

A Real Time Practical Approach for Multi Objective Job Shop Scheduling Using Fuzzy Logic Approach

¹Ramkumar, R., ²A. Tamilarasi and ³T. Devi

¹Department of Computer Applications,
Maharaja Engineering College, Avinashi, Coimbatore, Tamil Nadu

²Department of Computer Applications,
Kongu Engineering College, Perundurai, Erode, Tamil Nadu

³Department of Computer Science and Engineering,
Bharathiyar University, Coimbatore, Tamil Nadu

Abstract: Multi objective Job Shop scheduling is a difficult task in both theoretical and practical solving issues. **Problem statement:** In the present scenario the modern Engineering and Industrial manufacturing units are facing lot of problems in many aspects such as machining time, raw material movement, man power requirement, electricity demand and customer satisfaction. **Approach:** A triangular fuzzy membership function is used to represent customer priority and due date. **Results:** A fuzzy rule-based system is developed which determines the study to be allocated to N number of machines with M number of Jobs in the following premise variables: size of the job, workload on the shop floor and the priority of the job. Multi objective fuzzy job shop scheduling problems are formulated as three-objective ones which not only maximize the minimum agreement index but also maximize the average agreement index and minimize the maximum fuzzy completion time. **Conclusion/Recommendations:** The study is analyzed on real-world data obtained from a printing company and the results are found satisfactory.

Key words: Job shop scheduling, fuzzy logic, multiple objectives, rule base, multi criteria scheduling

INTRODUCTION

Fuzzy set theory has been utilized to develop hybrid scheduling approaches and it can be useful in modeling and solving job shop scheduling problems with uncertain processing times, constraints and set-up times. These uncertainties can be represented by fuzzy numbers that are described by using the concept of an interval of confidence (Wan and Yen, 1999). These approaches usually are integrated with other methodologies (e.g., search procedures, constraints and relaxation).

Job Shop problems are known to be NP hard and probably as a consequence of this scheduling has been an active area of research for many years (ElMaraghy *et al.*, 2000). However, real-world scheduling problems are usually very different from the mathematical models studied by researchers in academia. Actual firms place a higher priority on meeting due-dates than on typical research objectives such as minimizing flow-time.

Scheduling: Scheduling is the process to allocation of limited resources to tasks over time. It is a decision-

making process that has as a goal the optimization of one or more objectives, for each operation of the process; this means the determination of one or more intervals.

We consider generally the resources of two types: renewable or consumable. Renewable resources become available again after use (Reuse of machine, file, processor, personal), whereas non renewable resources disappear after use (money, raw materials.).

The basic unit of the job-shop process is the operation, there are three primary attributes of each operation are given as part of the description of a particular job-shop problem (Saghafian and Hejazi, 2005).

- Identifying the operation with a particular job
- Identifying the operation with a particular machine
- Representing the processing time of the operation

A job-shop is the set of all the machines that are identified with a particular set of operations. The process consists of the machine, the jobs (operations) and a statement of the disciplines that restrict the

Corresponding Author: Ramkumar, R., Department of Computer Applications, Maharaja Engineering College, Avinashi, Coimbatore, Tamil Nadu

manner in which operations can be assigned to specific points on the time scale of the appropriate machine (Chaabene *et al.*, 2007).

A four parameter notation will be used to identify the scheduling problems written as:

$$n/m/G/F_{\max}$$

- n = Denotes the job arrival process
- m = Denotes the number of machines
- G = Denotes sequence of the job assign to the machines
- F_{max} = Denotes process to be evaluated

There are several types of constraints related to the due dates.

