Determining Rules in Fuzzy Multiple Heuristic Orderings for Constructing Examination Timetables

Hishammuddin Asmuni
Faculty of Computer Science & Information System, Universiti Teknologi Malaysia,
81310 Skudai, Johor, Malaysia,
hishamudin@utm.my

Edmund K. Burke, Jonathan M. Garibaldi
School of Computer Science & IT, University of Nottingham, Nottingham, NG8 1BB, UK
{ekb,jmg}@cs.nott.ac.uk

Barry McCollum
School of Computer Science, Queen's University Belfast, Belfast, BT7 1NN, UK
b.mccollum@qub.ac.uk

This paper presents alternative methods for producing fuzzy models for the *fuzzy multiple heuristic ordering* technique that we previously introduced for the construction of examination timetables. The effects of altering the rules within the fuzzy inference system were investigated. Four alternative rule 'tuning' approaches are described in detail and their results are presented and compared.

Keywords: Timetabling, fuzzy sets, fuzzy model identification, fuzzy rule tuning

1 Introduction

In the process of examination timetable construction, the order in which exams are assigned to time slots has been shown to have a major effect on the eventual solution [6]. An assessment of how difficult it is to place a given exam into a timetable (in effect, some measure of how hard it is to satisfy the constraints relevant to the particular exam) is often used to guide the order of placement. The usual strategy is to place the most difficult exams first, on the basis that it is better to leave the easier exams until later in the process when there are fewer time slots remaining. There are many different criteria that may be used when assessing this difficulty.

A common approach has been to employ graph based heuristics (a heuristic is an approximate rule or a 'rule-of-thumb' [5]) to provide a quantitative indication of difficulty. This measure is then used to determine the order in which the exams are assigned into the timetable and, hence, are referred to as 'heuristic orderings'. Examples of such heuristics are the number of other exams in conflict with the given exam, the number of students enrolled on each exam, etc. For detailed descriptions of these heuristic orderings, please refer to Asmuni *et al.* [3]. In our previous papers [1, 3], we explored how fuzzy techniques [7] could be employed to combine multiple heuristics within the construction of examination timetables. Particularly, we proposed the use of fuzzy inference systems to combine multiple heuristics *simultaneously* in order to provide a measure of the difficulty of placing each exam. This measure was then used to order (rank) the exams for assignment. Various combinations of heuristics were investigated in the construction process. In order to investigate the wider applicability of this novel fuzzy approach, the techniques were also applied to the domain of course timetabling [2]. However, to date, we have only been concerned with tuning the membership functions of the fuzzy variables, utilising a set of pre-defined (and fixed) fuzzy rules. This paper presents a series of experiments which were undertaken to explore the influences of tuning the fuzzy rules in fuzzy inference systems for which the membership functions had been fixed.

2 Method

The following heuristic orderings have been widely used to determine the order of placement of examinations in *sequential construction* algorithms:

Largest Degree (*LD*): Exams are ranked in descending order by the number of exams in conflict — i.e. priority is given to exams with the greatest number of exams in conflict.

Largest Enrollment (*LE*): Exams are ranked in descending order by the number of students enrolled in each of the exam.

Least Saturation Degree (*SD*): Exams are ranked in increasing order by the number of valid time slots remaining in the timetable for each exam.

Largest Coloured Degree (*LCD*): This heuristic is based on *LD*. For this heuristic, only exams which have been already assigned to the schedule are considered to cause conflict.

Weighted Largest Degree (*WLD*): This heuristic is also based on *LD*. Besides the number of exams, the total number of students involved in the conflict is also taken into account.

The general framework for constructing timetables using fuzzy inference systems utilising various combinations of these three heuristics has previously been described in Asmuni *et al.* [3]. In this study, the effect of two further fuzzy model determination techniques were investigated. Firstly, a simple enumerative search was implemented for tuning the fuzzy rules for the fuzzy system utilising a combination of the first three heuristics above (termed the *Fuzzy LD+SD+LE Model*) for which the membership functions had been previously tuned (see Section 2.1). Secondly, a random search was implemented to create fuzzy systems utilising combinations of all five heuristics above, in which the fuzzy model parameters (specifying both membership functions and rules) were determined by random selection (see Section 2.2). This allowed for a wider exploration of the total search space of alternative fuzzy models (which is vast).

