

Fuzzy job shop scheduling with lot-sizing

Sanja Petrovic · Carole Fayad · Dobrila Petrovic ·
Edmund Burke · Graham Kendall

Published online: 4 December 2007
© Springer Science+Business Media, LLC 2007

Abstract This paper deals with a problem of determining lot-sizes of jobs in a real-world job shop-scheduling in the presence of uncertainty. The main issue discussed in this paper is lot-sizing of jobs. A fuzzy rule-based system is developed which determines the size of lots using the following premise variables: size of the job, the static slack of the job, workload on the shop floor, and the priority of the job. Both premise and conclusion variables are modelled as linguistic variables represented by using fuzzy sets (apart from the priority of the job which is a crisp value). The determined lots' sizes are input to a fuzzy multi-objective genetic algorithm for job shop scheduling. Imprecise jobs' processing times and due dates are modelled by using fuzzy sets. The objectives that are used to measure the quality of the generated schedules are average weighted tardiness of jobs, the number of tardy jobs, the total setup time, the total idle time of machines and the total flow time of jobs. The developed algorithm is analysed on real-world data obtained from a printing company.

Keywords Job shop scheduling · Fuzzy rule-based system · Lot-sizing · Batching · Fuzzy multi-objective genetic algorithm · Real-world application · Dispatching rules

S. Petrovic (✉) · C. Fayad · E. Burke · G. Kendall
School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road,
Nottingham NG8 1BB, UK
e-mail: sxp@cs.nott.ac.uk

C. Fayad
e-mail: cxf@cs.nott.ac.uk

E. Burke
e-mail: ekb@cs.nott.ac.uk

G. Kendall
e-mail: gxk@cs.nott.ac.uk

D. Petrovic
Faculty of Engineering and Computing, Coventry University, Priory Street, Coventry CV1 5FB, UK
e-mail: D.Petrovic@coventry.ac.uk

1 Introduction

The scheduling of jobs and the control of their flow through a production process is essential to modern production/manufacturing companies. Ever since the first results of modern scheduling theory appeared some 50 years ago, scheduling has attracted a lot of attention from both academia and industry. The diversity of scheduling problems, their large-scale dimension and dynamic nature make them computationally very complex and difficult to solve. Various machine configurations that exist in real production/manufacturing environments have been studied in the literature (Leung 2004; Pinedo 2002), starting from the simplest case of a single machine, through a flow shop which includes machines in a series, to the most complex job shop environment (Blazewicz et al. 1996).

A job shop scheduling problem is an optimisation problem in which a given number of jobs have to be allocated over time to machines, under some constraints. For example, each job is assigned a predefined order of processing on the machines. In this paper, we consider two issues that are important in real-world job shop scheduling: lot-sizing and batching. Lot-sizing deals with the decision of when and how to split a job into lots, while batching refers to the scheduling of jobs with similar production requirements (Potts and Van Wassenhove 1995).

The concept of splitting jobs into lots is introduced with the aim of partially satisfying a customer's demand by producing and delivering, in the first lot, *enough* amount of a product, so that the customer is not out of stock of requested product while awaiting the whole job to be completed. The remaining part of the job will be delivered at some later date. In job-shop scheduling applications, the creation of lots permits the overlapping of different operations of the same job on parallel machines and may therefore reduce the throughput time (Potts and Van Wassenhove 1995). The sizes of lots have to be carefully chosen in order to avoid a schedule with a large number of lots of very small size that need large setup time on machines. The initial research on lot-sizing was related to the economic order quantity problem, where demand for a product is assumed to be stationary, i.e. an order for a product is continuously placed with a constant rate of demand within an infinite planning horizon (Wilson 1934). Some other models that considered more complex assumptions were proposed. The economic lot scheduling problem takes into consideration capacity constraints, such as machines capacity, inventory space capacity, etc. (Sarker and Newton 2002). A quite major step was made by introducing the dynamic demand conditions. The Wagner-Whitin problem subdivides the finite planning horizon into discrete periods to which different demands may be assigned (Wagner and Whitin 1958). Most popular current approaches to lot-sizing include capacitated (Sarker and Newton 2002), discrete (Oueniche and Boctor 1998) or a combination of these two approaches (Drexel and Kimms 1997). Literature on fuzzy approaches to lot-sizing is rather limited. Lee et al. (1991), and Karapapilidis et al. (2000) used fuzzy sets to model uncertain demand expressed by linguistic terms, while the objectives to be minimised are the setup cost and the inventory cost.

Batching is of particular importance in manufacturing systems in which changing the type of product to be processed on a machine incurs a setup time/cost. In order to reduce the setup time/cost, a set of jobs is grouped into families on the basis of their production requirements. As a result, no setup is needed when jobs of the same family are scheduled consecutively. On the other hand, the scheduling of jobs that belong to different families, one after the other, incurs setup time/costs. The size of a batch appears to be a sensitive issue. Large batches improve the efficiency of the production line as they save time and also increase the machine utilization. However, batches may group together jobs of different priorities, which means that jobs of high priorities can be processed in different batches. If the

batches are of large size, this will lead to tardiness of high priority jobs. Therefore, customer service may be improved by having smaller batches. The relatively recent survey by Potts and Kovalyov (2000) gives the extensive literature on models that integrate scheduling with batching decisions. They review models of batch scheduling developed for all classical machine settings. Agnetis et al. (2004) deal with the problem of finding a schedule that minimises the number of setups when jobs are subject to precedence constraints.

This paper focuses on the job shop problem which involves batching and lot-sizing in a printing company Sherwood Press (Nottingham, UK). This job shop problem involves a wide range of parameters such as job release dates, processing times, due dates, priorities of jobs, machine setup times, etc. Deterministic production scheduling models and algorithms assume that all parameters are well-known and precisely defined. However, as in almost all real-world production systems, in this printing company activities are fraught with uncertainties, and this prevents the application of deterministic scheduling theory. The deterministic scheduling models and algorithms have been extended to tackle stochastic cases, mainly to models that assume that processing times are random variables with specified probability distributions (Dempster et al. 1982). However, probabilistic characteristics of scheduling parameters are often lacking in practice. Information about production/manufacturing processes can be both imprecise and/or incomplete, or sometimes does not even exist. In these cases, application of the standard methods of probability theory becomes difficult and often inappropriate. Fuzzy sets provide an appropriate tool for handling imprecise information (Zadeh 1965). Fuzzy scheduling models have recently attracted increased interest among the scheduling research community (Slowinski and Hapke 2000). Imprecise scheduling parameters have been represented as fuzzy numbers and operations on them have involved fuzzy arithmetic. Parameters that are most often represented as fuzzy numbers are processing times and due dates (e.g. Kuroda and Wang 1996; Ishibuchi et al. 1994; Ishii et al. 1992), but there are models that deal with some other scheduling parameters by employing fuzzy sets (e.g. fuzzy job precedence relations, Ishii and Tada 1995, or machine breakdowns, Li et al. 1994). Fuzzy sets have also been used to represent flexible constraints whose violation has to be minimised. Most often, the models included flexible temporal constraints, where the best schedule requested the least relaxation of release dates or due dates constraints (Fargier 1996). As an attempt to bridge the existing gap between the scheduling theory and practice, fuzzy scheduling models that utilise multicriteria approaches have been developed and reported in the literature. Most of these models were restricted to single machine scheduling problems (e.g. Han et al. 1994; Adamopoulos and Pappis 1996), while only a few papers considered a more complex production environment such as a job shop (e.g. Sakawa and Kubota 2000; Tsujimura et al. 1993).

