Fuzzy extensions to Integer Programming models of cell-formation problems in machine scheduling

G. Papaioannou[+], **J.M. Wilson**[*]

Management Science and Information Systems Group,

Business School, Loughborough University,

Loughborough, LE11 3TU, UK

[+]Corresponding author. Tel: +44(0)1509228842; fax: +44(0)1509223960

Email addresses: g.papaioannou@lboro.ac.uk (G. Papaioannou),

j.m.wilson@lboro.ac.uk (J.M. Wilson)

Author's final version– To cite this article: Papaioannou, G., and Wilson, J.M., "Fuzzy extensions to Integer Programming models of cell-formation problems in machine scheduling", Ann Oper Res (2009) 166: 163181.

DOI: 10.1007/s10479-008-0423-1

The original publication is available at www.springerlink.com

[*]The authors would like to thank Prof. J. Liu and Dr M. Sonmez for useful discussions and the anonymous referees for their suggestions that improved the exposition.

Abstract

Cell formation has received much attention from academicians and practitioners because of its strategic importance to modern manufacturing practices. Existing research on cell formation problems using integer programming (IP) has achieved the target of solving problems that simultaneously optimise: (a) cell formation (b) machine-cell allocation, and (c) part-machine allocation.

This paper will present extensions of the IP model where part-machine assignment and cell formation are addressed simultaneously, and also a significant number of constraints together with an enhanced objective function are considered. The main study examines the integration of inter-cell movements of parts and machine set-up costs within the objective function, and also the combination of machine set-up costs associated with parts revisiting a cell when part machine operation sequence is taken into account. The latter feature incorporates a key set of constraints which identify the number of times a part travels back to a cell for a later machine operation.

Due to two main drawbacks of IP modelling for cell formation, i.e. (a) only one objective function can be involved and (b) the decision maker is required to specify precisely goals and constraints, fuzzy elements like fuzzy constraints and fuzzy goals will be considered in the proposed model.

Overall the paper will not only include an extended and enhanced integer programming model for assessing the performance of cell formation, but also perform a rigorous study of fuzzy integer programming and demonstrate the feasibility of achieving better and faster clustering results using fuzzy theory.

Keywords: cellular manufacturing system; machine operation sequence; integer programming; uncertainty; fuzzy models

Notation

• Index Set

i	machine type index	$i = 1, \dots, M$
j	part index	$j = 1, \dots, P$
q	cell index	$q = 1, \dots, C$
k	machine instance index	$k = 1, \dots, KM$
z, r	machine operation indices	$z, r = 1, \dots, ZOPER$
l	membership functions index	$l = 0, \dots, MF$
t	additional cell index for min-bounded sum operator	$t = 1, \dots, C$

• Input Parameters

E_{MIN}	minimum number of machines allowed in a cell
E_{MAX}	maximum number of machines allowed in a cell
$NCELLS$	number of cells in the system
$M_{j,q}$	cost of allocating part j in cell q
A_j	cost for part j traveling back to an already visited cell
$UTIL_{i,j}$	utilisation of machine i by part j
$KTYPES_i$	number of machines instances for each machine type i
$SETUP_{i,j}$	set-up cost of machine i needed to process part j
$ZOPER$	number of machine operations
$ZTYPES_j$	number of different operations (machine types) required by part j
$L_{j,z}$	for part j the machine used for the z^{th} machine operation in sequence
$UTILMIN$	minimum amount of utilisation in $UTIL_{i,j}$ matrix
$UTILMAX$	largest amount of machine utilisation used
γ	parameter for fuzzy modelling

- **Decision Variables**

$x_{i,j,q}$	amount of processing by machines of type i for part j in cell q
$y_{i,k,q}$	=1 if k^{th} machine instance of type i is assigned to cell q, 0 otherwise
$w_{j,q}$	=1 if part j is processed in cell q, 0 otherwise
v_q	=1 if cell q is formed, 0 otherwise
$s_{i,j,q}$	integer number of machines of type i that will be used by part j in cell q
$extra_{q,j,L_{j,z}}$	=1 if after the z^{th} operation of part j in cell q the part leaves cell q but returns later, 0 otherwise
$xx_{L_{j,z},j,q}$	=1 if part j is processed in cell q for z^{th} machine operation, 0 otherwise
λ	minimum value of all membership functions
a_l	extra variable used for '$\tilde{and}$' operator
ω	extra variable used for '$\tilde{and}$' operator
u_t	extra variable used for 'min-bounded sum' operator

Cellular manufacturing has been a prosperous research area for the last three decades and received a lot of attention from academicians because of its strategic importance to 'modern' industrial and manufacturing areas. The design of cellular manufacturing systems has been called Cell Formation (CF). CF is the process of grouping parts with similar design features or processing requirements into parts families and machines into machine cells. Extensive reviews of CF problems can be found, for example, in (Wemmerlov and Hyer (1986), Wemmerlov and Hyer (1989), Singh (1993)).

Mathematical Programming formulations involve a wide range of manufacturing data. Several types of integer programming formulations have been proposed over the past years. In most of these formulations parts are assigned to individual machines and individual machines are allocated into cells simultaneously. A number of major results in the literature have as a main criterion the minimisation of intercellular movements and have been discussed by: Purcheck (1975), Kusiak (1987), Kusiak and Heragu (1987), Wei and Gaither (1990), Sankaran (1990), Zhu, Heady and Reiners (1995), Selim, Askin and Vakharia (1998), Wang (1998), Foulds, French and Wilson (2006). However, none of the studies attempted to handle the minimisation of intercellular movements when machine set up costs and the part machine operation sequence is taken into account.

Applying mathematical programming models to solve the cellular manufacturing problem is a challenging task as decision makers find it difficult to specify goals and constraints because some of the parameters involved cannot be estimated precisely. A number of researchers dealt with the formation of parts families (Xu and Wang (1989), Gill and Bector (1997), Ben-Arien and Triantaphyllou (1992)), whereas others considered fuzzy data within traditional approaches (Chu and Hayya (1991), Zhang and Wang (1992), Ravichandran and Rao (2001)). Moreover, other authors (Tsai, Chu and Barta (1997)) considered the application of fuzzy models for measuring uncertainty via a new proposed operator minimising the cost of exceptional elements. None of these studies have attempted to deal with fuzzy goals and constraints when the part machine operation sequence is taken into account and also to measure the performance of a number of operators towards the performance of a deterministic and rigidly defined CF model when a number of system attributes are taken into account .