- r_i = The ready-time, release time or arrival time. This is the time at which the job is released to the job by some external resources.
- d_i = The due date. This is the time at which the processing of the last operation to be completed.
- p_i = The processing time. It is assumed that at the time of scheduling the exact Processing times are unknown
- P_i = $\sum_j^{a_i} P_{i,j}$ be the total processing Time
- a_i = d_i - r_i

Multi objective job shop scheduling: Single Objective scheduling is related to maximizing a single function subject to a number of constraints. However, it has been increasingly recognized that many real-world decision-making problems involve multiple, no commensurable and conflicting objectives which should be considered simultaneously (Carlsson and Fuller, 2001). As an extension, multi objective scheduling is defined as a means of optimizing multiple objective functions subject to a number of constraints (Azadeh *et al.*, 2008), i.e.:

$$\begin{cases} \text{Max}[f_1(x), f_2(x), \dots, f_n(x)] \text{ subject to} \\ G_i(x) \leq 0, i = 0, 1, 2, \dots, p \end{cases}$$

where, f_i(x) are objective functions, i=1,2,...m and g_i(x) ≤ 0 are system constraints, i = 1,2,...p. When the objectives are in conflict, there is no optimal solution that simultaneously maximizes all the objective functions. For this case, we employ a concept of Pareto solution, which means that it is impossible to improve any one objective without sacrificing on one or more of the other objectives.

Introduction to fuzzy logic: Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than more precise form (because of membership function values taken between 0 and 1). In contrast with “crisp logic”, where binary sets have binary logic, the fuzzy logic variables may have a membership value of not only 0 or 1 that is the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values of classic propositional logic. Fuzzy logic is being incorporated worldwide in applications to accomplish these goals, primarily in the control mechanisms designed to make them work (Kangrang and Chaleeraktragoon, 2007). Appliances with fuzzy logic controllers provide the consumer with optimum settings that more closely approximate human perceptions and reactions than those associated with standard control systems. Products with fuzzy logic monitor user defined settings and then automatically set the equipment to function at the user’s preferred level for a given task.

Existing approach: Modern Fuzzy logic control for job-shop scheduling allows the modeling of scheduling knowledge with linguistic variables defined by membership functions showing the degree of strictness of the data and the reasoning about the imprecise data by using fuzzy rules. This means of representation and processing of job that is transparent and comprehensible for the user. While the presented fuzzy based approach is innovative within the field of multi valued scheduling, that already exist several approaches to solve general or local scheduling problems by techniques founded based on fuzzy set theory and fuzzy logic.

Proposed approach: A search begins with better job sequence and job order to satisfy the customer needs. The Customer Priority (CP) and Due Date (DD) consists of membership function. Each membership function is represented by the sequence of job orders, the sequence of operations and the set of machines to be used to accomplish to finish the entire operation. The proposed approach is based on the following Fig. 1.

The aim of this problem is to determine the optimal speed of each machine and an optimal schedule with respect to an objective function consisting of the minimum degree of satisfaction among all jobs and costs of machine time. The mechanisms of modulated algorithms are similar as illustrated in the following steps and flow of operations is shown in Fig. 1, this algorithm generates feasible active schedules.

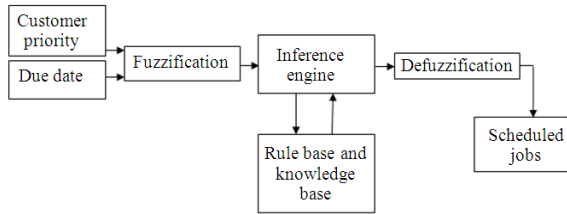


Fig. 1: Basic structure of Fuzzy JSP

Some studies have shown that the optimal schedules can be found from the set of active schedules. Therefore, there is a chance to get optimal schedules higher from the set of active schedules.

Fuzzy logic is an extension of Boolean with many-valued (values from 0-1) logic dealing with the concept of partial truth which denotes the extent to which a proposition is true. Fuzzy Inference Engine (FIE) are conceptually very simple. They consist of an input, processing and an output stage. The appropriate membership functions are used for truth values. The processing stage invokes each appropriate rule and generates a corresponding result (Xiaoying *et al.*, 2009). It then combines the results. Finally, the output stage converts the combined result back into a specific output value.