In our approach, fuzzy sets are applied to represent processing time of jobs that are uncertain due to both machines and human factors, and also to represent due dates that are flexible and allow the decision maker (scheduler) to express his/her attitude toward the tardiness of jobs. A novel fuzzy approach to lot-sizing is developed. In order to mimic the way that the scheduler decides on the lots and lot-sizing, fuzzy IF-THEN rules are introduced which conclude how to change lot sizes initially set based on imprecise data about the size of the job, its slack time, workload on the shop floor and the (crisp) priority of the job. A fuzzy bi-objective genetic algorithm was developed which evolved the decision on which dispatching rules to be used for scheduling jobs on each machine taking into consideration uncertain processing times and due dates of jobs (Fayad and Petrovic 2005). This genetic algorithm (GA) is enriched with three more objectives and used to test the developed fuzzy approach to lot-sizing of jobs. The objectives used to measure the quality of generated schedules are average weighted tardiness of jobs, the number of tardy jobs, the total setup time, the total idle time of machines and the total flow time of jobs.

The paper is organized as follows. Section 2 introduces a job shop problem and the real-world problem in Sherwood Press. Section 3 describes the fuzzy rule-based system developed for lot-sizing, while a fuzzy GA for job shop scheduling is presented in Sect. 4. Section 5 discusses the results obtained on real-world data obtained from Sherwood Press, followed by some concluding comments.

2 Problem statement

The notation used in the problem statement and throughout the paper is as follows:

N	total number of jobs.
J_j	job, $j = 1, \dots, N$.
M	total number of machines.
M_i	machine, $i = 1, \dots, M$.
a	time needed for setup when different families of jobs are processed one after the other.
r_j	release date of job J_j , $j = 1, \dots, N$, which denotes when a job can start its processing.
d_j	due date of job J_j , $j = 1, \dots, N$.
\tilde{d}_j	fuzzy due date of job J_j , $j = 1, \dots, N$.
C_j	completion time of job J_j , $j = 1, \dots, N$.
\tilde{C}_j	fuzzy completion date of job J_j , $j = 1, \dots, N$.
C_{\max}	makespan $C_{\max} = \max(C_1, \dots, C_j, \dots, C_N)$.
T_j	tardiness of job J_j , $j = 1, \dots, N$, $T_j = \max(C_j - d_j, 0)$.
w_j	importance of job J_j , $j = 1, \dots, N$.
(i, j)	operation of job J_j processed on machine M_i , $j = 1, \dots, N$ and $i = 1, \dots, M$.
p_{ij}	processing time of operation (i, j) needed for the required number of items q_j .
\tilde{p}_{ij}	fuzzy processing time of operation (i, j) needed for the required number of items q_j .
C_{ij}	completion time of operation (i, j) , $i = 1, \dots, M$, $j = 1, \dots, N$.
s_{ij}	start time of operation (i, j) , $i = 1, \dots, M$, $j = 1, \dots, N$.
$S_{AT}, S_{NT}, S_{ST}, S_{IT}, S_{TF}$	satisfaction grades of the objectives: average weighted tardiness, the number of tardy jobs, the total setup time, the total idle time of machines, the total flow time, respectively.
Φ	overall quality of the schedule measured as an aggregation of the satisfaction grades of the objectives.

The job shop scheduling problem considers N jobs $\{J_1, \dots, J_j, \dots, J_N\}$ with given release and due dates r_j, d_j , $j = 1, \dots, N$, respectively, that have to be scheduled on a set of M machines $\{M_1, \dots, M_i, \dots, M_M\}$. Each job consists of a chain of operations on which precedence constraints are imposed. Jobs may have different order of processing on the machines. Each operation of job J_j processed on machine M_i is represented by an ordered pair (i, j) , while its processing time is denoted by p_{ij} . If the job does not require processing on a machine, the processing time of the corresponding operation is equal to 0. Each job belongs to a certain family.

The scheduling problem is formulated as follows. Find a non-preemptive sequence of operations of N jobs on each of M machines subject to the following constraints: (1) the

precedence constraints which ensure that the processing sequence of operations conforms to the predefined order and (2) the capacity constraints which ensure that a machine processes one job at a time.

The following objectives are considered:

- (i) Minimise the average weighted tardiness of jobs in schedule s

$$AT(s) = \frac{1}{N} \sum_{j=1}^N w_j T_j. \tag{1}$$

- (ii) Minimise the number of tardy jobs in schedule s

$$NT(s) = \sum_{j=1}^N u_j, \quad \text{where } u_j = \begin{cases} 1 & \text{if } T_j > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

- (iii) Minimise the total setup time in schedule s

$$ST(s) = \sum_{i=1}^M \sum_{j=1}^N a X_{ij}, \tag{3}$$

where

$$X_{ij} = \begin{cases} 1 & \text{if } p_{ij} \neq 0 \text{ and job } J_j \text{ does not belong to the same family as its predecessor} \\ & \text{in the sequence of jobs on machine } M_i, \\ 0 & \text{otherwise.} \end{cases}$$

X_{ij} is referred to as the changeover coefficient and parameter a denoted the time needed to make a machine ready for a changeover of the job families.

- (iv) Minimise the total idle time of machines in schedule s , where the idle time is defined as time during which the machines are used neither for processing of jobs nor for setup

$$IT(s) = \sum_{i=1}^M \left(C_{ia} - \sum_{j=1}^N p_{ij} - \sum_{j=1}^N a X_{ij} \right) \tag{4}$$

where (i, a) and C_{ia} is the last operation on machine M_i and its completion time, respectively, $\sum_{j=1}^N p_{ij}$ is the total processing time of all jobs on machine M_i , and $\sum_{j=1}^N a X_{ij}$ is the total time required for setup on machine M_i .

