*IIE Transactions* (1997) **29**, 533–547

# Modeling and analysis of a manufacturing cell formation problem with fuzzy mixed-integer programming

CHANG-CHUN TSAI[1], CHAO-HSIEN CHU[2] and THOMAS ARNOLD BARTA[3]

[1] *Department of Industrial Management Science, National Cheng Kung University, Tainan, Taiwan, ROC*
[2] *Department of Management, 300 Carver Hall, Iowa State University, Ames, IA 50011, USA*
[3] *Department of Industrial and Manufacturing Systems Engineering, 205 Engineering Annex, Iowa State University, Ames, IA 50011, USA*

Received December 1994 and accepted July 1996

Cell formation (CF) has received much attention from academicians and practitioners because of its strategic importance to modern manufacturing practice. In this paper a sophisticated mixed-integer programming (MIP) model is proposed to simultaneously form manufacturing cells and minimize the cost of dealing with exceptional elements. Also, we illustrate how a fuzzy mixed-integer programming (FMIP) approach can be used to solve the CF problem in a fuzzy environment, propose a new fuzzy operator, and examine the impact of different membership functions and operators on computational performance. Our study shows that FMIP not only provides a better and more flexible way of representing the problem domain, it also leads to improved overall performance.

## 1. Introduction

Applying mathematical programming models to solve real-world problems is a challenging task because decision makers find it difficult to specify goals and constraints exactly and because the parameters used in these models cannot be estimated precisely. Over the past 25 years, fuzzy set theory has been applied to many disciplines, including operations research, control theory, and artificial intelligence/expert systems, dealing with situations or problems involving ambiguities and fuzziness. Fuzzy mathematical programming (FMP) is one area where the use of fuzzy set theory has been explored widely. For instance, FMP has been applied to problems regarding transportation [1], location planning [2], project networks [3], resource allocations [4], air pollution regulations [5], and media selection for advertising [6]. Although there has been intense study of the application of fuzzy set theory to industrial engineering [7] and to operations management [8], very few studies have attempted to use FMP in the design of manufacturing systems. This paper will focus attention on applying FMIP to the design of cellular manufacturing systems, specifically in terms of cell formation (CF) problems.

Cell formation is the process of grouping parts with similar design features or processing requirements into parts families and the corresponding machines into machine cells. Over the past decade, the problem has received much attention because of its strategic importance to modern manufacturing practice. Extensive review of CF problems can be found, for example, in [9–11]. Most studies, however, have focused on the process of forming manufacturing cells. If any exceptional element (EE) existed, it was removed manually [9, 12] or dealt with after initial cells were formed [13]. No study has attempted to handle EEs automatically during manufacturing CF. Furthermore, most studies have focused on using traditional analytical or heuristic methods to model the problem; a few studies have attempted to model the problem in a fuzzy environment, but their scope has been limited. Most studies form only part families [14–16] or form part families and machine cells sequentially [17]. Other studies apply only traditional approaches to fuzzy data [16], and some studies deal only with fuzzy variables [15, 17]. No study has attempted to deal with the CF problem with fuzzy goals, fuzzy constraints, or fuzzy parameters. Fuzzy constraints are the most important component of the CF problem because most algorithms are more sensitive to the number of cells that they can form and to the number of machines or parts allowed in each cell; that is, if these parameters are selected improperly, clustering results may be unacceptable [9].

The purposes of this study are twofold: first, to develop a more relatively sophisticated mixed integer programming (MIP) model able simultaneously to form manufacturing cells and to minimize the cost of eliminating EEs; secondly, to use FMIP to model CF problems in a fuzzy environment. Because membership functions and operators tend to influence computational performance [18, 19], this study also will assess the relative performances of two different types of membership function. An improved fuzzy operator will also be proposed. Its performance will be evaluated and contrasted with that of three other commonly used operators.

## 2. Notation used in the formulations

The following notation was used to model CF problems.

### Index Set

- $i$ machine index; $i = 1, ..., m$
- $j$ part index; $j = 1, ..., n$
- $k$ cell index; $k = 1, ..., c$
- $l$ index of membership functions; $l = 0, ..., c$
- $s$ index of fuzzy constraints; $s = 1, ..., c$

### Parameters

- $a_{ij}$ = 1 if part $j$ needs to be processed by machine $i$; 0 otherwise
- $A_i$ periodic cost of acquiring machine type $i$
- $C_i$ periodic capacity of machine type $i$
- $D_j$ periodic forecast demand for part $j$
- $I_j$ incremental cost for moving a unit of part $j$ within two cells
- $NM$ maximum number of machine types allowed in each cell
- $P_0$ tolerance value for the fuzzy objective function (cost)
- $P_r$ tolerance values for the fuzzy constraints ($NM$)
- $P_{ij}$ processing time of machine type $i$ needed to produce part $j$
- $\gamma$ parameter used in fuzzy modeling
- $S_j$ incremental cost of subcontracting a unit of part $j$ for an operation
- $SP$ set of pairs $(i, j)$ such that $a_{ij} = 1$
- $UC_{ij}$ utilization capacity of machine type $i$ for parts $j$. Value can be calculated from $P_{ij} \times D_j/C_i$
- $U_D(x)$ membership function of aggregated results
- $U_S(x)$ $s$th membership function
- $Z^0$ optimal solution using the maximum value of $NM$
- $Z^1$ optimal solution using the minimum value of $NM$

### Decision Variables

- $IC_k$ =1 if cell $k$ is formed; 0 otherwise
- $M_{ijk}$ number of machines $i$ dedicated to cell $k$ for producing part $j$
- $O_{ijk}$ units of part $j$ to be subcontracted as a result of machines type $i$ not being available within cell $k$
- $Q_i$ number of machines type $i$ needed to process corresponding parts in machine cell
- $R_{ik}$ number of machines type $i$ to be dedicated in cell $k$
- $U_{ijk}$ =1 if $X_{ik} = 1$ and $Y_{jk} = 0$; 0 otherwise
- $V_{ijk}$ =1 if $Y_{jk} = 1$ and $X_{ik} = 0$; 0 otherwise
- $X_{ik}$ =1 if machine $i$ is assigned to cell $k$; 0 otherwise
- $Y_{jk}$ =1 if part $j$ is assigned to cell $k$; 0 otherwise
- $Z_{ijk}$ number of inter-cell transfers required by part $j$ as a result of machine type $i$ not being available within part cell $k$
- $\lambda$ minimum value of all membership functions
- $\alpha_l$ extra variables used in the fuzzy And operator

## 3. Traditional model

The proposed model is an extension of the model used in [13], in which two major weaknesses can be found. First, the CF stage must occur separately, before optimization through elimination of EEs. The goodness of the results is highly dependent on the quality of the clustering. Secondly, the model does not consider machine capacity when accepting part transfers. The proposed model remedies both deficiencies. Not only is the best decision arrived at regarding the assignment of parts and machines to cells such that the total cost of dealing with EEs can be minimized, but also the available capacity and the investment cost of the machines are considered. The needed number of machines can consequently be determined and minimized. The formulation is as follows:

$$\text{Min} \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} \quad (1)$$

subject to:

$$\sum_{k=1}^{c} X_{ik} = 1, \quad \forall i; \quad (2)$$

$$\sum_{k=1}^{c} Y_{jk} = 1, \quad \forall j; \quad (3)$$

$$\sum_{i=1}^{m} X_{ik} \leq NM, \quad \forall k; \quad (4)$$

$$X_{ik} - Y_{jk} + \frac{1}{D_j} Z_{ijk} + \frac{1}{D_j} O_{ijk} + \frac{1}{UC_{ij}} M_{ijk} - V_{ijk} = 0, \quad \forall (i,j) \in sp, \ \forall k; \quad (5)$$

$$\sum_{(i,j)\in SP} M_{ijk} \leq R_{ik}, \quad \forall i, \ \forall k; \quad (6)$$

$$Q_i \leq \sum_{(i,j)\in SP} UC_{ij}(1 - \sum_k V_{ijk}) + 1, \quad \forall i; \quad (7)$$

$$\sum_k \sum_{(i,j)\in SP} \frac{P_{ij}}{C_i} Z_{ijk} \leq Q_i - \sum_{(i,j)\in SP} UC_{ij}(1 - \sum_k V_{ijk}), \quad \forall i; \quad (8)$$

$$X_{ik},\ Y_{jk},\ U_{ijk},\ V_{ijk} = 0 \text{ or } 1; \quad R_{ik},\ Q_i = \text{ general integer.} \tag{9}$$