The membership function of a fuzzy set corresponds to the indicator function of the classical sets. It is a curve that defines how each point in the input space is mapped to a membership value or a degree of truth between 0 and 1. The most common shape of a membership function is triangular, although trapezoidal and bell curves are also used. The input space is sometimes referred to as the universe of discourse. The processing stage which is called inference engine is based on a collection of logic rules in the form of IF-THEN statements where the IF part is called the “antecedent” and the THEN part is called the “consequent”. Typical fuzzy inference systems have dozens of rules. These rules are stored in a knowledgebase (Al-Odienat and Al-Lawama, 2008).

Fuzzifying the inputs is the act of determining the degree of closeness to which they belong to each of the appropriate fuzzy sets via membership functions. Once the inputs have been fuzzified (linguistic variables), the degree to which each part of the antecedent has been satisfied for each rule is known. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one value that represents the result of the antecedent for that rule. The implication function then modifies that output fuzzy set to the degree specified by the antecedent. Since decisions are based

on the testing of all of the rules in an FIE, the results from each rule must be combined in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. The input for the defuzzification process is the aggregated output fuzzy set and the output is a single value.

MATERIALS AND METHODS

Fuzzy decision making: FDM consists of three main steps:

Definition of quality criteria: According to the requirements of the optimization problem, quality criteria of the model have to be chosen. Total operation costs represented by functions or integrals J_i . These quality criteria are related to controlled variables y_i .

Transformation into fuzzy membership functions:

The quality criteria $J_i(y_i)$ are translated into fuzzy membership functions $\mu_i(y_i)$. This step is done in close cooperation with the different stakeholders and resembles the inherent expert knowledge. Thereby, a membership degree equal to one means optimal fulfillment of the Objective, equal to zero means unacceptable fulfillment.

Weighted optimization and decision making:

Quality criteria are described by fuzzy goals $\mu_{G1}, \dots, \mu_{Gn}$ and fuzzy constraints $\mu_{C1}, \dots, \mu_{Cm}$. Since both fuzzy goals and fuzzy constraints are desired to have a maximum degree of membership functions.

Implementation using Fuzzy MATLAB Simulink:

The fuzzy logic toolbox allows doing several things, but the most important thing is to create and edit fuzzy inference systems. We can create these systems using graphical tools or command-line functions, or you can generate them automatically using either clustering or adaptive neuro-fuzzy techniques. The toolbox also runs own stand-alone C programs directly, without the need for Simulink. This is made possible by a stand-alone Fuzzy Inference Engine that reads the fuzzy systems saved from a MATLAB session. We customize the stand-alone engine to build fuzzy inference into our code. All provided code is ANSI compliant. Because of the integrated nature of the MATLAB environment, create our own tools to customize the Fuzzy Logic Toolbox, such as the Optimization Toolbox, shown in Fig. 2.

JSP editor: The JSP Editor handles the high-level issues for our system: Number of inputs and outputs variables to be used in our system. The fuzzy logic Toolbox doesn't limit the number of inputs.

