikawa and H.Tokumaru, "Line balancing using a genetic evolution model", *Control Engineering Practice*, vol.3, no.1, pp.69-76, 1995.

[199] T.Alander and J.Lampinen, "On implementing CAD systems based on existing simulators and optimisation by genetic algorithms", *Proc.of the 3rd Int.Conf. on Genetic Algorithms, Optimisation Problems, Fuzzy Logic, Neural Networks and Rough Sets: MENDEL '97*, pp. 7-11, 1997.

[200] Y.J.Cao and Q.H.Wu, "Mechanical design optimisation by mixed-variable evolutionary programming", in *Proc.of the 1997 IEEE Int.Conf.on Evolutionary Computation*, pp.443-446, IEEE, New York, NY, USA, 1997.

[201] K.Rasheed, H.Hirsch and A.Gelsey, " A genetic algorithm for continuous design search space", *Artificial Intelligence in Engineering*, vol.11, pp.295-305, 1997.

[202] C.A.Coello and A.D.Christiansen, "An approach to multi objective optimisation using genetic algorithms", in in *ANNIE'95: Artificial Neural Networks in Engineering. Proceedings*, pp.410-416,1995.

[203] S.S.Altus, I.M.Kroo and P.J.Cage, "A genetic algorithm for scheduling and decomposition of multidisciplinary design problems", *Transactions of the ASME*, vol.118, no.4, pp.486-489, 1996.

[204] A.C.Thornton and A.L.Johnson, "CADET: a software support toll for constraint processes in embodiment design", *Research in Engineering Design*, vol.8, no.1, pp.1-13, 1996.

[205] S.E.Carlson, "Genetic algorithm attributes for component seeleection", *Research in Engineering Design*, vol.8, no.1, pp.33-51, 1996.

[206] M.Iannuzzi and E.Sandgren, "Optimal tolerancing: the link betweeen design and manufacturing productivity", in *Proc.of the 6th Int.Conf.on Design Theory and Methodology*, pp.29-42, DE-vol.68, ASME, New York, 1994.

[207] J.G.Ziegler and N.B.Nichols, " Optimum settings for automatic controllers", *Transactions of the ASME*, vol.64, pp.759-768, 1942.

[208] A.H.Jones and P.B.D.M.Oliveira, " Genetic autotuning of PID controllers", in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.141-145, IEE Conf.Publ. no.411, IEE, London, England, 1995.

[209] P.B.D.M.Oliveira and A.H.Jones, "Robust co-evolutionary design of SISO Smith Predictor PID controllers", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.504-509, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[210] A.H.Jones and B.Porter, "On-line genetic tuning of digital PID controllers", *Proc.of the IASTED Int.Conference*, pp.32-36, IASTED, Anaheim, CA, USA, 1994.

[211] A.H.Jones, N.Ajlouni, S.B.Kenway and P.B.D.M.Oliveira, "Genetic design of robust PID controllers to deal with prescribed plant uncertainties through a process of competitive co-evolution", *Proc.of the 1996 IEEE Int.Symposium on Intelligent Control*, pp.372-377, IEEE, New

York, NY, USA,1996.

[212] R.Krohling, "Design of PID controller for disturbance rejection: a genetic optimisation approach", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.498-503, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[213] C.Vlachos, J.T.Evans and D.Williams, "PI controller tuning for multivariable processes using genetic algorithms", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.43-49, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[214] M.Salami and G.Cain, "An adaptive PID controller based on genetic algorithm processor", in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.88-93, IEE Conf.Publ. no., IEE, Stevenage, England, 1995.

[215] M.Salami, "A multiple genetic algorithm processor for a PID controller system", in *Proc.of the 2nd Int.Conf. on Genetic Algorithms, Optimisation Problems, Fuzzy Logic, Neural Networks and Rough Sets: MENDEL '95*, pp. 133-138, 1995.

[216] X.M.Qi and T.C.Chin, "Genetic algorithms-based fuzzy controller for high order systems", *Fuzzy Sets and Systems*, vol.91, pp.279-284, 1997.

[217] J.Kim and B.P.Ziegler, "Hierarchical distributed genetic algorithms: a fuzzy logic controller application", *IEEE Expert*, vol.11, no.3, pp.76-84, 1996.