- (v) Minimise the total flow time defined as the total time that jobs spend on the shop floor in a schedule s

$$TF(s) = \sum_{j=1}^N (C_j - r_j); \quad j = 1, \dots, N. \tag{5}$$

2.1 A real-world job shop problem

A job shop problem in Sherwood Press involves 18 machines, which are grouped into 7 work centres: Printing, Cutting, Folding, Card-inserting, Embossing and Debossing, Gathering, Finishing (Stitching, Trimming and Packaging). Each job has a predetermined order

of processing on machines in some or all work centres. Jobs are grouped into families on the basis of their colour requirements. The time needed for setup of the machines when different families of jobs are processed one after the other is 20 minutes ($a = 20$).

The processing times of jobs are uncertain and are represented by fuzzy sets. The estimation of the processing time of each operation is obtained, taking into account the nature of the machine in use. While some machines are automated and can be operated at different speeds, others are staff-operated and therefore the processing times are staff-dependent. Uncertain processing time \tilde{p}_{ij} is modelled by a triangular membership function represented by a triplet $(p_{ij}^1, p_{ij}^2, p_{ij}^3)$, where p_{ij}^1 and p_{ij}^3 are lower and upper bounds of the processing time, while p_{ij}^2 is a so-called modal point (Klir and Folger 1998). The values p_{ij}^1 , p_{ij}^2 and p_{ij}^3 are interpreted as pessimistic, moderate and optimistic processing times of job J_j on machine M_i . An example of a fuzzy processing time is shown in Fig. 1.

Following the scheduling practice in Sherwood Press, we classify jobs into three groups on the basis of their priorities and the corresponding tolerance of their tardiness. A job of high priority (*Priority 1*) is expected to be processed on time with no delay, and therefore there is a crisp due date defined for it. A job of medium priority (*Priority 2*) has a due date that can be extended up to two additional days, whereas a job of low priority (*Priority 3*) is given a tolerance of delay of up to one week on top of the originally set due date. The imprecise due dates \tilde{d}_j , $j = 1, \dots, N$, are represented by trapezoidal fuzzy sets illustrated in Fig. 2.

Fig. 1 Example of fuzzy processing time

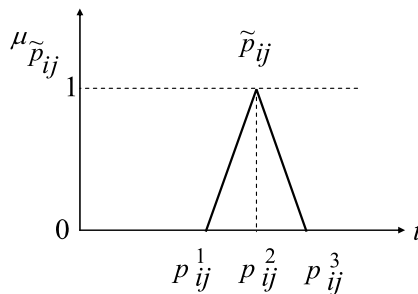
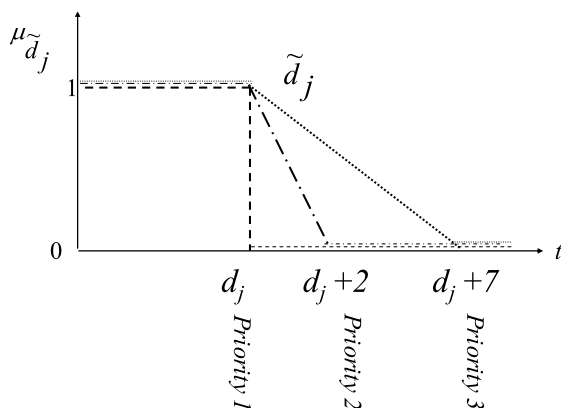


Fig. 2 Example of fuzzy due date



3 Fuzzy lot-sizing

In order to deliver at least part of the job ordered on time, jobs are split into lots. Four variables are identified to be important for the decision of whether to split up a job into lots and on lot-sizing: *size of job*, static *slack of job*, the *workload* on the shop floor and *priority of job*. The static slack represents the available time for the job processing, namely it is defined for job J_j as $d_j - r_j - \sum_{i=1}^M p_{ij}$. In order to mimic the way that the scheduler decides on lots, fuzzy IF-THEN rules are defined, which derive the size of each lot based on imprecise values of the identified variables (apart from the priority of the job which is a crisp number). *Workload* takes into consideration the number of items to be processed on the shop floor and is described by two linguistic terms: *Not Large* and *Large*. *Small*, *Medium* and *Large* are the linguistic terms used to describe *size of the job*. The priority of the job indicates its importance and is expressed by three crisp numbers: 1, 2 or 3 (introduced in Sect. 2.1). The slack of the job is described by three linguistic terms: *Small*, *Medium* and *Large*. These membership functions are defined in collaboration with Sherwood Press following their policy for lot-sizing and are given in Fig. 3.

It would be very difficult to define all 54 ($3 \times 3 \times 2 \times 3$) fuzzy rules, which determine the size of lots for all possible values of the four variables in the premises of the rules. To overcome this difficulty, we define fuzzy rules for each of the four variables independently (Petrovic 1991). The starting point in the lot-sizing decision is to split each job into two lots of equal size, i.e. 50% of the job size. A conclusion variable *change of lot* is introduced in the fuzzy rules, which indicates a necessary change of the current size of the lots, based on the truth value of the premise of the rule. It is described by the linguistic values *Large Negative*, *Medium Negative*, *Small*, *Medium Positive*, *Large Positive*. The membership functions of these fuzzy sets are given in Fig. 4.

The fuzzy rules are given in Fig. 5. The rules reflect the policy of the company. For example, if the size of the job is small or its slack is small then such a job should be processed as much as possible in the first lot. On the other hand, if a job is large and/or its slack is large, then its first lot should be smaller than the second lot, i.e. the processing of the major part of the job can be postponed. The workload on the shop floor and the priority of the job also influence the decision on lots, although to a smaller extent.

The defined four groups of fuzzy rules can be applied either together, resulting in one value of the conclusion variable *change of lot*, or in sequence, one by one, resulting in the change of lot sizes after the application of each group of the rules. We decided to adopt the second approach, which gives us the opportunity to express the preference of the production manager toward the importance of the variables for the decision on lots. The size of the job and its slack can be very important for the decision. For example, if the *size of job* is *Small* enough (i.e. the size of the job belongs to the fuzzy set *Small* with a membership degree close or equal to 1) and its *slack* is *Small* enough, then there is no need to split the job into lots. Therefore, if the increment of the initial size of the lot after the firing of the rules from Group I and II is greater or equal to 20%, then the job will not be split up into lots. If this is not the case, the *workload* and the *priority of job* are taken into consideration to further change the size of the lots. In addition, the policy of the company is not to have lots of size that is smaller than 10% of the total size of the job.