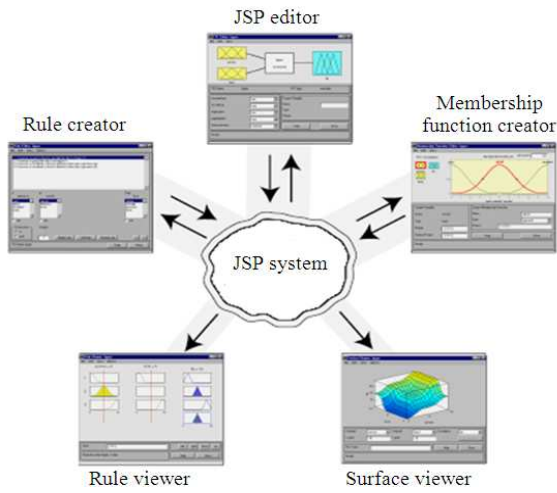


Fig. 2: JSP system using fuzzy tool

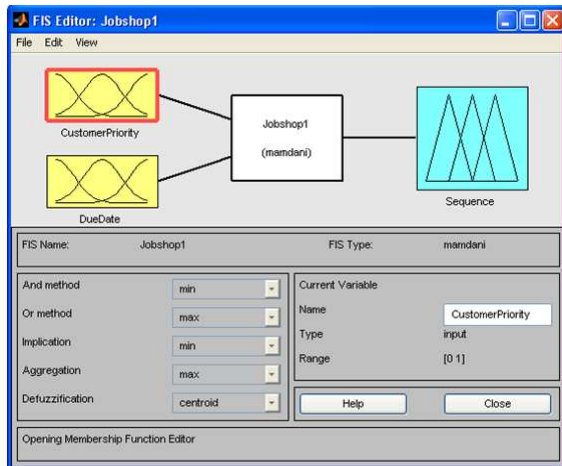


Fig. 3: Fuzzy editor for JSP problem

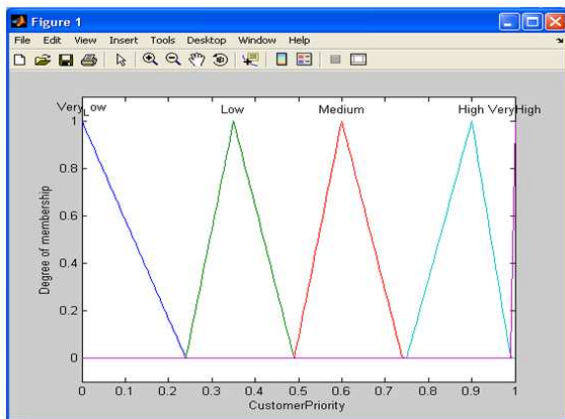


Fig. 4: Output screen of membership function for customer priority

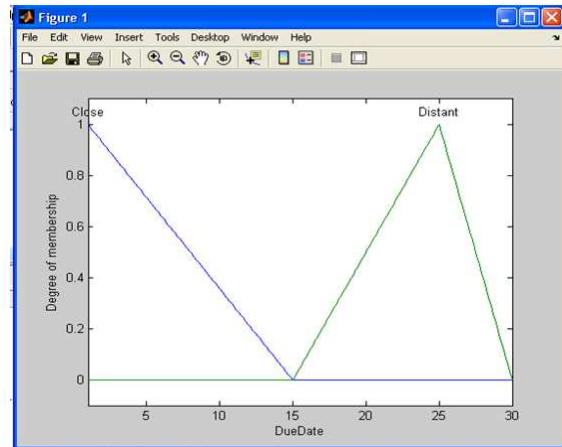


Fig. 5: Output screen of membership function for due date

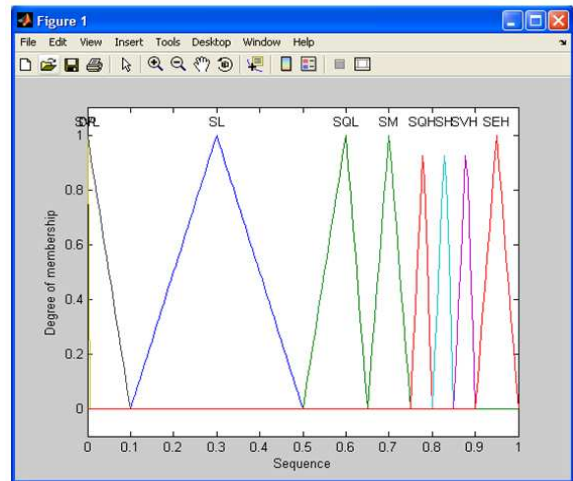


Fig. 6: Output screen of membership function for job

Table 1: Transformation for fuzzy membership functions

Rank	Attribute grade	Membership functions
CLOSE (C)	1	[-1, 1, 15, 0]
DISTANT (D)	2	[15, 25, 30, 0]