[218] Y.S.Tarng, Z.M.Yeh and C.Y.Nian, "Genetic synthesis of fuzzy logic controllers in turning", *Fuzzy Sets and Systems*, vol.83, no.3, pp.301-310, 1996.

[219] H.Polheim and P.Marenback, "Generation of structured process models using genetic programming", in *Evolutionary Computing. AISB Workshop. Selected Papers*. pp.102-109, Springer-Verlag, Berlin, Germany, 1996.

[220] B.M.McKay, M.J.Willis, H.G.Hiden, G.A.Montague and G.W.Barton, "Identification of industrial provesses using genetic programming", in *Proc.of the Conf.on Identification in Engineering Systems*, pp.510-519, Univ.of Wales, Swansea, UK. 1996.

[221] C.R.Reeves, P.Dai and K.J.Burham, "A hybrid genetic algorithm for system identification", in *Proc.of the Int.ICSC Symposia on Intelligent Industrial Automation and Soft Computing*, pp.B278-B283, Int.Comp.Science Conventions, Millet, Canada, 1996.

[222] Q.Xia and S.Macchietto, "Design and synthesis of batch plants - MINLP solution based on a stochastic method", *Computers & Chemical Engineering*, vol.21, suppl.issue, pp.S697-S702, 1997.

[223] D.G.Kim, D.Corne and P.Ross, "Industrial plant pipe-route optimisation with genetic algorithms, in *Proc.of the 4th Int.Conf.on PPS from Nature*, pp.1012-1021, Springer-Verlag, Berlin, Germany, 1996.

[224] R.O.Bowden, J.D.Hall and J.M.Usher, "Integration of evolutionary programming and simulation to optimise a pull production system", *Computers & Industrial Engineering*, vol.31, no.1/2, pp.217-220, 1996.

[225] P.Chen, T.Toyota and M.Nasu, "Self-organisation method of symptom parameters for for failure diagnosis by genetic algorithms, *Proc. of the 1996 IEEE Industrial Electronics Conference - IECON*, pp.829-835, IEEE, 1996.

[226] Z.R.Petrovic and S.L.Ivanovic, "Integration of neural networks and genetic algorithms: an example of machine noise diagnosis", in *Proc.of the 1st World Congress on Intelligent Manufacturing Processes and Systems*, pp.962-971, Univ.of Puerto Rico, Sa Juan, Puerto Rico, 1995.

[227] C.-R.Guzman and M.A.Kramer. "Remote diagnosis and monitoring of complex industrial systems using a genetic algorithm approach, in *Proc.of the IEEE Int.Symposium on Industrial Electronics*, pp.363-376, IEEE, 1994.

[228] H.Kim, N.Koichi and M.Gen, "A method for maintenance scheduling using GA combined with SA", *Computers & Industrial Engineering*, vol.27, no.1-4, pp.477-480, 1994.

[229] N.Viswanadham, S.M.Sharma and M.Taneja, "Inspection allocation in manufacturing systems using stochastic search techniques", *IEEE Transactions on Systems, Man, & Cybernetics - Part A: Systems ans Humans*, vol.26, no.2, pp.222-230, 1996.

[230] S.Patro and W.J.Kolarik, "Neural networks and evolutionary computation for real-time quality control of complex processe", *Proc. of the 1997 IEEE Annual Reliability and Maintainability Symposium*, pp.327-332, IEEE, 1997.

[231] C.G.Lu, D.Morton, Z.Wang, P.Myler and M.H.Wu, "A genetic algorithm solution of inspection path planning system for multiple tasks inspection on co-ordinate measuring machine (CMM), in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.436-441, IEE Conf.Publ. no.414, IEE, London, England, 1995.

[232] K.L.Mak and Y.S.Wong, "Design of integrated production production-inventory-distibution systems using genetic algorithms", in *GALESIA '95: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.454-460, IEE Conf.Publ. no.414, IEE, London, England, 1995.

[233] S.M.Disney, M.M.Naim and D.R.Towill, "Development of a fitneess measure for an inventory and production control system", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.351-355, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[234] D.Gong, G.Yamazaki and M.Gen, "Evolutionary program for optimal design of material distribution system", in *Proc.of the 1996 IEEE Int.Conf.on Evolutionary Computation*, pp.139-143, IEEE, New York, NY, USA,1996.

