Fuzzy Clustering with an Application to Scheduling

SiXin Cheng

A Thesis

in

The Department

of

Mechanical Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montréal, Québec, Canada

March 1999

0-612-39104-3

ABSTRACT

Fuzzy Clustering with an Application to Scheduling

SiXin Cheng

Usually, the generation of an optimal schedule is a costly and time-consuming process. This process requires expensive computational software and hardware. Scheduling problem modeling using human expert knowledge is promising and flexible in dealing with real world applications. Unfortunately, human expert knowledge may not be available in all cases, and human experts may not be able to explain their knowledge explicitly. A new scheduling decision learning approach is introduced in this thesis. A subtractive clustering based system identification method is developed to learn the scheduling decision mechanism from an existing schedule. It is utilized to build a fuzzy expert model. The existing schedule can be an optimal schedule developed using an optimization method or a schedule generated by a human expert. The fuzzy expert model is then used to generate new schedules for other problems following the decision mechanism it learned. The implementation of this method is demonstrated by modeling a single machine weighted flowtime problem. Furthermore, selective subtractive clustering and modified subtractive clustering algorithms are developed and used to improve knowledge extraction. Those algorithms can also be used to model nonlinear and spiral systems using the clustering based system identification, such as function approximation applications and pattern classification applications when the information about the system is scarce.

ACKNOWLEDGEMENTS

The author is extremely thankful to his supervisor Dr. Kudret Demirli for his professional supervision, intellectual guidance, and patience, throughout this study. It was a pleasure working with him.

I would like to specially thank to my friend Packirisamy Muthukumaran and all my other friends for their help in correcting the text.

This thesis is dedicated to the author's wife ChunHua Zhang, father and mother, sister, sister in law for their help, love and encouragement to achieve excellence. Without them this would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

p_j is processing time

w_j is priority

G is data group (cluster)

G_c is center of data group (cluster)

μ is degree of membership

m is weight factor for Fuzzy C-means algorithm

A is fuzzy set in the input space

B is fuzzy set in the output space

Z is consequent parameter vector

P is potential value

P_1^* is maximum potential

$\bar{\epsilon}$ is accept ratio

$\underline{\epsilon}$ is reject ratio

η is squash factor

γ is the shape factor

r_a is cluster radius

d_{min} is the minimum distance between two cluster centers

c_j^i is the cluster center i in $j'th$ dimension

π_j is quantitative index

ν is penalty factor

δ is degree of association

Γ_{ij} is the *core* of membership function for variable j in rule i

Γ_j is the *support* of membership function for variable j in rule i

ω^i is the firing strength

β is the weighted firing strength

S^t is target sequence

S^p is predicted sequence

a is weight factor for sequential error

σ is sequential error

σ_t is training sequential error

σ_v is validation sequential error

ϕ is representation factor

Chapter 1

Introduction

1.1 Background and scope of this work

There are a number of applications in which modeling of systems is important. Modeling of systems may include fuzzy system identification and selection of reasoning mechanism. Conventional models fail to represent the complete behavior of systems when information is not sufficient. Mathematical modeling becomes more difficult when there are uncertainties and ambiguities within system. Too many variables and unpredictable interactions are known factors to modeling failure. Solving real world problems is computational demanding in general. In some other cases, model could be developed using expert knowledge about the system. Even though the model developed by the expert is able to represent the complete behavior of the system, in some cases, it is still costly to adapt it to changed requirements. An alternative method to handle this situation is to model the system using input and output data of the system. The data could be obtained from perfect mathematical modeling, from expert knowledge, or from experimental analysis of the system. As these data also have uncertainties and vagueness, an appropriate approach is essential to model the system correctly.

Zadeh's [1] proposal of linguistic approach was effective and versatile in modeling ill-defined systems with fuzziness or fully defined systems with realistic approximations. Zadeh's approach was later expanded into fuzzy systems modeling by Tong [2], Pedrycz [3], Trojan et al. [4], Sugeno and Yasukawa [5], Yager and Filev [6, 7, 8], Mamdani and Assilian [9], Bezdek [10] and Bezdek et al. [11], Emami et al. [12] and Chiu [13, 14]. To further the above work, this work modeled an input-output relationship with IF-THEN-ALSO rules using a modified subtractive clustering technique.

In general, system identification includes function approximation or output classification. In either case, the system identification has to be automatic in order to be versatile and amenable for further learning. The clustering technique is used in this work to generate rules automatically. Each cluster represents a group of associated data in a data space and a rule in a knowledge base of the system. The model which is built based on the set of clusters can be used to predict system behaviors. Identification of fuzzy systems invariably uses fuzzy partitioning or fuzzy grouping in the data space based on certain measures.

Most real world scheduling problems are usually very complex. Generating a quality schedule is usually a costly and time-consuming process. To generate an optimal schedule using optimization method usually requires expensive computational software and hardware. In some real world cases, a schedule may be generated by a human expert. It is also possible to solve scheduling problem by modeling human expert knowledge (Turksen et al. [15], Caprihan et al. [16], and Ben-Arieh and Lee [17]). But, modeling human expert knowledge is usually limited by the availability of human expert and/or their knowledge.

Each schedule has a goal to accomplish. The goal is to find an optimal solution for certain objective functions. In this thesis, a subtractive clustering based system identification method and the Sugeno reasoning mechanism are developed and used to solve scheduling problems. These methods can be used to build fuzzy expert model from an existing schedule. The existing schedule could be a schedule developed using an optimization method or by a human expert. It could be an optimal solution for certain scheduling objective functions. The schedule is used to generate models, and the model is then used to create new schedules when there are new jobs. These new schedules optimize the same scheduling objective function as the schedule that the model learned. The Sugeno reasoning mechanism is computationally efficient in building models and generating new schedules. The fuzzy expert models can quickly incorporate new jobs into a new schedule. In order to facilitate the modeling of scheduling problems, a sequential distance measure is introduced.

In this thesis, the subtractive clustering based system identification method is demonstrated through solving a single machine weighted flow time problem. The method learns the schedule decision from the input and output of a schedule. The WSPT sequence optimizes the weighted flow time objective function in single machine cases. Successful capturing of job sequence means that the method can learn from a schedule and then generates an optimal schedule. An example of this process is to learn a WSPT sequence and to generate an optimal sequence for a single machine weighted flowtime problem.

The subtractive clustering based system identification method can be used to model systems with multiple inputs and outputs. For multiple-machine scheduling models, sub-optimal solution can be obtained by decomposing multiple-machine models into single machine problems. There are two ways to build models in the multiple machine cases using the subtractive clustering based system identification method. One of them is to build a fuzzy expert model for all machines. Another is to build one fuzzy expert model for each individual machine. In either case, the model generates a new schedule when new inputs are presented. The new schedule yields sub-optimal solutions following the existing schedule it has learned.

The successful identification of a system over a data space requires sufficient representation over the entire range of data space. In general, more data points may result in a model better representing the system. But in some situations, the model may have to be built from scarce data without losing the global behavior of the system. The minimum number of data points required for a perfect identification of the system is called *complete data.* When the data space is densely populated like in the case of *complete data*, the system behavior at any part of the input domain could easily be predicted using the system behavior at nearby points. But when the scarcity of the data increases the data is no longer complete to enable a prefect identification. It is called *scarce data*, where the system behavior a point may have to be modeled using the system response at farther away points. In those cases of system identification with *scarce data*, a new strategy of predicting the system behavior have to be used. A cluster of data in an input-output data space represents a grouping of data points with related behavior. And hence each cluster represents a particular system behavior in a phenomenological domain, which is the behavior of the cluster center. Each cluster is responsible for capturing a particular aspect of the system behavior. Thus a higher number of clusters may represent a system.

Subtractive clustering algorithm may lead to faulty representation of scarce data for complex system. In this thesis, modified subtractive clustering and selective subtractive clustering methods are developed to improve the clustering performance in modeling systems when the information about the system is scarce. The WSPT surface is a nonlinear and spiral surface. Successfully modeling WSPT problem with scarce data indicates that these two algorithms can be used to model nonlinear and spiral system with scarce training data.

1.2 Organization of the thesis

Scheduling approaches are introduced in Chapter 2. Clustering methods and clustering based system identification methods are discussed in Chapter 3. A clustering based system identification method is used for scheduling decision learning problems, and a scheduling sequencing learning application is presented in Chapter 4. Selective subtractive clustering and modified subtractive clustering algorithms are discussed in Chapter 5. The training and validation of sequential learning application are demonstrated in Chapter 6. Conclusions and possible applications are discussed in Chapter 7.

Chapter 2

Scheduling

2.1 Scheduling terminology and job attributes

Scheduling has been examined in the operations research literature since the early fifties (Conway et al. [18]). It has been defined by Baker [19] as *"the allocation of resources over time to perform a collection of tasks."* Many of the early developments in this field were related to problems in manufacturing. Hence most scheduling terminology and vocabulary are derived from manufacturing domain.

Scheduling is usually a multi-criteria based decision making process. The general decisions are sequencing, timing and routing. Specifying the order of jobs through the resources is called sequencing. Specifying the time, which includes starting time and completion time on resources is referred to as timing. Specifying the resource for each job is called routing.

A *task*, usually called *job*, can be a single item or a batch of items that needs processed together on resources. The resources are generally machines, manpower, and facilities. The processing of a particular job on a particular machine is called an *operation*. A task may have more than one operation that requires multiple resources.

The static attributes of a job are *processing times* of operations, *ready time*, *due date*, *priority*, etc. There are also changing attributes such as *slack time*, *remaining processing time*, *remaining number of operations*, etc. *Processing time* is the time required to complete an operation of a job. *Ready time* is the point of time a job is available for its next process. *Due date* is the customer demanded delivery date of the products. It is usually negotiable between manufacturer and customer. *Priority* is the preference of a job. The preference may indicate a job to be processed earlier than others. *Priority* of a job may be constant or may change during a scheduling process. *Slack time* is the maximum time left for a job before delivery minus *remaining processing time*. A value smaller than zero for *slack time* means that a job is overdue already. *Remaining processing time* and *remaining number of operations* are also changed during scheduling process. *Remaining processing time* is the total processing time minus time spent on all completed operations. *Remaining number of operations* is the number of unprocessed operations.

2.2 Problem classifications

Classifications of scheduling problems can be done on resources and tasks. A scheduling model may contain one resource, or several resources of one type, or several different types of resources. If it contains one or several resources of one type, tasks (or jobs) are likely to be single staged. Multi-resource model usually applies to multistage tasks. So a scheduling model can be an *one stage single resource model*, an *one stage parallel resource model*, an *multi-stage flow shop model*, an *multi stage job shop model* or an *multi-stage open shop model*. The models can be *static* or *dynamic*, and *deterministic* or *stochastic*. *Static model* refers to the one that the number of jobs and their *ready time* are given and fixed (French [20]). In *dynamic model* case, jobs arrive over a period of time. In the *deterministic case*, all scheduling information are predetermined and do not include stochastic factor. In the *stochastic case*, scheduling information is random in nature.

Most real world scheduling problems are very complex and far from being completely solvable due to their combinatorial nature. As a basic component of complicated scheduling models, single-machine problems help us understand a variety of scheduling issues in a tractable model. It is a basic building block in the development of understanding of comprehensive scheduling concepts. It facilitates the modeling of complicated systems. In case of complicated multi-machine problems, job sequencing may be determined by one highest or most restricted machine, such as a bottleneck machine or an expensive processor. Modeling a bottleneck machine or an expensive processor itself as a single machine may approximate the entire system scheduling. In some case, the level of decision may dictate that the processing facility should be modeled in the aggregate, as a single resource (Baker [19]). The modeling methodology presented in this thesis is demonstrated on a single machine scheduling problem. However the methodology also applies to multiple machine sequencing problems.