Table 2: Transformation for fuzzy membership functions

Rank	Attribute grade	Membership functions
Very Low (VL)	1	[-0.50, 0, 0.24, 0]
Low (L)	2	[0.24, 0.35, 0.49, 0]
Medium (M)	3	[0.49, 0.60, 0.74, 0]
High (H)	4	[0.75, 0.90, 0.99, 0]
Very High (VH)	5	[0.99, 1, 1, 0]

However, the number of inputs may be limited by the available memory of our machine. We will check the diagram updated to reflect the new names of the input and output variables. There is now a new variable in the workspace called JSP that contains all the information about this system. The window will look something like this shown in Fig. 3.

Table 3: Sequence Priority

Customer Priority	Due date	
Very Low (VL)	Reject	Reject
Low (L)	Sequence quite high	Sequence very low
Medium (M)	Sequence high	Sequence low
High (H)	Sequence very high	Sequence quite low
Very High (VH)	Sequence extremely high	Sequence medium

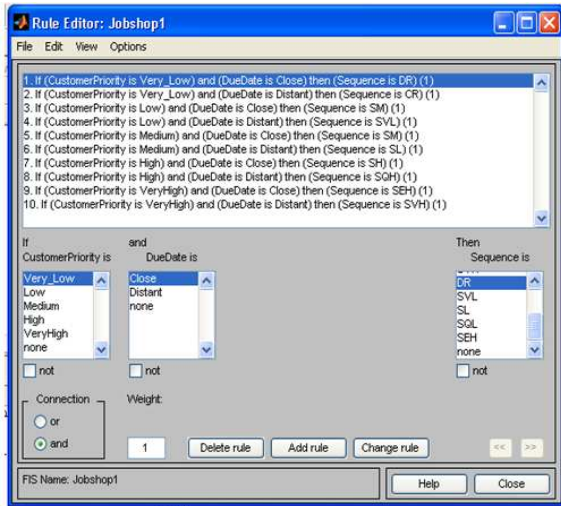


Fig. 7: Output screen of rule editor for JSP

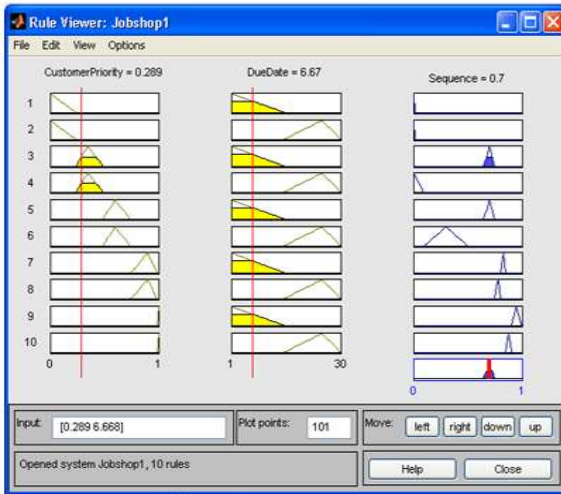


Fig. 8: Output screen of rule viewer for JSP

Membership function editor: The Membership Function Editor shares some features with the FIS Editor. The Membership Function Editor is the tool that lets, display and edits all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system. We are using the three

membership function such as Customer Priority (CP), Due Date (DD) and sequencing priority. The membership function output screen shown in Fig. 4-6 respectively. The sequence priority details as given above Table 3. The relevance of due-date to the scheduler was assumed to be in terms of ‘close’ and ‘distant’.

Hence given the universe $U = [-\infty, 30] \in Z$ and the fuzzy sets $C = \text{CLOSE}$ and $D = \text{DISTANT}$, The relevance of customer priority was deemed to be the set of ‘customer ratings’, {Very Low, Low, Medium, High, Very High}, with membership of the fuzzy set CP [14]. The membership functions values can be defined as below Table 1 and 2 respectively.