[235] D.J.Stockton and L.Quinn, "Aggregate production planning using genetic algorithms", *Proc. of the Inst.of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol.209, no.B3, pp.201-209, 1995.

[236] D.Wang and S.-C.Fang, "A genetics-based approach for aggregated production planning in a fuzzy eenvironment", *IEEE Transactions on Systems, Man, Cybernetics - Part A: Systems ans Humans*, vol.27, no.5, pp.636-645, 1997.

[237] W.Feng, G.B.Burns, and D.K.Harrison, "Using genetic algorithms bounded by dynamic linear constraints for marketing/production joint decision making", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.339-344, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[238] A.G.Garavelli, O.G.Okogbaa amd N.Violante, "Global manufacturing systems: a model supported by genetic algorithms to optimise production planning", *Computers & Industrial Engineering*, vol.31, no.1/2, pp.193-196, 1996.

[239] X.Jinxing, "An application of genetic algorithms for general dynamic lotsizing problems", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.82-87, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.

[240] H.A.Rao and P.Gu, "Entropic measure to determine reconfiguration using integrated systems design", in *Proc.of the 1995 IEEE Int.Conf. on Systems, Man & Cybernetics*, pp.518-523, IEEE, New York, NY, USA, 1995.

[241] N.Kubota, T.Fukuda and K.Shimojima, "Virus-evolutionary algorithm for a self-organising manufacturing system", *Computers & Industrial Engineering*, vol.30, no.4, pp.1015-1026, 1996.

[242] Y.Kawauchi, M.Inaba and T.Fukuda, "Genetic evolution and self-organisation of cellular robotic system", *JSME International Journal, Series C: Dynamics, Control, Robotics, Design and Manufacturing*, vol.38.no.3, pp.501-509, 1995.

[243] L.Zhao, Y.Tsujimura and M.Gen, "Genetic algorithm for robot selection and workstation assignment problem", *Computers & Industrial Engineering*, vol.31, no.3/4, pp.599-602, 1996.

[244] Y.Hamaguchi, H.Inoue, T.Jinguji, H.Yoshida and M.Shiozawa, "Transportation scheduling system based on evolution algorithm and super parallel computer", *Steps Forward: Proc.of the Second World Congress on Intelligent Transport Systems '95*, pp.2027-2030, Vehicle,Road & Traffic Int.Soc., Tokyo, Japan, 1995.

[245] M.Furukawa, M.Watanabe, A.Mizoe, T.Watanabe and Y.Kakazu, "Evolutionary computation applied to the logistic CIM system", in *Proc.of the Int.Conf. on Advance in Production Systems, APMS'96*, pp.319-324, Kyoto Univ., Kyoto, Japan, 1996.

[246] W.Mergenthaler, W.Stadler, H.Wilbertz and N.Zimmer, "Optimizing automotive manufacturing sequencees using simulated annealing and genetic algorithms", *Control Engineering Practice*, vol.3, no.4, pp.569-573, 1995.

[247] S.Sette, L.Boullart and L.V.Langenhore, "Optimising a production process by a neural network/genetic algorithm approach", *Engineering Applications in Artificial Intelligence*, vol.9, no.6, pp.681-689, 1996.

[248] B.A.Peters and M.Rajasekharan, "A genetic algorithm for determining facility design and configuration of single-stage flexible electronic assembly systems", *Journal of Manufacturing Systems*, vol.15, no.5, pp.316-324, 1996.

[249] P.G.Bachhouse, A.F.Fotheringham and G.Allan, "A comparison of a genetic algorithm with an experimental design technique in the optimisation of a production process", *Journal of the Operational Reseearch Society*, vol.48, no.3, pp.247-254, 1997.

[250] F.Corno, P.Prinetto, M.Rebaudengo, M.S.Reorda and S.Bisotto, "Optimising area loss in flat glass cutting", in *GALESIA '97: Genetic Algorithms in Engineering Systems: Innovations and Applications. Conference Proceedings*, pp.450-455, IEE Conf.Publ. no.446, IEE, Stevenage, England, 1997.