Multi-objective fuzzy job shop scheduling

Manjeet Singh and Gürsel A. Süer*

Industrial and Systems Engineering,
Ohio University,
Athens, OH, 45701, USA
Fax: (740) 593-0778
E-mail: ms312507@ohio.edu
E-mail: suer@ohio.edu
*Corresponding author

Feyzan Arikan

Industrial Engineering Department,
Gazi University,
Celal Bayar Boulevard, 06570 Maltepe-Ankara, Turkey
E-mail: farikan@gazi.edu.tr

Abstract: The job shop scheduling problem (JSSP) deals with determining schedule for each resource/machine under job and machine flow restrictions such that the selected objective function is satisfied. Multi-objective scheduling is widely used to obtain desirable results in the existence of more than one performance measure in scheduling problems. The paper focuses on multi-objective scheduling in a job shop environment. One of the useful methods in multi-objective environment is the use of fuzzy operators in modelling the system. Fuzzy operators provide the model with characteristics where user can input desired bounds for all of the performance measures with suitable membership functions. In this study, three mathematical models are presented and combined as a multiple objective scheduling model where the considered three objectives are minimising number of tardy jobs, total tardiness and maximum tardiness, respectively. For the solution of the model, fuzzy programming is utilised by using min operator and augmented max-min operator. The model with augmented max-min operator found only non-dominated solutions. Then mathematical model optimising two performance measure is also discussed. The quality between solutions obtained from math model optimising two performance measures model versus three performance measures is also discussed.

Keywords: job shop scheduling; fuzzy theory; non-dominated solutions.


Biographical notes: Manjeet Singh is a PhD student at the Department of Industrial and Systems Engineering at Ohio University. He received his BE in Mechanical Engineering from Maharishi Dayanand University of Rohtak, India and his MS degree from the Department of Industrial and Systems Engineering at Ohio University. His research interests are in math modelling and optimisation of scheduling systems, manufacturing system design, supply chain management, network optimisation, vehicle routing and capacity

Copyright © 2014 Inderscience Enterprises Ltd.
planning. He also has thorough knowledge of operations research techniques including linear and mixed-integer programming, heuristics, evolutionary algorithm, stochastic optimisation, multi-objective decision making, and discrete event simulation. He has published over six papers in journals and conference proceedings and made over three conference presentations.

Gürsel A. Süer is a Professor in the Industrial and Systems Engineering Department at Ohio University. He has obtained his BSIE and MSIE degrees from Middle East Technical University, Ankara, Turkey and his PhD in IE from Wichita State University. He is on the editorial board of various journals. Currently, he serves as the Manufacturing Area Editor of the *Computers and Industrial Engineering Journal*. He has co-chaired two Computers and Industrial Engineering Conferences (1997-Puerto Rico, 2005-Istanbul). He initiated Group Technology Conferences held in 2000-Puerto Rico, 2003-Ohio, 2006-Netherlands and 2009-Japan. He has consulted various companies and carried out several funded projects. He has published over 110 papers in journals, edited books, conference proceedings and made over 100 technical presentations.

Feyzan Arikan received her PhD in Industrial Engineering from Gazi University, Ankara, Turkey in 2002. She has been an Assistant Professor at the Department of Industrial Engineering, Gazi University since 2005. She has served as the European Credit Transfer System (ECTS) Coordinator Assistant of the department. She has worked as the accreditation committee representative of the department in The Accreditation Commission of Engineering Faculty, Gazi University. Her research interests include fuzzy mathematical modelling, multiple criteria decision making, manufacturing cell design and supplier selection. She has received several rewards sponsored by TUBITAK and Gazi University.

1 Introduction

With the severe pressure among competitive enterprises, time is becoming one of the most significant success factors. Through well-designed scheduling, enterprise can maximise their profits by increasing production efficiency and resource utilisation. Job-shop scheduling is concerned with optimisation of schedule on machines by determining an optimal/good processing order of the jobs on each machine to achieve specific objectives. In diverse application fields like production planning, job-shop problem has engaged many researchers’ attention to finding the optimal way to arrange activities to meet certain requirements or objectives. Previous scheduling research has focused on regular performance measures as the objective function, where the optimal schedule determination is relatively simple. Some commonly used performance measures are makespan (maximum completion time), number of tardy jobs etc. However, in more complex environments, multi-objective scheduling becomes relevant and therefore is widely used in tackling such issues. Considering other alternative optimisation criteria, more complicated objective functions and corresponding constraints are developed for solving practical problems. Among many different types of objectives, the due date of a job has special contributions to manage scheduling and number of tardy jobs is one of the important objectives related to due date. A job is defined as a tardy job when a job is completed after its due date (completion time > due date). The number of tardy jobs is a
Multi-objective fuzzy job shop scheduling

A statistic which indicates all the tardy jobs in the system. The equation (1) shows the number of tardy jobs in a system of \( n \) jobs.

\[
n_T = \sum_{i=1}^{n} n_i \quad \forall n_i \in (0, 1)
\]

(1)

\[
n_i = \begin{cases} 1, & \text{if tardy} \\ 0, & \text{else} \end{cases}
\]

(2)

While increasing the satisfactory percentage of on-time shipments by minimising this objective, it may cause schedules to have some extremely tardy jobs. This results in impractical solutions which cannot be applied in the real life cases. For this reason, minimising the total tardiness and maximum tardiness are proposed as fairly suitable objectives to optimise a schedule. Total tardiness is defined as the summation of tardiness of all of the jobs in the system (it is either positive for tardy jobs or zero for non-tardy jobs). The equation (3) shows the formula to compute total tardiness in the system and the equation (4) shows tardiness for a job \( i \), where \( c_i \) is the completion time of job \( i \) and \( d_i \) is the due date of job \( i \). Maximum tardiness is the tardiness of most tardy job in the system. Therefore, it is maximum among the individual tardiness values as shown by equation (5).

\[
TT = \sum_{i=1}^{n} t_i
\]

(3)

\[
t_i = \max \{0, (c_i - d_i)\}
\]

(4)

\[
T_{\max} = \max (t_1, t_2, \ldots, t_n)
\]

(5)

However, besides the number of tardy jobs and/or total tardiness, another important performance measure for manufacturers is maximum tardiness. Because only minimising the number of tardy jobs and/or total tardiness does not necessarily minimise maximum tardiness of a job. Since unacceptable tardiness may cause delay with large cost, minimising maximum tardiness is another significant objective in addition to the first two objectives.