2.3 Scheduling objectives

Scheduling is a decision-making process that has a goal to accomplish. The goal is usually called objective or performance measure. The objectives or performance measures vary from manufacturer to manufacturer and sometimes from day to day.

Most scheduling objectives are related to the *flow time* and the *due date*. *Flow time* of a job is the amount of time between arrival and departure of a job from the system. *Due date* is the customer demanded delivery date of the products. They reflect the cost of storage, materials handling, lost sale, and rejections by missing due date. *Flow based* objectives are concerned about reducing turnaround time of jobs through the shop. The turnaround time is usually reflected in the cost of storage and materials handling. *Due date* based objective is to reduce the amount and frequency by which the individual completion time to the promised time. *Tardiness* and *lateness*, both are due date based objectives. *Tardiness exceeds* the due date, the *lateness* does not. The promised time is the customer required delivery time, which is usually negotiable between customer and manufacture. Missing due date usually causes some penalties. A *weight based* objective assigns different importance levels to different jobs. A non-weight-based objective can be considered as a special case of *weight based* objective where jobs are of equal importance. The weight is usually related to the cost. The most common objectives are introduced next.

Makespan is the length of the time to complete all jobs. *Flow time* includes both waiting time and processing time. *Mean flow time* is an important performance measure. Minimizing the *mean flow time* will lower the waiting time of jobs or work-in-process inventory in a system. *Mean flowtime* is a special case of *weighted mean flowtime* where different jobs are equally importance. *Weighted mean flowtime* incorporates both the flowtime and the weight factor into the performance measure. *Lateness* is the amount of time by which the completion time of a job has exceeded its due date. *Tardiness* is the lateness of a job that fails to meet the due date. *Minimizing maximum lateness and tardiness* is an important objective when customers tolerate smaller tardiness but become rapidly and progressively more upset for larger ones. *Minimizing the number of tardy jobs* maximizes schedule performance when missing due date is completely unacceptable. The *due date* is called the *deadline*.

Mean tardiness performance measure only considers jobs that miss the due date. When using this performance measure, there is no benefit driven from early completing jobs. The multi-criteria objective links different single objective into cost penalty. The global objective is to minimize the total cost penalty.

Multi-criteria objectives are most often used in real world situations, it is either explicitly in a mathematical model or implicitly by a human expert scheduler. *Weighted early-tardy objective* is a objective where jobs have different penalty factors. Both early and later completed jobs have penalty. This kind of situation is very common in *just-in-time* manufacturing processes. *Weighted early-tardy-flowtime* is an objective describing the conflicting objectives of inventory level and customer satisfaction. *Mean flowtime* represents *work-in-process* inventory, and total earliness represents finished product inventory. Tardiness represents customer satisfaction.

2.4 Scheduling approaches

Approaches to solving scheduling problems include operations research, dispatching rules, and fuzzy logic approaches.

2.4.1 Operations research approaches

Operations research includes mathematical-analytical methods and heuristic methods. Mathematical-analytical methods developed in the context of the operations research can be characterized theoretically and yields optimal solutions. These methods have their own limitations, particularly in modeling complex real world applications (Slany [21]). The most famous and efficient methods are *dynamic programming* and *branch and bound.* Heuristics search uses all the available information and knowledge toward a solution along the most promising path, omits the less promising ones (Murty [22]). Heuristics methods, which lead to approximate solutions, may not be the optimal solution. The most famous and efficient methods are *beam search, tabu search, simulated annealing, genetic algorithm*, etc. Many practitioners seem to prefer simpler heuristic methods to solving combinatorial optimization problems because:

- there is no guarantee that special algorithm such as branch and bound will give the effective performance on all large-scale problems.
- the real world applications are often complex
- the data available often has unknown errors.

Many heuristic methods do involve some type of search to look for a good approximate solution. Sometime the search for a good approximate solution may be slow and unrewarding (Murty [22]).

2.4.2 Algorithmic and dispatching rules

Scheduling algorithm consists of a set of conditions and rules. If all conditions for a particular algorithm are met and the rules are applied properly, an optimal schedule can be generated. These kinds of algorithms are *Shortest Processing Time* (SPT), *Earliest Due Date* (EDD), *Weight Shortest Processing Time* (WSPT), Moore's algorithm for special objectives of single machine problem, and Johnson's algorithms deal with the special objective of special of two-machine or three-machine problems.

WSPT algorithm, for example, is a simple algorithm which sorts jobs in an increasing order according to the ratio of priority (job's weight) over processing time. In a single machine problem, WSPT algorithm optimizes the *weighted sum of*

Processing time (p_j)	*Priority* (w_j)	*Ratio* ($\frac{p_i}{w_i}$)	*Sequence*
2	5	0.4	1
4	8	0.5	2
4	7	0.57	3
6	9	0.67	4
7	10	0.7	5
5	7	0.714	6
8	10	0.8	7
4	5	0.8	7
6	7	0.857	8
6	6	1	9
10	10	1	9
2	2	1	9
9	9	1	9
10	9	1.11	10
5	4	1.25	11
12	9	1.33	12
14	10	1.4	13
15	10	1.5	14
6	2	3	15
14	2	7	16

Table 2.1: An example of WSPT sequence

completion times and the *weighted mean flow times*. A WSPT sequence is listed in table 2.1.

For a WSPT sequence, the ratio is generated using processing time and priority directly. The sequence generated by the ratio is indirectly related to the two input variables. There is also a nonlinear relationship between the *processing time*, the *priority* and job sequence as it is demonstrated in Section 5.1. WSPT sequence is a typical sequencing problem with a nonlinear and indirect relationship between inputs and output. In this work, the WSPT algorithm is utilized to generate training and validation sets of jobs for scheduling sequence learning.

2.4.3 Fuzzy logic approaches

Fuzzy control has emerged as an attractive application area of Fuzzy Set Theory (Zadeh [1]). Fuzzy control, which is found on fuzzy logic, is attractive because fuzzy logic is able to model the human thinking process. The significant part of fuzzy logic controller is a knowledge base that contains fuzzy logic rules and a fuzzy database. An example of fuzzy linguistic rule defined by Ben-Arieh and Lee [17] is as follows:

> ``IF the Processing time is *low* AND Queue Length is *Long* AND Slack Time is *Zero* AND Machine Breakdown Rate is *Very Small* THEN Select Factor is *Medium*''

where *Low*, *Long*, etc. are linguistic variables. There are two possible approaches to using fuzzy theory in the development of scheduling systems. One is subjective extraction of system behavior knowledge from the experts via protocol interviews (Turksen et al. [15], Caprihan et al. [16], and Ben-Arieh and Lee [17]). Another is objective extraction of system behavior knowledge from input-output data via fuzzy cluster analysis (Turksen et al. [23]), neural network, etc.

In *subjective approaches*, fuzzy knowledge base can be built manually based on expert knowledge (Turksen et al. [15], Caprihan et al. [16], and Ben-Arieh and Lee [17]). The fuzzy sets can be defined by discrete event simulation results (Ben-Arieh and Lee [17]) or by expert knowledge in (Turksen et al. [15], Caprihan et al. [16]) expressed in a linguistic form. Mamdani model is chosen by Caprihan et al. [16] and Ben-Arieh and Lee [17] to model the reasoning process. The output is a fuzzy set and a suitable defuzzification method is used to transfer fuzzy output to a crisp control value. The fuzzy rule base is built by using all the possible combinations of fuzzy linguistic input variables. The input variables selected by Turksen et al. [15] are *priority, slack time* and *requested start time* for job release decisions; *priority, slack time* and *remaining processing time* for job dispatching decisions. The output variable is the *selectability.* The input variables selected by Caprihan et al. [16] are *relative opportunity gain, relative work-in-process.* The output variable is *switching confidence level.* The input variables selected by Ben-Arieh and Lee [17] are *processing time, queue length, slack time* and *machine break down rate.* The output variable is *selected factor.* A job releasing decision and dispatching decision problem in a job shop environment is solved by Turksen et al. [15]. A part routing decision problem in a job shop environment is solved by Ben-Arieh and Lee [17]. And a single machine with two queue sequencing problem is solved by Caprihan et al. [16]. Rule base obtained by subjective approach may have redundant rules, ineffective rules, conflicting rules and missing rules. Even without those rules, a desired performance has to be obtained by the tuning of fuzzy sets.

In *objective approaches*, system knowledge is extracted via fuzzy clustering techniques. When input-output dimensions are high, subjective approaches usually lead to a huge rule base. Eliminating redundant and ineffective rules is impossible (Turksen [23]). Fuzzy C-means clustering technique is used to minimize number of rules. The structure of fuzzy rules is:

> ``IF the lot size is A AND the slack time is B AND the priority is C, THEN the lateness is D.''

where the input variables are *lots size, slack time, priority.* The output variable is *lateness* performance measure. For the same scheduling problem in Turksen et al. [15], the number of rules is reduced to 7, compared with 343 in subjective approach. Input variables *remaining process time* and *request start time* are changed to *lots size.* Consequence part of the rules are changed from *selectability* to *lateness* performance measure.

2.5 Summary

Although operations research approaches can solve problems of moderate sizes and yield optimal solutions for some scheduling problems, they are computationally demanding in solving real life scheduling problems. Many heuristic methods, which involve some type of search, yield good approximate solutions for some large size scheduling problems. Sometimes, the search for a good approximate solution may be slow and unrewarding. Dispatching rules quickly generates an optimal schedule of a special model with particular objective functions. They are not very general but rather problem specific. Subjective approaches model scheduling problems using human expert knowledge. These approaches are promising and flexible in dealing with real world applications. But they are often limited by the knowledge of human experts. Sometimes human experts may or may not be available, or human experts may or may not be able to explain domain knowledge explicitly. Compared to these approaches, modeling a scheduling problem from a schedule itself (input and output data) is more appropriate. The objective approaches take the objective function as output data of a schedule.

Clustering based system identification methods are to be discussed in Chapter 3. These methods can be used to learn schedule decision mechanism from input and output data of an existing schedule. The inputs of a schedule are attributes of a job.

The outputs are the decisions of each job, such as a sequence decision, a routing decision or a timing decision. The existing schedule may be an optimal schedule or a schedule generated by a human expert. Each schedule achieves certain schedule objective functions. A fuzzy expert model is built by learning the scheduling decision from an existing schedule. It can be used to generate optimal or sub-optimal schedule based on the existing schedule it learned.

Chapter 3

Fuzzy clustering and fuzzy clustering based system identification

3.1 Background

System identification based on clustering technique may include structure and parameter identifications. It is generally that system structure is identified before the parameter identification (Sugeno and Yasukawa [5], Emami et al. [12]). Depending on the different clustering techniques and reasoning mechanisms, structure and parameters of the system can be identified at the same time (Chiu [13, 14]).

In this Chapter, an overview of fuzzy clustering methods and clustering based system identification methods are introduced. Subtractive clustering based system identification method and parametric optimization method is also presented. Before doing this, two types of fuzzy models are introduced in the next section.

3.2 The structure of fuzzy model

A fuzzy model could be of types of either multi-input and multi-output (MIMO) or multi-input and single-output (MISO).

The fuzzy model of multi-input and multi-output (MIMO) systems of m inputs and r outputs defined with n rules can be written as

R^1 : IF x_1 is A_1^1 AND x_2 is $A_2^1, \cdots, x_m$ is A_m^1 THEN y_1 is B_1^1 AND y_2 is $B_2^1, \cdots, y_r$ is B_r^1

ALSO

...