Three types of cost associated with EEs are to be minimized in (1). The first cost is that of duplicating a machine. This cost subsumes the purchase, maintenance, salvage, and machine life. The second cost includes inter-cell transfers for the EE. The last cost is that of subcontracting. Constraints (2) and (3) ensure that each machine and part is assigned to only one cell. Constraint (4) prevents the assignment of more than $NM$ machines to each cell. This constraint also prevents all machines and parts from being assigned to a single cell. Constraint (5) combines the two equations

$$X_{ik} - Y_{jk} + U_{ijk} - V_{ijk} = 0, \quad \forall\,(i,j) \in SP,\ \forall\,k \text{ and}$$
$$Z_{ijk} + O_{ijk} + (C_i/P_{ij})M_{ijk} = D_j U_{ijk}, \quad \forall\,(i,j) \in SP,\ \forall\,k,$$

and ensures that an EE either is an *exceptional machine* (a machine to be duplicated) or an *exceptional part* (a part to be transferred or subcontracted). Furthermore, the constraint guarantees that the demand of exceptional part $j$ can be shared by the duplicated machine $i$, transfer within cells, or subcontracting. Constraint (6) calculates the number of machines of type $i$ needed to be dedicated within cell $k$ to producing the EEs, where $M_{ijk}$ is a real variable representing the utilization capacity of a machine type $i$ dedicated to process part $j$ in cell $k$. Constraint (7) determines the number of machine type $i$ needed in each cell. The constraint sums the utilization capacity of machine type $i$ for all relative parts ($\sum_{(i,j)\in SP} UC_{ij}$), not the EEs ($1 - \sum_k V_{ijk}$). Constraint (8) ensures that the number of inter-cell transfers between machines of type $i$ do not exceed the available machine capacity. The maximum available capacity of machine type $i$ (machine unit) for the relative parts to be transferred is equal to $C_i/P_{ij}$ times the right-hand side of (8).

## 4. Fuzzy models

In a fuzzy environment, mathematical programming models must take into consideration fuzzy constraints, vague goals, and ambiguous parameters. Many FMP approaches have been developed for these combinations and can be classified into different categories according to different criteria. For instance, vague versus ambiguous [18, 20–28]; symmetric versus asymmetric [19, 21, 28–32]; and traditional versus interactive [18, 33]. A detailed discussion of the FMP procedure can be found in [18, 19], for example.

Although FMP is different from other fuzzy applications such as fuzzy inference and fuzzy ranking, two factors – membership functions and fuzzy operators – common to other fuzzy applications deserve attention. Membership functions are used to incorporate fuzziness or to represent the linguistic variables for applications of fuzzy set theory. Many types of membership function have been used in practice [2, 18, 32, 34–36]: linear nonincreasing [3–5] and triangular [1, 6] are two of the most popular functions in use. To manipulate the fuzzy numbers in fuzzy sets, a variety of fuzzy operators extended from the traditional (crisp) mathematical operators have been developed. For instance, let $\widetilde{A}$ and $\widetilde{B}$ be two fuzzy sets in $\boldsymbol{X}$, the fuzzy sum $\widetilde{C} = \widetilde{A} + \widetilde{B}$ is defined as $\widetilde{C} = \left\{\left(x, \mu_{\tilde{A}+\tilde{B}}(x)\right) | x \in X\right\}$ where $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ are the membership functions of $x$ in $\widetilde{A}$ and $\widetilde{B}$ and $\mu_{\tilde{A}+\tilde{B}}(x) = \mu_{\tilde{A}}(x) + \mu_{\tilde{B}}(x) - \mu_{\tilde{A}}(x) \cdot \mu_{\tilde{B}}(x)$. In FMP, fuzzy operators are used to transform FMP to traditional MP so that the FMP can be solved via traditional MP software. Table 1 summarizes the operators that often have been applied on FMP. The first five operators have simple linear forms after transformation. Their characteristics can be found in [19, 34, 37, 38].

The aforementioned traditional model assumes that objective functions and constraints can be defined precisely; however, in practice, it is quite difficult for the decision maker to specify the exact goals and constraints when modeling the problem. For example, the right-hand side of constraint (4) is often fuzzy and can be expressed as

$$\sum_{i=1}^{m} X_{ik} \,\tilde{\leq}\, NM, \quad \forall\,k \quad \text{or} \quad \sum_{i=1}^{m} X_{ik} \cong NM, \quad \forall\,k. \tag{10}$$

According to Werners [31], (1) can be fuzzified as:

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,\,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,\,j)\in SP} S_j O_{ijk} \,\tilde{\leq}\, Z^0 = Z^1 - P_0. \tag{11}$$

Here, the value of $Z^0$ is the feasible value of the best goal, which can be obtained by solving the traditional model with the maximum value of $NM$. $Z^1$ is the feasible value of the worst goal, which can be obtained by solving the same model with the minimum value of $NM$. To transform the fuzzy model to a traditional formulation, two tolerance values, $P_r$ and $P_0$, must be determined in advance. $P_r$, for (10), is normally determined by the decision maker, according to problem characteristic or experience. $P_0$ for (11) can be determined from the budget limit and its allowance, or it can be set as a value equal to $Z^0$ subtracted from $Z^1$. Fuzzy objective function (11) and fuzzy constraint (10) then can be transformed to the conventional formulation by means of a suitable operator. In this paper, we estimated the $P_0$ values according to the average machine cost and the percentage of EEs of each data set. Two membership functions (linear nonincreasing and triangular) and four operators (‘max–min’, ‘añd’, ‘add’, and a newly proposed operator) will be examined. Additionally, if the membership function is linear and nonincreasing, the transformation formula in Zimmermann [19] will be used; otherwise, the formula in Yang and

**Table 1.** Summary of common operators used in the FLP studies

| *Operator* | *Formulation*[a] | *Compensatory*[b] | *Format after transformation* | *References* |
|---|---|---|---|---|
| 'Max-min' | $U_D = \min U_S$ | No | Linear | [1, 3–6] |
| 'Min-bound-sum' | $U_D = \gamma \min U_S + (1-\gamma)\min\left(1, \sum_{S=0}^{t} U_S\right)$ | Positive | Linear | [43] |
| 'Compensatory and' | $U_D = \gamma \min U_S + (1-\gamma)\max U_S$ | Positive | Linear | [44] |
| 'Fuzzy and' ('añd') | $U_D = \gamma \min U_S + (1-\gamma)/(t+1) \times \sum_{S=0}^{t} U_S$ | Positive | Linear | [38, 44] |
| 'Add' | $U_D = \sum_{S=0}^{t} U_S$ | Negative | Linear | [5] |
| 'Product' | $U_D = \prod_{S=1}^{t} U_S$ | Negative | Nonlinear | [16, 36] |
| $\gamma$ | $U_D\left(\prod_{S=0}^{t} U_S\right)^{1-\gamma}\left[1 - \prod_{S=0}^{t}(1-U_S)\right]^{\gamma}$ | Positive | Nonlinear | [43] |

[a]The objective is to maximize $U_D$.
[b]The definitions of compensatory and of negative compensatory are adapted from [37].

Ignizio [39] will be used. Consequently, there are eight cases needing to be examined for (10) and (11). The first two cases are the combinations of 'max–min' operator and different membership functions, cases 3 and 4, the combinations of the 'añd' operator and different membership functions. Cases 5 and 6 are the combinations of the 'add' operator and different membership functions. Cases 7 and 8 combine the proposed operator with different membership functions. The generic forms can be found in Appendix A. Equivalent formulations for each case are summarized below. Please note that, for each case, complete formulation includes those equations noted below as well as (2), (3), and (5)–(9).

**Case 1:**
'Max–min' operator and linear nonincreasing membership function. The equivalent formulation can be obtained as

$$\text{Max } \lambda \tag{12}$$

subject to

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} + \lambda P_0 \leq Z^0 + P_0; \tag{13}$$

$$\sum_{i=1}^{m} X_{ik} + \lambda P_r \leq NM + P_r, \quad \forall k; \tag{14}$$

$$\sum_{k=1}^{c} IC_k \geq 2; \tag{15}$$

$$\sum_{i=1}^{m} X_{ik} \geq 2IC_k, \quad \forall k; \tag{16}$$

$$0 \leq \lambda \leq 1. \tag{17}$$

Constraint (15) prevents the formation of fewer than two cells, and constraint (16) prevents the assignment of fewer than two machines in each cell formed. There, the $IC_k$ variables in (15) and (16) are added either to allow a lower-bound value of the number of machines in each cell when there is a linear nonincreasing membership function or to prevent the grouping of parts and machines into exactly $C$ cells when there is a triangular membership function. Consequently, the number of cells permitted ranges from 2 to $C$, and when linear nonincreasing membership is used, the number of machines allowed in each cell ranged from 2 to $NM + P_r$.

**Case 2:**
'Max–min' operator and triangular membership function. The equivalent LP formulation consists of (12)–(15), (17), and (18):

$$\sum_{i=1}^{m} X_{ik} - NM \times IC_k - \lambda P_r \geq -P_r, \quad \forall k. \tag{18}$$

The difference between cases 1 and 2 is that the triangular membership function requires one additional constraint (18) to define the minimum value of $\sum_{i=1}^{m} X_{ik}$.