Many scheduling systems are prevalent in manufacturing industry today. Job-shop is one of the most widespread and challenging production systems. In a job shop problem, a set of jobs must be processed on a set of machines. Each job consists of a sequence of task operations. The sequence of machines for each job is predefined along with the processing times on each machine. All operations of a job must be processed using any of the machines only once (if required) in the given order.

Since only one objective cannot satisfy the needs of real life problems, multi-objective scheduling is an important field. For instance, minimising the total tardiness could be the objective but the manufacturer is also interested in controlling the number of tardy jobs and maximum tardiness simultaneously. To solve such problems the emergence of multi-objective scheduling has been a more suitable idea. The results obtained from multi-objective scheduling are more balanced than single objective solutions; therefore solutions are essentially more flexible.

In this study, three novel mathematical models are presented and combined as a multiple objective scheduling model where the considered objectives are optimising number of tardy jobs, total tardiness and maximum tardiness. For the solution of the
model, fuzzy programming is utilised by using min operator, augmented max-min operator and its weighted version. The rest of the paper is organised as follows. In Section 2, literature is reviewed. In Section 3, mathematical models for each objective are discussed separately. In Section 4, after preliminary definitions of fuzzy mathematical programming (FMP), fuzzy linear programming is explained. Thereafter, fuzzy mathematical models which find a fuzzy optimal solution for all three objectives for set bounds is discussed. The capability of different fuzzy mathematical models to produce non-dominated solutions is compared and discussed in detail in Section 5. Another important discussion is the procedure used to generate non-dominated solutions for three performance measures. Finally, the conclusions and future research are presented in Section 6.

2 Literature review

Average flow time, makespan, number of tardy jobs, total tardiness, cost factors related with production, inventory and labour and their variations are significant scheduling decisions criteria. The first exploration on multi-criteria scheduling was introduced in 1970, however, most of the previous research only focuses on one criterion at a time since it is hard to solve scheduling problems (Bagchi, 1999). There are three main approaches to solve the multi-criteria scheduling problems. The first class is the optimisation of one performance measure given a fixed value for a second performance measure (Babayigit and Süer, 2003). In the second class, Mondal and Sen (2001) accounts for a linear combination or weighted additive representation of the different criteria as a unique objective function. With the aim to obtain non-dominated solutions, the third class considered various criteria simultaneously (Stein and Wein, 1997; Süer and Allard, 2009). An extensive review of the literature on multiple objectives scheduling problems can be found in Nagar et al. (1995), T’Kindt et al. (2003), Hoogeveen (2005) and Lei (2009).

To solve the multi-objective mathematical programming problems, Zimmermann (1978) developed a fuzzy linear programming model to obtain non-dominated solutions for a scheduling problem with various criteria. The introductions to fuzzy decision (Bellman and Zadeh, 1970) and fuzzy linear programming formulation (Zimmerman, 1978) have been applied to many multiple objective optimisation problems to obtain compromise solutions. In the multi-objective model, a compromise solution is an efficient solution and ‘as close as possible’ to the ideal solutions of each objective function (Zeleny, 1973; Zimmermann, 1985).

When Zimmermann (1978) fuzzy linear programming employed for the multiple objective programming problems, fuzzy intervals are represented by using membership functions and are defined by ideal and anti-ideal solutions of the objective functions. Fuzzy operators in mathematical programming can be used to describe the relationship between each membership function (see e.g. in Luhandjula, 1982; Pedrycz, 1983; Werners, 1988; Yager, 1980, 1988; Zimmermann and Zysno, 1980). Zimmermann (1978) proposed the min operator and product operator as two aggregation operators to aggregate the objective functions. In the literature, min-operator corresponds to the set-theoretical intersection in fuzzy mathematical modelling. Although it does not guarantee a non-dominated solution, it is the most frequently used aggregate operator because of the ease of computation (Buckley, 1987; Zimmermann, 1985).
meantime, Zimmerman proposed product operator as a second operator by to aggregate the objective functions. The set of solutions to fuzzy problem is the Pareto optimal set for any multiple objective programming problem when the product operator is used to combine the membership functions (Buckley, 1987). However, product operator is rarely used because the resulting fuzzy problem is a non-linear optimisation problem and difficult to solve when the product operators are used.

To avoid the disadvantage of the min operator, Süer et al. (2008) has employed the arithmetical average aggregation operator instead of the min operator. By applying arithmetical average aggregation operator, the solution of the mathematical model gives non-dominated solution (Lee and Li, 1993). However, there is one advantage in the solution: the high performance objectives or goals are paid a very high attention to, while the low performance objectives are neglected. This unbalanced solution is not desirable in compromise programming (Zeleny, 1973).

An improved max-min model was generated to manage possibilistic interest rate risk in a basic bank hedging decision problem (Lai and Hwang, 1993). In the fuzzy model, a convex combination of max and min operators, which is a modified version of Zimmermann and Zyno’s γ operator, was used to solve the model (Lai and Hwang, 1996). Lai and Hwang’s (1993) improved max-min model gives both non-dominated and balanced solution for fuzzy goal programming problems.

An extensive review and classification of the approaches for the fuzzy multiple objective optimisation approaches can be found in Arikan and Gungor (2007). Applications of different fuzzy operators for fuzzy optimisation can be found in Süer et al. (2008, 2009).