ALSO

R^n : IF x_1 is A_1^n AND x_2 is $A_2^n, \cdots, x_m$ is A_m^n THEN y_1 is B_1^n AND y_2 is $B_2^n, \cdots, y_r$ is B_r^n

But in the case of MISO systems, fuzzy rule base will be

R^1 : IF x_1 is A_1^1 AND x_2 is $A_2^1, \cdots, x_m$ is A_m^1 THEN y_1 is B_1^1

ALSO

...

ALSO

R^n : IF x_1 is A_1^n AND x_2 is $A_2^n, \cdots, x_m$ is A_m^n THEN y_1 is B_1^n

where R^i represents $i'th$ rule, x_1, x_2, $\cdots$, x_m are m input variables and y_1, y_2, $\cdots$, y_r are r outputs. A_i^j for $(i = 1, 2, \cdots, m; j = 1, 2, \cdots, n)$ are *linguistic* fuzzy sets of *antecedents* and B_i^j for $(i = 1, 2, \cdots, r; j = 1, 2, \cdots, n)$ are linguistic fuzzy sets of *consequent.*

In these models, both antecedent and consequent parts of the IF-THEN-ALSO rules consist only of fuzzy sets. The output of the model is a fuzzy set. Thus a suitable defuzzification method which translates fuzzy output to crisp control value is required. In other type of models which is the combination of fuzzy and nonfuzzy modeling proposed by Takagi-Sugeno-Kang [24], called Sugeno model, the consequent part is expressed as a linear combinations of antecedents as given below for both MIMO and MISO systems. In the Sugeno model, a MIMO system with n rules, m antecedents and r consequents can be expressed as

R^1 : IF x_1 is A_1^1 AND x_2 is $A_2^1, \cdots, x_m$ is A_m^1

THEN $y_1 = Z_{10}^1 + Z_{11}^1 x_1 + \cdots + Z_{1m}^1 x_m$ AND, $\cdots, y_r = Z_{r0}^1 + Z_{r1}^1 x_1 + \cdots + Z_{rm}^1 x_m$

ALSO

$\cdots$

ALSO

R^n : IF x_1 is A_1^n AND x_2 is $A_2^n, \cdots, x_m$ is A_m^n

THEN $y_1 = Z_{10}^n + Z_{11}^n x_1 + \cdots + Z_{1m}^n x_m$ AND, $\cdots, y_r = Z_{r0}^n + Z_{r1}^n x_1 + \cdots + Z_{rm}^n x_m$

But in the case of MISO systems, Sugeno fuzzy rule base will be

R^1 : IF x_1 is A_1^1 AND x_2 is $A_2^1, \cdots, x_m$ is A_m^1 THEN $y_1 = Z_{10}^1 + Z_{11}^1 x_1 + \cdots +$

$Z^1_{1m}x_m$

ALSO

$\cdots$

ALSO

R^n : IF x_1 is A^n_1 AND x_2 is $A^n_2, \cdots, x_m$ is A^n_m THEN $y_1 = Z^n_{10} + Z^n_{11}x_1 + \cdots +$

$Z^n_{1m}x_m$

where Z^k_{ij} $(i = 1, 2, \cdots, r; j = 0, 1, 2, \cdots, m; k = 1, 2, \cdots, n)$ are consequent regression parameters which are optimized by linear least square estimation as given by Takagi and Sugeno [24]. Both fuzzy and Sugeno model are possible through clustering technique.

3.3 Fuzzy clustering methods

Clustering is a process to obtain a partition G of a set E of N objects x_i $(i = 1, 2, ..., N)$ using a resemblance or disemblance measure, such as a distance measure d between x_i and x_j, where $i, j = 1, 2, ..., N$. The measure is generally of distance or of sequential norms like euclidean, diagonal, mahlanobis, Hamming etc. A partition G is a set of disjoint or partial overlapped subsets of E and the element G_c of G is called a *cluster* and the centers of the clusters are centroids or prototypes.

The purpose of clustering is to distill natural groupings of data from a large data set, such that a concise representation of system's behavior is produced. Each cluster essentially identifies a region in the data space that contains a sufficient mass of data to support the existence of a fuzzy input/output relationship. The sufficiency of data requires large amount of training data for each system behavior in order to build a fine model (Chi [25]). But the real world data is often imprecise and/or incomplete. Thus effective grouping data to capture system behavior based on scarce data becomes an important issue.

3.3.1 Fuzzy C-means clustering algorithm

The well known *fuzzy C-means* (FCM) clustering algorithm is proposed by Dunn [26] and later developed by Bezdek [10] and Bezdek et al. [11]. FCM enables us to make flexible partitions of a finite data set. FCM is used in many fields such as data analysis and image segmentation (Imai et al. [27]).

FCM needs a priori knowledge of the number of clusters. The FCM will form iteratively a suitable cluster pattern in such a way that an objective function dependent of cluster locations is minimized. The objective function is defined as

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} {\mu_k^i}^m \|x_k - v_i\|^2 \tag{3.1}$$

where n is the number of data points, c is the number of clusters, x_k in R^p is the $k'th$ $p-$dimensional data point, $v_i \in R^p$ is the $i'th$ $p-$dimensional cluster center, μ_k^i is the degree of membership of $k'th$ data in the $i'th$ cluster, and m is a weight factor greater than one. The degree of membership μ_k^i is defined as

$$\mu_k^i = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|}\right)^{\frac{2}{(m-1)}}}. \tag{3.2}$$

FCM requires a desired number of clusters c and an initial guess position for each cluster center $v_i, (i = 1, 2, \cdots, c)$. The FCM output rules depend strongly on the choice of these initial values. The constraint on membership degree causes the FCM to generate memberships that can be interpreted as degrees of *sharing* but not as degrees of *typicality* (Krishnapuram and Leller [28]). Thus, the memberships in a given cluster of two points that are equidistant from the prototype of the cluster can be significantly different. Also memberships of two points in a given cluster can be equal even though the two points are arbitrarily far away from each other.

3.3.2 Mountain clustering method

The autogeneration capability for determining the number and initial location cluster centers through search techniques was introduced by Yager and Filev [6, 7] in *mountain clustering* method. This method can provide approximate estimation of cluster centers based on a search measure called the mountain function. A data point with the highest mountain function represents a cluster center. In order to find cluster centers, data space is discretized into grid points and each grid point is assumed to be a potential candidate for cluster centers. Grid points could be evenly or unevenly spaced to reflect a priori knowledge. The potential value for each grid point based on its distance measure to the actual data points is estimated in order to select those with high potential values as cluster centers. Once a cluster center with highest mountain function is selected, all the grid points are penalized in proportion to the distance from the cluster center. Hence, the grid points that are close to the cluster center are penalized more than the farther ones. This potential penalization reduces the possibility of forming other clusters close to the previous one. It also increases the possibility of new cluster centers creation away from the previous center so that the entire data space is represented with minimum clusters. The grid point with the new maximum mountain function is selected as a new cluster center and all the grid points are penalized in proportion to the distance from the new cluster center. This procedure of identifying cluster centers and penalizing all grid points continue as long as the mountain function falls below a threshold potential value.

Even though this method is simple, its application becomes limited due to a large amount of computation involved when the dimensions of data space are large.

3.3.3 Subtractive clustering method

Chiu [13, 14] proposes a *subtractive clustering* method with improved computational effort in which data points themselves are considered as the candidates for cluster centers instead of grid points. By using this method, the computation is simply proportional to the number of data points and is independent of the dimension of the problem. In this method, a data point with the highest potential, which is a function of the distance measure, is considered as a cluster center. The data points that are close to new cluster center are penalized in order to facilitate the emergence of new cluster centers.

In subtractive clustering method, the potential of each data point is estimated by the following equation:

$$P_i = \sum_{j=1}^{n} e^{-\alpha\|x_i - x_j\|^2}, \tag{3.3}$$

where

$$\alpha = \frac{\gamma}{{r_a}^2},$$

P_i is the potential of $i'th$ data point, n is the total number of data points, x_i and x_j are data vectors in data space including both input and output dimensions, γ is the *shape factor* of the a cluster. It is a positive constant and is selected as 4, and r_a is a positive constant defining the neighborhood range of the cluster or simply the radius of hypersphere cluster in data space. The potential is a function of its distances to all other data points (also includes itself). The more the neighborhood data points, the higher the potential value is. Each time a cluster center is obtained, the revising of the potential is done using the following equation:

$$P_i = P_i - P_k^* e^{-\beta\|x_i - x_k^*\|^2}, \tag{3.4}$$

where

$$\beta = \frac{4}{{r_b}^2},$$

and

$$r_b = \eta * r_a \tag{3.5}$$

P_k^* is the potential of *k'th* cluster center, x_i is *i'th* data point being subtracted and x_k^* is *k'th* cluster center. A positive constant, r_b, defines the efficient subtractive range. *Squash factor* η is a positive constant greater than 1. The positive constant r_b is somewhat greater than r_a. It avoids closely spaced cluster centers.

Similar to Yager and Filev's mountain method, the process of acquiring new cluster center is based on potential value in relation to an upper acceptance threshold $\bar{\epsilon}$, lower rejection threshold $\underline{\epsilon}$, and the relative distance criterion. The upper acceptance threshold $\bar{\epsilon}$ and lower rejection threshold $\underline{\epsilon}$ are ratio between 1 to 0. The upper acceptance threshold $\bar{\epsilon}$ should be a ratio greater than the lower rejection threshold $\underline{\epsilon}$. A data point with the potential greater than the upper acceptance threshold is directly accepted as a cluster center. The acceptance of a data point with a potential between the upper and the lower thresholds depends on the relative distance equation, defined as:

$$\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1 \tag{3.6}$$

where d_{min} is the shortest distance between the candidate cluster center and all previously found cluster centers. Relative distance function, which tries to balance between data potential and distance to all previously found clusters, avoids the emerging of new clusters close to the existing ones. The proportion of P_k^* to P_1^*, where P_1^* the maximum potential, may be small. Hence relative distance function very often positions clusters far away from the previous clusters. After several potential evolutions, relative distance function will improve the distribution of cluster centers.

Once the clusters are formed in the input and output space. They are projected into each dimension. Chiu [13, 14] proposes the following equation for assigning exponential type membership degree in the input space.

$$\mu_i^j = e^{-\alpha\|x_i - x_j\|^2}, \tag{3.7}$$

where

$$\alpha = \frac{\gamma}{{r_a}^2},$$

and $\|x_i - x_j\|$ is the distance measure between the *i'th* data point and *j'th* cluster. γ is a positive constant determining fuzziness in the cluster. The value γ is important in generating meaningful fuzzy sets. As an initial guess, Chiu [13] uses $\gamma = 4$ to make fuzzy sets close to triangular membership functions. The value $\gamma = 4$ is chosen in this work. This parameter can also be used to tune the fuzzy sets in order to absorb small variations in the system after it is identified (Demirli and Muthukumaran [29]).