2.1 Tuning Fuzzy Rules with Fixed Membership Functions

The objective of these experiments was to investigate whether tuning the fuzzy rules would offer any improvement in performance over the previously tuned and then fixed set of fuzzy rules. For this purpose, the best membership functions identified in experiments reported in [1] were implemented for the respective data sets. As we used the fuzzy multiple ordering that considered three heuristics simultaneously (i.e. *Fuzzy LD+SD+LE Model*), the rules shown in Fig. 6 of [1] were used as the benchmark fuzzy rule-set.

The original fuzzy rule-set is detailed in Table 1, where the number in the cell represents the rule number. As an example, rule 22 is read as:

IF *LE* is *small* AND *SD* is *medium* AND *LD* is *high* THEN *examweight* is *small*

In the tuning process, the only modification allowed was in the consequence part of each rule.

Table 1: Fuzzy Rule set for *Fuzzy LD+LE+SD Model*

<table>
<tr><th rowspan="4">LE</th><th colspan="9">LD</th></tr>
<tr><th colspan="3">S</th><th colspan="3">M</th><th colspan="3">H</th></tr>
<tr><th colspan="3">SD</th><th colspan="3">SD</th><th colspan="3">SD</th></tr>
<tr><th>S</th><th>M</th><th>H</th><th>S</th><th>M</th><th>H</th><th>S</th><th>M</th><th>H</th></tr>
<tr><td>S</td><td>S^{1}</td><td>VS^{4}</td><td>VS^{7}</td><td>S^{10}</td><td>S^{13}</td><td>VS^{16}</td><td>M^{19}</td><td>S^{22}</td><td>S^{25}</td></tr>
<tr><td>M</td><td>S^{2}</td><td>S^{5}</td><td>VS^{8}</td><td>H^{11}</td><td>M^{14}</td><td>M^{17}</td><td>H^{20}</td><td>M^{23}</td><td>M^{26}</td></tr>
<tr><td>H</td><td>H^{3}</td><td>S^{6}</td><td>S^{9}</td><td>H^{12}</td><td>M^{15}</td><td>M^{18}</td><td>VH^{21}</td><td>H^{24}</td><td>M^{27}</td></tr>
</table>

VS=very small
S=small
M=medium
H=high
VH=very high

Six possible values for the consequence part were defined: *not_in_use*, *very small*, *small*, *medium*, *high* and *very high*. If the *not_in_use* value was assigned to the consequence part of a rule, that means the rule was not applicable. For each rule, it's consequence part was changed by assigning one of the six possible values in the sequence of *not_in_use*, *very small*, *small*, *medium*, *high* and *very high*, iteratively. Considering 27 fuzzy rules and six possible values that could be assigned to the consequence part of each rule, there were 162 possible sets of fuzzy rules. Although the initial number of rules was 27, the number of rules might be reduced if, by doing so, the solution quality improved. Each rule-set was tested over three runs of the sequential construction algorithm. Hence, at the end of the experiment, for each data set, three timetable solutions were obtained for each of the 162 rule-sets. The fuzzy rules set with the lowest penalty cost were selected as the 'best' set of rules for that specific data set. Two experiments were conducted:

- *Tuned Fuzzy Rules 1*— The set of tuned fuzzy rules that gave the best current solution quality was kept and used as the initial set of rules for the next step of the tuning process. A simple deepest descent enumerative search algorithm was employed in this experiment.
- *Tuned Fuzzy Rules 2*— Each of the rules was changed in isolation, no changes made in the earlier iterations were taken into account. Hence, after each change to the consequence part of any rule, the rules were reinstated to the initial configuration as shown in Table 1, before moving to the next iteration (i.e. setting another value in a consequence part).

2.2 Randomly Generated Fuzzy Models

The aim of this experiment was to examine alternative approaches for determining fuzzy models and to explore a larger space of possible fuzzy models. Instead of using fuzzy models in which either the membership functions or the rules were fixed, a stochastic approach was utilised to define the fuzzy model. In order to make the resultant fuzzy systems more manageable, only combinations of three heuristics selected from the five specified above (*LD*, *SD*, *LE*, *LCD* and *WLD*) were generated (the rule-sets for fuzzy systems featuring five heuristics would be enormous).