In this study, three mathematical models are combined as a multiple objective scheduling model where the considered objectives are minimising number of tardy jobs ($n_T$), minimising total tardiness ($TT$) and maximum tardiness ($T_{\text{max}}$). For the solution of the model, fuzzy programming is utilised by using min operator, augmented max-min operator and its weighted version.

Fuzzy due dates, processing times, makespan and flow time are generally utilised to express the uncertain time parameters in the fuzzy scheduling problem literature. Fuzzy logic is also widely used in literature to generate aggregate dispatch rules in scheduling and assignment of precedence relation between jobs. Fuzzy multi-objective scheduling research is concentrated on job shop scheduling problems (JSSP) (Lei, 2009). Fuzzy job shop scheduling studies can also be found in Guiffrida and Nagi’s (1998) literature review for the fuzzy set theory (FST) applications in production management research. A wide review of the fuzzy scheduling problems can be found in Slowinski and Hapse (2000).

In the realm of multiple objective flow-shop scheduling problems, Lei (2009) mentioned 38 studies which generally considered binary combination of the following performance criteria: minimising makespan, total flow time, machine idle time, maximum earliness and total tardiness.

The JSSP is classified as one of the most difficult NP-complete problems. To solve job shop scheduling hierarchically, Brandimarte (1993) pointed out an integrated approach by adopting the dispatching rules for solving the machine selection problem and solving the sequencing problem using the different Tabu search simultaneously. With respect to the fuzzy job shop problem, Allet (2003) introduced a resolution method (the horizon method) in the deterministic case (fixed data and strict constraints). Xia and Wu (2005) used hybrid approach combined particle swarm optimisation and simulated
annealing for the multi-objective flexible job-shop scheduling problem. Zhang and Wu (2010) proposed the divide-and-conquer methodology to solve large JSSPs with the objective of minimising total weighted tardiness. By using a simulated annealing procedure, promising decomposition policy for the operation set were searched and all of the sub-problems are sequentially solved by a particle swarm optimisation algorithm. Aiming at minimising makespan, Ploydanai and Mungwattana (2010) investigated the machines with different machine availability constraint and developed new algorithm based on a non-delay scheduling heuristic by adding machine availability constraint to solve JSSP. Shafia et al. (2011) presented flexible simulated annealing algorithm to find near optimal solutions for a robust JSSP.

The mathematical models are based on the basic idea of disjunctive mathematical modelling taken from Baker (1974). The application of mathematical modelling to a multi-objective environment wherein three objectives are optimised simultaneously for a job shop problem has not been tried before (to the best of our knowledge). These objectives share a relationship and therefore affect the total satisfaction of the user because in practical situations the system is generally considered from a multi-objective point of view. Therefore, this study provides an introduction to solving such problems. The methodology employed finds all of the non-dominated solutions of the domain searched. This is done to ensure the findings of efficient fuzzy models lies within this set of non-dominated solutions irrespective of the bounds used and thereby also validates the solutions.

3 Preliminary definitions

Before presenting the mathematical models and fuzzy models for JSSP some basic definitions in multiple objective optimisation and FST (Zadeh, 1965) are introduced here.

3.1 Basic definitions for multiple-objective optimisation

For the multiple objective programming problem with $S$ objective functions $f_s(x) = c_s x$, $s = 1, 2, \ldots S$:

$$\max z = \left[ f_1(x), f_2(x), \ldots, f_s(x) \right]_T$$

s.t. $x \in X$  \hspace{1cm} (6)

where $X = \{x \in R^n : Ax \leq b, x \geq 0 \}$, $c_s = (c_{s1}, \ldots, c_{sn}) \in R^n$, $s = 1, \ldots S$; $b = (b_1, \ldots, b_m) \in R^m$ and $A$ is a $R^{m \times n}$ matrix. All of the objective functions, subject to the constraint set, in the problem (6) cannot be optimised simultaneously. This happens due to the functions’ conflicting nature.

**Definition 3.1.1:** The ideal solution of problem (6) is the vector whose components are the maximum value of each objective function obtained by solving each of them individually subject to the given constraint set. It can be represented as follows:

$$z^* = \left[ f_1^*(x), f_2^*(x), \ldots, f_S^*(x) \right] = \left[ \max f_1(x), \max f_2(x), \ldots, \max f_S(x) \right]$$

(7)

Therefore, the anti-ideal solution vector or negative ideal solution of the same problem is defined as
Multi-objective fuzzy job shop scheduling

\[ z^* = [f_1^*(x), f_2^*(x), \ldots, f_s^*(x)] = [\max f_1(x), \max f_2(x), \ldots, \max f_s(x)] \]  

**Definition 3.1.2:** The solution \( x^* \) to the problem mentioned in (6) is efficient (Pareto optimal or non-dominated) if and only if there does not exist any \( x \in X \) such that \( f_i(x) \geq f_i(x^*) \) for all \( s = 1, 2, \ldots, S \) and \( z_s(x) > z_s(x^*) \) for at least one \( s \) (Tabucannon, 1988).

### 3.2 Basic definitions in FST

**Definition 3.2.1:** A fuzzy subset \( \tilde{A} \) of a universe of discourse \( X \) is defined by a membership function \( \mu_A(x) : X \rightarrow [0, 1] \) which associates with each element \( x \) of \( X \), a number \( \mu_A(x) \) in the interval \([0, 1]\), where \( \mu_A(x) \) represents the grade of membership of \( x \) in \( \tilde{A} \). Formally, \( \tilde{A} \) can be written as:

\[ \tilde{A} = \{(x, \mu_A(x)) | x \in X\} \]  

**Definition:** The intersection of two fuzzy subsets denoted \( \tilde{A} \cap \tilde{B} \) and defined by the min-operator is shown as (Zadeh, 1965):

\[ \tilde{A} \cap \tilde{B} = \{(x, \min(\mu_A(x), \mu_B(x))) | x \in X\} \]  