The subtractive clustering algorithm is summarized below:

```
Initialize parameters.
Calculate potential for each point using Equation (3.3).
Set the maximum potential as $P_1^*$
Pick up data point with the current maximum potential $P_k^*$ as the
cluster candidate.
Accepting operation:
```

```
if $P_k^* > \bar{\epsilon} * P_1^*$,
    Accept $x_k^*$ as a cluster center, subtract potential for each point using Equation (3.4) and continue.
Rejecting operation:
else if $P_k^* < \underline{\epsilon} P_1^*$
    Reject $x_k^*$ and end the clustering process.
Discard operation:
else
    Let $d_{min}$ = shortest of the distances between $x_k^*$ and all previously found cluster centers.
    If $\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1$
        Accept $x_k^*$ as a cluster center, subtract potential for each point using Equation (3.4) and continue.
    else
        Reject $x_k^*$ and set the potential at $x_k^*$ to 0.  Select the data point with the next highest potential as the new $x_k^*$ and retest
    endif
endif
```

Subtractive clustering has four parameters, *accept ratio* $\bar{\epsilon}$, *reject ratio* $\underline{\epsilon}$, *cluster radius* r_a and *squash factor* η (or r_b). These parameters have influence on the *number of rules* and *error performance measures.* Large values of $\bar{\epsilon}$ and $\underline{\epsilon}$ will result in small number of rules. Conversely small values of $\bar{\epsilon}$ and $\underline{\epsilon}$ will increase the number of rules. A large value of r_a generally results in fewer clusters that lead to a coarse model. A small value of r_a can produce excessive number of rules that results in an over-defined system. The optimal parameters suggested by Chiu [13, 14] are $1.25 \leq \eta \leq 1.5$ and $0.15 \leq r_a \leq 0.30$.

3.4 Fuzzy clustering based system identification

For clustering based system identification, there are two ways that clustering can be use in extracting system knowledge. The clustering of data space could be done in output space only or in combined input-output space as explained by Demirli et al. [30].

3.4.1 Clustering in output space and projection onto input space (COSPI)

In this approach, clustering is done in output space. Cluster is formed only in output dimensions. The distance measure for clustering could depend on the density of the data or the closeness to the required objective function only in the output space of r dimensions for MIMO type or one dimension in MISO type. The degree of association of any data point in an output space to a cluster depends on the distance measure to that cluster center. Once clusters in the output space are formed, they are projected onto the input space of m dimensions as shown in Figure 3.1. Membership function of a data point is maintained in its projections onto input space. These projections transfer the convex fuzzy sets of the output space into non-convex ones in the input space as shown in Figure 3.1. These non-convex input fuzzy sets need to be approximated to convex sets for further evaluation. As the cluster formation depends only on the outputs, the variables in the input space do not influence the reasoning mechanisms behind the cluster formation. Thus this method ensures that insignificant input variables do not participate in the reasoning mechanism. On the other hand, if the significant input variables do not participate in the reasoning mechanism, a totally faulty model may evolve. This method is ideal for systems of large dimensions. But it is not as accurate because it does not include the input space into reasoning process. Hence the fuzzy sets obtained using this method may require more tuning in the parameter identification phase.

3.4.2 Clustering in input-output space (CIOS)

Clustering of data is done in the combined input-output space of $(m + r)$ dimensions as shown in Figure 3.2. Distance measure used for the association of a data point to a cluster is a combined distance in input-output space where output variables are considered as other dimensions in the data space. Once the clusters are formed in the input-output space, they are projected into each dimension. These projections transfer convex fuzzy sets in combined space into convex fuzzy sets in every dimension of input and output spaces. Therefore no further approximation of fuzzy sets is needed. The CIOS method leads to a more accurate system than using COSPI method. The fuzzy sets obtained using CIOS method requires less tuning because input variables contribute equally to the reasoning process, i.e., fuzzy cluster

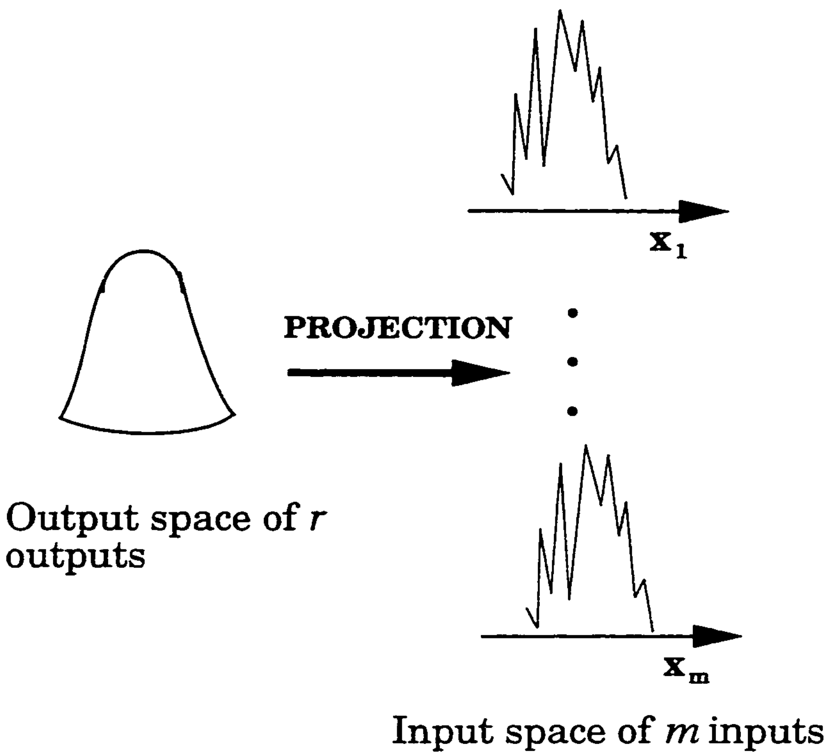


Figure 3.1: Clustering in output space and projection onto input space (COSPI)

formation when compared to output variables. The CIOS method have difficulty when dimensions of the system is very large, because of the high computational demand. Thus in dealing with system with large dimensionality, CIOS method requires clustering algorithm which is computationally efficient and is independent of the dimensions of data space.

3.4.3 Elimination of insignificant inputs

Clusters should be formed on the right training data. The training data may contain noise and insignificant variables. If the whole data space is used to identify a system, it could lead to a faulty model of the system. Elimination of insignificant variables

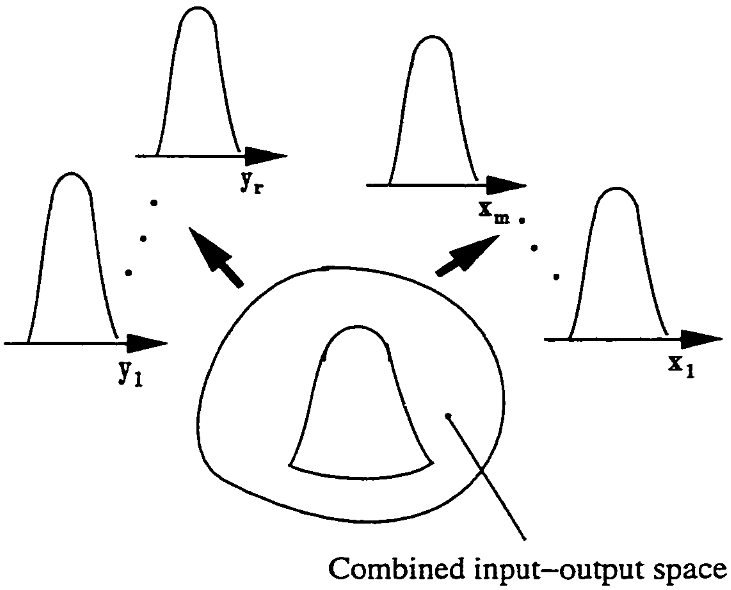


Figure 3.2: Clustering in input-output space (CIOS)

is important and it should be done before system identification. Significance test needs to be carried out on each dimension of the data space to eliminate insignificant variables. It ensures the real relevance between the model and the data space. It is appropriate to use COSPI method to build models for significant data identification, because COSPI method is independent of input dimension. Using COSPI method, different models are built and insignificant input variables are identified and then eliminated.

In this thesis, the elimination procedure is carried out in two steps. First, different models (rule bases) were created for various cluster radii. The radii are chosen so that the model can cover the entire output space. The insignificance tests are carried out for all of these models. The variables that are insignificant in all the models are eliminated form further identification of the actual system.

Different models are built by using COSPI method. The subtractive clustering method is used to form clusters in the output space only. The *quantitative index* π_j, as a measure of insignificance proposed by Emami et al. [12], is used in this work to identify insignificance of input variables. The quantitative index π_j is defined as:

$$\pi_j = \prod_{i=1}^{n} \frac{\Gamma_{ij}}{\Gamma_j}, j = 1, 2, \ldots, m \tag{3.8}$$

where Γ_{ij} represents the *core* of membership function for variable j in rule i, and Γ_j represents the *support* of variable j in rule i. The smaller the quantitative index π_j is, the more significant the variable j is. The selection of significant inputs is explained in the following section.

Subtractive clustering method partitions output space into several clusters. Each cluster represents a membership function in output space of a rule. Membership functions are exponential functions as defined in Equation (3.7). For each membership function, the *core* is formed by approximating membership degrees with value greater than 0.90 to 1. The *Support* is formed by taking of membership degrees greater than 0.05. As mentioned in Section 3.4.1, when projecting output clusters onto input space, the membership functions are not always convex. For variable identification, only the *core* Γ_{ij} and *support* Γ_j need to be projected to calculate quantitative index. The quantitative index calculated for all the input variables of each models are calculated using Equation (3.8). The input variable which scores very high in insignificance measure is eliminated from further modeling.

3.5 Subtractive clustering based system identification

In Sugeno and Yasukawa [5], Emami et al. [12], COSPI method is used to identified system where FCM is used to extract system knowledge from input and output data. FCM needs a priori knowledge of the number of clusters. Thus an optimal structure of the model (number of clusters or rules) should be identified before the parameter identification. Mamdani type model is used. Because the clustering is formed only in output space, the tuning of fuzzy set is required.

In Chiu [13, 14], a subtractive clustering based system identification method is introduced. The CIOS method and the *linear least square* (LSE) method are used for system identification from input and output data. CIOS method is used to identified structure (number of rules) and the antecedent parts of the model. As mentioned in Section 3.4.2, the CIOS method leads to a more accurate system than using COSPI method, because input variables and output variables contribute equally to the reasoning process. Subtractive clustering algorithm is used to extract system knowledge from input and output data. With a given set of parameters for the subtractive clustering algorithm, the number of clusters is automatically generated. Each cluster defines a rule. For Sugeno model, after projecting the clusters into the each input dimension, the antecedent of the model is constructed. The structure of the model and the antecedent parts of the model are identified at the same step. The consequent parameters are identified using LSE method. As pointed out by Takagi and Sugeno [24], given a set of rules with fixed antecedent parameters, optimizing the parameters in the consequent equations with respect to training data reduces to a LSE problem. LSE method can solve this problem and the solution is always globally optimal. Sugeno zero order and first order models are investigated in Chiu [13, 14].

In this thesis, the insignificant variable identification is eliminated by using COSPI method introduced in Section 3.4.3. Subtractive clustering based system identification method, which includes CIOS and LSE methods, is used to identify system from input and output data. Both Sugeno ([24]) first order and Demirli and Muthukumaran ([29]) second order models are investigated. The identification of consequent parameters of Sugeno first order and second order model are introduced in the next section.