In the implementation, the first step was to randomly select which three heuristic orderings would be considered simultaneously. The next step was to create a set of fuzzy rules for the selected heuristic orderings, which were also selected in random fashion. Any rule contain at least one antecedent, up to a maximum of three antecedents. The last step was to choose centre points (*cp*) for membership functions for all of the fuzzy variables. The *cp* parameter defines the right-hand point of the *small*, the centre-point of the *medium* and the left-hand point of the *high* membership functions, as illustrated in Fig. 1. As a fuzzy system with three input and one output variables was implemented, four *cp* points needed to be randomly chosen.

Integer values were used to encode the heuristic orderings and fuzzy rules, as shown in Table 2. An example is presented in Figure 2 to show how the random fuzzy model was created. In *STEP 1*, the three heuristic orderings chosen are identified as *LE*, *SD* and *WLD*. Based on these heuristic orderings, the randomly generated rules are translated into '*IF ... THEN ...*' form. The rules

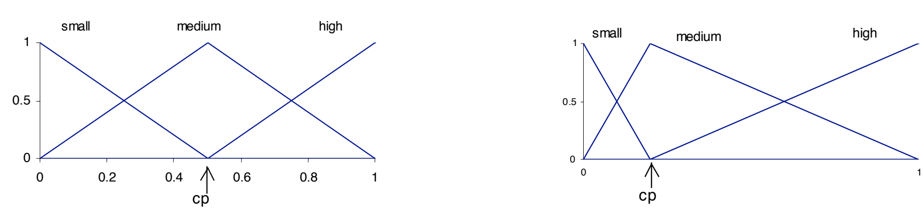

Figure 1: An illustration of the effect of the *cp* parameter on the membership functions of a variable

Table 2: Integer codes assigned to fuzzy model parameters

Heuristic	*LD*	*SD*	*LE*	*LCD*	*WLD*
Heuristic Code	1	2	3	4	5

Antecedent linguistic variable	not_inuse	small	medium	high
Antecedent Code	0	1	2	3

Consequence linguistic variable	not_inuse	very small	small	medium	high	very high
Consequence Code	0	1	2	3	4	5

are presented in a two dimensional array. Each row of the array represents one rule. In each row, the first column corresponds to the antecedent for the first heuristic ordering, the second column represent the antecedent value for the second heuristic and the third column holds the value for the third heuristic; the last column corresponds to the consequence part (i.e. *examweight*). In the example, three rules are randomly generated and their translated form are given. Notice that *Rule 2* only consists of two antecedents as *SD* is set to *not_in_use* (antecedent code = 0). This rule generation was performed without any concern as to the meaning of the rule — any rule was accepted even if it contradicted the 'common sense' of the relevant heuristic ordering. Next, in *STEP 3*, four *cp* points are randomly picked and a graphical representation of the membership functions that are generated is given. Again, the first three elements of the array are corresponds to the membership functions for the three chosen heuristic ordering in the sequence order; while the last element represents the *cp* point for *examweight*.

In order to evaluate this stochastic approach to fuzzy model determination, two experiments were performed as follows:

- *Random Model 1*: Experiments were performed for 100 iterations for each data set. In each iteration, a new fuzzy model was created by randomly choosing the heuristic orderings, 27 fuzzy rules and the four *cp* points for the membership functions. Each fuzzy model was tested three times within the sequential constructive algorithm.
- *Random Model 2*: Experiments were conducted for 1000 iterations for nine of the data sets, while for *CAR-F-92*, *CAR-S-91* and *UTA-S-92*, the experiments were run for 100 iterations. For this experiment, the heuristic orderings and *cp* points were randomly choosen only once for each data set. Initially, the fuzzy rule-set was empty. In the first iteration, a fuzzy rule was randomly created and specified to be the first rule. Having created the fuzzy model, the sequential constructive algorithm was run three times. The best timetable constructed was set as a 'benchmark'. For each of the remaining iterations, a fuzzy rule was randomly created and appended to the list of rules. The sequential constructive algorithm was again run three times with the new fuzzy model (i.e. only the rules were changed). The rule was kept if a better solution was obtained with this new fuzzy model, and the new best solution was recorded as the new benchmark; otherwise, the newly added rule was removed. This process continued until the number of iteration exceeded the maximum number of iterations allowed for the particular data set.