**Definition:** The union of two fuzzy subsets denoted \( \tilde{A} \cup \tilde{B} \) and defined by the max-operator is shown as (Zadeh, 1965):

\[ \tilde{A} \cup \tilde{B} = \{(x, \max(\min(\mu_A(x), \mu_B(x)))) | x \in X\} \]

**Definition:** Let \( \mu_s(x) \) be the membership function of fuzzy sets which are to be aggregated in the logic of the fuzzy min-operator (12) and fuzzy augmented max-min-operator (13) (Lai and Hwang, 1993, 1996). Then, the membership functions of the resulting fuzzy sets are defined as in (12) and (13).

\[ \mu_{\min} = \min_s \mu_s(x) \]  

\[ \mu_{\text{augmented max-min}} = \min_s \mu_s(x) + \delta \sum_s w_s \mu_s(x) \]  

where \( \delta \) is a adequately small positive number, and \( w_s \) is the relative importance of the objective \( s \) and \( \sum w_s = 1 \). The augmented max-min operator is an extension of Zimmermann’s ‘min’ operator. Augmented max-min operator is defined below in (14) where the objectives have equal importance.

\[ \mu_{\text{augmented max-min with equal importance}} = \min_s (\mu_s(x)) + \left( \sum_s \mu_s(x) \right) / S. \]

### 3.3 Fuzzy mathematical programming

To construct the FMP model, consider the crisp and fuzzy formulation mentioned here:
(crisp formulation)
\[
\min f_s(x) = c_s^T x, \quad \forall s, 
\]
s.t.
\[
x \in X 
\]
(fuzzy formulation)
find \( x \) such that
\[
c_s^T x \leq z_s, \quad \forall s, 
\]
\( x \in X \)

In model (16), each of the \( s \) rows are represented by a fuzzy set and the linear membership functions of which are \( \mu_s(x) \) where \( c_s \) denotes the \( s^{th} \) row of \( c \) and \( \mu_s(x) \) is the linearly decreasing function and shows the degree to which \( x \) satisfies the fuzzy inequality \( c_s x \leq z_s \). The \( lb_s \) and \( ub_s \) are the lower and upper bounds of the \( s^{th} \) objective function’s \( z_s \) aspiration level. Then, it can be shown as follows (17)

\[
\mu_s(x) = \begin{cases} 
1 & c_s x \leq lb_s \\
\frac{(ub_s - c_s x)}{(ub_s - lb_s)} & lb_s < c_s x \leq ub_s \\
0 & c_s x > ub_s 
\end{cases} 
\]

The membership function of the fuzzy decision set of the model (16) is defined by fuzzy min-operator:

\[
\mu_0(x) = \min_s \{ \mu_s(x) \} 
\]

If the decision maker (DM) is interested in an optimal solution, then Zimmermann (1987) suggested maximising solution to (18)

\[
\max_{x \in X} \mu_0(x) = \max_{x \in X} \left( \min_s \{ \mu_s(x) \} \right) 
\]

From the expression (19) it is understood that the larger values of \( \mu_s(x) = \frac{(ub_s - c_s x)}{(ub_s - lb_s)} \) provide the higher the constraints’ satisfaction. Introducing a new variable \( \lambda \) is such that \( \lambda = \min_s \{ \mu_s(x) \} \). Zimmermann’s max-min model is shown as follows:

\[
\max \lambda 
\]
s.t.
\[
\lambda \leq \mu_s(x), \quad \forall s, \\
x \in X \text{ and } \lambda \in [0, 1] 
\]

If the fuzzy decision set is defined by Lai and Hwang’s (1993, 1996) max-min operator then the equivalent mathematical model (21) is shown as follows, let \( \lambda_s = \min_s \{ \mu_s(x) \} \)

\[ \text{(20)} \]
Multi-objective fuzzy job shop scheduling

\[
\begin{align*}
\max & \quad \lambda + \delta \sum_{s} w_{s} \mu_{s}(x) \\
\text{s.t.} & \quad \lambda \geq \mu_{s}(x), \quad \forall s, \\
& \quad x \in X \\
& \quad \lambda \in [0, 1]
\end{align*}
\] (21)

4 Mathematical models for the JSSP

The problem considered is a job shop problem. In a job shop, the sequence of jobs on each machine may be different, so if a job is at a certain position on one machine then it may be at a different position for all the other machines considered in the system. Jobs are processed on each machine only once. The variables and parameters used are defined below:

Indices

\begin{itemize}
  \item \(i\) \quad \text{machine index}
  \item \(j\) \quad \text{job index.}
\end{itemize}

Parameters

\begin{itemize}
  \item \(n\) \quad \text{total number of jobs}
  \item \(d_{j}\) \quad \text{due date of job} \(j\)
  \item \(r\) \quad \text{upper limit set for number of tardy jobs}
  \item \(q\) \quad \text{upper limit set for total tardiness}
  \item \(s\) \quad \text{upper limit set for total earliness}
\end{itemize}

Decision variables

\begin{itemize}
  \item \(y_{ij}\) \quad \text{starting time of job} \(j\) \text{ on machine} \(i\)
  \item \(c_{ij}\) \quad \text{completion time of job} \(j\) \text{ on machine} \(i\)
  \item \(r_{j}\) \quad \text{ready time of job} \(j\)
  \item \(n_{j}\) \quad 1 \text{ if job} \(j\)\text{ is tardy, } 0 \text{ otherwise}
  \item \(t_{j}\) \quad \text{tardiness of job} \(j\)
  \item \(e_{j}\) \quad \text{earliness of job} \(j\)
  \item \(n_{T}\) \quad \text{total number of tardy jobs}
  \item \(TT\) \quad \text{total tardiness}
  \item \(T_{\text{max}}\) \quad \text{maximum tardiness}
  \item \(k\) \quad \text{total satisfaction}
\end{itemize}
41 satisfaction for total tardiness
k2 satisfaction for number of tardy jobs
k3 satisfaction for maximum tardiness
eej earliness quotient (= 1 if early, = 0 else).