Before we explain the mechanism of our fuzzy rule-based system, we give a formal syntax of fuzzy rules. Each fuzzy rule R_s contains one imprecise variable in the premise and an imprecise conclusion and has the following form:

$$R_s: \text{ IF } x \text{ is } A_s \text{ THEN } y \text{ is } B_s \quad s = 1, \dots, S \quad (6)$$

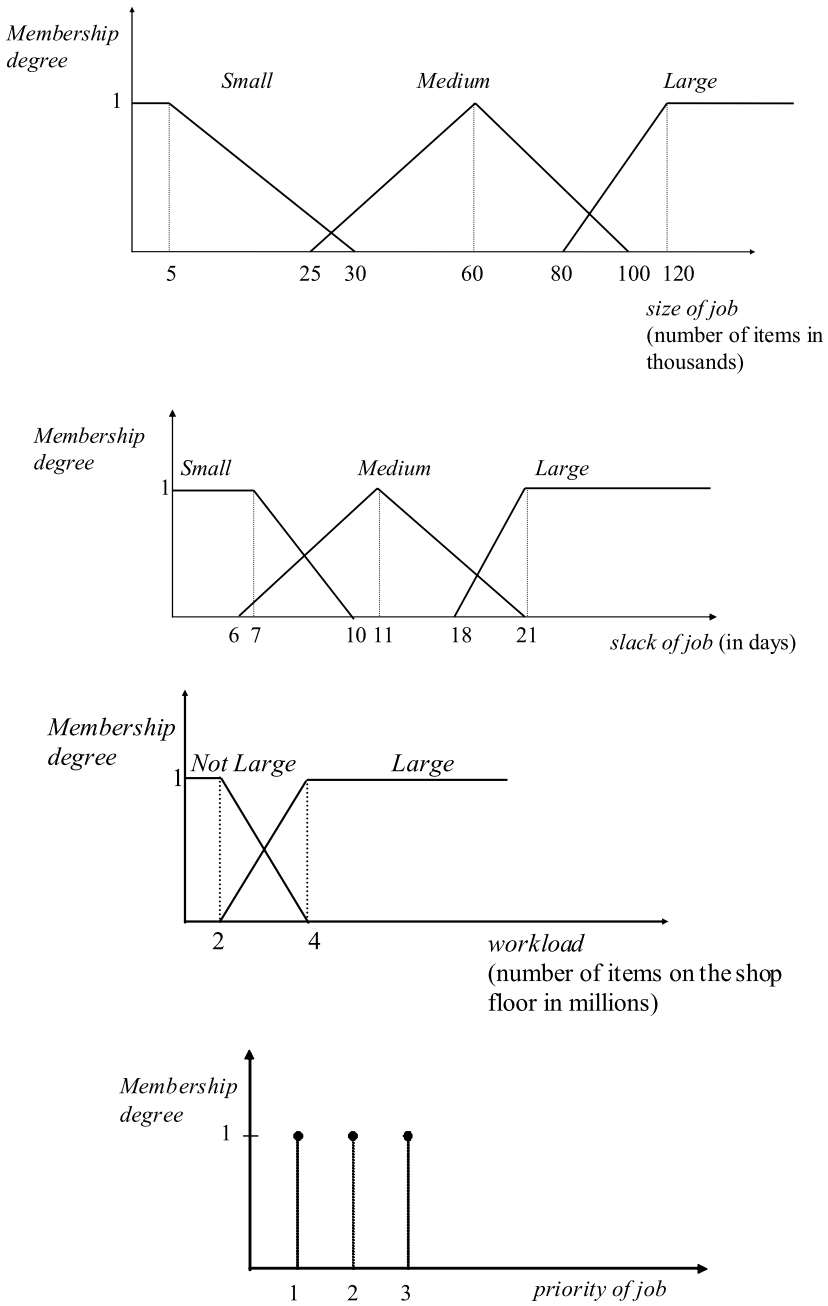


Fig. 3 Linguistic terms for the variables *size of job*, *slack of job*, *workload*, and *priority of job*

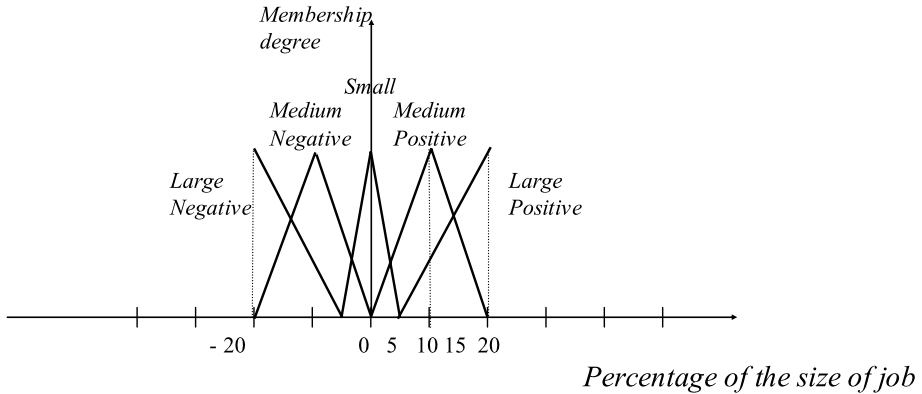


Fig. 4 Membership function of the linguistic terms of the conclusion variable

- | | |
|-------------------|--|
| <i>Group I:</i> | |
| 1. | IF size of job is <i>Small</i> THEN change of lot is <i>Large Positive</i> |
| 2. | IF size of job is <i>Medium</i> THEN change of lot is <i>Small</i> |
| 3. | IF size of job is <i>Large</i> THEN change of lot is <i>Large Negative</i> |
| <i>Group II:</i> | |
| 1. | IF slack of job is <i>Small</i> THEN change of lot is <i>Large Positive</i> |
| 2. | IF slack of job is <i>Medium</i> THEN change of lot is <i>Small</i> |
| 3. | IF slack of job is <i>Large</i> THEN change of lot is <i>Large Negative</i> |
| <i>Group III:</i> | |
| 1. | IF workload is <i>Not Large</i> THEN change of lot is <i>Medium Positive</i> |
| 2. | IF workload is <i>Large</i> THEN change of lot is <i>Medium Negative</i> |
| <i>Group IV:</i> | |
| 1. | IF priority of job is 3 THEN change of lot is <i>Medium Negative</i> |
| 2. | IF priority of job is 2 THEN change of lot is <i>Medium Positive</i> |
| 3. | IF priority of job is 1 THEN change of lot is <i>Small</i> |

Fig. 5 Fuzzy rules for lot-sizing

where x and y are imprecise variables, A_s and B_s , $s = 1, \dots, S$ are fuzzy sets, and S is the total number of rules.