The aim of the paper is threefold: first, to produce a comprehensive integer programming (IP) model able to assign parts to machines and machines to cells simultaneously and to minimise the cost of intercellular movements, the set-up cost of machines and the cost of parts revisiting a cell for a later machine operation; second, to use fuzzy IP (FIP) to model CF via a number of fuzzy operators and membership functions addressing the uncertainty of certain elements; third, to assess the performance of deterministic versus fuzzy models as the scale of the problem increases.

1 Deterministic model

The proposed model is an extension of a model developed by Foulds, French and Wilson (2006), the first to simultaneously optimise cell formation, machine-cell allocation and part machine allocation; where costs of setting up machines for parts to be processed and *part machine operation sequences* are now taken into account. Each part has a machine operation sequence which may lead to several intercell movements. The objective function is enhanced to minimise simultaneously the number of distinct allocations of parts to cells, the set-up costs of machines and also the number of times a part travels back to an already visited cell in order to be processed.

The complete formulation of the mathematical programming model is shown below:

$$Min \ \ (\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q} \times w_{j,q}) + \sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j} \times \sum_{q=1}^{C} s_{i,j,q}) + \sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j \times extra_{q,j,i})) \tag{1}$$

subject to

$$\sum_{q=1}^{C} y_{i,k,q} = 1 \ \ \forall \ i,k \tag{2}$$

$$\sum_{q=1}^{C} x_{i,j,q} = UTIL_{i,j} \ \ \forall \ i,j \tag{3}$$

$$x_{i,j,q} \leq s_{i,j,q} \ \ \forall \ i,j,q \tag{4}$$

$$x_{i,j,q} \geq UTILMIN \times s_{i,j,q} \ \ \forall \ i,j,q \tag{5}$$

$$\sum_{j=1}^{C} x_{i,j,q} \leq \sum_{k=1}^{KM} y_{i,k,q} \ \ \forall \ i,q \tag{6}$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} \leq v_q \times E_{MAX} \ \ \forall \ q \tag{7}$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} \geq v_q \times E_{MIN} \ \ \forall \ q \tag{8}$$

$$v_{q+1} \leq v_q \ \ \forall \ q \tag{9}$$

$$\sum_{q=1}^{C} q \times y_{i,k,q} \leq \sum_{q=1}^{C} q \times y_{i,k+1,q} \ \ \forall \ i,k \tag{10}$$

$$x_{i,j,q} \leq UTIL_{i,j} \times w_{j,q} \quad \forall\ i,j,q \tag{11}$$

$$x_{i,j,q} \leq UTILMAX \times xx_{i,j,q} \quad \forall\ i,j,q \tag{12}$$

$$x_{i,j,q} \geq UTILMIN \times xx_{i,j,q} \quad \forall\ i,j,q \tag{13}$$

$$xx_{L_{j,z},j,q} + xx_{L_{j,r},j,q} - \sum_{zz=z+1}^{r-1} xx_{L_{j,zz},j,q} \leq extra_{q,j,L_{j,z}} + 1 \quad \forall\ q,j,z,r \tag{14}$$

$$y_{i,k,q},\ v_q,\ w_{j,q},\ extra_{q,j,i},\ xx_{i,j,q} = 0\ or\ 1; \quad 0 \leq x_{i,j,q} \leq 1; \quad s_{i,j,q}\ \ integer \tag{15}$$

Objective function, (1), combines a mixture of the following three requirements to:

- *Minimise number of distinct cells used by each part*
- *Minimise set-up costs when allocating parts to machines*
- *Minimise number of times a part revisits a cell for a later machine operation.*

Constraint (2) ensures that k^{th} machine of type i must be assigned to exactly one cell. Constraint (3) handles the requirements for processing part j on machine i: the number of machines or fraction thereof required to process part j in cell q is equal to the utilisation of machine i required to process part j in cell q. Constraint (4) ensures that the total number of machines (in terms of machine utilisation) required to process part j in cell q is less than or equal to the integer number of machines of type i that will used by part j in cell q. Constraint (5) forces variable s to get the value 0 whenever x variable is *zero* and is not strictly necessary but aids branch and bound. Constraint (6) ensures that the total number of machines of type i used in cell q should be less than or equal to the number of machine instances of type i assigned to cell q. Constraints (7), (8) limit the number of machines in each cell. Constraint (9) ensures that cells are formed in successive numerical order. Constraint (10) assigns duplicate machines when needed to lower numbered cells in successive numerical order. Constraints (9) and (10) are included to eliminate certain symmetries. Constraint (11) picks which cells are used by parts. Both constraints (12) and (13) ensure that whenever a part uses a machine or a fraction thereof ($x_{i,j,q} > 0.0$) variable $xx_{i,j,q}$ is assigned the value 1, otherwise it is assigned to 0. The *key constraint* (14) picks out the number of times a part travels back to a cell for a later machine operation. A part j, whose z^{th} machine operation is processed in cell q, could revisit the cell q for a later machine operation i.e. $(z+r)^{th}$, only when the 2^{nd} machine operation $(z+1)$ is not processed within the same cell. In this case the value of $extra_{q,j,L_{j,z}}$ is assigned to 1.

It is assumed that the machine utilisation for processing a part j is equal to the processing time of part j in machine i. For this reason no time element is considered for the current model. Moreover, for each machine of type i only a maximum of a unity of its capacity can be spent on processing a part j.

Figure 1 provides a visual representation of the CF model which was solved by running `Xpress-MP` mathematical programming software. The data used are presented in Table 1. Each item in the square boxes, i.e. M_i^k, denotes the instance k of machine of type i currently used within a cell. Also the elements in the arrows have a certain explanation, e.g. 2(0.3) describes that part 2 is using 0.3 capacity units of the machine that the arrow is pointing at. According to this figure all parts follow a certain route and it is worth noting that the dotted line represents the route of part 2 which revisits a cell in order to be fully processed.