In both experiments, non-applicable rules (rule in which the consequence part was *not_in_use* or rules in which all the antecedents were *not_in_use*) were removed. As the fuzzy rules were randomly selected, in the case of experiment with *Random Model 1*, it was possible to have a fuzzy model that contained less than 27 rules.

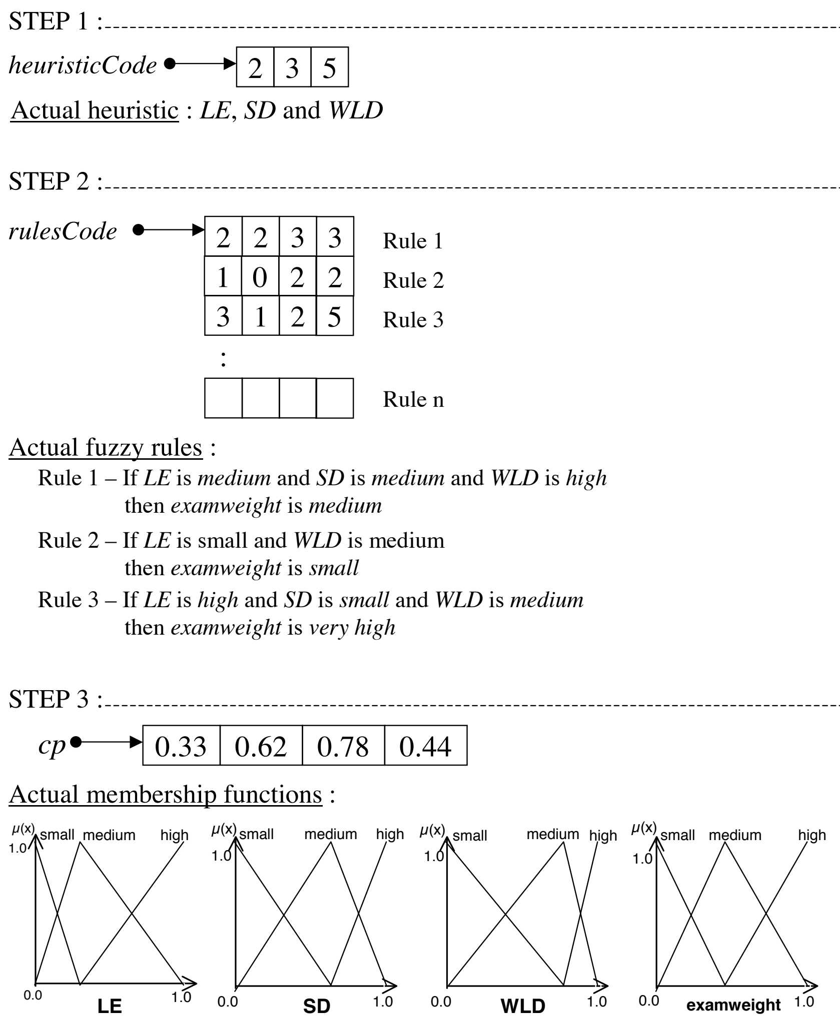

Figure 2: An example of defining a random fuzzy model

3 Implementation

3.1 Problem Description

The experiments were carried out with 12 benchmark data sets made publicly available by Carter *et al.* [6]. Table 3 reproduces the problem characteristics. A proximity cost function was used to measure the timetable quality. The maximum capacity for each time slot was not taken into account. Only feasible timetables were accepted. The penalty function was taken from Carter *et al.* [6]. It is motivated by the goal of spreading out each student's examination schedule. If two exams scheduled for a particular student are t time slots apart, the weight is set to $w_t = 2^{5-t}$ where $t \in \{1, 2, 3, 4, 5\}$. The weight is multiplied by the number of students that sit for both of the scheduled exams. The average penalty per student is calculated by dividing the total penalty by total number of students. The following formulation was used (adapted from Burke *et al.* [4]), in which the goal is to minimize:

$$\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s_{ij} w_{|p_j - p_i|}}{T},$$

Table 3: Examination Timetabling Problem Characteristics

Data Set	Number of time slots	Number of exams	Number of students	Conflict density
CAR-F-92	32	543	18419	0.14
CAR-S-91	35	682	16925	0.13
EAR-F-83	24	190	1125	0.27
HEC-S-92	18	81	2823	0.42
KFU-S-93	20	461	5349	0.06
LSE-F-91	18	381	2726	0.06
RYE-F-92	23	486	11483	0.08
STA-F-83	13	139	611	0.14
TRE-S-92	23	261	4360	0.18
UTA-S-92	35	622	21266	0.13
UTE-S-92	10	184	2750	0.08
YOR-F-83	21	181	941	0.29

where N is the number of exams, s_{ij} is the number of students enrolled in both exam i and j, p_i is the time slot where exam i is scheduled, and T is the total number of students; subject to $1 \leq |p_j - p_i| \leq 5$.