**Case 3:**
The 'añd' operator and the linear nonincreasing membership function. The equivalent LP formulation consists of (19), (15), (16), and (20)–(23).

$$\text{Max } \alpha + (1-\gamma)\frac{1}{c+1}\sum_{l=0}^{c}\alpha_l \tag{19}$$

subject to

$$\sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} + \alpha P_0 + \alpha_0 P_0 \le Z^0 + P_0; \tag{20}$$

$$\sum_{i=1}^{m} X_{ik} + \alpha P_r + \alpha_k P_r \le NM + P_r, \quad \forall k; \tag{21}$$

$$\alpha + \alpha_l \le 1, \quad l = 0, \ldots, c; \tag{22}$$

$$\alpha_l \ge 0, \quad 0 \le \alpha \le 1, \quad \gamma < 1. \tag{23}$$

Because no simple rule can be applied to decide the value of $\gamma$, determining it becomes a major bottleneck in the use of the 'añd' operator. An experiment has been conducted to identify the nature of the $\gamma$ value. Results are discussed below.

**Case 4:**
The 'añd' operator and triangular membership function. The equivalent LP formulation can be expressed as (24), (15), (20), (21), (25), (26), and (23).

$$\text{Max } \alpha + (1-\gamma)\frac{1}{2c+1}\sum_{l=0}^{2c}\alpha_l \tag{24}$$

subject to

$$\alpha + \alpha_l \le 1, \quad l = 0, \ldots, 2c; \tag{25}$$

$$\sum_{i=1}^{m} X_{ik} - NM \times IC_k - \lambda \times P_r - \alpha_{c+k} \times P_r \ge P_r, \ \forall k, \tag{26}$$

**Case 5:**
'Add' operator and linear nonincreasing membership function. The equivalent LP formulation consists of (27), (15), (16), (28), and (29):

$$\text{Min } \left( \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} - Z_0 \right) \Big/ P_0 + \sum_{s=1}^{c} \frac{S_s}{P_r} \tag{27}$$

subject to:

$$\sum_{i=1}^{m} X_{ik} - S_k \le NM, \quad \forall k; \tag{28}$$

$$S_s \le P_r, \quad s = 1, \ldots, c. \tag{29}$$

**Case 6:**
'Add' operator and triangular membership function. The equivalent LP formulation consists of (30), (15), (28), (31), and (32):

$$\text{Min } \left( \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} - Z_0 \right) \Big/ P_0 + \sum_{s=1}^{2c} \frac{S_s}{P_r} \tag{30}$$

subject to

$$\sum_{i=1}^{m} X_{ik} + S_{c+k} \ge NM \times IC_k, \quad \forall k; \tag{31}$$

$$S_s \le P_r, \quad s = 1, \ldots, 2c. \tag{32}$$

**Case 7:**
The proposed operator (denoted as 'add–min') and linear nonincreasing membership function. The proposed operator applies the 'min' operator for the fuzzy constraints. Thus, the aggregated membership function becomes

$$U_D = 0.5(U_G + \min_{s=1} U_s). \tag{33}$$

The range of $U_D$ is [0, 1]. The model after aggregation is a linear form, and the compensatory property is better than the 'max–min' and 'add' operators. The equivalent transformed formulation is

$$\text{Max } U_G + \lambda \tag{34}$$

subject to

$$U_G = 1 - \left( \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} - Z_0 \right) \Big/ P_0; \tag{35}$$

$$1 - \left( \sum_{i=1}^{m} X_{ik} - NM \right) \Big/ P_r \ge \lambda, \quad \forall k. \tag{36}$$

If we insert constraint (35) into (34), the fuzzy objective function becomes (37). The fuzzy constraints include (14)–(17).

$$\text{Min } \sum_k \sum_i A_i R_{ik} + \sum_k \sum_{(i,j)\in SP} I_j Z_{ijk} + \sum_k \sum_{(i,j)\in SP} S_j O_{ijk} - \lambda P_0. \tag{37}$$

**Case 8:**
The proposed operator and triangular membership function. The objective function of the equivalent LP formulation is the same as in (37), and the constraints are (14), (15), (17), and (18).

## 5. Data sets for numerical computations

To evaluate the performance of the proposed fuzzy models, four data sets with different percentages of EEs are used. Here, the percentage of EEs is defined as their number divided by the total number of elements in a data set. Data set I, adapted from [13], has the highest percentage of EEs (28.6%) and can be formed into two or three cells. Table 2 lists the processing time of each part, the costs involved, the part demand, and the machine capacity of this data set. Data set II, with an intermediate EE percentage (18.8%), also can be formed into two or three cells. The machine / part matrix of this data set was from [40], but the other data were generated randomly by a computer program on the basis of the mean value and the standard deviation of data set I. The percentage of EEs of data set III is 3.3%, the lowest percentage. Two, three, or four cells can be formed in this data set. The machine / part matrix of data set III was from [12], and the other related data were also generated randomly. The percentage of EEs of data set IV is 6.87%. Three or four cells can be formed in this data set. The machine / part matrix of data set IV was from [41], and the other related data were also generated randomly. For the first three data sets, the maximum number of machines allowed (*NM*) in each cell is set as 'no more than four' or 'around four'. Thus, for the traditional model, the right-hand side of (5) is a fixed number 'four'; under the fuzzy environment, however, we can use fuzzy intervals to represent these numbers. We have also relaxed the *NM* value into 'six' in the first three data sets for the crisp model (i.e., non-fuzzy model) to see its possible impacts on the solution quality. Similarly, the *NM* value for data set IV was first set at 'nine' and relaxed to 'twelve' for exploration purposes.

## 6. Computational results

All models were solved by running the LINDO (linear interactive and discrete optimizer) package on a PC with Intel's 120 MHz Pentium processor. Clustering performances were measured in terms of EE numbers, executing CPU time, and total costs of dealing with EEs. Table 3 summarizes computational results from ten different cases. Cases 1 to 8 already have been discussed. Case 9 is the proposed traditional (crisp) CF model and case 10 is a two-phase sequential model, in which an MIP (whose objective is to minimize the number of exceptional elements) was first used to form the manufacturing cells, the modified formulation (to consider capacity restriction for part transfers) from [13] was then used as a post-processor to deal with the exceptional elements. The last two cases were included for purposes of comparison. Several observations can be made on the basis of the information in Table 3.

First, despite the fact that the sequential approach and the proposed traditional (crisp) model are more computationally efficient (i.e., consume less CPU time), their clustering results cannot compete with the proposed fuzzy models. For instance, under the comparable settings (i.e., formulations with same *NM* restriction and cell number), the sequential approach and the proposed crisp model always resulted in a larger number of exceptional elements and significantly higher cost of dealing with EEs (see the first row of cases 9 and 10 of Table 3). If we carefully design the fine-tuning process, the solution of the traditional model can improve significantly. For instance, in data sets I and II, its solutions are even better

**Table 2.** Numerical values for data set I

| | *Parts* | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Machines* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | $A_i$($) | $C_i$ |
| 1 | 2.95 | 0 | 2.2 | 0 | 0 | 0 | 0 | 0 | 0 | 4.61 | 50 784 | 2000 |
| 2 | 2.76 | 5.18 | 1.89 | 3.89 | 0 | 5.14 | 0 | 0 | 0 | 0 | 67 053 | 2000 |
| 3 | 5.54 | 4.29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 43 944 | 2000 |
| 4 | 2.91 | 0 | 0 | 1.97 | 2.59 | 4.01 | 0 | 2.7 | 0 | 0 | 67 345 | 2000 |
| 5 | 0 | 0 | 0 | 4.28 | 0 | 4.51 | 0 | 0 | 0 | 0 | 42 414 | 2000 |
| 6 | 1.92 | 0 | 0 | 0 | 0 | 0 | 2.23 | 0 | 5.52 | 0 | 75 225 | 2000 |
| 7 | 0 | 0 | 0 | 0 | 3.4 | 0 | 1.16 | 4.72 | 0 | 2.49 | 52 741 | 2000 |
| 8 | 0 | 5.32 | 0 | 0 | 0 | 0 | 0 | 3.75 | 3.85 | 0 | 63 523 | 2000 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 4.04 | 0 | 0 | 1.83 | 50 632 | 2000 |
| $S_j$($) | 4.20 | 4.30 | 3.50 | 4.40 | 5.00 | 3.90 | 4.40 | 4.60 | 5.00 | 5.00 | | |
| $D_j$ | 32 128 | 27 598 | 20 651 | 11 340 | 18 707 | 17 040 | 46 196 | 45 384 | 16 409 | 22 000 | | |
| $I_j$($) | 3.70 | 2.80 | 2.80 | 3.30 | 2.80 | 3.50 | 2.80 | 2.60 | 3.40 | 3.20 | | |