2 Fuzzification for CF

In the aforementioned deterministic model it is assumed that both the objective function and all related constraints can be defined precisely. In practice it is very difficult for the decision maker to specify the exact goals and constraints when modelling the problem. Tools for experimenting with changes in both coefficients and constraints by doing either sensitivity or postoptimality analysis are well established for linear programming (LP) models. For IP these tools are less well developed because the absence of continuity precludes the natural extension of these tools from LP to IP, however certain experimentation is still possible. In what follows the authors consider aspects of fuzzification within CF problems.

Although in the current model there are a lot of elements that could be fuzzy such as set up costs and utilisation amounts, the authors consider the fuzziness concept on the number of machines included in cells. The latter is chosen based upon the model's main operation which is the creation of cells with a specific number of machines in them. The resulting analysis can thus be considered as just one example of analysing fuzziness out of a range of possible elements that could be made fuzzy. Thus the analysis is to some extent exploratory and intended to show the possibilities of extending CF problems to incorporate more realism. The analysis will show ways in which a range of configurations can be offered to decision makers and this range could be extended by introducing fuzziness into other parameters, of which the number of machines in a cell is one.

In the deterministic model, equation (7), the maximum, E_{MAX}, number of machines allowed in each of the cells has been precisely specified. What will happen if this maximum number of machines varies between a range, and thus is uncertain? The range assumed is defined by the *upper bound* of the maximum number of machines and the *lower bound* of the maximum number of machines. The following two fuzzy equations describe the situation where the maximum number of machines takes values between a range. Depending upon the *type of the membership function* used, either both equations 16, 17 are utilised or only one is employed as will be seen later.

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} \lessapprox v_q \times E_{MAX} \quad \forall\ q \tag{16}$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} \gtrapprox v_q \times E_{MAX} \quad \forall\ q \tag{17}$$

The objective function, equation (1), can also be be fuzzified (Werners (1987) and Lai and Hwang (1992)) as follows:

$$\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q} \times w_{j,q}) + \sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j} \times \sum_{q=1}^{C} s_{i,j,q}) + \sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j \times extra_{q,j,i}) \lessapprox D^0 = D^1 - P_0 \tag{18}$$

where D^0 is the feasible value of the best goal, which can be obtained by solving the deterministic model with the total number of machines instances in the system. D^1 is the feasible value of the worst goal which can be obtained by solving the deterministic model with the minimum number of machines (E_{MIN}).

In order to transform the fuzzy model to its equivalent traditional formulation three tolerance values, P_0, P_{R1} and P_{R2} are used within the objective function (18) and constraints (16) and (17) respectively. The value for parameter P_0 can be determined as the value equal to $(D^1 - D^0)$, whereas the parameter values P_{R1} and P_{R2} depend upon the decision maker's opinion.

For the fuzzy incorporation within the current model two membership functions, linear non-increasing (Wiedey and Zimmermann (1978), Zimmermann (1991)) and triangular (Yang and Ignizio (1991)) will be considered. For the transformation of the fuzzy formulation to MP formulation, fuzzy aggregation operators will be used. Table 3 summarizes a number of operators that have been applied before in fuzzy mathematical programming. The first three operators have linear forms after transformation whereas the last two are non-linear and thus more difficult to handle. Moreover, all operators except the 'min' classical operator allow some type of compensation; either a positive or negative (Kim, Lee and Lee (1993)). For example, the 'fuzzy and' operator (Werners (1988)) combines the minimum and maximum operator with the arithmetic mean and allows compensation between the membership values of the aggregated sets leading to very good results with respect to empirical data (**?**). For the current study 'min', 'fuzzy and' and 'min-bounded sum' operators will be examined. Last but not least, objectives and constraints are treated equally and there is no difference between them, therefore their relationship is fully *symmetric* (Zimmermann (1991), Lai and Hwang (1992)).

Before continuing with the fuzzy model formulation, an illustration of the membership functions involved is given in Figure 2. The membership function for the objective function (18) is linear non-increasing (no other type of membership function can be assumed) and can be seen in Figure 2(a). Considering the fuzzy constraints, if constraint (16) is added only, the membership function is linear non-increasing and can be seen in Figure 2(b); However, if both constraints, (16), (17), are involved the membership function is triangular as presented in Figure (2(c)).

The mathematical presentation for the linear non-increasing membership function of objective (18) is as follows:

$$\mu_0(x)=\begin{cases} 1, & if \;\; c^T x \leq D^0 \\ 1-\frac{c^T x - D^0}{P_0}, & if \;\; D^0 \leq c^T x \leq D^0+P_0 \\ 0, & if \;\; c^T x > D^0+P_0 \end{cases}$$

In a similar way the rest of the membership functions can be expressed mathematically.

3 Defuzzification for CF

In this section a number of combinations of membership functions and aggregation operators are performed and the fuzzy CF model is converted into an IP model.

3.1 Min operator

• Linear non-increasing membership function

Applying the 'min' operator the following formulation is obtained:

$$Max \;\; \lambda \tag{19}$$

subject to

$$\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q}\times w_{j,q})+\sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j}\times\sum_{q=1}^{C}s_{i,j,q})+\sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j\times extra_{q,j,i})+\lambda P_0 \leq D^0+P_0 \tag{20}$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q}+\lambda P_{R1} \leq v_q \times E_{MAX}+P_{R1} \;\; \forall \, q \tag{21}$$

$$0 \leq \lambda \leq 1 \tag{22}$$

Besides those equations noted above, equations (2)-(6) and (8)-(15) are added here as well. From equation (21), the number of machines allowed in a cell ranges from E_{MIN} (crisply defined by the decision maker as in the deterministic model) to $v_q \times E_{MAX}+P_{R1}$.

- **Triangular membership function**

The equivalent MP formulation consists of equations (19) to (22) and equation (23) below.

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} - \lambda P_{R2} \geq v_q \times E_{MAX} - R_{R2} \ \ \forall\ q \tag{23}$$

It also includes crisp constraints (2)-(6) and (9)-(15).