4.1 Mathematical model for minimising number of tardy jobs

This section presents the mathematical model used to minimise the number of tardy jobs in a JSSP.

Objective function:

\[
\text{Minimise } n_T
\]  \hspace{2cm} (22)

Subject to:

\[
y_{ij} - y_{ij} \geq p_{ij} \quad \forall (i, j) \rightarrow (k, j) \in A
\]  \hspace{2cm} (23)

\[
c_{ij} - y_{ij} = p_{ij} \quad \forall i \text{ in last operation of job } j
\]  \hspace{2cm} (24)

\[
y_{ij} - y_{ij} \geq p_d \text{ or } y_{ij} - y_{ij} \geq p_d \quad \forall (i, l) \text{ and } (i, j), i = 1, \ldots, m.
\]  \hspace{2cm} (25)

\[
c_{ij} - d_j = t_j - e_j \quad \forall i \text{ in last operation of job } j
\]  \hspace{2cm} (26)

\[
\mu * n_j \geq t_j \quad \forall j \in N
\]  \hspace{2cm} (27)

\[
n_j \leq t_j \quad \forall j \in N
\]  \hspace{2cm} (28)

\[
eej * \mu \geq e_j \quad \forall j \in N
\]  \hspace{2cm} (29)

\[
eej \leq e_j \quad \forall j
\]  \hspace{2cm} (30)

\[
eej + n_j \leq 1 \quad \forall j \in N
\]  \hspace{2cm} (31)

\[
y_{ij} \geq r_j \quad \forall (i, j) \in N
\]  \hspace{2cm} (32)

\[
\sum_{j=1}^{n} n_j = n_T
\]  \hspace{2cm} (33)

The objective function [equation (22)] is to minimise the number of tardy jobs in the system. The first set of constraints [equation (23)] signifies the sequence of machines for a job. It says that operation \((k, j)\) starts only after the completion of operation \((i, j)\). The second set of constraints [equation (24)] calculates the completion time of a job with the help of the sum of the starting time and processing time for the last operation of a job. The third set of constraints [equation (25)] is called disjunctive constraints. It signifies that there exists some order for the operations of the jobs on a particular machine. The fourth constraints [equation (26)] signifies that \(t\) is positive if completion time is greater than due date and negative if completion time is less than due date. This in turn determines whether the job is tardy or non-tardy in the fifth and sixth constraints [equations (27) and (28)]. The seventh and the eighth constraints [equations (29) and (30)] in conjunction with the fourth constraint determine whether the job is early or not.
The ninth constraint [equation (31)] ensures that either the job will be tardy or early. The tenth constraint [equation (32)] ensures that each job is scheduled after their respective ready times. The eleventh constraint [equation (33)] computes the total number of tardy jobs in the schedule.

4.2 Mathematical model for minimising total tardiness

This section presents a mathematical model that can be used to minimise the total tardiness of jobs in a JSSP.

Objective function:

\[
\text{Minimise } TT
\]  

Subject to:

The first ten constraints are the same as the first ten set of constraints [equations (23)–(32)] in Model 4.1.

\[
\sum_{j=1}^{n} t_j = TT
\]  

The objective function [equation (34)] is to minimise the total tardiness \( TT \). The first ten constraints are the same as in Section 4.1. The 11th constraint [equation (35)] computes the total tardiness of the schedule.

4.3 Mathematical model for minimising maximum tardiness

This section presents a mathematical model that can be used to minimise the maximum tardiness of jobs in JSSP.

Objective function:

\[
\text{Minimise } T_{\text{max}}
\]  

Subject to:

The first ten constraints are the same as the first ten set of constraints [equations (23)–(32)] in Model 4.1.

\[
T_{\text{max}} \geq t_j \quad \forall j
\]  

The objective function is to minimise the maximum tardiness of a schedule \( T_{\text{max}} \). The eleventh constraint (equation 37) generates the total earliness in the schedule.

4.4 Fuzzy mathematical model-1 (max-min operator)

This model was originally proposed by Zimmermann (1978). In this paper, his approach is used for minimising total earliness, total tardiness and number of tardy jobs.

This section presents a fuzzy mathematical model [equation (20)] that builds on previously mentioned single objective mathematical models. This model is used to minimise the number of tardy jobs, total tardiness and total earliness.
Objective function:

Maximise \( k \) \hspace{1cm} (38)

Subject to:

\[ k \leq k_1, k \leq k_2, k \leq k_3 \] \hspace{1cm} (39)

\[ k_1 = \frac{q - T}{q} \] \hspace{1cm} (40)

\[ k_2 = \frac{r - W}{r} \] \hspace{1cm} (41)

\[ k_3 = \frac{s - T_m}{s} \] \hspace{1cm} (42)

\[ \sum_{j=1}^{n} n_j = n_f \] \hspace{1cm} (43)

\[ T_m = \max (t_1, t_2, \ldots, t_p) \] \hspace{1cm} (44)

\[ \sum_{j=1}^{n} t_j = TT. \] \hspace{1cm} (45)

The objective function [equation (38)] is to maximise \( k \) (satisfaction level). The first set of constraints [equation (39)] sets the optimum value for \( k \) based on the values of \( k_1, k_2 \) and \( k_3 \). The second constraint [equation (40)] computes \( k_1 \). The third constraint [equation (41)] determines \( k_2 \) and the fourth constraint [equation (42)] calculates \( k_3 \). The fifth constraint [equation (43)] defines total number of tardy jobs in the system as the summation of all the tardy jobs. The sixth constraint [equation (44)] defines total earliness as the summation of the earliness of all jobs. The seventh constraint [equation (45)] defines the total tardiness as the summation of tardiness of all jobs. The constraints given in equations (23)–(32) are also included in this model.