We are using triangular membership function types for the output. The fuzzy inputs of C, D and CP are combined to produce an output which is a sequence priority. First, we have to set the Range (and the Display Range) to [0 30], to cover the output range. Initially, the cheap membership function will have the parameters [0 5 10], the average membership function will be [10 15 20] and the generous membership function will be [20 25 30]. Our system should look something like this shown in Fig. 6.

Rule editor: Constructing rules using the graphical rule editor interface is fairly self-evident. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor allows to construct the rule statements automatically, by clicking on and selecting one item in each input variable, one item in each output and one connection item. Our proposed system will look like this as shown in Fig. 7.

Rule viewer: The Rule Viewer displays a roadmap of the whole fuzzy inference process. It’s based on the fuzzy inference diagram as described. We will see a single figure window with 10 small plots nested in it. The three small plots across the top of the figure represent the antecedent and consequent of the first rule. Each rule is a row of plots and each column is a variable. The first two columns of plots (the ten yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule. The third column of plots (the three blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule. If we click once on a rule number, the corresponding rule will be displayed at the bottom of the figure. The Rule Viewer allows you to interpret the entire fuzzy inference process at once. The Rule Viewer also shows how the shape of certain membership functions influences the overall result. The result screen referred Fig. 8.

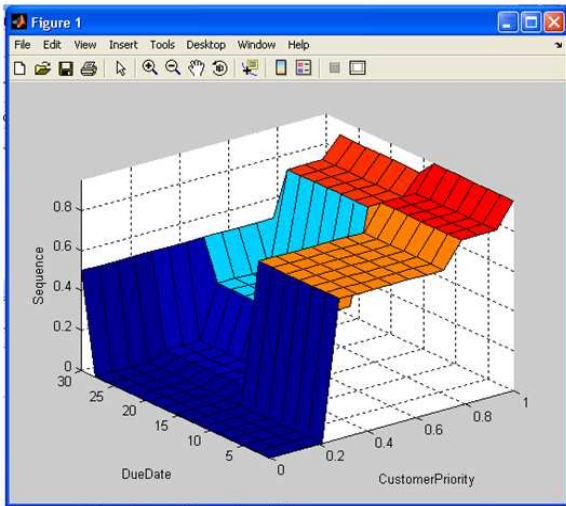


Fig. 9: Output screen of surface viewer for JSP

Surface viewer: We can see the entire output surface of our system, that is, the entire span of the output set based on the entire span of the input set and we need to open up the surface viewer. By opening the Surface Viewer, we are presented with a two-dimensional curve that represents the mapping from service quality.

In our system contain two-input one-output, as they generate three-dimensional plots that MATLAB can adeptly manage. When we move beyond three dimensions overall, we start to encounter trouble displaying the results. Accordingly, the Surface Viewer is equipped with pop-up menus that let we select any two inputs and any one output for plotting. Just below the pop-up menus are two text input fields that we determine how many x-axis and y-axis grid lines are used in our systems. The Surface Viewer has a special capability that is very helpful in our cases with two (or more) inputs and one output: we can actually grab the axes and reposition them to get a different three-dimensional view on the data. The dimensional view of our system is given below in Fig. 9.

RESULTS

Simulink result: One of the great advantages of the Fuzzy Logic Toolbox is the ability to take fuzzy systems directly into Simulink and test them out in a simulation environment. A Simulink block diagram for this system is shown in Fig. 10. It contains a simulink block called the fuzzy logic controller block. The fuzzy logic toolbox library contains the fuzzy logic controller and fuzzy logic controller with rule viewer blocks. It also includes a membership functions sub library that contains Simulink blocks for the built-in membership functions.