3.2 Fuzzy and ($\tilde{\text{and}}$) operator

- **Linear non-increasing membership function**

The equivalent MP formulation is as follows:

$$Max \ \ \omega + (1-\gamma) \times \frac{1}{c+1}\sum_{l=0}^{MF} a_l \tag{24}$$

subject to

$$\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q} \times w_{j,q}) + \sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j} \times \sum_{q=1}^{C} s_{i,j,q}) + \sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j \times extra_{q,j,i}) +$$

$$\omega \times P_0 + a_0 \times P_0 \leq D^0 + P_0 \tag{25}$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} + \omega \times P_{R1} + a_1 \times P_{R1} \leq v_q \times E_{MAX} + P_{R1} \ \ \forall\ q \tag{26}$$

$$\omega + a_l \leq 1 \tag{27}$$

$$\omega \leq 1, \ a_l \geq 0, \ \gamma < 1 \tag{28}$$

Also crisp constraints (2)-(6) and (8)-(15) are included in this model as well.

- **Triangular membership function**

The objective function for this formulation is the same as equation (24) and only one new constraint is added as follows:

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} - \omega \times P_{R2} - a_2 \times P_{R2} \geq v_q \times E_{MAX} - R_{R2} \ \ \forall\ q \tag{29}$$

Also constraints (25)-(28), (2)-(6) and (9)-(15) are included here as well.

3.3 Min-bounded sum operator

- **Linear non-increasing membership function**

The equivalent LP formulation is as follows:

$$Max \ \ \gamma \times \omega + (1-\gamma)\sum_{t=1}^{C} u_t \tag{30}$$

subject to

$$\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q} \times w_{j,q}) + \sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j} \times \sum_{q=1}^{C} s_{i,j,q}) + \sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j \times extra_{q,j,i}) + \omega \times P_0 \leq D^0 + P_0 \quad (31)$$

$$u_t \leq [\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q} \times w_{j,q}) + \sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j} \times \sum_{q=1}^{C} s_{i,j,q}) + \sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j \times extra_{q,j,i}) - D^0]/P_0 + [\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,t} - v_t \times E_{MAX}]/R_{R1} \quad \forall\, t \quad (32)$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} + \omega \times P_{R1} \leq v_q \times E_{MAX} + P_{R1} \quad \forall\, q \quad (33)$$

$$u_t \leq 1 \quad \forall\, t \quad (34)$$

$$0 \leq \omega \leq 1,\ \gamma < 1 \quad (35)$$

Also equations (2)-(6) and (8)-(15) are included in the above formulation.

- **Triangular membership function**

For this case one more membership function is included, therefore equation (32) is differently formulated (see equation (36)) and constraint (37) is added.

$$u_t \leq [\sum_{j=1}^{P}\sum_{q=1}^{C}(M_{j,q} \times w_{j,q}) + \sum_{i=1}^{M}\sum_{j=1}^{P}(SETUP_{i,j} \times \sum_{q=1}^{C} s_{i,j,q}) + \sum_{q=1}^{C}\sum_{j=1}^{P}\sum_{i=1}^{M}(A_j \times extra_{q,j,i}) - D^0]/P_0 + [\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,t} - v_t \times E_{MAX}]/R_{R1} + [v_t \times E_{MAX} - \sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,t}]/R_{R2} \quad \forall\, t \quad (36)$$

$$\sum_{i=1}^{M}\sum_{k=1}^{KM} y_{i,k,q} - \omega \times P_{R2} \geq v_q \times E_{MAX} - R_{R2} \quad \forall\, q \quad (37)$$

Objective function (30) and constraints (31), (33)-(35) are preserved and added here. Also equations (2)-(6) and (9)-(15) are included in this formulation.

4 Models Assessment

To assess the results of each of the aggregation operators within the CF model four data sets are used. The first data set (DS 1) was created by adopting numerical values from Foulds, French and Wilson (2006) and randomly generated some more needing to meet the requirements of the current model. The numerical values of DS 1 are presented in Table 2. The remaining data sets were randomly generated by a computer program developed with `MatLab`. The sets were chosen in order to provide variety of parameter values in the models.

The first three data sets have the same size as DS 1 in terms of the number of machine types and parts used. DS 4 has a different size with both machines types and parts equal to nine. DS 2 is similar to DS 1 because of the total number of machine instances, but utilisation and set-up cost matrices differ as they were randomly generated. DS 3 has a greater number of machine instances, therefore its computation is more intense. For DS 4, most of the input parameters have dissimilar values with the rest of the data sets.

Last but not least the maximum number of machines allowed in a cell takes different values depending upon the total number of machine instances in each of the data sets and the number of cells allowed to be created. When fuzzy models are used, fuzzy intervals are used to define those numbers; upper bound of maximum number of machines and lower bound of maximum number of machines allowed in a system is the interval specified when triangular membership function is used. Sometimes depending upon the size of the problem, the value of the lower bound of maximum number of machines becomes equal to the minimum number of machines rigidly defined by the decision maker. For the linear non-increasing membership function the upper bound of the maximum number of machines is also specified. The minimum number of machines is explicitly acquired in order to avoid creating a single cell with all machines in it.

5 Performance of Models

All models were solved by running `Xpress-MP` package on a PC with Intel 3.20 GHz Pentium 4 Processor. Clustering performances were measured in terms of number of cells created, distinct cells used by each part, number of times a part visits a cell for later machine operation, CPU time and total cost of dealing with intercell movements and set-up costs.

Table 5 summarises the computational results for all four data sets. For each data set, six cases for the fuzzy models (three operators and two membership functions), plus two cases for the proposed deterministic model are examined. The final case (case 8) where the deterministic model is utilised it examines a possible impact on the solution quality when the maximum number of machines is relaxed when the upper bound is specified.

From Table 5, it is observed that the CPU times for both 'fuzzy and' and 'min-bounded sum' operators vary but they differ marginally, whereas 'min' operator requires more execution time. However, the clustering results for all operators are different and this can be verified by the number of cells created, the distinct number of cells used by each part and the number of later revisits of parts to an already visited cell. The performance of 'min' operator is not promising especially when a linear non-increasing membership function is employed as it requires the longest time to process (especially when problem size increases i.e DS's 3, 4), and the clustering results are not better than those produced from the other two operators. The performance of the latter slightly improves when a triangular membership function is assumed. Overall and as can be seen from the tables the computation time for each problem is very long for the 'min' operator. For two of the data sets the algorithm fails to converge after excessive computational time.