4.5 Fuzzy mathematical model-2 (augmented max-min model)

This model has been originally proposed by Lai and Hwang (1996). The objective function is given in equation (46). All of the constraints remain the same as in the model presented in Section 4.4.

Maximise \( k + (k_1 + k_2 + k_3) / 3 \) \hspace{1cm} (46)

4.6 Numerical illustration (interpretation of a fuzzy solution)

This section shows an example showing the computation of fuzzy function and its interpretation using graphical representation. The linear shape is used for each fuzzy membership function. All of the jobs are ready at time zero. The example problem chosen is a four-job three-machine problem. The order of jobs is given in Table 1(a). Sequence
of jobs for each machine is given in Table 1(b). Table 2 summarises the results obtained for the given sequence shown in Figure 1. The values of performance measures and the fuzzy objective functions for the given sequence are given in Table 3. Figure 2 shows the computation of satisfaction levels for each performance measure for the example problem.

Table 1(a)  Processing order for each job for example problem 1

<table>
<thead>
<tr>
<th>Job</th>
<th>Processing Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Machine 1, 2, 3</td>
</tr>
<tr>
<td>2</td>
<td>Machine 1, 3, 2</td>
</tr>
<tr>
<td>3</td>
<td>Machine 1, 2, 3</td>
</tr>
<tr>
<td>4</td>
<td>Machine 1, 3, 2</td>
</tr>
</tbody>
</table>

Table 1(b)  Data for the example problem 1

<table>
<thead>
<tr>
<th>Job</th>
<th>Machine 1</th>
<th>Machine 2</th>
<th>Machine 3</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>33</td>
</tr>
</tbody>
</table>

Table 2  Results for each job for example problem 1

<table>
<thead>
<tr>
<th>Due date</th>
<th>Completion time</th>
<th>Tardiness</th>
<th>Maximum tardiness</th>
<th>Number of tardy jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1</td>
<td>35</td>
<td>12</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Job 2</td>
<td>20</td>
<td>25</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Job 3</td>
<td>30</td>
<td>33</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Job 4</td>
<td>33</td>
<td>34</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1  Gantt chart for the example problem 1 (see online version for colours)

Table 3  Performance measures for sequence given in Figure 1

<table>
<thead>
<tr>
<th>Measure</th>
<th>Value</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>Satisfaction level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0.5</td>
</tr>
<tr>
<td>$TT$</td>
<td>7</td>
<td>0</td>
<td>20</td>
<td>0.6</td>
</tr>
<tr>
<td>$T_{max}$</td>
<td>9</td>
<td>0</td>
<td>15</td>
<td>0.4</td>
</tr>
</tbody>
</table>
5 Experimentation

In this section, the experimentation performed is discussed in detail.

5.1 Experiment 1

The purpose of this experiment is to test if fuzzy mathematical models can find the known unique solution for the given problem. The problem chosen is fabricated such that it has a unique solution in a multi-objective environment. The problem that will be used for experimentation with the models developed in this paper is six-job three-machine JSSP. The order of processing through the three machines, for each job, is known [shown in Table 4(a)] and the processing times of the six jobs on the machines are known [shown in Table 4(b)].

Table 4(a)  Processing order for each job in example 2

<table>
<thead>
<tr>
<th>Machines</th>
<th>J1</th>
<th>J2</th>
<th>J3</th>
<th>J4</th>
<th>J5</th>
<th>J6</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>2</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>m2</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>m3</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>d_j</td>
<td>33</td>
<td>24</td>
<td>35</td>
<td>18</td>
<td>31</td>
<td>24</td>
</tr>
</tbody>
</table>
The results obtained from different models are shown in Table 5. The first three models are single objective models and the last three are fuzzy multi-objective models. The results are shown only for the respective performance measure. The problem has a unique optimal solution for each performance measure; it is zero for all the three performance measures for this problem. All fuzzy models could find the unique optimal solution to the problem where $n_T$, $TT$ and $T_{\text{max}}$ take a value of zero.

### Table 4(b) Six-job three-machine job shop problem 2

<table>
<thead>
<tr>
<th>Job 1</th>
<th>Machine 2</th>
<th>Machine 3</th>
<th>Machine 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 2</td>
<td>Machine 1</td>
<td>Machine 2</td>
<td>Machine 3</td>
</tr>
<tr>
<td>Job 3</td>
<td>Machine 1</td>
<td>Machine 2</td>
<td>Machine 3</td>
</tr>
<tr>
<td>Job 4</td>
<td>Machine 3</td>
<td>Machine 2</td>
<td>Machine 1</td>
</tr>
<tr>
<td>Job 5</td>
<td>Machine 3</td>
<td>Machine 2</td>
<td>Machine 1</td>
</tr>
<tr>
<td>Job 6</td>
<td>Machine 1</td>
<td>Machine 3</td>
<td>Machine 2</td>
</tr>
</tbody>
</table>

The results obtained from different models are shown in Table 5. The first three models are single objective models and the last three are fuzzy multi-objective models. The results are shown only for the respective performance measure. The problem has a unique optimal solution for each performance measure; it is zero for all the three performance measures for this problem. All fuzzy models could find the unique optimal solution to the problem where $n_T$, $TT$ and $T_{\text{max}}$ take a value of zero.

### Table 5 Results for different models for example 2

<table>
<thead>
<tr>
<th>Models</th>
<th>Performance measures</th>
<th>Satisfaction levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n_T$</td>
<td>$TT$</td>
</tr>
<tr>
<td>Min $n_T$</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Min $TT$</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Min $T_{\text{max}}$</td>
<td>- 0</td>
<td>-</td>
</tr>
<tr>
<td>Max $k$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max $(k1 + k2 + k3) / 3$</td>
<td>0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

### 5.2 Experiment 2

In this experiment, first non-dominated solutions for a JSSP are determined. Then, fuzzy mathematical models are used to test which model(s) will produce non-dominated solutions.