**Table 3.** Computational results for crisp and fuzzy models

(a) Data set I: Problem size $(m \times n) = 9 \times 10$; % of EE = 28.6%; $c = 3$; $NM = 4$; $P_r = 2$; $P_0 = \$166\ 000$

| *Case* | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Operator* | *Membership functions* | {*Machines/parts*} | *Number of machines needed*[a] | *Inter-cell movement (M) and subcontract (S)*[b] | *CPU time (H:M:S)* | *No. of EE* | *Total cost of dealing with EE ($)* |
| 1. 'Max–min' | Linear nonincreasing | {1,2,3,4,5/1,2,3,4,5,6}<br>{6,7,8,9/7,8,9,10} | 1(2);2(4);3(2);4(2);5(2);6[1];7[1];8[1]<br>6(2);7(3);8(2);9(2);1[1];4[1] | M: 2(1177); 8(941)<br>S: 2(3870) | (24:34:29) | 5 | 332 001 |
| 2. 'Max–min' | Triangular | Same as case 1 | Same as case 1 | Same as case 1 | (16:31:12) | 5 | 332 001 |
| 3. 'Añd' | Linear nonincreasing | {2,4,5,7,8/2,'4,5,6,8,9}<br>{1,3,6,9/1,3,7,10} | 2(3);4(3);5(2);7(3);8(4);3[1]<br>1(3);3(2);6(2);9(2);2[1];4[1];7[1] | M: 1(2792); 9(13 641)<br>S: 9(2769) | (5:18:15) | 7 | 301 638 |
| 4. 'Añd' | Triangular | Same as case 3 | Same as case 3 | Same as case 3 | (0:44:43) | 7 | 301 638 |
| 5. 'Add' | Linear nonincreasing | Same as case 3 | Same as case 3 | Same as case 3 | (0:27:45) | 7 | 301 638 |
| 6. 'Add' | Triangular | Same as case 3 | Same as case 3 | Same as case 3 | (0:26:24) | 7 | 301 638 |
| 7. 'Add–min' | Linear nonincreasing | Same as case 3 | Same as case 3 | Same as case 3 | (0:14:56) | 7 | 301 638 |
| 8. 'Add–min' | Triangular | Same as case 3 | Same as case 3 | Same as case 3 | **(0:08:28)** | **7** | **301 638** |
| 9. Traditional Model | $NM = 4$ (default) | {1,2,3,5/1,2,3,4,6}<br>{4,7,8/5,8}<br>{6,9/7,9,10} | 1(2);2(4);3(3);5(2);4[1];6[1];8[1]<br>4(2);7(3);8(2)<br>6(2);9(2);1[1];7[1] | M: 2(1280);6(15 997); 9(16 421)<br>S: 2(3767) | (0:21:06) | 9 | 441 221 |
| | $NM = 6$ | {2,3,4,5,7,8/1,2,4,5,6,8}<br>{1,6,9/3,7,9,10} | 2(4);3(3);4(3);5(2);7(3);8(3);1[1];6[1]<br>1(2);6(2);9(2);7[1] | M: 3(20 662) | **(0:11:02)** | **6** | **236 604** |
| 10. Sequential Model | $NM = 4$ (default) | {1,2,3/1,2,3}<br>{4,5/4,5,6}<br>{6,7,8,9/7,8,9,10} | 1(2);2(3);3(3)<br>4(2);5(2);2[2]<br>6(2);7(4);8(4);9(2);1[1];4[1] | M: (27 598);5(18 707); 6(2276);8(940)<br>S: 1(32 128) | (0:04:56) | 8 | 527 237 |
| | $NM = 6$ | {2,3,4,5,6,8/1,2,4,6,8,9}<br>{1,7,9/3,5,7,10} | 2(4);3(3);4(3);5(2);6(2);8(4);1[1];6[1]<br>1(2);7(3);9(2);7[1] | M: 3(20 662);5(18 707); 8(19 961) | (0:03:05) | 5 | 340 882 |

(b) Data set II: Problem size $(m \times n) = 9 \times 9$; % of EE = 18.8%; $c = 3$; $NM = 4$; $P_r = 2$; $P_0 = \$121\ 000$

| *Case* | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Operator* | *Membership functions* | {*Machines/parts*} | *Number of machines needed*[a] | *Inter-cell movement (M) and subcontract (S)*[b] | *CPU time (H:M:S)* | *No. of EE* | *Total cost of dealing with EE ($)* |
| 1. 'Max–min' | Linear nonincreasing | {1,2,6,9/1,2,6,9}<br>{3,4,5,7,8/3,4,5,7,8} | 1(2);2(4);6(3);9(2);5[1];4[1]<br>3(2);4(3);5(3);7(2);8(4);1[1] | | (2:43:52) | 3 | 168 200 |
| 2. 'Max–min' | Triangular | Same as case 1 | Same as case 1 | | (1:34:45) | 3 | 168 200 |
| 3. 'Añd' | Linear nonincreasing | Same as case 1 | Same as case 1 | | (0:19:32) | 3 | 168 200 |
| 4. 'Añd' | Triangular | Same as case 1 | Same as case 1 | | (0:07:22) | 3 | 168 200 |
| 5. 'Add' | Linear nonincreasing | Same as case 1 | Same as case 1 | | (0:04:07) | 3 | 168 200 |
| 6. 'Add' | Triangular | Same as case 1 | Same as case 1 | | (0:03:32) | 3 | 168 200 |
| 7. 'Add–min' | Linear nonincreasing | Same as case 1 | Same as case 1 | | (0:02:53) | 3 | 168 200 |

**Table 3.** (*Continued*)

| *Case* | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Operator* | *Membership functions* | {*Machines/parts*} | *Number of machines needed*[a] | *Inter-cell movement (M) and subcontract (S)*[b] | *CPU time (H:M:S)* | *No. of EE* | *Total cost of dealing with EE ($)* |
| 8. 'Add–min' | Triangular | Same as case 1 | Same as case 1 | | **(0:02:08)** | **3** | **168 200** |
| 9. Traditional Model | $NM = 4$ (default) | {1,5/1,5}<br>{2,6,9/2,6,9}<br>{3,4,7,8/3,4,7,8} | 1(2);5(3);2[1];8[1]<br>2(3);6(3);9(2);4[1]<br>3(2);4(3);7(2);8(3);5[1] | M: 2(15 224);4(7236) | (0:06:28) | 6 | 289 465 |
| | $NM = 6$ | {1,3,4,5,7,8/1,3,4,5,7,8}<br>{2,6,9/2,6,9} | 1(2);3(2);4(3);5(4);7(2);8(4);2[1]<br>2(3);6(3);9(2);4[1] | M: 2(15 221) | **(0:02:16)** | **3** | **145 524** |
| 10. Sequential Model | $NM = 4$ (default) | {1,5/1,5}<br>{2,6,9/2,6,9}<br>{3,4,7,8/3,4,7,8} | 1(2);5(3);2[1];8[1]<br>2(3);6(3);9(2)<br>3(2);4(3);7(2);8(3);5[1] | M: 4(7236)<br>S: 2(15 224) | (0:01:05) | 6 | 252 290 |
| | $NM = 6$ | {1,2,6,9/1,2,6,9}<br>{3,4,5,7,8/3,4,5,7,8} | 1(2);2(4);6(3);9(2);5[1]<br>3(2);4(4);5(3);7(2);8(4);1[1] | M: 2(15 224) | **(0:01:03)** | **3** | **161 930** |