The 'fuzzy and' operator arrives at acceptable clustering results and the required CPU time is quite low for either non-increasing or triangular membership functions when smaller data sets are utilised. It is worth noting that the 'min-bounded sum' operator results in significant CPU time reduction when larger data sets are used (i.e. DS 3, DS 4) compared to the other two operators but it has two weaknesses: a) it is time consuming to obtain one of its constraints, in which all the membership functions must be summed up and, b) clustering results seem to be affected because all the membership functions of the constraints are added within one constraint when formulation takes place. It is particularly encouraging that on the data sets considered, for the linear non-increasing version of the 'min-bounded sum' operator only small amounts of CPU time are required and these are substantially lower that those required for the deterministic versions of the problem. Constraints (31), (32) may be helping to reduce integrality gap and aid convergence.

Moreover, for both 'fuzzy and' and 'min-bounded sum' operators experiments were carried out in order to determine a proper γ parameter value (γ parameter can take values between the range [0.0, 0.9]). Table 4 summarises the results of using DS 1 for the 'min-bounded sum' operator. According to the table the best γ values for cases 5, 6 are 0.4 and 0.5 respectively. It has been decided by the researchers that the best γ value should be determined by the lowest executing CPU time and not the total cost. The latter varies in very small ranges because of the machine set-up costs configuration. For determining the remaining best γ values a similar process was followed.

For considering the combinatorial explosion on the CPU times of all operators used a comparison between DS 1 (the smallest data set recorded in Table 5) and DS 4 (the largest data set) will now be made. As the size of the problem increases computation intensifies especially for the 'min' operator where no compensation between the membership functions of the aggregated sets is performed. For the 'fuzzy and' operator the CPU time increases significantly as the size of the problem becomes larger especially when a linear non-increasing membership function is employed. For the 'min-bounded sum' operator things are different as CPU time is kept low regardless how big the problem is. Although CPU times are very promising for the latter, clustering results are not so good (see DS 4) compared with the 'fuzzy and' operator. Therefore, the 'fuzzy and' operator, when a triangular membership function is employed, can be characterised as a good operator with promising clustering results and reasonable CPU times even when data sets become larger.

For cases 7 and 8, the deterministic model is utilised and it can be observed that more effort with trial and error is needed till an appropriate number of maximum number of machines allowed in a cell is decided for obtaining a good solution. Also as the size of the problem increases (see DS 3 or DS 4) the CPU time increases significantly. However, employing a good operator (in terms of clustering results and CPU time) the use of the fuzzy model is quite straightforward. It can flexibly adjust the number of machines, given a tolerance value, thus avoiding time consuming trial and error.

6 Summary and Conclusions

This paper proposes an efficient integer mathematical programming formulation where parts are assigned to machines and machines to cells simultaneously, when an enhanced objective function is taken into account. Fuzzy integer programming models are also developed to form manufacturing cells when uncertainty is taken into account. Two membership functions with three operators were applied and the results compared. From all the computational analyses the following conclusions can be drawn:

- It is a difficult and a very important issue for the decision maker to choose the appropriate number of machines allowed in a cell. If the size of the problem is small then there are some chances with trial and error that a good solution may be obtained. However, once the problems size increases, trial and error is not effective. Thus, fuzzy mathematical programming is a more promising alternative methodology.

- The fuzzy mathematical programming provides a more flexible way of representing the problem and it leads in most cases to better clustering results, especially when the 'fuzzy and' operator was employed. The CPU time depends upon the operator used. Although the 'min' operator is the most frequently used method, it did not perform well especially when a linear non-increasing membership function was used. The 'min-bounded sum' operator shortened the CPU significantly and outperformed the rest of the operators as well as the deterministic model, especially when a bigger problem was considered.

- The time advantage of using a fuzzy model rather than a deterministic model becomes significant once a larger scale model is used and the tolerance value of the constraints becomes bigger.

- The triangular membership function performed better than the linear non-increasing membership function for the CF problem and it seems to be more appropriate for modelling the constraints when a number of machines, either maximum or minimum are involved.

For the purpose of this study small to medium sized data sets were used because with mathematical programming as the size of the problem increases computation intensifies. It will be possible to solve some larger sized problems, but ultimately a limit will be reached when computation times become excessive. For this reason the development of a heuristic algorithm will be the subject of future research. In this way the performance of fuzzy models could be compared against heuristic outputs even when large data sets are utilised. Also the specifications of the problem can be enhanced by considering a multiple fuzzy linear objective function.

References

Ben-Arieh, D. and E. Triantaphyllou. (1992). Quantifying data for group technology with weighted fuzzy features, International Journal of Production Research, 30, 1285-1299.

Chu, C.H. and J.C. Hayya. (1991). A fuzzy clustering approach to manufacturing cell formation, International Journal of Production Research, 29, 1457-1487.

Foulds, L.R., French, A.P. and J.M. Wilson. (2006). The sustainable cell formation problem: manufacturing cell creation with machine modification costs, Computers & Operations Research, 33, 1010-1032.

Gill, B. and R. Bector. (1997). A fuzzy linguistic approach to data quantification and construction of distance measures for the part family formation problem, International Journal of Production Research, 35, 2565-2578.

Kim, M.H., Lee, J.H. and Y.J. Lee. (1993). Analysis of fuzzy operators for high quality retrieval, Information Processing Letters, 46, 252-256.

Kusiak, A. (1987). The generalized group technology concept, International Journal of Production Research, 25, 561-569.

Kusiak, A. and S.S. Heragu. (1987). Group Technology, Computers in Industry, 9, 83-91.

Lai, Y.-J. and C.-L. Hwang. (1992). Fuzzy Mathematical Programming - Methods and Applications, Springer-Verlag, Berlin.