3.5.1 Consequent identification for Sugeno first order model by LSE

The Sugeno model is a special model in which consequent parts are linear combinations of antecedents as shown in Section 3.2. The consequent parameter vector Z for first order model with n rules, two antecedents, and one consequent is defined as

$$Z = [Z_{10}^{1}, \cdots, Z_{10}^{n}, Z_{11}^{1}, \cdots, Z_{11}^{n}, Z_{12}^{1}, \cdots, Z_{12}^{n}]^{T} \tag{3.9}$$

Once the antecedent part is identified, the output y for the given input vector (x_1, x_2) can be predicted using the inference mechanism defined by the following equation:

$$y = \frac{\sum_{i=1}^{n}(\mu_1^i \wedge \mu_2^i) \cdot (Z_{10}^i + Z_{11}^i \cdot x_1^i + Z_{12}^i \cdot x_2^i)}{\sum_{i=1}^{n}(\mu_1^i \wedge \mu_2^i)} \tag{3.10}$$

where firing strength $\mu_1^i = A_1^i(x_1)$ and $\mu_2^i = A_2^i(x_2)$. This can be rewritten as

$$y = \sum_{i=1}^{n} \beta_i (Z_{10}^i + Z_{11}^i \cdot x_1^i + Z_{12}^i \cdot x_2^i)$$

$$= \sum_{i=1}^{n}(Z_{10}^{i} \cdot \beta_i + Z_{11}^{i} \cdot x_1^i \cdot \beta_i + Z_{12}^{i} \cdot x_2^i \cdot \beta_i) \tag{3.11}$$

where

$$\beta_i = \frac{\mu_1^i \wedge \mu_2^i}{\sum_{j=1}^{n}(\mu_1^i \wedge \mu_2^i)} \tag{3.12}$$

The intersection operator $\wedge$ takes minimum of μ_1^i and μ_2^i as the firing strength for the inputs (x_1, x_2) of the $i'th$ rule. The β_i is the weighted firing strength for inputs (x_1, x_2). When enough training data is obtained, the consequent parameter vector Z is estimated by pseudo-inverse solution given by

$$Z = (X^T X)^{-1} X^T Y \tag{3.13}$$

where

$$Y = [y_1, \cdots, y_m]^T \tag{3.14}$$

$$X = \left| \begin{array}{cccccc} \beta_{11}, & \beta_{21}, & \beta_{11} \cdot x_1^1, & \beta_{21} \cdot x_1^1, & \beta_{11} \cdot x_2^1, & \beta_{21} \cdot x_2^1 \\ \vdots & & & & & \vdots \\ \beta_{1m}, & \beta_{2m}, & \beta_{1m} \cdot x_1^1, & \beta_{2m} \cdot x_1^1, & \beta_{1m} \cdot x_2^1, & \beta_{2m} \cdot x_2^1 \end{array} \right| \tag{3.15}$$

$$\beta_{ij} = \frac{\mu_{1j}^i \wedge \mu_{2j}^i}{\sum_j(\mu_{1j}^i \wedge \mu_{2j}^i)}. \tag{3.16}$$

The β_{ij} is the weighted firing strength for inputs (x_1, x_2) of $j'th$ rule. The matrix X is the input matrix of known values obtained using weighted firing strengths and input parts of training data. It is of the order of $(m \times 3 \cdot n)$. Vector Y is the known output vector of training data and is of the order of $(m \times 1)$ and the regression parameter vector Z is of the order of $(3 \cdot n)$.

3.5.2 Consequent identification for Sugeno second order model by LSE

The second order Sugeno model uses the same antecedent parameters as the first order Sugeno model. The difference is in the consequent part which is expressed in Equation (3.17):

$$Y = Z_{10}^i + Z_{11}^i \cdot x_1 + Z_{12}^i \cdot x_2 + Z_{13}^i \cdot {x_1}^2 + Z_{14}^i \cdot {x_2}^2 \tag{3.17}$$

thus

$$y = \frac{\sum_{i=1}^{n}(\mu_1^i \wedge \mu_2^i) \cdot (Z_{10}^i + Z_{11}^i \cdot x_1^i + Z_{12}^i \cdot x_2^i + Z_{13}^i \cdot {x_1^i}^2 + Z_{14}^i \cdot {x_2^i}^2)}{\sum_{i=1}^{n}(\mu_1^i \wedge \mu_2^i)} \tag{3.18}$$

which can be rewritten as

$$\begin{aligned} y &= \sum_{i=1}^{n} \beta_i (Z_{10}^i + Z_{11}^i \cdot x_1^i + Z_{12}^i \cdot x_2^i + Z_{13}^i \cdot {x_1^i}^2 + Z_{14}^i \cdot {x_2^i}^2) \\ &= \sum_{i=1}^{n} (Z_{10}^i \cdot \beta_i + Z_{11}^i \cdot x_1^i \cdot \beta_i + Z_{12}^i \cdot x_2^i \cdot \beta_i + Z_{13}^i \cdot {x_1^i}^2 \cdot \beta_i + Z_{14}^i \cdot {x_2^i}^2 \cdot \beta_i) \end{aligned} \tag{3.19}$$

and β_i is obtained by using Equation (3.12). Consequent parameter vector is give by Equation (3.13), where

$$Z = [Z_{10}^1, \cdots, Z_{10}^n, Z_{11}^1, \cdots, Z_{11}^n, Z_{12}^1, \cdots, Z_{12}^n, Z_{13}^1, \cdots, Z_{13}^n, Z_{14}^1, \cdots, Z_{14}^n]^T \tag{3.20}$$

$$X = \begin{vmatrix} \beta_{11}, & \beta_{21}, & \beta_{11} \cdot x_1^1, & \beta_{21} \cdot x_1^1, & \beta_{11} \cdot x_2^1, & \beta_{21} \cdot x_2^1, & \\ & & & \beta_{11} \cdot {x_1^1}^2, & \beta_{21} \cdot {x_1^1}^2, & \beta_{11} \cdot {x_2^1}^2, & \beta_{21} \cdot {x_2^1}^2 \\ \vdots & & & \cdots & & & \vdots \\ \beta_{1m}, & \beta_{2m}, & \beta_{1m} \cdot x_1^1, & \beta_{2m} \cdot x_1^1, & \beta_{1m} \cdot x_2^1, & \beta_{2m} \cdot x_2^1, & \\ & & & \beta_{1m} \cdot {x_1^1}^2, & \beta_{2m} \cdot {x_1^1}^2, & \beta_{1m} \cdot {x_2^1}^2, & \beta_{2m} \cdot {x_2^1}^2 \end{vmatrix} \tag{3.21}$$

where the β_{ij} is obtained using Equation (3.16). The matrix X is of the order of $(m \times 5 \cdot n)$ and vector Z is regression parameter vector of the order of $(5 \cdot n)$.

3.6 System optimization

With a set of parameters, the clusters extracted using subtractive clustering algorithm may not be the optimal representation of training data. Thus the antecedent part built based on these clusters may not be the optimal either. Although optimal consequent parameters are obtained from LSE for the antecedent parts, the model might just be a local optimal model. Thus optimization of the model requires searching a set of optimal parameters for subtractive clustering algorithm.

In this thesis, the optimization of the model is carried out by a partial enumerated parametric search. The parametric search is performed over *accept ratio* $\bar{\epsilon}$, *reject ratio* $\underline{\epsilon}$, *squash factor* η and *cluster radius* r_a. The enumeration is partial because there are different step sizes for the *cluster radius* r_a. The different step sizes are closely related to the error measure. The following heuristic rule is used to control the step size:

> If error is *large*, make a jump by *large* step size Else make a small jump by *small* step size

The value of the *large* error, the value of *large* and *small* jump step sizes are determined from simulation results. For different problems, there are different values of error measures and step sizes of parameters that could be obtained through simulation.

3.7 Sugeno reasoning process

Once an optimal model is obtained, it is used to make control actions or decisions. The reasoning algorithm of the Sugeno model is demonstrated using an example introduced by Takagi and Sugeno [24] in Figure 3.3.

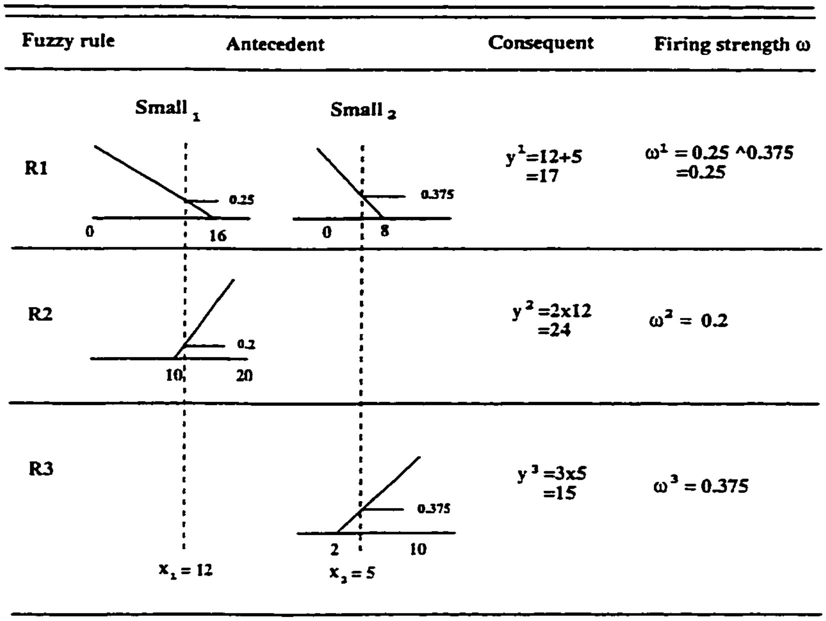


Figure 3.3: Example of Sugeno reasoning process

The example in Figure 3.3 is a Sugeno first order model with three rules:

R^1 : IF x_1 is $small_1$ AND x_2 is $small_2$ then $y = x_1 + x2$

R^2 : IF x_1 is big_1 then $y = 2 \times x1$

R^3 : IF x_2 is big_2 then $y = 3 \times x2$

Figure 3.3 shows the reasoning process by each rule when given crisp inputs $x_1 = 12$, $x_2 = 5$. The “antecedent” column in Figure 3.3 shows the membership functions of the fuzzy sets “$small_1$” and “ $small_2$” in the antecedent parts. The “Consequent” column shows the calculation of y^i. The calculation of the firing strength ω^i is shown in the fourth column. The value inferred by the rules is

$$\begin{aligned} y &= \frac{\omega^1 \times y^1 + \omega^2 \times y^2 + \omega^3 \times y^3}{\omega^1 + \omega^2 + \omega^3} \\ &= \frac{0.25 \times 17 + 0.2 \times 24 + 0.375 \times 15}{0.25 + 0.2 + 0.375} \\ &= 17.8 \end{aligned}$$

3.8 The summary of the modeling method

To summarize the modeling method introduced in this chapter, the steps of modeling procedures are

1. Eliminate insignificant variables by using COSPI method
2. Initialize parameters for subtractive clustering method
3. Build fuzzy expert model by using CIOS and LSE methods
    - (a) Form clusters from training data
    - (b) Project clusters onto the input space and get antecedent parts of a model
    - (c) Identify consequent parameter parts and complete the model
4. Evaluate model, and then

    IF good model is found THEN validate this model with more data

```
5. Change parameters, then GOTO 3.
```

Subtractive clustering based system identification method is introduced in this chapter. It can be used to model system based on input and output data.

Chapter 4

Scheduling decision learning

4.1 Introduction

Two fuzzy logic based approaches to solving scheduling problems are discussed in Chapter 2. These are: subjective approaches and objective approaches. In subjective approaches, human expert knowledge is used to model scheduling problems. These approaches are promising and flexible in dealing with real world applications. But the quality of the schedule is limited by the human expert's knowledge. Sometimes, human expert does not exist, or may not be available, or an expert may not be able to explain domain knowledge explicitly. In objective approaches, FCM is used to extract scheduling knowledge and to optimize the scheduling model. In these approaches, scheduling problems are solved by modeling input and output data. The output is a scheduling objective function.