(c) Data set III: Problem size $(m \times n) = 14 \times 24$; % of EE = 3.3%; $c = 4$; $NM = 4$; $P_r = 2$; $P_0 = \$209\ 660$

| *Case* | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Operator* | *Membership functions* | {*Machines/parts*} | *Number of machines needed*[a] | *Inter-cell movement (M) and subcontract (S)*[b] | *CPU time (H:M:S)* | *No. of EE* | *Total cost of dealing with EE ($)* |
| 1. 'Max–min' | Linear nonincreasing | {1,4,5,7,13/1,2,6,7, 17–20,23}<br>{2,3,10,11/3,4,21,24}<br>{6,8,9,12,14/5,8–16,22} | 1(3);4(5);5(6);7(7);13(5);12[1]<br>2(2);3(2);10(2);11(3)<br>6(7);8(8);9(5);12(1);14(4);13[1] | M: 7(1427) | (1:55:06) | 2 | 98 938 |
| 2. 'Max–min' | Triangular | Same as case 1 | Same as case 1 | Same as case 1 | (0:24:16) | 2 | 98 938 |
| 3. 'Añd' | Linear nonincreasing | {1,4,5,7,13/1,2,6,7,8, 17–20,23}<br>{2,3,10,11/3,4,21,24}<br>{6,8,9,12,14/5,9–16,22} | 1(3);4(5);5(6);7(7);13(6);12[1]<br>2(2);3(2);10(2);11(3)<br>6(7);8(8);9(5);12(1);14(4) | M: 8(9924) | (0:46:34) | 2 | 67 777 |
| 4. 'Añd' | Triangular | Same as case 3 | Same as case 3 | Same as case 3 | (0:32:54) | 2 | 67 777 |
| 5. 'Add' | Linear nonincreasing | Same as case 9 (default) | Same as case 9 (default) | Same as case 9 (default) | (0:26:05) | 2 | **209 660** |
| 6. 'Add' | Triangular | Same as case 3 | Same as case 3 | Same as case 3 | (0:23:45) | 2 | 67 777 |
| 7. 'Add–min' | Linear nonincreasing | Same as case 3 | Same as case 3 | Same as case 3 | (0:17:32) | 2 | 67 777 |
| 8. 'Add–min' | Triangular | Same as case 3 | Same as case 3 | Same as case 3 | **(0:10:31)** | **2** | **67 777** |
| 9. Traditional Model | $NM = 4$ (default) | {1,12,13/6,7,8,18}<br>{2,3,10,11/3,4,21,24}<br>{4,5,7/1,2,17,19,20,23}<br>{6,8,9,14/5,9–16,22} | 1(3);12(2);13(4);7[2]<br>2(2);3(2);10(2);11(3)<br>4(5);5(6);7(5)<br>6(7);8(8);9(5);14(4);13[2] | | (0:37:22) | 2 | 209 660 |
| | $NM = 5$ | {1,4,5,7,13/1,2,6,7,8, 17–20,23}<br>{2,3,10,11,12/3,4,21,24}<br>{6,8,9,14/5,9–16,22} | 1(3);4(5);5(6);7(7);13(6);12[1]<br>2(2);3(2);10(2);11(3);12(1)<br>6(7);8(8);9(5);14(4) | M: 8(9924) | **(0:11:21)** | **2** | **67 777** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10. Sequential Model | $NM = 4$ (default) | {1,12,13/6,7,8,18}<br>{2,3,10,11/3,4,21,24}<br>{4,5,7/1,2,17,19,20,23}<br>{6,8,9,14/5,9–16,22} | 1(3);12(2);13(6)<br>2(2);3(2);10(2);11(3)<br>4(5);5(6);7(7)<br>6(7);8(8);9(5);14(4) | M: 7(44 794);23(39 667) | (0:07:08) | 2 | 146 436 |
| | $NM = 5$ | {1,4,5,7,13/1,2,6,7,8, 17–20,23}<br>{2,3,10,11,12/3,4,21,24}<br>{6,8,9,14/5,9–16,22} | 1(3);4(5);5(6);7(7);12(6);12[1]<br>2(2);3(2);10(2);11(3);12(1)<br>6(7);8(8);9(5);14(4) | M: 8(9924) | **0:06:38** | 2 | **67 777** |

(d) Data set IV: Problem size $(m \times n) = 30 \times 50$; % of EE = 6.87%; $c = 4$; $NM = 9$; $P_r = 4$; $P_0 =$ \$650 966

| *Case* | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Operator* | *Membership functions* | {*Machines/parts*} | *Number of machines needed*[a] | *Inter-cell movement (M) and subcontract (S)*[b] | *CPU time (H:M:S)* | *No. of EE* | *Total cost of dealing with EE (\$)* |
| 1. 'Max–min' | Linear nonincreasing | – | – | – | >48:00:00 | – | – |
| 2. 'Max–min' | Triangular | – | – | – | >48:00:00 | – | – |
| 3. 'Añd' | Linear nonincreasing | {1–6,8–13/1–18}<br>{18–23/28–38}<br>{7,14–17/19–27}<br>{24–30/39–50} | 1(7);2(4);3(5);4(4);5(7);6(5),8(8);9(6); 10(6);11(4);12(5);13(7);7[3]<br>18(4);19(4);20(5);21(6);22(5);23(3)<br>7(1);14(6);15(5);16(6);17(4)<br>24(5);25(6);26(3);27(4);28(3);29(6);30(6) | M: 12(9561) | (20:32:56) | 4 | 146 852 |
| 4. 'Añd' | Triangular | {1–6,8–13/1–18}<br>{14–23/19–38}<br>{7,24–30/39–50} | 1(7);2(4);3(5);4(4);5(7);6(5),8(8);9(6); 10(6);11(4);12(5);13(7),7[3]<br>14(6);15(5);16(6);17(4);18(4);19(4); 20(5);21(6);22(5);23(3)<br>7(1);24(5);25(6);26(3);27(4);28(3); 29(6);30(6) | M: 12(9561) | (11:42:34) | 4 | 146 852 |
| 5. 'Add' | Linear nonincreasing | Same as case 3 | Same as case 3 | Same as case 3 | (7:56:39) | 4 | 146 852 |
| 6. 'Add' | Triangular | Same as case 4 | Same as case 4 | Same as case 4 | (4:32:19) | 4 | 146 852 |
| 7. 'Add–min' | Linear nonincreasing | Same as case 3 | Same as case 3 | Same as case 3 | (0:47:53) | 4 | 146 852 |
| 8. 'Add–min' | Triangular | Same as case 4 | Same as case 4 | Same as case 4 | (**0:45:07**) | **4** | **146 852** |
| 9. Traditional Model | $NM = 9$ (default) | {1–5,9,10,11,13/1–11,17, 18}<br>{6,7,8,12,14–17/12–16, 19–27}<br>{18–23/28–38}<br>{24–30/39–50} | 1(7);2(4);3(5);4(4);5(7);9(6);10(7); 11(4);13(7)<br>6(3);7(2);8(5);12(2);14(6);15(5);16(6); 17(4)<br>18(4);19(4);20(5);21(6);22(5); 23(3);6[2];7[2];8[2];12[2]<br>24(5);25(6);26(3);27(4);28(3);29(6);30(6) | M: 9(10 603);7(3769) | (10:25:56) | 9 | 650 966 |
| | $NM = 12$, $c = 3$ | {1–6,8–13/1–18}<br>{14–23/19–38}<br>{7,24–30/39–50} | 1(7);2(4);3(5);4(4);5(7);6(5),8(8);9(6); 10(6);11(4);12(5);13(7),7[3]<br>14(6);15(5);16(6);17(4);18(4);19(4);20(5); 21(6);22(5);23(3)<br>7(1);24(5);25(6);26(3);27(4); 28(3);29(6);30(6) | M: 12(9561) | (**1:40:54**) | **4** | **146 852** |

**Table 3.** (*Continued*)

| *Case* | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Operator* | *Membership functions* | {*Machines/parts*} | *Number of machines needed*[a] | *Inter-cell movement (M) and subcontract (S)*[b] | *CPU time (H:M:S)* | *No. of EE* | *Total cost of dealing with EE ($)* |
| 10. Sequential Model | $NM = 9$ (default) | {1–5,9,10,11,13/1–4,6–11,17,18} | 1(7);2(4);3(5);4(4);5(7);9(6);10(7);11(4);13(7) | M: 4(15 010);5(3664);6(12 018);7(28 460);9(15 662) S: 5(10 436) | (0:41:02) | 3 | 695 102 |
| | | {6,7,8,12,14–17/5,12–16,19–27} | 6(4);7(2);8(5);12(3);14(6);15(5);16(6);17(4);1[1];13[1] | | | | |
| | | {18–23/28–38} | 18(4);19(4);20(5);21(6);22(5);23(3);6[1];7[1];8[1];12[2] | | | | |
| | | {24–30/39–50} | 24(5);25(6);26(3);27(4);28(3);29(6);30(6) | | | | |
| | $NM = 12$ | {1–10,12,13/1–18} | 1(7);2(4);3(5);4(4);5(7);6(5),8(8);9(6);10(6);12(5);13(7),7(4) | M: 4(17 256);7(44 370);8(31 816) | (0:22:02) | 3 | 223 687 |
| | | {11,14–23/19–38} | 11(4);14(6);15(5);16(6);17(4);18(4);19(4);20(5);21(6);22(5);23(3) | | | | |
| | | {24–30/39–50} | 24(5);25(6);26(3);27(4);28(3);29(6);30(6) | | | | |

[a]i(no.): no. of machines which are needed for machine type *i*; *i*[no.]: no. of machine type *i* duplicated.
[b]m(no.): Units of part type *m* receive treatment of subcontract (S) or inter-cell movement (M).

than the best of fuzzy models. Although solutions of the sequential approach are improving, most of them yet remain the worst. Furthermore, the sequential approach requires one to predetermine the manufacturing cells and its solution is highly dependent on the quality of the cell formation. If the clustering result is good, it will consume less CPU time and result in a lower cost; otherwise, the computational efficiency may deteriorate. This indicates that our proposed crisp model is more convenient and robust than the sequential approach.