The firing of each group of fuzzy rules involves the following steps (Pedrycz and Gowide 1998):

1. *Fuzzification* of the crisp input value: The degree to which the crisp input value belongs to the fuzzy set in the premises of the rules is determined.
2. *Evaluation* of each rule: The truth-value of a rule premise is equal to the value of the membership function of its premise $\mu_{A_s}(x_0)$ for a given crisp input x_0 . In our fuzzy rule-based system, the truth-value of the conclusion is determined using the *cutting* (or *truncation*) method (other methods are also possible). The membership function of the fuzzy set in the conclusion is cut at the level of the obtained truth-value $\mu_{A_s}(x_0)$ of the corresponding premise. The obtained fuzzy set is denoted by $\mu_{A_s \rightarrow B_s(x_0)}(y)$ and has the

following membership function:

$$\mu_{A_s \rightarrow B_s(x_0)}(y) = \min\{\mu_{A_s}(x_0), \mu_{B_s}(y)\}. \quad (7)$$

3. *Aggregation* of rule conclusions: A final conclusion R is a fuzzy set obtained by aggregating conclusions of all the rules that have fired. The sum operator was adopted for aggregation:

$$\mu_R(y) = \mu_{A_1 \rightarrow B_1(x_0)}(y) + \mu_{A_2 \rightarrow B_2(x_0)}(y) + \dots + \mu_{A_S \rightarrow B_S(x_0)}(y). \quad (8)$$

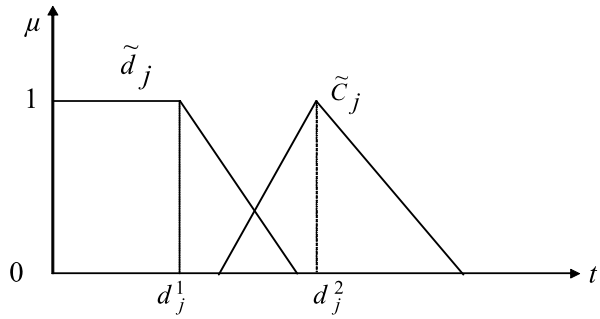
4. *Defuzzification*: A final fuzzy set R obtained in Step 3 is mapped into a crisp value \bar{y} . We apply here the centre of sums as a defuzzification method. This method is similar to the centre of gravity defuzzification method, but the overlapping areas of the truncated fuzzy sets in the rules are counted more than once (Ruspini et al. 1998).

$$\bar{y} = \frac{\sum_{s=1}^S (y \times \mu_R(y))}{\sum_{s=1}^S \mu_R(y)}. \quad (9)$$

4 A fuzzy genetic algorithm for job shop scheduling

A genetic algorithm (GA) is an iterative stochastic search procedure, which has been widely used for solving combinatorial optimisation problems (Goldberg 1989; Reeves 1995a; Sastry et al. 2005). It is motivated by biological systems and natural genetics. A genetic algorithm iteratively maintains a population of solutions called the generation. The fitness of a solution is determined by the value of the objective function. The solutions are represented as chromosomes which can be combined to produce offspring by crossover operations and modified by mutation operations in order to explore new areas of the solution space. Following the principles of natural selection, fittest solutions have a better chance of surviving to the next generation. The main characteristics of the GA developed for job shop scheduling with batching and lot-sizing are described below. Each chromosome consists of two sub-chromosomes of length M (number of machines), named *machines sub-chromosome* and *dispatching rules sub-chromosome*. The genes of the first sub-chromosome represent machines, while genes of the second one contain the dispatching rules to be used for sequencing operations on the corresponding machines. The machine sub-chromosome is initialized by randomly choosing machines M_i , $i = 1, \dots, M$. The *dispatching rules sub-chromosome* is initialised by choosing randomly one among the following six rules: *Early Due Date First*, *Shortest Processing Time First*, *Longest Processing Time First*, *Longest Remaining Processing Time First*, *Highest Priority First*, *Same Family of Jobs Together*. While the first four dispatching rules are well-established and widely used in the literature on job shop scheduling (Pinedo 2002), the last two are tailored to this application with the aim of reducing the flow time of jobs of higher priority and to schedule, whenever possible, jobs of the same families contiguously. An one-point crossover operator is applied on both sub-chromosomes of parents with a certain probability. Genes from two parent sub-chromosomes are combined in such a way as to preserve the absolute positions of the machines (dispatching rules) from one parent and the relative positions of the other parent, taking care that machines are not duplicated in the machine sub-chromosome (Reeves 1995b). Points are randomly chosen in both sub-chromosomes independently to increase the variation of coupling a machine with a dispatching rule. A mutation operator is applied independently in both sub-chromosomes,

Fig. 6 Satisfaction grade of completion time using area of intersection



where a randomly chosen pair of genes exchange their positions in a sub-chromosome. The roulette-wheel-selection technique is used for the selection of chromosomes which survive to the next iteration. The probability of survival of the chromosome is proportional to its fitness.

The fitness function is used to evaluate the quality of a given schedule driving the search towards *promising* areas in the solution space. In our problem, objectives are measured in different units i.e., they are incommensurable. For example, the average tardiness of jobs is given in time units, while the number of tardy jobs takes an integer value from the interval $[0, N]$. However, the objective values have to be used simultaneously to assess the quality of schedules. Satisfaction grades are introduced for each objective to reflect the decision maker preferences with respect to the achieved values of the objectives. The values of the objectives are mapped into satisfaction grades, which take values from the interval $[0, 1]$, where 0 represents full dissatisfaction and 1 denotes full satisfaction with the achieved objective value. The satisfaction grades of all the objectives are combined in an overall satisfaction grade.

The fuzzy processing time of job operations implies fuzzy completion times. Before we can assess the satisfaction grades of objectives that are due date related (average tardiness and number of tardy jobs), we need to establish a way to compare the fuzzy completion time of a job with its associated fuzzy due date, and calculate the likelihood that a job is tardy. We use the method of area of intersection introduced by Sakawa and Kubota (2000) which measures the intersection between two fuzzy values \tilde{C}_j and \tilde{d}_j (Fig. 6). The satisfaction grade of a fuzzy completion time of job J_j is defined in the following way:

$$S_T(\tilde{C}_j) = (\text{area } \tilde{C}_j \cap \tilde{d}_j) / (\text{area } \tilde{C}_j). \tag{10}$$

Parameter λ is introduced to define when a job is considered to be tardy. The number of tardy jobs in fuzzy environments is denoted by \overline{NT} and is calculated in the following way:

$$\overline{NT} = \begin{cases} 1 & S_T(\tilde{C}_j) \leq \lambda, \\ 0 & S_T(\tilde{C}_j) > \lambda, \end{cases} \quad 0 \leq \lambda \leq 1. \tag{11}$$

In our problem λ is set empirically to be 0.4.

The objectives (i) to (v) are transformed into objectives to maximize the corresponding satisfaction grades as follows.