Lai, Y.-J. and C.-L. Hwang. (1992). Interactive fuzzy linear programming, Fuzzy Sets and Systems, 45, 169-183.

Luhandjula, M.K. (1982). Compensatory operators in fuzzy linear programming with multiple objectives, Fuzzy Sets and Systems, 8, 245-252.

Mjelde, K.M. (1986). Fuzzy resource allocation, Fuzzy Sets and Systems, 19, 239-250.

Purcheck, G.F. (1975). Programming method for combinatorial group of an incomplete power set, Journal of Cybernetics, 5, 51-76.

Ravichandran, K.S. and K.C.S. Rao. (2001). A new approach to fuzzy part-family formation in cellular manufacturing systems, International Journal of Advanced Manufacturing Technology, 18, 591-597.

Sankaran, S. (1990). Multiple objective decision making approach to cell formation - A goal programming model, International Journal of Production Research, 13, 71-81.

Selim, M.S., Askin, R.G. and A.J. Vakharia. (1998). Cell Formation in Group Technology: Review Evaluation and Directions for Future Research, Computers and Industrial Engineering, 34, 3-20.

Singh, N. (1993). Design of Cellular Manufacturing Systems: An Invited Review, European Journal of Operational Research, 69, 284-291.

Tsai, C.C., Chu, C.H. and T.A. Barta. (1997). Modelling and analysis of a manufacturing cell formation problem with fuzzy mixed-integer programming, IIE Transactions, 29, 533-547.

Wang, J. (1998) A linear assignment algorithm for formation of machine cells and part families in cellular manufacturing, Computers and Industrial Engineering, 35, 81-84.

Wei, J.C. and N. Gaither. (1990). An optimal model for cell formation decisions, Decision Sciences, 21, 416-433.

Wemmerlov, U. and N.L. Hyer. (1986). Procedures for the Part/Machine Group Identification Problem in Cellular Manufacturing, Journal of Operations Management, 6, 125-147.

Wemmerlov, U. and N.L. Hyer. (1989). Cell Manufacturing in the US Industry: A Survey of Users, International Journal of Production Research, 27, 1511-1530.

Werners, B.M. (1987). An interactive fuzzy programming system, Fuzzy Sets and Systems, 23, 131-147.

Werners, B.M. (1988). Aggregation models in mathematical programming, in Mathematical Models for Decision Support, Mitra. G. and Greenberg, H.J.(eds.), Springer-Verlag, Berlin, Heidelberg, 295-305.

Wiedey, G. and H.J. Zimmermann. (1978). Media selection and fuzzy linear programming, Journal of the Operational Research Society, 31, 342-349.

Xpress-MP. Dash Optimisation, Blisworth, Northamptonshire, UK.

Xu, H. and H.P Wang. (1989). Part family formation for GT applications based on fuzzy mathematics, International Journal of Production Research, 9, 1637-1651.

Yang, T. and J.P. Ignizio. (1991). Fuzzy programming with nonlinear membership functions: Piecewise linear approximation, Fuzzy Sets and Systems, 41, 39-53.

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

Zhu, Z., Heady, R.B. and S. Reiners. (1995). An efficient zero-one formulation of the cell formation problem, Computers and Industrial Engineering, 28, 911-916.

Zimmermann, H.J. (1978). Fuzzy programming and linear programming with several objective functions, Fuzzy Sets and Systems, 1, 45-55.

Zimmermann, H.J. (1983). Fuzzy mathematical programming, Computers and Operations Research, 10, 291-298.

Zimmermann H.J. and P. Zysno. (1988). Decisions and evaluations by hierarchical aggregation of information, Fuzzy Sets and Systems, 10, 243-266.

Zimmermann, H.J. (1991). Fuzzy Set Theory and Its Applications, Kluwer Academic Publishers, Boston, MA.

Table 1: Numerical values for IP CF model

Part/Machines utilisation & Number of machine instances

Parts/M_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	$\sum_{j=1}^{10} UTIL_{i,j}$	$KTYPES_i$
M_1	0.0	0.3	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.5	1
M_2	0.0	0.0	0.0	1.0	0.5	0.0	0.9	0.0	0.1	0.6	3.1	4
M_3	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.2	0.1	0.0	1.2	2
M_4	0.0	1.2	0.0	0.9	0.5	0.4	0.0	0.0	0.0	0.0	3.0	3
M_5	0.4	0.2	0.0	0.8	1.1	1.0	0.6	1.0	1.0	0.0	6.1	7
M_6	0.0	0.7	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.3	1.8	2
M_7	0.0	0.5	0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.9	1

Set-up Costs

Parts/M_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
M_1	0.00	2.95	0.00	1.96	0.00	0.00	0.00	1.92	0.00	0.00
M_2	0.00	0.00	0.00	5.12	2.42	0.00	5.05	0.00	1.54	2.16
M_3	0.00	0.00	2.93	0.00	0.00	0.00	0.00	1.94	1.45	0.00
M_4	0.00	5.54	0.00	2.91	2.42	2.38	0.00	0.00	0.00	0.00
M_5	2.91	2.59	0.00	2.88	5.42	4.91	2.63	4.93	4.81	0.00
M_6	0.00	2.82	0.00	0.00	0.00	0.00	2.73	0.00	0.00	2.49
M_7	0.00	2.78	0.00	2.12	2.12	0.00	0.00	0.00	0.00	0.00

Part-Machine operation Sequence & ZTYPES

Sequence/Parts	1	2	3	4	5	ZTYPES
P_1	M_5					1
P_2	M_1	M_4	M_5	M_7	M_6	5
P_3	M_3					1
P_4	M_1	M_2	M_4	M_5	M_7	5
P_5	M_2	M_5	M_4	M_7		4
P_6	M_4	M_5				2
P_7	M_5	M_2	M_6			3
P_8	M_1	M_3	M_5			3
P_9	M_3	M_5	M_2			3
P_{10}	M_2	M_6				2