Secondly, although with some trial and error a better solution may be obtained using the proposed crisp model, the process is very subjective, tedious, and time-consuming. A general rule-of-thumb for obtaining a good solution is to set the *NM* value larger (see the contrast of cases 9 in Table 3). However, with a larger *NM* value, there is a risk of assigning machines and parts into a single cell. On the contrary, with a good operator such as the proposed 'add–min' (case 8), the use of the fuzzy model is very straightforward. It can flexibly adjust the *NM* values and cell number to find a good solution in a single run, thus avoiding, tedious and time-consuming trial and error. Also, the processing time of a good fuzzy model is always less than that of the proposed crisp model. The time advantage of using a fuzzy model over the sequential and traditional models becomes significant once the problem size and the tolerance value of the constraints (i.e., *NM* value and number of cells) becomes bigger, as it would take quite a long time to complete each run. This is evident from cases 9 and 10 of data set IV.

Thirdly, whereas the computational CPU times of the fuzzy models differed from case to case, clustering results, except for case 5 of data set III, were far better than those obtained for the traditional crisp model under the same (default) settings. This can be seen from the fact that EE number and total costs were smaller for all fuzzy models except case 5. The clustering results of case 5 actually were the same as those for the traditional model.

Fourthly, even though the 'max–min' operator was used most often in the literature, its performance was surprisingly unacceptable. Not only did it require the longest time to process, clustering results (in terms of cost and of EE number) were also always worse than those of the other operators. For instance, after more than 48 hours it still had not finished the computations for data set IV. Also, more than 24 hours and 16 hours were required to complete cases 1 and 2 of data set I, respectively, and the costs of dealing with EEs were far higher than those in the other fuzzy cases. Although the situation is improved in data set III, the results were still unsatisfactory. For instance, nearly 2 hours were required for using the linear nonincreasing membership function, and 25 minutes when a triangular membership function was used.

Fifthly, although the 'añd' operator often arrived at good clustering results and required shorter CPU time than the 'max–min' operator, the performance time of the former was still far worse than that of either the 'add' or the proposed operators. Meanwhile, determining a proper $\gamma$ value was difficult. Therefore, an experiment was needed to determine the best possible values of $\gamma$. Table 4 summarizes the results of using data set II. According to the table, the best $\gamma$ values for cases 3 and 4 are 0.8 and 0.2, respectively. Although they are not shown, the best values for data set I were 0.2 and 0.1, and for data sets III and IV were 0.3 and 0.8. The best value for $\gamma$ clearly depended on the data set. Nevertheless, when the best $\gamma$ values are used for comparison, the 'añd' operator still performed worse in terms of CPU time than the 'add' operator or the proposed operator.

Sixthly, although applying the 'add' operator can shorten CPU time, this operator has two basic weaknesses: (1) it is time-consuming to obtain the objective function because all membership functions must be

**Table 4.** Results of varying $\gamma$ values for the 'añd' operator

| | *Data set II (linear nonincreasing membership function)* | | | | *Data set II (triangular membership function)* | | | |
|---|---|---|---|---|---|---|---|---|
| *$\gamma$ value* | *NMc*[a] | *Executing CPU time (H:M:S)* | *No. of EEs* | *Cost of dealing with EE* ($) | *NMc*[a] | *Executing CPU Time (H:M:S)* | *No. of EEs* | *Cost of dealing with EE* ($) |
| 0.1 | (4,5) | (0:32:18) | 3 | 168 200 | (4,5) | (0:06:41) | 3 | 168 200 |
| 0.2 | (4,5) | (0:58:32) | 3 | 168 200 | (4,5) | (0:05:32) | 3 | 168 200 |
| 0.3 | (4,5) | (0:56:49) | 3 | 168 200 | (4,5) | (0:12:12) | 3 | 168 200 |
| 0.4 | (4,5) | (0:58:13) | 3 | 168 200 | (4,5) | (0:14:54) | 3 | 168 200 |
| 0.5 | (4,5) | (0:29:01) | 3 | 168 200 | (4,5) | (0:23:38) | 3 | 168 200 |
| 0.6 | (4,5) | (1:01:57) | 3 | 168 200 | (4,5) | (0:26:25) | 3 | 168 200 |
| 0.7 | (4,5) | (0:51:52) | 3 | 168 200 | (4,5) | (0:32:32) | 3 | 168 200 |
| 0.8 | (4,5) | (0:14:23) | 3 | 168 200 | (4,5) | (0:43:17) | 3 | 168 200 |
| 0.9 | (4,5) | (1:21:45) | 5 | 168 200 | (4,5) | (0:45:53) | 3 | 168 200 |

[a]number of machine types in each cell.

summed in the objective function, and (2) clustering results are affected by constraints because all membership functions of the constraints are put in the objective function. Poor clustering results can be seen from data set III of Table 3; the CF result of case 5 is the worst.

Seventhly, the proposed operator ('add–min') is obviously the most efficient no matter which membership function is used. For example, in data set I, the worst case of using the proposed operator was still better than the best cases of using the other three operators. It also is true that the proposed operator with any membership function consistently performed better than the conventional formulations did, and other operators often required more CPU time than the traditional model.

Finally, the triangular membership function performed better than the linear nonincreasing function for modeling the CF problems for one of two reasons: (1) it sets a lower bound on constraints, and thus the possible range of solutions was narrowed; (2) the triangular function may be more appropriate for representing the fuzzy constraints of the number of machines.

## 7. Sensitivity analysis

To verify the sensitivity of these computational results, two follow-up analyses were performed. The first examines the impact of $\gamma$ on the computational performance of the 'añd' operator. Table 4 summarizes results from data set II. As shown, though CPU times depend upon $\gamma$ values, clustering results are the same. Thus, the selection of $\gamma$ affects CPU time but not the clustering result.

The second analysis concerned the impact of $P_0$ on the computational performance of the proposed operator. The best case was chosen from the first three data sets (case 8), and the $P_0$ value ranged from $\frac{1}{2}P_0$ to $\frac{3}{2}P_0$ for each case. Results are summarized in Table 5. As can be seen, although $P_0$ varied, clustering results from three different data sets remained the same. Moreover, CPU time for the triangular membership function was always shorter than that of the linear nonincreasing membership function. The proposed operator is therefore robust, and the performance of the triangular membership function is better than that of the linear nonincreasing membership function.

## 8. Concluding remarks

This paper proposes an efficient mathematical programming formulation and corresponding FMIP models simultaneously to form manufacturing cells and to minimize the cost of eliminating EEs. Two membership functions with four operators, including a newly proposed operator, were applied and their results compared. Sensitivity analyses also were performed to test the robustness of the fuzzy models and of the proposed operator. From the computational analyses and sensitivity tests, a number of conclusions can be drawn.

1. Choosing appropriate values for the number of cells and the number of machines or parts allowed in each cell (*NM*) is vital to obtain a good solution to CF problems. When the number of parts and the machines needed (i.e., problem size) are relatively small, one may be able to use trial and error to find an acceptable solution without consuming much time. However, once the problem size becomes bigger, trial and error is not an effective approach. In this case, the fuzzy MP is a straightforward and more effective alternative.

2. Not only does the FMP approach provide a better and more flexible way of representing the problem domain than the traditional MP or sequential approaches, it also leads to improved clustering performance. The CPU time required for fuzzy models, however, depends on the operator used. Clearly, the proposed operator always outperforms the traditional model whereas the performance of other operators depends on the data used.

**Table 5.** Impact of different $P_0$ values on the performance of the proposed operator

| | *Data set I* ($P_0$ = $166 000) | | | | *Data set II* ($P_0$ = $121 000) | | | | *Data set III* ($P_0$ = $209 660) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_0$ *value* | *No. of cells* | *Executing CPU time (H:M:S)* | *No. of EEs* | *Cost of dealing with EE* ($) | *No. of cells* | *Executing CPU time (H:M:S)* | *No. of EEs* | *Cost of dealing with EE* ($) | *No. of cells* | *Executing CPU time (H:M:S)* | *No. of EEs* | *Cost of dealing with EE* ($) |
| $\frac{1}{3}P_0$ | 2 | (0:06:21) | 7 | 301 638 | 2 | (0:03:54) | 3 | 168 200 | 3 | (0:13:42) | 2 | 67 777 |
| $\frac{1}{2}P_0$ | 2 | (0:08:08) | 7 | 301 638 | 2 | (0:02:42) | 3 | 168 200 | 3 | (0:11:57) | 2 | 67 777 |
| $P_0$ | 2 | (0:08:28) | 7 | 301 638 | 2 | (0:02:08) | 3 | 168 200 | 3 | (0:10:31) | 2 | 67 777 |
| $\frac{3}{2}P_0$ | 2 | (0:15:46) | 7 | 301 638 | 2 | (0:02:16) | 3 | 168 200 | 3 | (0:15:46) | 2 | 67 777 |
| $0P_0$ (Traditional model) | 3 | (0:21:06) | 9 | 441 221 | 3 | (0:06:28) | 6 | 289 465 | 4 | (0:37:22) | 2 | 209 660 |

3. Performances (in terms of both clustering results and CPU time) of the popularly used 'max–min' operator are worse than those of the other operators, no matter which membership function is used. Thus, a frequently used method may not necessarily be the best.