Fig. 7 Satisfaction grade of number of tardy jobs

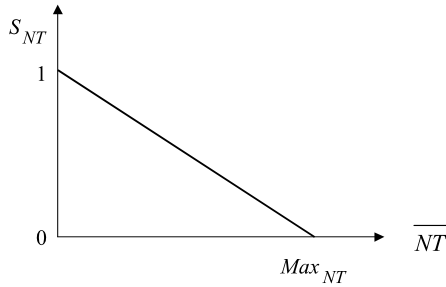
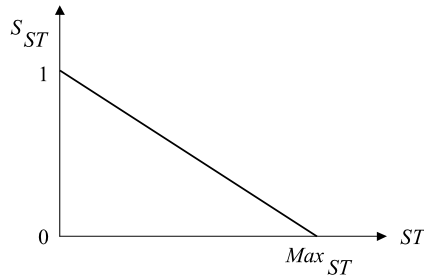


Fig. 8 Satisfaction grade of total setup time



(i) Satisfaction grade of average weighted tardiness of jobs S_{AT}

$$S_{AT} = \frac{1}{N} \sum_{j=1}^N w_j S_T(\tilde{C}_j). \tag{12}$$

(ii) Satisfaction grade of number of tardy jobs S_{NT} is presented in Fig. 7:

$$S_{NT} = \begin{cases} 1 & \text{if } \overline{NT} = 0, \\ (\text{Max}_{NT} - \overline{NT})/\text{Max}_{NT} & \text{if } 0 < \overline{NT} < \text{Max}_{NT}, \\ 0 & \text{if } \overline{NT} \geq \text{Max}_{NT} \end{cases} \tag{13}$$

where \overline{NT} is the number of tardy jobs calculated in expression (11), while Max_{NT} is equal to 20% of the total number of jobs and reflects a preference of the scheduler at Sherwood Press; there is full dissatisfaction when $\overline{NT} \geq \text{Max}_{NT}$, and full satisfaction when $\overline{NT} = 0$, while satisfaction degrees are linearly decreasing when \overline{NT} increases from 0 to Max_{NT} .

(iii) Satisfaction grade of total setup time S_{ST} (Fig. 8)

$$S_{ST} = \begin{cases} 1 & \text{if } ST = 0, \\ (\text{Max}_{ST} - ST)/\text{Max}_{ST} & \text{if } 0 < ST < \text{Max}_{ST}, \\ 0 & \text{if } ST \geq \text{Max}_{ST} \end{cases} \tag{14}$$

where ST is the total setup time calculated by the expression given in (3). A maximum setup time Max_{ST} is needed when all consecutive jobs belong to different families, i.e., $\text{Max}_{ST} = a \cdot N_{op}$, where N_{op} is the total number of operations on all machines.

Fig. 9 Satisfaction grade of total idle time

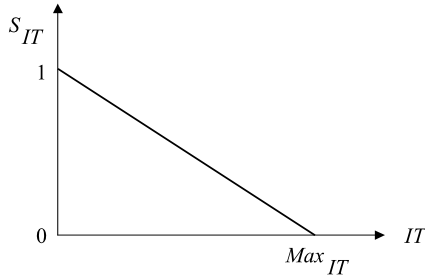
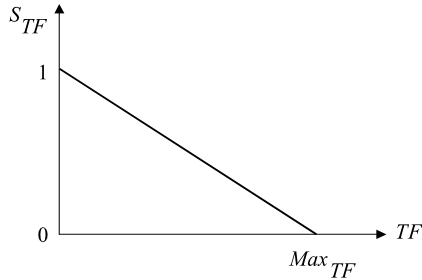


Fig. 10 Satisfaction grade of total flow time



(iv) Satisfaction grade of total idle times S_{IT} (Fig. 9)

$$S_{IT} = \begin{cases} 1 & \text{if } IT = 0, \\ (Max_{IT} - IT)/Max_{IT} & \text{if } 0 < IT < Max_{IT}, \\ 0 & \text{if } IT \geq Max_{IT} \end{cases} \quad (15)$$

where IT is the total idle time given in (4), while $Max_{IT} = \sum_{i=1}^M C_{ia}$ is the total time of machines being in use, (i, a) and C_{ia} is the last operation on machine M_i and its completion time, respectively.

(v) Satisfaction grade of total flow time S_{TF} (Fig. 10)

$$S_{TF} = \begin{cases} 1 & \text{if } TF = 0, \\ (Max_{TF} - TF)/Max_{TF} & \text{if } 0 < TF < Max_{TF}, \\ 0 & \text{if } TF > Max_{TF} \end{cases} \quad (16)$$

where TF is the total flow time which is calculated by the expression given in (5) while $Max_{TF} = \sum_{j=1}^N (C_{max} - r_j)$ is maximal flow time.

The satisfaction grade Φ of all objectives, used as the fitness function, is calculated as follows:

$$\Phi = (S_{AT} + S_{NT} + S_{ST} + S_{IT} + S_{TF})/5. \quad (17)$$

5 Performance of the fuzzy genetic algorithm on real-world data

Data sets obtained from Sherwood Press were used to test the performance of the developed fuzzy GA with lot-sizing and batching. The input data for each job are: identification number

Table 1 Lot sizes obtained using the fuzzy IF-THEN rules

Job	Size of job (in thousands)	Slack of job	Workload	Priority of job	Change of the first lot	First lot	Second lot
J_1	1	4	–	–	+30%	100%	0%
J_2	60	11	–	–	0%	50%	50%
J_3	70	10	6 388 763	3	–20%	30%	70%
J_4	70	19	6 388 763	3	–31%	19%	81%
J_5	140	9	6 388 763	1	–8%	42%	58%
J_6	120	10	6 388 763	2	+5%	55%	45%

of the job (*JobID*), the order of machines on which the job requires processing, the fuzzy processing time on each of the machines, the required quantity of the items, the family of the job, the release date, the due date and the priority of the job. The workload of the shop floor is calculated taking into consideration the required production of items. Batching is performed only on the printing machines (3 of them).

Two groups of tests were performed in order (a) to evaluate the performance of the fuzzy rule-based system for lot-sizing and (b) to evaluate the performance of the fuzzy GA which takes, as input data, the size of lots. The algorithm was implemented using Visual C++ in a Windows XP environment, and tests were run on a PC Pentium 4 CPU, 3.2 GHz with 2 GB of RAM.

5.1 Testing of the fuzzy rule-based system for lot-sizing

The fuzzy rules for lot-sizing are tested twofold using carefully chosen values for premise variables and using real-world data from Sherwood Press. Firstly, we tested the conditions for not splitting a job into lots. The results are shown in Table 1. For example, regarding its size and slack, job J_1 belongs to the fuzzy set *Small* (size) with a membership degree value of 1 and to the fuzzy set *Small* (slack) also with membership degree 1. It leads to a *Large Positive* increase of the first lot twice by 15% by firing the first two groups of rules. As the total size of the first lot is 80%, it would lead to the decision not to split that job into lots.