The scheduling objective function is the goal of a schedule to maximize or to minimize. It is used to evaluate the performance of a schedule. The scheduling objective function is achieved by the outputs of a schedule, which are also called decisions of jobs. The decisions are made based on the inputs of the schedule, which are also called attributes of jobs. The decisions are used to schedule jobs to maximize or to minimize certain scheduling objective functions. Thus modeling scheduling problem to achieve certain objectives from input and output data is to model job attributes and scheduling decisions.

In this thesis, a general and appropriate approach to solving scheduling problem is introduced. Subtractive clustering based system identification method is used to learn scheduling decisions from input-output data of a schedule. It builds a fuzzy expert model, where the inputs are job attributes and the outputs are scheduling decisions. Learning from a schedule overcomes the limitation of using human expert knowledge. A fuzzy expert model built by learning schedule decisions can be used to generate a new schedule when there are new jobs.

Subtractive clustering based system identification method introduced in Chapter 3 is used to learn scheduling decision mechanism from an existing schedule, and to build a fuzzy expert model. The fuzzy expert model is then used to generate new schedules following the decision mechanism it has learned. The existing schedule can be an optimal schedule generated by an optimization method, or a schedule generated by a human expert. A decision objective function, sequential distance error, is introduced for the sequence learning applications. The implementation of this method is demonstrated by modeling a single machine weighted flowtime problem.

A single machine problem is simple and important because it often is the fundamental element of more complex problems. Selecting a single machine weighted flowtime problem to verify the modeling method is due to the following reasons.

Firstly, the WSPT sequence optimizes the single machine weighted flowtime problem. Learning the WSPT sequencing mechanism is learning from an optimal sequence. Success in capturing the WSPT sequencing mechanism shows that an optimal schedule can be learned. The model can generate optimal schedule within certain scopes. Secondly, the fuzzy reasoning expressions of the input and output relationship can be observed intuitively. The testing data surface of a complete set of jobs is shown in Figure 4.1, where the x-axis is processing time, the y-axis is priority and the z-axis is job sequence. The data surface of the complete set of jobs is a spiral shaped data surface.

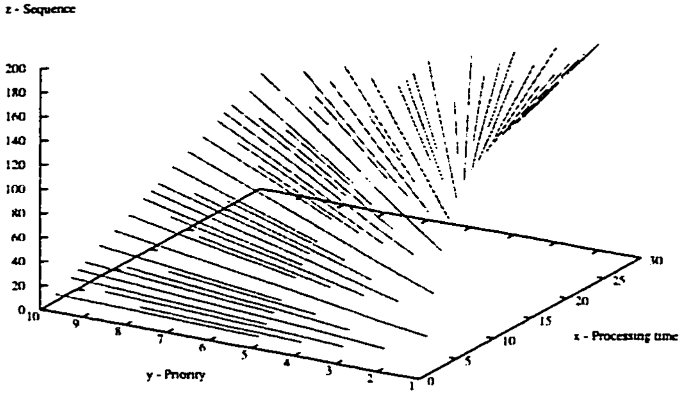


Figure 4.1: WSPT surface

Although, a single machine weighted flowtime problem is a simple scheduling problem, it is not an easy modeling problem. The WSPT surface shown in Figure 4.1 is a complex surface. There are nonlinear relationships between input and output. It is a very difficult problem for modeling by using clustering algorithm when global behavior the system requires to be predicted based on scarce system information. When using this method to learn scheduling decision to solving complicated scheduling problem for complicated objective functions, the level of difficult of modeling should not increase.

One of the objectives of this thesis is to learn scheduling decision mechanism from an existing schedule. Subtractive clustering based system identification method is used to extract and model decision mechanism from the schedule. The scheduling decisions are made based on the attributes of jobs. All attributes of a job may not be related to or may not be significant in scheduling decisions. Thus the elimination of the insignificant input variables introduced in Section 3.4.3 is required before the identification of the scheduling model.

The scheduling decision learning process includes the following steps:

step 1. Generate training data

1. Project attributes of jobs onto the scheduling decisions.
2. Eliminate insignificant variables (COSPI)

step 2. Build (Sugeno) model by subtractive clustering based system identification method.

1. Obtain number of rules and premise parameters using CIOS method.
2. Identify consequent parameters using LSE method.

A schedule can be a single machine schedule or a multi-machine schedule. A schedule for a multi-machine case can be decomposed to a schedule for each machine. Thus there can be two approaches to solving the multi-machine problem: A fuzzy expert model for each machine or for all machines. A fuzzy expert model for each machine makes the sequencing decision more accurate, because the sequencing decision may vary form machine to machine. The combination of single machine solutions makes a suboptimal solution. On the other hand, it is time consuming to build fuzzy expert model for every machine than one fuzzy expert model for all machines. When the scheduling problems include more than one decision, a model with multi-input and multi-output or a model with multi-input and single-output for each decision can be considered. Although subtractive cluster based system identification method and Sugeno reasoning mechanism can deal with MIMO cases. In this thesis, only a single machine, sequencing decision learning application is investigated.

Compared to the other fuzzy logic based approaches, the advantages and differences of scheduling decision learning approaches are summarized below:

1. Fuzzy logic based approaches use human expert knowledge to build a fuzzy expert model. The scheduling decision learning approaches learn a human expert's scheduling reasoning mechanism or the reasoning behind an optimization method by using input and output data.

2. Fuzzy logic approaches in Turksen [23] utilize special scheduling performance measures, such as lateness. The scheduling objective function is considered as the schedule outputs. It is different from the subjective approaches, which use schedule decisions as the schedule output. The scheduling decision learning approach is more general. Fuzzy expert model is build based on input and output of a schedule. The objective function is related to the attributes of jobs, and the outputs of a schedule are the scheduling decisions.

3. In objective approaches in Turksen [23], even though the model is built based on single objective function as an output, several scheduling objective functions are computed in order to evaluate the model. These scheduling objective functions are "tardiness", "earliness", "maximum lateness" and "WIP". In scheduling decision learning approach, the fuzzy expert model is evaluated by its capability in capturing scheduling decision mechanism. Although the modeling is independent of the scheduling objective functions, correctly capturing the scheduling decision mechanism means the model is capable of generating optimal or sub-optimal schedule for the same scheduling objective function as the training schedule.

4. Objective approaches reported by Turksen [23] and scheduling decision learning approaches both use clustering algorithm. Turksen [23] uses FCM to optimize the knowledge base. The scheduling decision learning approach uses subtractive clustering algorithm to optimally extract scheduling decision mechanism.

5. Fuzzy logic approaches optimize fuzzy expert model by tuning the fuzzy sets. The scheduling decision learning approach optimizes fuzzy expert model through parametric search.

6. In this thesis, different validation strategies are used. It is shown that the fuzzy expert model is robust with respect to large number of new jobs. In an objective approach ([23]), a third of total jobs is used for validation. It is used as the test data to validate "the goodness of the model". In verification of scheduling sequencing learning application, validation is to verify the ability of the model in capturing the sequencing mechanism.

An application of the scheduling decision learning, which is a learning of WSPT sequencing mechanism, is introduced in Section 4.2. A sequential objective function is presented in Section 4.2.1. The generation of training and validation data is discussed in Section 4.2.2. And the design of parametric search for WSPT sequence learning is discussed in Section 4.2.5.

4.2 An application of decision learning: Learning WSPT sequencing mechanism

4.2.1 Objective function of sequencing decision learning

An optimal sequence optimizes certain scheduling objectives or performance measures. The modeling of sequence should not miss any job from sequencing and should maintain right sequence of all jobs. In order to quantify the capability of the model in capturing sequential mechanism, a sequential error measure σ is introduced. It is defined as

$$\sigma = \sum_{j=1}^{n} \left| S_j^p - S_j^t \right|^a \tag{4.1}$$

where n is total number of training jobs, S_j^p is the *predicted sequence* using the model output for the $j'th$ job, and S_j^t is the *target sequence* or *training* or *validation sequence* for the $j'th$ job, and a is a penalty factor to be enforced on the difference in the sequence. The factor a is set to 1 because the model is required to be captured the training sequence correctly. Sequential measure $\sigma = 0$ implies that all the sequences are predicted correctly by the model. Larger value of σ corresponds to many situations, such as many jobs are sequenced incorrectly with small differences in sequencing, or a few jobs are sequenced incorrectly with large difference in sequence, or the combination of both conditions.

There are two sequential objectives in this work, σ_t and σ_v, where σ_t is *training sequential error* and is related to prediction errors of training jobs. The *validation sequential error* is σ_v, and it is related to the sequential prediction of new jobs. The smaller the validation sequential error σ_v is, the more robust the sequencing model. A small validation sequential error σ_v indicates that the model has a high generalization ability. Thus an optimal model should have the smallest validation error and a small number of rules. Two examples for the calculation of sequential error σ are given in Tables 4.1 and 4.2. The same target sequence number S_t indicates that more jobs has same selectability. For WSPT sequence, these jobs have same ratio of *priority* over *processing time.* The S_j^f is the *selectability factor* and the S_j^d is the *sequence decision.* The higher the selectability S_j^f of a job, the smaller the sequence decision S_j^d and the earlier the processing of a job are.

These two factors are explained in Section 4.2.6.

Example 1:

j	S_j^t	S_j^f	S_j^d	S_j^p
1	1	0.8	1	1
2	2	1.9	3	2
3	2	1.7	2	2
4	3	2.9	4	3

Table 4.1: Example of recreating job sequence and sequential error σ calculation, $\sigma = 0$

For the example 1 in Table 4.1, the sequential error σ is calculated as:

$$\sigma = \sum_{j=1}^{4} \left| S_j^p - S_j^t \right|^1$$

$$\begin{aligned}
&= \left|S^p_1 - S^t_1\right|^1 + \left|S^p_2 - S^t_2\right|^1 + \left|S^p_3 - S^t_3\right|^1 + \left|S^p_3 - S^t_3\right|^1 \\
&= |1-1|^1 + |2-2|^1 + |2-2|^1 + |3-3|^1 \\
&= 0
\end{aligned}$$

Example 2:

j	S^t_j	S^f_j	S^d_j	S^p_j
1	1	0.8	1	1
2	2	1.9	2	2
3	2	2.6	4	3
4	3	2.4	3	2

Table 4.2: Example of recreating job sequence and sequential error σ calculation, $\sigma = 2$

For the example 2 in Table 4.2, the sequential error σ is calculated as:

$$\begin{aligned}
\sigma &= \left|S^p_1 - S^t_1\right|^1 + \left|S^p_2 - S^t_2\right|^1 + \left|S^p_3 - S^t_3\right|^1 + \left|S^p_3 - S^t_3\right|^1 \\
&= |1-1|^1 + |2-2|^1 + |3-2|^1 + |2-3|^1 \\
&= 2
\end{aligned}$$

Once a model is built, the training or validation jobs become inputs to the model. The model gives each job a selectability factor S^f_j following the reasoning process introduced in Section 4.2.3. The selectability factor S^f_j shown in the second column of Tables 4.1 and 4.2 are real numbers. The sequence decision S^d_j shown in the fourth column is calculated based on the increased order of selectability factor S^f_j. The predicted sequence in the fifth column used in the calculation of sequential error is generated from target sequence S^t_j and sequence decision S^d_j. Taking the example 2 in Table 4.2, at first the S^d_j is sorted in an increased order. Then the S^t_j

become 1, 2, 3, and 2 which are the predict sequences S_j^p. It can be seen that $\sigma = 2$ is caused by the wrong sequence order of the third and the fourth jobs.