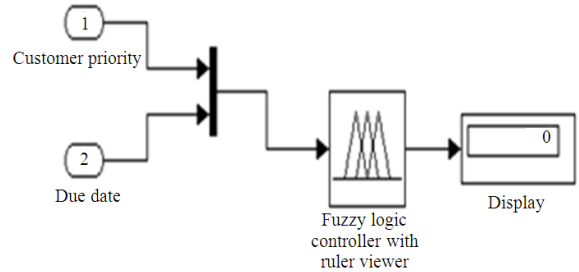


Fig. 10: Output screen of Simulink block for JSP

The fuzzy logic controller with rule Viewer block is an extension of the fuzzy logic controller block. It allows visualizing how rules are fired during simulation.

DISCUSSION

The proposed approach is used to solve the job shop scheduling problems to meet customer demands and make profit and improve the ability of autonomous systems to optimize their operations by the deployment of intelligent systems and with the optimizations of communications systems. For this reason, operations research analysts and engineers will continue this approach to pursuit well into the next century.

CONCLUSION

Fuzzy set theory allows the complexity of real life issues to be included within the confines and rigors of the mathematical model. In this study, a theoretical model has been presented which demonstrates how fuzzy decision making can support the dynamic scheduling process, enabling the conflicting priorities of multi-objectives to be managed effectively in polynomial time.

ACKNOWLEDGMENT

This project study is supported by University Grants Commission; New Delhi under the Major Research Project Grant (F.No. 33-65/2007(SR)) entitled "Job Shop Scheduling using Heuristic Methods".

REFERENCES

- Al-Odienat, A.I. and A.A. Al-Lawama, 2008. The advantages of PID fuzzy controllers over the conventional types. *Am. J. Applied Sci.*, 5: 653-658. DOI: 10.3844/ajassp.2008.653.658
- Azadeh, A., S.F. Ghaderi, Z. Javaheri and M. Saberi, 2008. A fuzzy mathematical programming approach to dea models. *Am. J. Applied Sci.*, 5: 1352-1357. DOI: 10.3844/ajassp.2008.1352.1357

- Carlsson, C. and R. Fuller, 2001. Optimization under fuzzy if-then rules. *Fuzzy Sets Syst.*, 119: 111-120. DOI: 10.1016/S0165-0114(98)00465-5
- Chaabene, M., M.B. Ammar and A. Elhajjaji, 2007. Fuzzy based energy management of a domestic photovoltaic panel. *Am. J. Applied Sci.*, 4: 60-65. DOI: 10.3844/ajassp.2007.60.65
- ElMaraghy, H., V. Patel and I.B. Abdallah, 2000. Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *J. Manufact. Syst.*, 19: 186-201. DOI: 10.1016/S0278-6125(00)80011-4
- Saghafian, S. and S.R. Hejazi, 2005. Multi-criteria group decision making using a modified fuzzy TOPSIS procedure. Proceedings of the International Conference on Computational Intelligence for modelling, Control and Automation and International Conference on Intelligent Agents. Web Technologies and Internet Commerce, Nov. 28-30, IEEE Xplore Press, Vienna, pp: 215-221. DOI: 10.1109/CIMCA.2005.1631471
- Wan, G. and P.C. Yen, 1999. A fuzzy logic system for dynamic job shop scheduling. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Oct. 12-15, IEEE Xplore Press, Tokyo, Japan, pp: 546-551. DOI: 10.1109/ICSMC.1999.812462
- Xiaoying, X., J. Lili and Z. Qiang, 2009. Optimization for multi-resources-constrained job shop scheduling based on three-level heuristic algorithm. Proceedings of the International Asia Conference on Informatics in Control, Automation and Robotics, Feb. 1-2, IEEE Xplore Press, Bangkok, pp: 296-300. DOI: 10.1109/CAR.2009.28
- Kangrang, A. and C. Chaleeraktragoon. 2007. A fuzzy-gas model for determining varied irrigation efficiency. *Am. J. Applied Sci.* 4: 339-345. DOI: 10.3844/ajassp.2007.339.345