$E_{MIN} = 6$, $E_{MAX} = 8$ **& NCELLS=3**

Table 2: Numerical values for DS 1

Part/Machines utilisation & Number of machine instances

Parts/M_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	$\sum_{j=1}^{10} UTIL_{i,j}$	$KTYPES_i$
M_1	0.0	0.3	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.5	1
M_2	0.0	0.0	0.0	1.0	0.5	0.0	0.9	0.0	0.1	0.6	3.1	4
M_3	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.2	0.1	0.0	1.2	2
M_4	0.0	1.2	0.0	0.9	0.5	0.4	0.0	0.0	0.0	0.0	3.0	3
M_5	0.4	0.2	0.0	0.8	1.1	1.0	0.6	1.0	1.0	0.0	6.1	7
M_6	0.0	0.7	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.3	1.8	2
M_7	0.0	0.0	0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0	0.4	1

Set-up Costs

Parts/M_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
M_1	0.00	2.95	0.00	1.96	0.00	0.00	0.00	1.92	0.00	0.00
M_2	0.00	0.00	0.00	5.12	2.42	0.00	5.05	0.00	1.54	2.16
M_3	0.00	0.00	2.93	0.00	0.00	0.00	0.00	1.94	1.45	0.00
M_4	0.00	5.54	0.00	2.91	2.42	2.38	0.00	0.00	0.00	0.00
M_5	2.91	2.59	0.00	2.88	5.42	4.91	2.63	4.93	4.81	0.00
M_6	0.00	2.82	0.00	0.00	0.00	0.00	2.73	0.00	0.00	2.49
M_7	0.00	0.00	0.00	2.12	2.12	0.00	0.00	0.00	0.00	0.00

Part-Machine operation Sequence & ZTYPES

Sequence/Parts	1	2	3	4	5	ZTYPES
P_1	M_5					1
P_2	M_1	M_4	M_5	M_6		4
P_3	M_3					1
P_4	M_1	M_2	M_4	M_5	M_7	5
P_5	M_2	M_5	M_4	M_7		4
P_6	M_4	M_5				2
P_7	M_5	M_2	M_6			3
P_8	M_1	M_3	M_5			3
P_9	M_3	M_5	M_2			3
P_{10}	M_2	M_6				2

$E_{MIN} = 3$, $E_{MAX} = 6$ & **NCELLS=5**

Table 3: Common Operators used in Fuzzy MP

Operator	Formulation*	Format after transform.	References
'**Min**'	$\mu_D = min \;\; \mu_S$ †	Linear	Mjelde (1986)
			Zimmermann (1978)
			Zimmermann (1991)
'**Fuzzy and**'	$\mu_D = \gamma min \; \mu_S + (1-\gamma)/(T+1) \times \sum_{S=0}^{T} \; \mu_S$	Linear	Werners (1988)
'**Min – bounded sum**'	$\mu_D = \gamma min \; \mu_S + (1-\gamma) min(1, \sum_{S=0}^{T} \; \mu_S)$	Linear	Luhandjula (1982)
'**Product**'	$\mu_D = \Pi_{S=1}^{T} \;\; \mu_S$	Nonlinear	Zimmermann (1978)
'γ'	$\mu_D = (\Pi_{S=0}^{T})^{1-\gamma} \;\; [1 - \Pi_{S=0}^{T}(1-\mu_S)]^{\gamma}$	Nonlinear	Luhandjula (1982)
			Zimmermann (1983)

* The objective is to maximise μ_D.

† μ_S is membership function of the S^{th} fuzzy constraint (S is index of fuzzy constraints, $S = 0, ..., T$).

Table 4: Results of varying γ values for the 'min-bounded sum' operator (DS 1)

Case 5: Linear non-increasing membership function			Case 6: Triangular membership function		
γ	CPU Time (secs)	Total Cost	γ	CPU time (secs)	Total Cost
0.1	97	237.01	0.1	18	237.01
0.2	23	237.01	0.2	27	231.47
0.3	28	231.47	0.3	63	231.47
0.4	**10**	231.47	0.4	19	237.01
0.5	12	231.89	**0.5**	**9**	231.47
0.6	202	231.59	0.6	55	231.47
0.7	549	231.47	0.7	103	237.01
0.8	325	237.01	0.8	77	231.47
0.9	135	231.47	0.9	47	212.47

Table 5: Computational results for fuzzy and crisp models

Case	Operator	Membership Functions	Cells Created	Distinct Cells Used by Each Part	Later Revisits of Parts	CPU Time (secs)	Total Cost
Data Set 1: Problem size $(M \times P) = (7 \times 10)$; NCELLS=5; $E_{MAX} = 6$; $P_0 = 32.91$; $P_{R1} = 2$; $P_{R2} = 3$, Machine Instances= 20							
1	‘**Min**’	Linear non-increasing	4	12	0	2010	211.47
2	‘**Min**’	Triangular	3	12	1	32	219.95
3	‘$\widetilde{\textbf{and}}$’	Linear non-increasing	4	12	0	40	217.01
4	‘$\widetilde{\textbf{and}}$’	Triangular	3	12	1	38	212.47
5	‘**Min – bounded sum**’	Linear non-increasing	3	14	0	10	231.47
6	‘**Min – bounded sum**’	Triangular	3	14	0	9	231.47
7	Deterministic Model	$E_{MAX} = 6$	4	12	0	109	219.92
8	Deterministic Model	$E_{MAX} = 8$	4	12	0	32	210.92
Data Set 2: Problem size $(M \times P) = (7 \times 10)$; NCELLS=5; $E_{MAX} = 6$; $P_0 = 33$; $P_{R1} = 2$; $P_{R2} = 3$; Machine Instances= 20							
Case	Operator	Membership Functions	Cells Created	Distinct Cells Used by Each Part	Later Revisits of Parts	CPU Time (secs)	Total Cost
1	‘**Min**’	Linear non-increasing	3	13	0	10875	229.24
2	‘**Min**’	Triangular	3	13	0	1149	229.24
3	‘$\widetilde{\textbf{and}}$’	Linear non-increasing	4	13	2	5913	225.82
4	‘$\widetilde{\textbf{and}}$’	Triangular	3	13	0	996	229.24
5	‘**Min – bounded sum**’	Linear non-increasing	3	14	3	35	236.82
6	‘**Min – bounded sum**’	Triangular	3	14	3	21	236.82
7	Deterministic Model	$E_{MAX} = 6$	4	13	2	4554	225.82
8	Deterministic Model	$E_{MAX} = 8$	3	12	1	433	220.24