4.2.2 Training and validation sets of jobs

In order to validate the model, a set of validation jobs is built for each set of training jobs. Each set of training jobs has three input variables and one output variable. Three input variables: *processing time*, *due date* and *priority*. Data for these variables are randomly generated by the ranges using for *processing time* 1 to 30, for *due date* 1 to 1500, and priority for 1 to 10. The output is a sequence of jobs, ordered using the WSPT algorithm which minimizes the *weighted sum of completion times* and the *weighted mean flow times*. A sequence decision is made based on the *processing time* and the *priority*. Due date variable has no influence on the WSPT sequence. The design and the use of due date variable are to check the algorithm of eliminating insignificant input variables.

Three sets of jobs: *complete set*, *moderate set* and *scarce set* are used in this thesis. The *complete set* of training jobs includes 300 non-repeated jobs as shown in Figure 4.2. Since 300 non-repeated jobs cover the entire training input space, they are the same for training and validation. The *moderate set* of jobs includes 200 non-repeated jobs for training and 271 non-repeated jobs for validation as shown in Figures 4.3 and 4.4, respectively. The *scarce set* of jobs includes 100 jobs for training and 214 for validation as shown in Figures 4.5 and 4.6, respectively. It should be noticed that training jobs are a subset of validation jobs for both *moderate set* and *scarce set*.

The output job sequence varies with the number of jobs. Even though the number of jobs varies, the relative sequencing of jobs remains consistent if the model

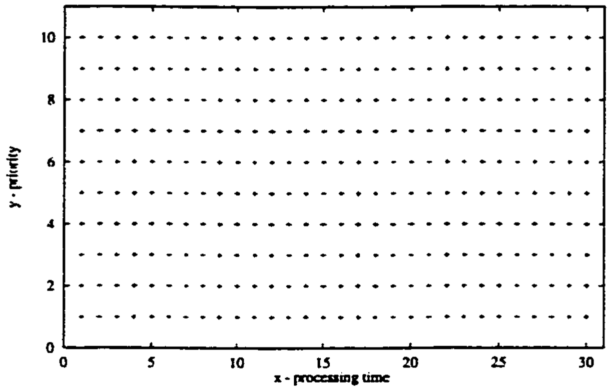


Figure 4.2: *Complete set* of jobs for modeling and validation

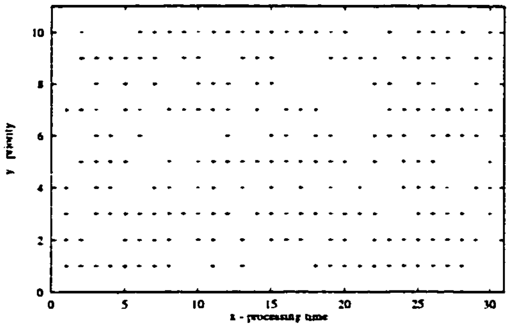


Figure 4.3: *Moderate set* of jobs for training

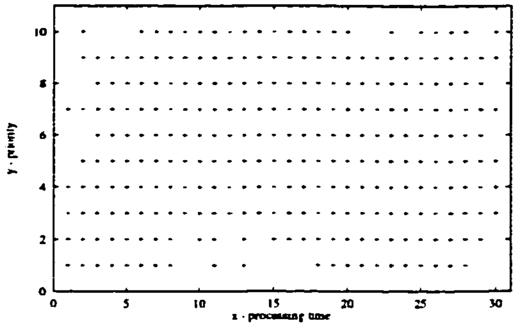


Figure 4.4: *Moderate set* of jobs for validation

captures the monotonocity of the variation. This monotonocity of sequencing with reference to all dimensions is the global behavior of the sequencing system. Any model, which is able to learn the monotonocity of the sequencing system, is an optimal model for such scheduling problem.

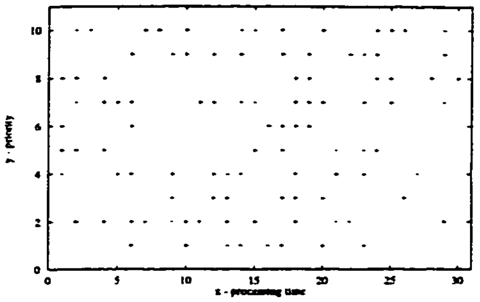


Figure 4.5: *Scarce set* of jobs for training

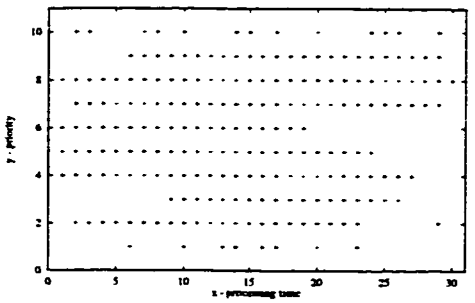


Figure 4.6: *Scarce set* of jobs for validation

4.2.3 Elimination of insignificant inputs for sequence learning

The method to eliminate the insignificant input variables is introduced in Section 3.4.3. For the sequence learning data, the elimination is done by the scarce set of jobs. 10 different models are developed using MISO type clustering as mentioned in Section 3.2, and using Chiu's [13] subtractive clustering method and COSPI method. Parameters are selected such that the *accept ratio* $\bar{\epsilon} = 0.3$, the *reject ratio* $\underline{\epsilon} = 0.0$ and *squash factor* $\eta = 2$ for different radii r_a ranging from 1.0 to 0.1 with a step size of 0.1. Since the data has been normalized, $r_a = 1.0$ is twice the size of the variable domain. The quantitative index calculated using Equation (3.8) for all the input variables of the ten models are given in Table 4.3.

It is shown in the table that the input variable *due date* scores very high in insignificance measure and thus are eliminated from further modeling. This is consistent since the sequence generated using WSPT does no have any relationship with the *due date*. The significant inputs are *processing time* and *priority*. The identification of the real system with the input variables of *processing time* and *priority*, and the output result of job sequence is explain in next section.

		π_j		
r_a	*number of rules*	x_1	x_2	x_3
1	2	0.608374	0.859732	0.296296
0.9	2	0.655172	0.859732	0.296296
0.8	2	0.762599	0.868659	0.296296
0.7	3	0.236669	0.813388	0.296296
0.6	4	0.193407	0.666535	0.0329218
0.5	4	0.148707	0.677012	0.0564374
0.4	4	0.232143	0.666077	0.0658436
0.3	7	0.0391319	0.483559	0.046875
0.2	10	0.00664274	0.161088	0
0.1	14	0.000271792	0.00410954	0

Table 4.3: Quantitative index π_j of various r_a for scarce data set at $\bar{\epsilon} = 0.3$, $\underline{\epsilon} = 0.0$ and $\eta = 2.0$ using subtractive clustering method by COSPI technique

4.2.4 Identification of the actual system

After eliminating insignificant input variables, the actual system is identified by using *processing time* and *priority* as input data and job sequence as the output data. The system identification process is performed using CIOS method with subtractive clustering method.

At first, the training data is clustered using subtractive clustering algorithm. Each cluster represents certain parts of system behaviors. Then the clusters are projected into each dimension in the input space. In the input space, each cluster forms antecedent parts of a rule. Thus the premise parameters of the model is identified. The model is completed by LSE method which identifies the optimal consequent parameters as explained in Section 3.5. The optimization of the model is introduced in the next section.

4.2.5 Design of the parametric search

The parametric search is designed to find the "best" models. The design of the parametric search is to determine the range of each parameter, the step sizes of the parameters, and the heuristic control rules for the change of step size. The parameters are *accept ratio* $\bar{\epsilon}$, *reject ratio* $\underline{\epsilon}$, *squash factor* η, and *cluster radius* r_a. The heuristic rule is similar to the rule in Section 3.6, where the error measure is training sequential error σ_t. They are obtained by the results of several small simulations. The ranges and searching step sizes are shown in Table 4.4, and the *large* sequential error σ_t is set to 80.

Parameter	*Start*	*End*	*Step size*	*Jump step size*
η	2.0	1.00	0.05	-
r_a	1.0	0.15	0.02	*0.1*
$\bar{\epsilon}$	1.0	0.00	0.10	-
$\underline{\epsilon}$	0.9	0.00	0.10	-

Table 4.4: Parameter ranges and changing step size for subtractive clustering

When *squash factor* η is greater than 2, the strong squash cause loss of the training job. The models are very coarse and can not capture training sequence correctly. Therefore it is not necessary to consider searching optimal model in *squash factor* η greater than 2. Thus *squash factor* η is set to a range between 2.0 to 1.0, at the step size of 0.05. The *Cluster radius* r_a is the strongest factor which affects the performance of model. It is the approximated resolution of the training data space. The simulation results show that the performance of the model is very sensitive to *cluster radius* r_a even with a very smaller step changes of 0.002. When *Cluster radius* r_a is close to or less than 0.15, the number of rules increases dramatically. This leads to an over-defined model with the large value of training sequential error σ_t. Thus cluster radius is set to a range of 1.0 to 0.15. When training sequential error σ_t is larger than 80, the zero training sequential error can not be obtain, even when the step size changes by 0.1 for *Cluster radius*. The *large* training sequential error σ_t is set to 80. The *large* step size for *cluster radius* r_a is set to 0.1, and the *small* step size r_a is 0.002. It is observed that the *accept ratio* $\bar{\epsilon}$ and the *reject ratio* $\underline{\epsilon}$ do not have big influence in the performance of model. The whole ranges between 1 to 0 of them are chosen with both at the step size of 0.1. The parametric search maximizes the possibility of finding optimal model. It also provides optimal parameter zones for sequence learning application under different availability of data using subtractive clustering algorithm.

The parametric search is summarized as follows:

1. Build a model by the method introduced in Section 4.1

2. Calculate decision related objective function, such as training sequential error σ_t for sequencing decision.

   IF $\sigma_t = 0$, use the validation data set to validate the model.