4. Although the 'añd' operator has the best compensatory property, its performance is worse than that of the proposed operators. The compensatory property therefore is not necessarily the only major factor needing to be considered when operators are selected for the CF problem.

5. The proposed operator always outperformed the other three operators, regardless of the measure used. It also was more stable and robust than the others, regardless of membership function or $P_0$ values.

6. The triangular membership function was a more appropriate membership function for solving the CF problem than the linear nonincreasing type of membership function.

Using MP to model CF problems has several advantages for practical applications. First, with a specific objective in mind, the MP can be used to obtain an optimal solution that aligns with that objective. Secondly, the analytical approaches found in the literature are predominately heuristic [9–11]. Without optimal solutions, it would be difficult to judge heuristics. Still another benefit of using MP is to clarify the embedded logic so that an efficient heuristic can be developed or an improved optimal algorithm created. The use of conventional MP for modeling, however, faces the 'curse of dimension' because as problem size increases, computation intensifies. Furthermore, MP models require their parameters and constraints to be precisely specified. In practice it is very difficult for decision makers to do that. Using FMP for modeling can resolve the second mentioned problem. The dimension problem remains unsolved as yet; therefore, efficient heuristic procedures or computational algorithms need to be developed before one can enjoy the benefits of FMP. Also, although this paper has demonstrated through examples that FMIP can be applied successfully to solve the designated problem, several issues require further study. For instance, the fuzziness considered in this paper was limited to fuzzy constraints. Several other parameters such as demand, processing time, inter-cell transfer cost, and subcontracting can be fuzzified further. Other objectives such as machine utilization and total similarity coefficient also can be considered as a fuzzy equation, or the problem can become a multiple fuzzy linear objective function.

## Acknowledgments

We gratefully acknowledge the valuable comments and suggestions from anonymous referees and Dr Y. Yih, editor of the Manufacturing Systems Modeling Department. This research was partly supported by a research grant from the Iowa Center for Emerging Manufacturing Technology and the National Sciences Council of Taiwan.

## References

[1] Chanas, S., Kolodziejczyk, W. and Machaj, A. (1984) A fuzzy approach to the transportation problem. *Fuzzy Sets and Systems*, **13**, 211–221.

[2] Deporter, E.L. and Ellis, K.P. (1990) Optimization of project networks with goal programming and fuzzy linear programming. *Computers and Industrial Engineering*, **19**, 500–504.

[3] Mjelde, K.M. (1986) Fuzzy resource allocation. *Fuzzy Sets and Systems*, **19**, 239–250.

[4] Sommer, G. and Pollatschek, M.A. (1978) A fuzzy programming approach to an air pollution regulation problem. *European Journal of Operational Research*, **10**, 303–313.

[5] Wiedey, G. and Zimmermann H.J. (1978) Media selection and fuzzy linear programming. *Journal of Operational Research Society*, **31**, 342–349.

[6] Darzentas, J. (1987) On fuzzy location model, in *Optimization Models Using Fuzzy Sets and Possibility Theory*, Kacprzyk, J. and Orlovski, S. A. (eds), D. Reidel, Dordrecht, MA. pp. 328–341.

[7] Evans, G.W., Karwowski, W. and Wilhelm, M.R. (eds) (1989) *Applications of Fuzzy Set Methodologies in Industrial Engineering,* Elsevier, New York.

[8] Karwowski, W. and Evans, G.W. (1986) Fuzzy concepts in production management research: a review. *International Journal of Production Research*, **24**, 129–147.

[9] Chu, C.H. (1989) Cluster analysis in manufacturing cellular formation. *OMEGA*, **17**, 289–295.

[10] Singh, N. (1993) Design of cellular manufacturing systems: an invited review. *European Journal of Operational Research*, **69** (3), 284–291.

[11] Wemmerlöv, U. and Hyer, N.L. (1986) Procedures for the part family/machine group identification problem in cellular manufacturing. *Journal of Operations Manufacturing*, **6**, 125–147.

[12] King, J.R. (1980) Machine–component group formation in group technology. *OMEGA*, **8** (2), 193–199.

[13] Shafer, S.M., Kern, G.M. and Wei, J.C. (1992) A mathematical programming approach for dealing with exceptional elements in cellular manufacturing. *International Journal of Production Research*, **30**, 1029–1036.

[14] Li, J., Ding, Z. and Lei, W. (1986) Fuzzy cluster analysis and fuzzy pattern recognition methods for formation of part families, in *Proceedings of the 14th North American Manufacturing Research Conference (NAMRC)*, Society of Manufacturing Engineering, Dearborn, MI. pp. 558–563.

[15] Xu, H. and Wang, H.P. (1989) Part family formation for GT applications based on fuzzy mathematics. *International Journal of Production Research*, **27**, 1637–1651.

[16] Zhang, C. and Wang, H.P. (1992) Concurrent formation of part families and machine cells based on the fuzzy set theory. *Journal of Manufacturing Systems*, **11**, 61–67.

[17] Chu, C.H. and Hayya, J.C. (1991) A fuzzy clustering approach to manufacturing cell formation. *International Journal of Production Research*, **29**, 1475–1487.

[18] Lai, Y.-J. and Hwang, C.-L. (1992) *Fuzzy Mathematical Programming – Method and Application*, Springer-Verlag, Berlin.

[19] Zimmermann, H.-J. (1991) *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, Boston, MA.

[20] Buckle, J.J. (1989) Solving possibilistic linear programming problems. *Fuzzy Sets and Systems*, **31**, 329–341.

[21] Carlsson, C. and Korhonen P. (1986) A parametric approach to fuzzy linear programming. *Fuzzy Sets and Systems*, **20**, 17–30.
[22] Dubois, D. (1987) Linear programming with fuzzy data, in *Analysis of Fuzzy Information*, vol. III (*Applications in Engineering and Science*), Bezdek, J.C. (ed.), CRC Press, Boca Raton, FL. pp. 241–263.
[23] Inuiguchi, M., Ichihashi, H.I. and Tanaka, H. (1990) Fuzzy programming: a survey of recent development, in *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*, Slowinski, R. and Teghem, J. (eds), Kluwer, Dordrecht, MA. pp. 45–70.
[24] Lai, Y.-J. and Hwang, C.-L. (1992) A new approach to some possibilistic linear programming problems. *Fuzzy Sets and Systems*, **49**, 121–133.
[25] Luhandjula, M.K. (1990) Multiple objective programming problems with possibilistic coefficients. *Fuzzy Sets and Systems*, **34**, 15–31.
[26] Negoita, C.V. and Sularia, M. (1976) On fuzzy mathematical programming and tolerances in planning. *Economic Computer and Economic Cybernetic Studies and Researches*, **1**, 3–14.
[27] Tanaka, H., Ichihashi, H. and Asai, K. (1984) A formulation of fuzzy linear programming problem based on comparison of fuzzy numbers. *Control and Cybernetics*, **13**(3), 185–194.
[28] Verdegay, J.L. (1984) A dual approach to solve the fuzzy linear programming problem. *Fuzzy Sets and Systems*, **14**, 131–141.
[29] Chanas, S. (1983) The use of parametric programming in fuzzy linear programming. *Fuzzy Sets and Systems*, **11**, 243–251.
[30] Fabian, C. and Stoica, M. (1984) Fuzzy integer programming, in *Fuzzy Sets and Decision Analysis*, Zimmermann, H.J., Zadah, L.A. and Gaines, B.R. (eds), TIMS/Studies in the Management Sciences, vol 20, Elsevier Science, Amsterdam pp. 123–131.
[31] Werners, B.M. (1987) An interactive fuzzy programming system. *Fuzzy Sets and Systems*, **23**, 131–147.
[32] Zimmermann, H.-J. (1985) Applications of fuzzy set theory to mathematical programming. *Information Sciences*, **36**, 29–58.
[33] Lai, Y.-J. and Hwang, C.-L. (1992) Interactive fuzzy linear programming. *Fuzzy Sets and Systems*, **45**, 169–183.
[34] Dombi, J. (1991) Membership function as an evaluation. *Fuzzy Sets and Systems*, **35**, 1–21.
[35] Leberling, H. (1981) On finding compromise solutions in multicriteria problems using the fuzzy min-operator. *Fuzzy Sets and Systems*, **6**, 105–118.
[36] Li, R.J. and Lee, E.S. (1991) An exponential membership function for fuzzy multiple objective linear programming. *Computers for Mathematical Applications*, **22**(12), 55–60.
[37] Kim, M.H., Lee, J.H. and Lee, Y.J. (1993) Analysis of fuzzy operators for high quality information retrieval. *Information Processing Letters*, **46**, 252–256.
[38] Werners, B.M. (1988) Aggregation models in mathematical programming, in *Mathematical Models for Decision Support*, vol. 48, Mitra, G. and Greenberg, H.J. (eds), Springer-Verlag, Berlin. pp. 295–305.
[39] Yang, T. and Ignizio, J.P. (1991) Fuzzy programming with nonlinear membership functions: piecewise linear approximation. *Fuzzy Sets and Systems*, **41**, 39–53.
[40] Gongaware, T.A. and Ham, I. (1981) Cluster analysis applications for group technology manufacturing systems, in *Proceedings of the 9th North American Manufacturing Research Conference (NAMRC)*, Society of Manufacturing Engineering, Dearborn, MI. pp. 503–508.
[41] Stanfel, L.E. (1985) Machine clustering for economic production. *Engineering Costs and Production Economics*, **9**, 73–81.
[42] Zadeh, L.A. (1965) Fuzzy sets. *Information and Control*, **8**, 338–353.
[43] Luhandjula, M.K. (1982) Compensatory operators in fuzzy linear programming with multiple objectives. *Fuzzy Sets and Systems*, **8**, 245–282.
[44] Zimmermann, H.-J. (1978) Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, **1**, 45–55.