Job J_2 is of *Medium* size (membership degree 1) and *Medium* slack (membership degree 1) which does not lead to *change of lot* according to rules from Groups I and II. On the other hand, fuzzy rules in Groups III and IV yield –10% and +10%, respectively, and consequently the final *change of lot* will be 0%, i.e., the job will be split into two lots of the same size. The decision on lot-sizing is also evaluated using real-world data. We can notice that for example, jobs J_3 and J_4 have the same *size* (70 000 items) and *Priority* 3 but the latter is of larger slack and therefore its first lot is smaller than for job J_3 .

5.2 Testing of the fuzzy genetic algorithm

Three different approaches to lot-sizing were analyzed.

- (i) The described fuzzy rule-based system is used where jobs can be split up into two lots.
- (ii) We investigated whether the higher number of lots (up to 3 lots) can improve the performance of the schedule. The algorithm was applied as follows. The fuzzy IF-THEN rules are firstly applied to determine whether to split the job into lots and to decide on the size of the two lots. Then, the second lot was considered as a new problem and the

Table 2 Parameters of the GA

Population size	50
Length of the chromosome	36 ($M = 18$)
Crossover probability	0.9
Mutation probability	0.2
Termination condition	200 iterations

Table 3 Number of lots

Approach to lot-sizing	Number of lots
2 lots	248
3 lots	261
No lots	158

Table 4 Average and best values of the aggregated satisfaction grades and satisfaction grades of the objectives

Approach to lot-sizing	Φ	S_{AT}	S_{NT}	S_{ST}	S_{IT}	S_{TF}
2 lots						
Average	0.43	0.38	0.36	0.38	0.32	0.70
Best	0.52	0.43	0.76	0.40	0.34	0.75
3 lots						
Average	0.43	0.38	0.36	0.36	0.32	0.70
Best	0.53	0.43	0.71	0.43	0.38	0.76
No lots						
Average	0.37	0.38	0.26	0.12	0.32	0.79
Best	0.47	0.43	0.65	0.13	0.36	0.83

same groups of fuzzy rules were used to decide whether to split it into lots and to determine the size of the two additional lots. The same idea can be employed to a higher number of lots, but too high number does not make sense in practice.

(iii) Jobs were processed as whole and were not split up into lots.

The developed fuzzy GA was run with the parameters given in Table 2. A series of initial experiments were used to tune the algorithm.

The algorithms were run 10 times. The number of lots determined by the three different approaches is given in Table 3 and the results obtained using real-world data for one month are given in Table 4 (the best results are shown in bold). This specific month was chosen because it was considered to be rather busy with 158 jobs and a *workload* of 6 388 763 required items.

The average and the best values of the fitness function and the satisfaction grades of 5 objectives are presented for each approach to lot-sizing.

The higher number of lots leads to the higher number of operations of smaller size in terms of the number of items to be processed, i.e. there is a higher number of operations to be scheduled than in the scheduling without lots. Although these schedules consist of higher number of operations, contrary to the expectation, the obtained average satisfaction grade of the total setup time is higher in the schedules with lots. The explanation for this lies in

Table 5 Batching with and without lots

Approach to lot-sizing	Number of batches	Number of operations	Batch average size
2 lots	172	308	1.8
3 lots	172	344	2
No lots	161	183	1.14

batching. Higher number of operations gives better opportunities for batching. There is a smaller number of batches (due to the fact that more jobs of the same family can be grouped together) and higher average size of batches in the schedules with lots than in the schedules without lots. The total number of batches, the total number of operations to be scheduled (on the printing machines that are relevant for batching) and the average size of batches are given in Table 5. It can be concluded that the batch average size increases with the increase of number of lots.

In the calculation of the due-date oriented objective values (average tardiness and the number of tardy jobs) in the schedules with lots, we consider each lot to be a separate job. Therefore, the results presented in Table 4 need to be interpreted carefully. It can be observed that the average tardiness values achieved in all approaches, with and without lots, are the same, while the best satisfaction grade of the number of tardy jobs is highest in the schedules with 2 lots (the average satisfaction grade is the same in the schedules with 2 and 3 lots).

In order to investigate the effect of splitting jobs into lots on tardiness, we also record the number of tardy jobs taking into consideration the completion of jobs in the first lots. Table 6 shows the average and best values obtained of the number of tardy jobs, the number of tardy first lots, the percentage of tardy jobs $100 \cdot (\frac{NT}{N})$ and the percentage of tardy first lots $100 \cdot (\frac{\text{number of tardy first lots}}{NT})$. The 2-lot approach gives the lowest number of tardy first lots. Also, the second (third) lots determine the ultimate quality of the schedule but the smaller number of tardy jobs in the first lots may increase the satisfaction grade of the customer.

Dispatching rules applied to sequence jobs on each machine have a different effect on the introduced objectives. In the developed GA, these dispatching rules are chosen with the same probability. Further experiments are performed when assigning a higher probability to some of the rules. For example, in order to increase the satisfaction grade of the total setup time, the probability of applying the dispatching rule *Same Family of Jobs Together* is increased (3 times). As expected, the total setup time was improved (for example, in the 2-lot approach, the satisfaction grade of the setup time was increased from 0.43 to 0.66), but at the expense of the fitness function decrement from 0.51 to 0.28 (best values). Similar experiments can be performed with other dispatching rules.

6 Conclusion

This paper considers a real-world job shop problem that is faced by a printing company. The processing times of jobs and their due dates are imprecise and modelled by fuzzy sets. A fuzzy rule-based system, which reflects the scheduling policy of the printing company, is developed to determine the lot sizes. A fuzzy multi-objective genetic algorithm is employed to generate schedules taking into consideration the following objectives: to minimize average tardiness, the number of tardy jobs, the setup times, the idle times of machines and the flow time of jobs. Fuzzy sets are used to represent satisfaction grades of the objectives taking into consideration the preferences of the scheduler on the shop floor. A genetic algorithm is

Table 6 Tardy jobs versus tardy lots using the different approaches

Approach to lot-sizing (total number of jobs)	Number of tardy jobs	Number of tardy first lots	Percentage of tardy jobs	Percentage of tardy first lots
2 lots (266)				
Average	15.4	8.5	5	55
Best	14	8	5	57
3 lots (299)				
Average	20.4	11.8	7	58
Best	22	11	7	50

developed to search for a schedule with maximum satisfaction grades of the objectives. The algorithm uses lot-sizing of jobs determined by the fuzzy rule-based system. In addition, the batching of jobs with similar characteristics is taken into consideration, which leads to the reduction of the total setup time of machines. The results obtained on real-world data from a printing company are given and analyzed. The obtained results support the application of fuzzy rule-based system to the lot-sizing. The success of the methodology described in this paper and the quality of the results has led to a collaborative programme of implementation within the company.