#### 5.2.1 Finding non-dominated solutions

The non-dominated solutions for another job shop problem [three machines, six jobs shown in Table 6(a) and 6(b)] were found using the below mentioned methodology. It is divided into five steps which will be illustrated by applying corresponding steps to the example problem mentioned below.

1. First the minimum values for individual performance measure were found using the mathematical models for optimising individual performance measures. Table 7 shows the results for this problem. The minimum value of $n_T$ and $TT$ were used to find non-dominated solutions.

2. The value for the number of tardy jobs was set to (2, 3 and 4) and then corresponding minimum values for total tardiness were found. Similarly, values for maximum
tardiness were also found setting the values for number of tardy jobs as 2, 3 and 4, respectively. The results are shown in Table 8. For example, third row of Table 8 shows that we minimise $TT$ subject to $n_T = 2$ (column 2) and we minimise $T_{\text{max}}$ subject to $n_T = 2$ (column 3).

3 The minimum values for $T_{\text{max}}$ were obtained by setting the values of $n_T$ and $TT$ to the values obtained from previous the step as shown in Table 9. The minimum values for $TT$ were also determined in a similar way as shown in Table 10. For example, third row of Table 9 shows that we minimise $T_{\text{max}}$ subject to $n_T = 2$ and $TT = 42$ (column 3).

4 Then the minimum values for $TT$ and $T_{\text{max}}$ were found setting the values of other two variables similar to the last step. This step was carried out for all the possible combinations with the extreme bounds obtained for the performance measures in the last step. The results for data point ($n_T = 2$, $TT = 42$), results for range ($n_T = 3$, $TT = 26$ to $n_T = 3$, $TT = 42$) and ($n_T = 4$, $TT = 22$ to $n_T = 4$, $TT = 42$) are given in Table 11.

5 Then the non-dominated solutions are identified as given in Table 12. This step was performed for three values of $n_T$ (2, 3 and 4), and all corresponding $TT$ and $T_{\text{max}}$ values.

**Table 6(a)** Six-job three-machine job shop problem 3

<table>
<thead>
<tr>
<th>Machines</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$J1$</td>
</tr>
<tr>
<td>$m1$</td>
<td>2</td>
</tr>
<tr>
<td>$m2$</td>
<td>5</td>
</tr>
<tr>
<td>$m3$</td>
<td>6</td>
</tr>
<tr>
<td>$d_j$</td>
<td>11</td>
</tr>
</tbody>
</table>

**Table 6(b)** Processing order for each job for example problem 3

<table>
<thead>
<tr>
<th>Job 1</th>
<th>Machine 2</th>
<th>Machine 3</th>
<th>Machine 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 2</td>
<td>Machine 1</td>
<td>Machine 2</td>
<td>Machine 3</td>
</tr>
<tr>
<td>Job 3</td>
<td>Machine 1</td>
<td>Machine 2</td>
<td>Machine 3</td>
</tr>
<tr>
<td>Job 4</td>
<td>Machine 3</td>
<td>Machine 2</td>
<td>Machine 1</td>
</tr>
<tr>
<td>Job 5</td>
<td>Machine 3</td>
<td>Machine 2</td>
<td>Machine 1</td>
</tr>
<tr>
<td>Job 6</td>
<td>Machine 1</td>
<td>Machine 3</td>
<td>Machine 2</td>
</tr>
</tbody>
</table>

**Table 7** Minimum values for each performance measure

<table>
<thead>
<tr>
<th>Min</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>2</td>
</tr>
<tr>
<td>$TT$</td>
<td>22</td>
</tr>
<tr>
<td>$T_{\text{max}}$</td>
<td>8</td>
</tr>
</tbody>
</table>
Table 8  Values for $T$ and $T_m$ setting $N_t$ (= 2, 3 and 4)

<table>
<thead>
<tr>
<th>Set</th>
<th>Find</th>
<th>Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>$TT$</td>
<td>$T_{max}$</td>
</tr>
<tr>
<td>$n_T = 2$</td>
<td>42</td>
<td>22</td>
</tr>
<tr>
<td>$n_T = 3$</td>
<td>26</td>
<td>11</td>
</tr>
<tr>
<td>$n_T = 4$</td>
<td>22</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 9  Values for $T_{max}$ (setting $n_T$ and $TT$)

<table>
<thead>
<tr>
<th>Set</th>
<th>Find</th>
<th>Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>$TT$</td>
<td>$T_{max}$</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 10  Values for $TT$ (setting $n_T$ and $T_{max}$)

<table>
<thead>
<tr>
<th>Set</th>
<th>Find</th>
<th>Find</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>$T_{max}$</td>
<td>$TT$</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 11  Values for $T_{max}$ [setting $n_T$ (= 2, 3, 4) and $TT$]

<table>
<thead>
<tr>
<th>$n_T$</th>
<th>$TT$</th>
<th>$T_{max}$</th>
<th>$n_T$</th>
<th>$TT$</th>
<th>$T_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>42</td>
<td>22</td>
<td>4</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>11</td>
<td>4</td>
<td>25</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>11</td>
<td>4</td>
<td>26</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>12</td>
<td>4</td>
<td>27</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>29</td>
<td>12</td>
<td>4</td>
<td>28</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>12</td>
<td>4</td>
<td>29</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>32</td>
<td>12</td>
<td>4</td>
<td>30</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>12</td>
<td>4</td>
<td>32</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>34</td>
<td>12</td>
<td>4</td>
<td>33</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>12</td>
<td>4</td>
<td>34</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>37</td>
<td>13</td>
<td>4</td>
<td>36</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>38</td>
<td>13</td>
<td>4</td>
<td>37</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>14</td>
<td>4</td>
<td>38</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>14</td>
<td>4</td>
<td>39</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>15</td>
<td>4</td>
<td>40</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
<td>15</td>
<td>4</td>
<td>41</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>10</td>
<td>4</td>
<td>42</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Following are the results of all of the non-dominated solutions obtained for the example problem. Also, Figure 3 shows the surface plot of all the non-dominated solutions along with the interaction of the three performance measures.