## Appendix A. Generic forms of selected fuzzy operators

### Notation Used:

The following notation is used in the transformation of fuzzy models:

$U_G(x)$: the membership function of the objective function. Because the linear nonincreasing function is more suitable, $U_G = 1 - (f(x) - Z^0)/P_0$, where $P_0$ is the tolerance value for $f(x)$.

$U_{C_l}(x)$: the membership function of the constraints. Two function types are considered in this paper:
(1) Non-increasing function: $U_{C_l} = 1 - (AX - b)/P_r$, where $P_r$ is the tolerance value for the fuzzy constraints.
(2) Triangular function. The following two formulae are used: $U_{C_l} = 1 - (AX - b)/P_r$ for the upper bound and $U_{C_l} = 1 - (b - AX)/P_r$ for the lower bound.

**1. 'Max–min' operator:**
Proposed by Zadeh [42], this operator uses 'min' to define the intersection of an aggregated rule, i.e.,

$$U_D(X) = U_G(X) \wedge U_{C_l}(X) = U_G(X) \cap U_{C_l}(X). \quad \text{(A1)}$$

The aggregated membership function becomes

$$U_D(X) = \min[U_G(X), U_{C_l}(X)], \quad \forall j. \quad \text{(A2)}$$

To find the maximum value of $U_D(x)$, the model can be defined as

$$\text{Max } \lambda \quad \text{(A3)}$$

subject to

$$\lambda \leq U_G(x); \quad \text{(A4)}$$

$$\lambda \leq U_{C_l}(X), \quad \forall j; \quad \text{(A5)}$$

$$0 \leq \lambda \leq 1. \quad \text{(A6)}$$

**2. 'Min-bound-sum' operator:**
Proposed by Luhandjula [43], this operator uses (A7) as the aggregation rule:

$$U_D = \gamma \times U_{G \cap C_l} + (1 - \gamma) \times U_{G \cup C_l}. \quad \text{(A7)}$$

It then uses the 'min' operation to represent the intersection and the 'bounded-sum' operation to represent the union; the aggregated membership function thus becomes

$$U_D = \gamma \min U_S + (1 - \gamma) \min(1, \sum_{S=0}^{t} U_S). \quad \text{(A8)}$$

The equivalent form after using the operator is

$$\text{Max } \gamma\lambda + (1 - \gamma)u \quad \text{(A9)}$$

subject to

$$\lambda \leq U_G(x); \tag{A10}$$

$$\lambda \leq U_{C_l}(x), \quad \forall j; \tag{A11}$$

$$u \leq 1; \tag{A12}$$

$$u \leq U_G + \sum_l U_{C_l}(x); \tag{A13}$$

$$0 \leq \lambda \leq 1; \quad \gamma \leq 1. \tag{A14}$$

**3. 'Compensatory and' operator:**
This operator uses the same aggregation rule as the 'min-bound-sum' operator, but it then uses the 'min' operation to represent the intersection and the 'max' operation to represent the union. Thus the result of $U_D(x)$ becomes

$$U_D = \gamma \min U_S + (1-\gamma) \max U_S. \tag{A15}$$

The crisp equivalent model after using this operator is:

$$\text{Max } \gamma\lambda + (1-\gamma)u \tag{A16}$$

subject to

$$\lambda \leq U_G(x); \tag{A17}$$

$$\lambda \leq U_{C_l}(x), \quad \forall j; \tag{A18}$$

$$u \leq U_G + MY_0; \tag{A19}$$

$$u \leq U_{C_l} + MY_l, \quad \forall l; \tag{A20}$$

$$\sum_{S=0}^{t} Y_S \leq t; \tag{A21}$$

$$0 \leq \lambda \leq 1; \quad \gamma < 1, Y_S = 0, 1; \quad M \text{ is a large number.} \tag{A22}$$

**4. 'Añd' operator:**
To address the deficiencies of aggregation rule (A7), Werners [38] suggested modification of the membership function of the resulting fuzzy set as

$$U_D = \gamma \min U_S + (1-\gamma)/(t+1) \times \sum_{S=0}^{t} U_S. \tag{A23}$$

The equivalent model after this operator is used becomes

$$\text{Max } \lambda + (1-r)/(t+1) \times \sum_{S=0}^{t} \alpha_S \tag{A24}$$

subject to

$$\lambda + \alpha_0 \leq U_G(x); \tag{A25}$$

$$\lambda + \alpha_l \leq U_{C_l}(x), \quad \forall l; \tag{A26}$$

$$\lambda + \alpha_s \leq 1, \quad \forall s; \tag{A27}$$

$$\lambda, \alpha_s \geq 0; \quad \delta < 1. \tag{A28}$$

**5. 'Add' operator:**
Proposed by Sommer [5], this operator first modifies the fuzzy constraints $AX \gtrsim b$ into $AX - S_l \leq b$ and $S_l \gtrsim 0$. As a result, the membership function of the constraint set becomes $U'_{C_l} = 1 - S_l/P_r$. Therefore all membership functions are summed up in the objective functions; i.e.

$$\text{Max } U_G + \sum_{l=1}^{t} U'_{C_l}. \tag{A29}$$

The equivalent LP model can be obtained as

$$\text{Min } (f(x) - Z^0)/P_0 + \sum_{l=1}^{t} S_l/P_r \tag{A30}$$

subject to

$$AX - S_l \leq b, \quad \forall j; \tag{A31}$$

$$S_l \leq P_r, \quad \forall j. \tag{A32}$$

## Biographies

Chang-Chun Tsai is an Associate Professor in the production management group of the Department of Industrial Management Science at National Cheng Kung University (NCKU) in Taiwan. He received his B.S. and M.B.A. from NCKU, an M.S. in industrial engineering from the University of Wisconsin at Madison, and a Ph.D. in Industrial and Manufacturing Systems Engineering at Iowa State University. His current research focuses on the design and operation of cellular manufacturing systems, group technology, and applications of fuzzy set theory.

Chao-Hsien Chu is an Associate Professor of Operations Management at Iowa State University. He received a Ph.D. in Business Administration from the Pennsylvania State University. Dr Chu has served on the faculties of Baruch College (CUNY), Pennsylvania State University and Tatung Institute of Technology (Taiwan) and was a visiting professor of Management Sciences at University of Tsukuba, Japan. Dr Chu's research interests are in cellular manufacturing, Japanese manufacturing management techniques, and applied artificial intelligence: expert systems, fuzzy logic, genetic algorithms, and neural networks for manufacturing applications. He has published articles in *Journal of Operations Management, International Journal of Production Research, OMEGA, International Journal of Operations and Production Management, Decision Support Systems*, among others. He is on the Editorial Review Board of *International Journal of Production and Operations Management* and *Journal of End User Computing*.

Thomas A. Barta is a Professor of Industrial and Manufacturing Systems Engineering at Iowa State University. His research interests are in simulation and knowledge-based systems. He is a registered professional engineer, a senior member of IIE, and a member of ASEE, AAAI, IEEE, MTM Association, and SCS.