In most real-world production environments, scheduling is on-going process where circumstances in both external business environments and in internal production environments may change dynamically. In our future research work, we will focus on issues regarding the dynamic scheduling, namely how to respond to disturbances such as machine breakdowns, delays in the arrival of materials, arrival of new orders, changes in current orders, etc.

Acknowledgements The authors would like to thank the Engineering and Physics Science Research Council (EPSRC), UK, for supporting this research (Grant No. GR/R95319/01 and GR/R95326/01). We would also like to acknowledge the support of our industrial collaborator Sherwood Press Ltd, Nottingham.

References

- Adamopoulos, G. I., & Pappis, C. P. (1996). Fuzzy-linguistic approach to a multi-criteria sequencing problem. *European Journal of Operational Research*, 92(3), 628–636.
- Agnetis, A., Alfieri, A., & Nicosia, G. (2004). A heuristic approach to batching and scheduling a single machine to minimise setup costs. *Computers & Industrial Engineering*, 46(2), 793–802.
- Blazewicz, J., Domschke, W., & Pesch, E. (1996). The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*, 93, 1–33.
- Dempster, M. A. H., Lenstra, J. K., & Rinnooy Kan, A. H. G. (Eds.). (1982). *Deterministic and stochastic scheduling*. Dordrecht: Reidel.
- Drexl, A., & Kimms, A. (1997). Lot sizing and scheduling—Survey and extensions. *European Journal of Operational Research*, 99, 221–235.
- Fargier, H. (1996). Fuzzy scheduling: Principles and experiments. In D. Dubois, H. Prade, & R. R. Yager (Eds.), *Fuzzy information engineering, a guided tour of applications*. New York: Wiley.
- Fayad, C., & Petrovic, S. (2005). A genetic algorithm for the real world job-shop scheduling. In M. Ali & F. Esposito (Eds.), *Lecture notes in artificial intelligence: Vol. 3533. Innovations in applied artificial intelligence* (pp. 524–533). New York: Springer.
- Goldberg, D. (1989). *Genetic algorithms in search, optimisation, and machine learning*. Reading: Addison-Wesley.
- Han, S., Ishii, H., & Fujii, S. (1994). One machine scheduling problem with fuzzy due dates. *European Journal of Operational Research*, 79(1), 1–12.

- Ishibuchi, H., Yamamoto, N., Misaki, S., & Tanaka, H. (1994). Local search algorithms for flow shop scheduling with fuzzy due-dates. *International Journal of Production Economics*, 33, 53–66.
- Ishii, H., & Tada, M. (1995). Single machine scheduling problem with fuzzy precedence relation. *European Journal of Operational Research*, 87(2), 284–288.
- Ishii, H., Tada, M., & Masuda, T. (1992). Two scheduling problems with fuzzy due-dates. *Fuzzy Sets and Systems*, 6(3), 336–347.
- Karapapilidis, N., Pappis, C., & Adamopoulos, G. (2000). Fuzzy set approaches to lot sizing. In *Scheduling under fuzziness* (pp. 291–304). Heidelberg: Physica.
- Klir, G., & Folger, T. (1998). *Fuzzy sets, uncertainty and information*. New Jersey: Prentice Hall.
- Kuroda, M., & Wang, Z. (1996). Fuzzy job shop scheduling. *International Journal of Production Economics*, 44(1–2), 45–51.
- Lee, Y. Y., Kramer, B. A., & Hwang, C. L. (1991). A comparative study of three lot-sizing methods for the case of fuzzy demand. *International Journal of Operations and Production Management*, 11(7), 72–80.
- Leung, J. (Ed.). (2004). *Handbook of scheduling: Algorithms, models, and performance analysis*. New York: Chapman & Hall/CRC Press.
- Li, Y., Luh, P. B., & Guan, X. (1994). Fuzzy optimization-based scheduling of identical machines with possible breakdown. In *Proceedings—IEEE international conference on robotics and automation* (pp. 3447–3452). San Diego, USA, May 1994.
- Ouenniche, J., & Boctor, F. (1998). Sequencing, lot sizing and scheduling of several products in job shops: The common cycle approach. *International Journal of Production Research*, 36(4), 1125–1140.
- Pedrycz, W., & Gowide, F. (1998). *An introduction to fuzzy sets—analysis and design*. Cambridge: MIT Press.
- Petrovic, D. (1991). Decision support for improving systems reliability by redundancy. *European Journal of Operational Research*, 55, 357–367.
- Pinedo, M. (2002). *Scheduling: Theory, algorithms, and systems* (2nd edn.) New Jersey: Prentice Hall.
- Potts, C., & Kovalyov, M. (2000). Scheduling with batching: A review. *European Journal of Operational Research*, 120, 228–249.
- Potts, C. N., & Van Wassenhove, L. N. (1995). Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *Journal of Operational Research Society*, 43(5), 395–406.
- Reeves, C. (1995a). Genetic algorithms. In V. J. Rayward-Smith (Ed.), *Modern heuristic techniques for combinatorial problems* (pp. 151–196). London: McGraw-Hill International.
- Reeves, C. (1995b). A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, 22(1), 5–13.
- Ruspini, E. H., Bonissone, P., & Pedrycz, W. (Eds.). (1998). *Handbook of fuzzy computation*. Bristol: Institute of Physics Publishing (IOP).
- Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. *European Journal of Operational Research*, 120(2), 393–407.
- Sarker, R., & Newton, C. (2002). A genetic algorithm for solving economic lot size scheduling problem. *Computers & Industrial Engineering*, 42, 189–198.
- Sastry, K., Goldberg, D., & Kendall, G. (2005). Genetic algorithms. In E. K. Burke & G. Kendall (Eds.), *Search methodologies: Introductory tutorials in optimization and decision support techniques* (pp. 97–125). New York: Springer.
- Slowinski, R., & Hapke, M. (Eds.). (2000). *Scheduling under fuzziness*. Heidelberg: Physica.
- Tsujimura, Y., Park, S., Change, I. S., & Gen, M. (1993). An efficient method for solving flow shop problem with fuzzy processing times. *Computer and Industrial Engineering*, 25, 239–242.
- Zadeh, L. A. (1965). Fuzzy sets. *Information & Control*, 8, 338–353.
- Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management Science*, 5, 89–96.
- Wilson, R. H. (1934). A scientific routine for stock control. *Harvard Business Review*, 13, 116–128.