### Figure 3  Surface plot of non-dominated solutions (see online version for colours)

<table>
<thead>
<tr>
<th>$n_T$</th>
<th>$TT$</th>
<th>$T_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>42</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>8</td>
</tr>
</tbody>
</table>

#### 5.2.2 Performance of fuzzy mathematical models with respect to non-dominated solutions

The fuzzy mathematical models were tried to replicate the results (non-dominated solutions) by altering the bounds for each performance measure. Below are the results obtained with all the fuzzy models along with corresponding bounds for the variables. The fuzzy model \textit{(augmented max-min)} provided only the non-dominated solutions and was successfully able to replicate all the non-dominated solutions by varying bounds as shown in Table 13. The non-dominated solution is dependent on the bounds set on the performance measures. The set of non-dominated solutions provided in Step 5 were taken as the references for bounds and some random bounds were also chosen. Numbers in the proximity of ideal solutions, depending on the problem, were also used as bounds. The bounds can be varied, from the minimum required to get a feasible non-dominated solution to any larger number, in accordance with the output required. Tighter bounds resulted in non-dominated solutions closer to the bounds for the corresponding performance measure.
Table 13  Non-dominated solutions replicated by fuzzy mathematical models

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>MAX ( k + (k_1 + k_2 + k_3) / 3 )</th>
<th>MAX ( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear bounds</td>
<td>-</td>
<td>( n_T )</td>
<td>( TT )</td>
</tr>
<tr>
<td>( n_T )</td>
<td>( TT )</td>
<td>( T_{\text{max}} )</td>
<td>( n_T )</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>20</td>
<td>-</td>
</tr>
</tbody>
</table>

6  Comparing the two performance measures v/s three performance measures model

This section compares two types of models, model with satisfaction levels of two performance measures \((n_T \text{ and } TT)\) in the objective function versus the model with satisfaction levels of three performance measures \((n_T, TT \text{ and } T_{\text{max}})\) in the objective function. This experimentation in this section is divided into three steps.

1. The bounds for \(n_T\) and \(TT\) were set to find out the corresponding non-dominated solutions for only these two performance measures. The values of all the three performance measures were recorded, i.e., the value for \(T_{\text{max}}\) was also recorded from the corresponding solution. Also the values for satisfaction level of \(n_T\) and \(TT\), \(k_1\) and \(k_2\), were recorded. The model used was same as the model in Section 4.4 except that the objective function used was to minimise \(k_1 + k_2\) instead of \(k\). The readings are shown in Table 14.

2. In this step, the bounds for \(n_T\) and \(TT\) were taken from Step 2 and bounds for \(T_{\text{max}}\) were chosen arbitrarily to replicate non-dominated solutions. The values of all the three performance measures were recorded. Also the values for satisfaction level of \(n_T\) and \(T, k_1\) and \(k_2\), were recorded. The model used was same as the model in Section 4.4 except that the objective function used was to minimise \(k_1 + k_2 + k_3\) instead of \(k\). The readings are shown in Table 15.

3. In this step, the bounds from Step 2 was used to calculate the corresponding satisfaction level, \(k_3\), for \(T_{\text{max}}\) for the values of the \(T_{\text{max}}\) obtained in Step 1. Then the summation of all three satisfaction levels, \(k_1 + k_2 + k_3\), is calculated. The data calculated is shown in Table 16.

4. In this step, we analyse the results obtained from Step 2 and Step 3. The results in Table 15 and Table 16 are arranged in descending order of the values of \(k_1 + k_2 + k_3\). After careful examination the last columns of Table 15 and Table 16 closely, it can be concluded that on an average the satisfaction value in the last column of Table 15 is greater than that obtained in Table 16. This means that on an average satisfaction level obtained using the model which optimises three performance measures is greater than satisfaction level obtained using the model which optimises two performance measures.
Table 14  Calculating satisfaction for two performance measures (using max $k1 + k2$)

<table>
<thead>
<tr>
<th>Bounds</th>
<th>Results</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>$TT$</td>
<td>$n_T$</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 15  Calculating satisfaction for three performance measures (using bounds from Table 14)

<table>
<thead>
<tr>
<th>Bounds</th>
<th>Results</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>$TT$</td>
<td>$T_{max}$</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 16  Calculating satisfaction for $T_{max}$ and total satisfaction for two performance measure case

<table>
<thead>
<tr>
<th>Using bounds for $T_{max}$ from Table 15</th>
<th>Results</th>
<th>Calculating $k3$ and $k1 + k2 + k3$ using bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_T$</td>
<td>$TT$</td>
<td>$T_{max}$</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>

7 Conclusions and future work

This paper illustrated novice mathematical models for minimising number of tardy jobs, total tardiness and maximum tardiness. It also provided three fuzzy multi-objective mathematical model formulations for job shop scheduling. Generally, multi-objective environment gives the users a choice between solutions according to their needs. However, the methods/models seldom incorporate the user’s bounds as inputs; therefore the fuzzy modelling is a more practical approach in modelling of scheduling systems. In experiment 1, it was shown that all three fuzzy mathematical models could find the unique optimal solution with respect to all three performance measures. In experiment 2, a procedure that we followed to find non-dominated solutions was described first. Then, an experiment was carried out to evaluate the performance of two models. One of the models identified non-dominated solutions for each and every interval tested. These intervals represent user’s needs. On the other hand, the third model was not that successful in finding all the non-dominated solutions. As result, we recommend to use...
**Multi-objective fuzzy job shop scheduling**

An augmented max-min model to solve problems with multiple objectives. A comparison of model optimising two versus three performance measures provided conclusive results which showed that on average the latter is better in providing better overall satisfaction value. The user should judge whether additional modelling complexity and computational complexity is worth the improvement in the overall satisfaction level.

References