3.2 Results

Table 4 shows a comparison of the results obtained using fixed and tuned fuzzy rules. The first column indicates the penalty cost for the timetable solution of each data set that has been constructed with a set of fixed fuzzy rules (extracted from the ninth column of Table 2 in [1]). In the next two columns, the qualities of the timetable solutions produced using the sequential constructive algorithm with *Tuned Fuzzy Rules 1* and *Tuned Fuzzy Rules 2* approaches are given. In the fifth and sixth columns, the qualities of the timetable solutions produced by the experiments outlined in Section 2.2 are given. In Table 4, the best results across all experiments for each data set is highlighted in bold font. It can be seen that, in all data sets, better solutions were produced by tuning the fuzzy rules (either by *Tuned Fuzzy Rules 1* or *Tuned Fuzzy Rules 2*), compared to the approach that only used fixed fuzzy rules (column one), the only exception being *KFU-S-93* for which no improvement was observed. These results show that tuning the fuzzy rules produced considerably better timetable solutions. Although the results only show small improvements (in the range of 0.01 to 1.43), this evidence indicates that performing fuzzy rule tuning does give a considerable performance advantage.

In [1], we demonstrated that combinations of three heuristic ordering generally produced better solutions compared to combinations of two heuristic orderings. However, in two cases (*CAR-F-92* and *EAR-F-83*), it was observed that two heuristics ordering outperformed three heuristics ordering. We argued then that this may have been rectified if the fuzzy rules were to be tuned. As these fuzzy rules tuning experiments were performed only with the *Fuzzy LD+SD+LE Model*, the result obtained in this experiment confirmed that statement in which it can be observed that the *EAR-F-83* data set now has a penalty cost equal to 36.16. This penalty cost value is smaller than the penalty cost incurred when the *Fuzzy SD+LE* was used — i.e. 36.99 (see Table 2 in [1]). Although the result produced by the *Fuzzy SD+LE* model for the *CAR-F-92* (see Table 2 in [1]) still outperformed the result obtained in this experiment, overall the results indicate the potential of expanding the tuning of the fuzzy model to also incorporate tuning the fuzzy rules.

If we now compare the results obtained for rule tuning (the third and fourth columns) to the results obtained by random model generation (the fifth and sixth columns) it can be seen that the best result for eight of the data sets was obtained by fuzzy models that were developed using tuned

Table 4: A comparison of results

Data Set	Fixed Fuzzy Rules (from [1])	*Tuned Fuzzy Rules 1*	*Tuned Fuzzy Rules 2*	*Random Model 1*	*Random Model 2*
CAR-F-92	4.52	4.51	4.51	4.59	**4.32**
CAR-S-91	5.24	**5.19**	**5.19**	5.58	5.54
EAR-F-83	37.11	**36.16**	36.64	40.93	37.05
HEC-S-92	11.71	11.61	**11.60**	12.55	12.31
KFU-S-93	15.34	15.34	15.34	15.74	**15.03**
LSE-F-91	11.43	**11.35**	**11.35**	12.58	12.65
RYE-F-92	10.30	10.02	10.05	10.58	**9.75**
STA-F-83	159.15	159.09	160.79	159.22	**158.64**
TRE-S-92	8.64	8.62	**8.47**	9.24	8.79
UTA-S-92	3.55	**3.52**	**3.52**	3.69	4.31
UTE-S-92	27.64	27.64	**27.55**	29.77	29.10
YOR-F-83	40.68	**39.25**	39.79	43.88	42.30

membership functions with tuned fuzzy rules. Although experiments which applied *Random Model 1* did not produce any best results, the experiments that used *Random Model 2* actually produced four best results. These best results were obtained using the following fuzzy models:

CAR-F-92:

- heuristic orderings: *LCD*, *LE* and *SD*
- *cp* points for membership functions: 0.550, 0.110, 0.296, and 0.132
- number of fuzzy rules: 16

KFU-S-93:

- heuristic orderings: *WLD*, *SD* and *LE*
- *cp* points for membership functions: 0.021, 0.721, 0.351, and 0.095
- number of fuzzy rules: 48

RYE-F-92:

- heuristic orderings: *LE*, *WLD* and *LCD*
- *cp* points for membership functions: 0.679, 0.358, 0.001, and 0.708
- number of fuzzy rules: 9

STA-F-83:

- heuristic orderings: *WLD*, *SD* and *LE*
- *cp* points for membership functions: 0.309, 0.739, 0.408, and 0.595
- number of fuzzy rules: 17

One possible reason why only four best results were found is the fact that the number of iterations in the experiments (100 for *Random Model 1* and 1000 for *Random Model 2*) was quite small when compared to the huge search space that needs to be explored in order to find the 'optimal' fuzzy model. Taking a different view, one should notice that performing tuning of membership functions and fuzzy rules in separate stages is better than performing both membership functions and fuzzy rules tuning at the same stage (as in the *Random Model 2* approach). It also worthy of mention that only 16, 9 and 17 fuzzy rules are required to produced the best solutions obtained for *CAR-F-92*, *RYE-F-92* and *STA-F-83* respectively. This indicates that not all possible rules are require to be utilised in such a fuzzy system in order to get a good solution. Fewer rules makes the fuzzy model more understandable for the developer and user. Therefore, a more sophisticated optimisation approach could almost certainly be devised to tackle the tuning process more systematically.

4 Conclusion

The overall aim of this paper was to investigate the effect of altering the fuzzy rules for fuzzy multiple heuristic ordering in measuring the difficulty of assigning exams into time slots. Having implemented two alternative approaches for altering the fuzzy rules (one for tuning the rules once membership functions have been tuned and fixed and the other for randomly generating alternative rule-sets), the results obtained demonstrated that better solutions can be generated. Unfortunately the space of all possible fuzzy models is vast and there remains much scope for the investigation of more sophisticated methods to efficiently search this vast space in order to find effective fuzzy models.

Acknowledgements. This research work is supported by the Universiti Teknologi Malaysia (UTM) and the Ministry of Science, Technology and Innovation (MOSTI) Malaysia.

References

[1] H. Asmuni, E. K. Burke, and J. M. Garibaldi. A Comparison of Fuzzy and Non-Fuzzy Ordering Heuristics for Examination Timetabling. In A. Lotfi, editor, *Proceedings of 5th International Conference on Recent Advances in Soft Computing 2004*, pages 288–293, 2004.

[2] H. Asmuni, E. K. Burke, and J. M. Garibaldi. Fuzzy Multiple Heuristic Orderings for Course Timetabling Problem. In B. Mirkin and G. Magoulas, editors, *Proceedings of the 2005 UK Workshop on Computational Intelligence (UKCI2005)*, pages 302–309, London, UK, September 2005. Birkbeck University of London. workshop date: 5-7 September 2005.

[3] H. Asmuni, E. K. Burke, J. M. Garibaldi, and Barry McCollum. Fuzzy Multiple Heurisitc Orderings for Examination Timetabling. In E. K. Burke and M. Trick, editors, *Practice and Theory of Automated Timetabling V (PATAT 2004, Pittsburg USA, August 2004, Selected Revised Papers)*, volume 3616 of *Lecture Notes in Computer Science*, pages 334–353, Berlin, 2005. Springer.

[4] E. K. Burke, Y. Bykov, J. Newall, and S. Petrovic. A Time-Predefined Local Search Approach to Exam Timetabling Problems. *IIE Transactions*, 36(6):509–528, June 2004.

[5] Edmund K. Burke and Graham Kendall, editors. *Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques.* Springer, 2005.

[6] M. W. Carter, G. G. Laporte, and S. Y. Lee. Examination Timetabling: Algorithmic Strategies and Applications. *Journal of the Operational Research Society*, 47:373–383, 1996.

[7] L. A. Zadeh. Fuzzy Sets. *Information and Control*, 8:338–353, 1965.