... continues on next page

Table 5 – continues from previous page

Data Set 3: Problem size $(M \times P) = (7 \times 10)$; NCELLS=5; $E_{MAX} = 8$; $P_0 = 21$; $P_{R1} = 2$; $P_{R2} = 5$; Machine Instances= 27

Case	Operator	Membership Functions	Cells Created	Distinct Cells Used by Each Part	Later Revisits of Parts	CPU Time (secs)	Total Cost
1	‘**Min**’	Linear non-increasing	-	-	-	$>$ 57 hours	-
2	‘**Min**’	Triangular	3	12	1	6966	246.91
3	‘$\widetilde{\mathbf{and}}$’	Linear non-increasing	-	-	-	$>$ 57 hours	-
4	‘$\widetilde{\mathbf{and}}$’	Triangular	3	12	1	10431	246.91
5	‘**Min – bounded sum**’	Linear non-increasing	3	13	1	17	251.49
6	‘**Min – bounded sum**’	Triangular	3	13	1	51	267.75
7	Deterministic Model	$E_{MAX} = 8$	4	12	2	16499	264.14
8	Deterministic Model	$E_{MAX} = 10$	3	12	0	8341	262.14

Data Set 4: Problem size $(M \times P) = (9 \times 9)$; NCELLS=5; $E_{MAX} = 8$; $P_0 = 35$; $P_{R1} = 4$; $P_{R2} = 5$; Machine Instances= 28

Case	Operator	Membership Functions	Cells Created	Distinct Cells Used by Each Part	Later Revisits of Parts	CPU Time	Total Cost
1	‘**Min**’	Linear non-increasing	-	-	-	$>$ 50 hours	-
2	‘**Min**’	Triangular	3	11	0	25448	243.27
3	‘$\widetilde{\mathbf{and}}$’	Linear non-increasing	4	11	1	114171	249.69
4	‘$\widetilde{\mathbf{and}}$’	Triangular	3	11	0	2752	243.27
5	‘**Min – bounded sum**’	Linear Non-Increasing	3	12	5	9	263.69
6	‘**Min – bounded sum**’	Triangular	3	12	5	49	263.69
7	Deterministic Model	$E_{MAX} = 8$	4	11	1	48974	257.99
8	Deterministic Model	$E_{MAX} = 12$	3	10	1	688	245.11

Figure captions

Figure 1. Part/Machine cell allocation.

Figure 2(a). Objective function (18)

Figure 2(b). Fuzzy constraint (16)

Figure 2(c). Fuzzy constraints (16), (17)

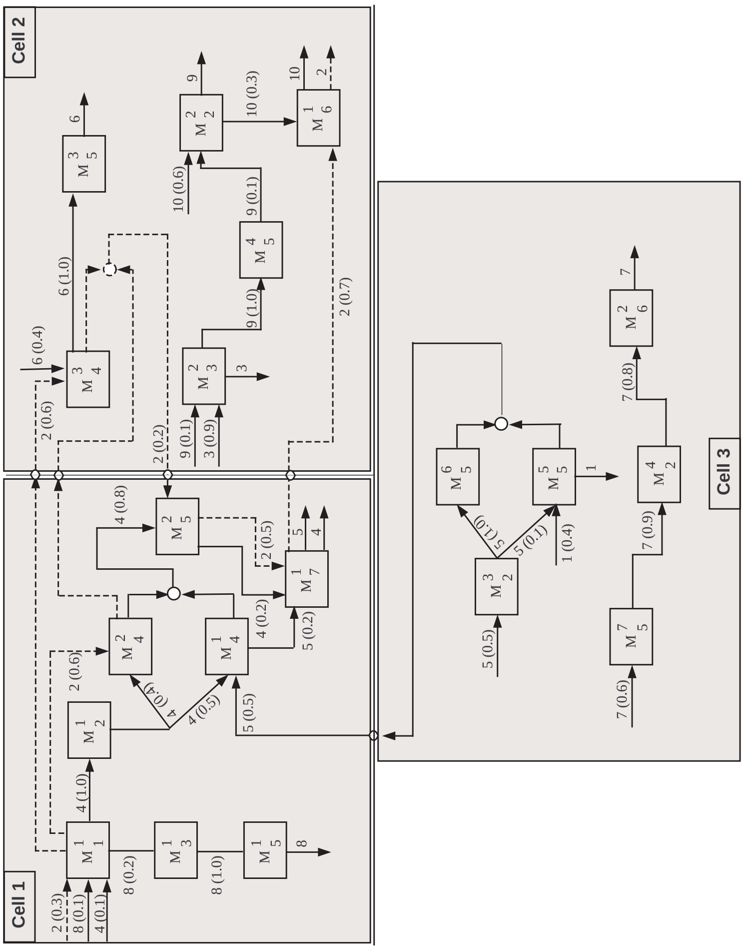
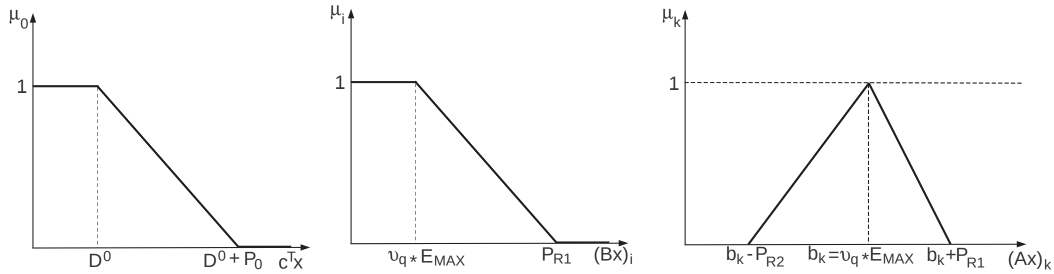


Figure 1: Part/Machine Cell Allocation



(a) Objective function (18) (b) Fuzzy constraint (16) (c) Fuzzy constraints (16), (17)

Figure 2: Membership Functions