3. Change parameters and GOTO step 1.

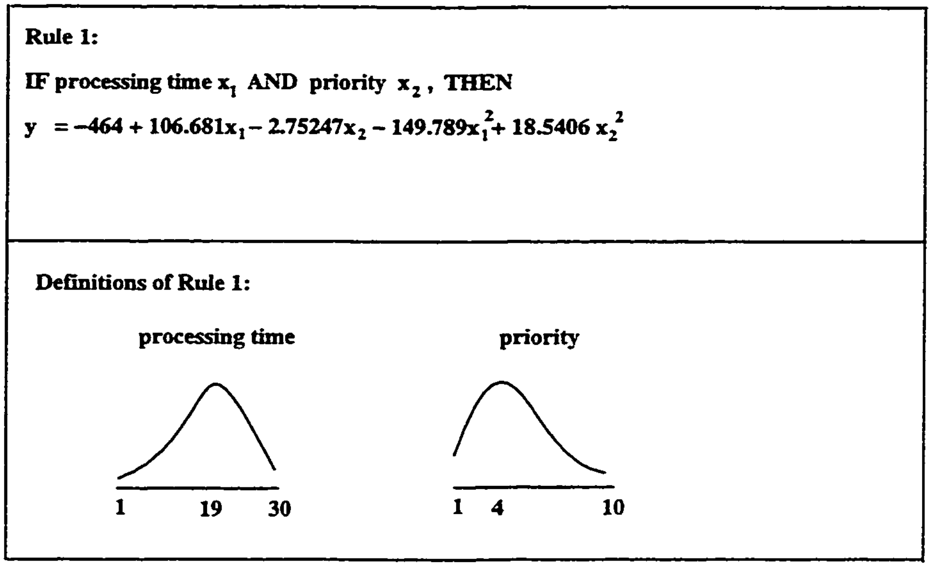
4.2.6 An example of sequencing a set of n jobs by using a fuzzy model

A model built on *moderate set* of training jobs is shown in Table 4.5. It has 28 rules. Each row in the table is a rule. The cluster centers c_1^i and c_2^i in the table correspond to two inputs. A cluster center c_j^i, with a constant cluster radius r_a of 0.686 and a constant γ of 4 denotes an exponential membership function of the *j'th*

R^i	c_1^i	c_2^i	Z_{10}^i	Z_{11}^i	Z_{12}^i	Z_{13}^i	Z_{14}^i
R^1	19	4	-464.068	106.681	-2.75247	-149.789	18.5406
R^2	18	6	-1506.75	109.267	-2.44744	63.8293	1.94465
R^3	12	7	-580.572	-4.77163	0.47448	228.055	-18.9729
R^4	13	4	1171.72	-30.6327	0.252848	-210.191	8.32617
R^5	24	6	52.5247	49.6784	-0.357705	-302.639	23.8561
R^6	25	3	-3340.87	295.919	-5.70899	-267.819	47.2687
R^7	22	8	-1317.68	92.3413	-1.30168	-47.2102	2.39879
R^8	15	9	-1464.35	-76.9768	2.89509	400.78	-22.0178
R^9	17	2	159.612	-62.5559	3.14218	241.749	-56.3247
R^{10}	8	5	391.65	-19.9539	0.325979	-102.137	4.11916
R^{11}	7	8	2341.22	65.9094	-5.00835	-725.493	50.5205
R^{12}	8	3	145.625	208.474	-13.1006	-511.555	89.6537
R^{13}	10	2	986.636	-159.752	8.3921	191.792	-97.8135
R^{14}	4	6	-171.342	-70.6293	7.39875	98.0263	-11.6443
R^{15}	23	1	-2376.51	211.043	-3.90833	-278.771	72.1845
R^{16}	29	8	1210.02	-385.935	6.75099	1030.66	-59.8659
R^{17}	30	5	2722.76	-160.332	2.1732	81.8658	-11.454
R^{18}	10	10	-371.554	64.9991	-1.35866	118.376	-12.9243
R^{19}	23	10	-2778.8	-150.628	1.91891	1009.44	-44.6444
R^{20}	4	3	-344.224	0.909577	1.04236	233.688	-25.6694
R^{21}	12	5	-297.842	-18.7597	1.00216	287.7	-32.6859
R^{22}	14	3	-556.877	205.292	-8.403	-281.138	59.2984
R^{23}	29	2	3221.29	-239.469	4.24781	260.087	-56.6914
R^{24}	3	9	-2119.15	-21.4475	0.18347	589.807	-34.4994
R^{25}	24	4	2669.9	-194.026	4.03823	117.558	-29.1462
R^{26}	17	7	226.49	-9.26089	0.313932	-64.8188	6.47827
R^{27}	28	10	5421.97	515.985	-8.37818	-2701.18	136.149
R^{28}	13	1	-992.934	111.687	-5.53212	-100.351	89.678

Table 4.5: Sugeno second order model with 28 rules, $r_a = 0.686$ and $\gamma = 4$

input of R^i of the antecedent part. Linear regression parameters $Z^i_{10}, Z^i_{11}, Z^i_{12}, Z^i_{13}$ and Z^i_{14} define the consequent linear equation of rule R^i. The example of the rule R^1 is shown in Figure 4.7.



Rule 1:

IF processing time x_1 AND priority x_2, THEN

$y = -464 + 106.681x_1 - 2.75247x_2 - 149.789x_1^2 + 18.5406\,x_2^2$

Figure 4.7: The rule R^1 in Table 4.5

Two jobs are used to demonstrate the reasoning process of sequence prediction by this model. The Sugeno reasoning mechanism is used. The reasoning process of these two jobs on rule R^1 are show in Figures 4.8 and 4.9. In Figure 4.8, the job 1 has a *processing time* of 2 and a *priority* of 5. The firing strength ω^1_1 is 0.0708 and the rule output y^1_1 is -504.147. In Figure 4.9, the job 2 has a *processing time* of 10 and a *priority* of 4. The firing strength ω^1_2 is 0.4761 and the rule output y^1_2 is -24.9865.

The firing strengths ω^i_1 and ω^i_2, and the rule of outputs y^i_1 and y^i_1 for these two jobs are listed in the Table 4.6. For job 1, the sum of firing strength is 6.9384 and weight sum of rule output is 53.9406. The selectability factor of job1 is

$$y_1 = \frac{\sum_{i=1}^{n} y_1^i \cdot \omega_1^i}{\sum_{i=1}^{n} \omega_1^i} = \frac{53.9406}{6.9384} = 7.77422$$

For job 2, the sum of firing strength is 11.6783 and weight sum of rule output is 671.061. The selectability factor of job2 is

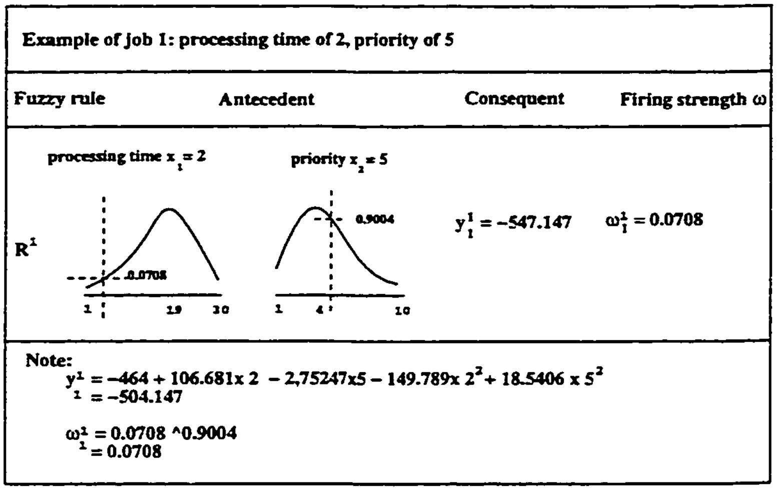
$$y_2 = \frac{\sum_{i=1}^{n} y_2^i \cdot \omega_2^i}{\sum_{i=1}^{n} \omega_2^i} = \frac{671.061}{11.6783} = 57.4624$$

Considering these two jobs only, the predicted sequence number of job 1 is 1, and the predicted sequence number of job 2 is 2. This sequencing result is the same as a WSPT sequence.

A set of 40 jobs sequenced by the model is shown in Table 4.7. The sixth column is model output, the selectability factor S_j^f. The seventh column is the predicted sequence number S_j^d.

4.3 Summary

A scheduling decision learning approach is discussed in this chapter. Subtractive clustering based system identification method is used to learn scheduling decision mechanism from existing schedules. The method successfully captures the sequencing mechanism of WSPT algorithm from input and output data. It is observed during the system validation that models built on the scarce data have low generalization abilities. To improve the model, two clustering algorithms are introduced in the next chapter.



Example of job 1: processing time of 2, priority of 5

Fuzzy rule Antecedent Consequent Firing strength ω

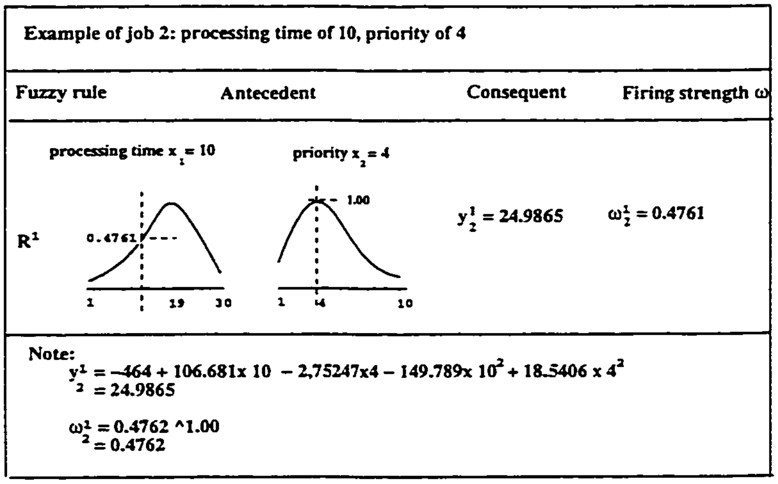
$y_1^1 = -547.147$ $\omega_1^1 = 0.0708$

Note:

$y_1^1 = -464 + 106.681 x\, 2 - 2{,}75247 x 5 - 149.789 x\, 2^2 + 18.5406\ x\ 5^2$
$= -504.147$

$\omega_1^1 = 0.0708\ \hat{}\, 0.9004$
$= 0.0708$

Figure 4.8: Example of reasoning process of R^1 for job 1, processing time of 2, priority of 5



Example of job 2: processing time of 10, priority of 4

Fuzzy rule Antecedent Consequent Firing strength ω

$y_2^1 = 24.9865$ $\omega_2^1 = 0.4761$

Note:

$y_2^1 = -464 + 106.681 x\ 10 - 2{,}75247 x 4 - 149.789 x\ 10^2 + 18.5406\ x\ 4^2$
$= 24.9865$

$\omega_2^1 = 0.4762\ \hat{}\, 1.00$
$= 0.4762$

Figure 4.9: Example of reasoning process of R^1 for job 2, processing time of 10, priority of 4

R^i	y_1^i	ω_1^i	$y_1^i \cdot \omega_1^i$	y_2^i	ω_2^i	$y_2^i \cdot \omega_2^i$
R^1	-547.147	0.0707941	-38.7348	24.9865	0.476081	11.8956
R^2	-930.244	0.0957882	-89.1064	-372.389	0.556324	-207.169
R^3	77.7344	0.400014	31.0948	27.8128	0.49684	13.8185
R^4	268.668	0.329998	88.6598	183.135	0.920845	168.639
R^5	-766.343	0.0118589	-9.08797	-315.321	0.165984	-52.3384
R^6	-2929.24	0.00785194	-23.0002	-1267.56	0.127253	-161.301
R^7	-1314.29	0.0256036	-33.6504	-674.896	0.267293	-180.395
R^8	-153.27	0.212572	-32.5809	-693.775	0.143271	-99.3976
R^9	-152.301	0.127253	-19.3808	-85.9259	0.638288	-54.8455
R^{10}	-54.6578	0.71903	-39.3006	-117.931	0.925223	-109.112
R^{11}	88.5541	0.49684	43.9972	405.835	0.288364	117.028
R^{12}	193.739	0.71903	139.304	308.546	0.925223	285.474
R^{13}	-785.677	0.49684	-390.356	-569.521	0.732799	-417.345
R^{14}	-83.9808	0.925223	-77.701	68.0366	0.71903	48.9203
R^{15}	-1559.3	0.0175854	-27.4209	-617.045	0.212572	-131.167
R^{16}	4121.82	0.0012564	5.17865	1190.57	0.0366008	43.5757
R^{17}	2533.77	0.000759047	1.92325	1480.96	0.0256036	37.9179
R^{18}	21.7824	0.143271	3.12078	409.286	0.0609349	24.9398
R^{19}	858.709	0.0175854	15.1007	-769.74	0.0609349	-46.904
R^{20}	188.47	0.732799	138.111	293.15	0.71903	210.784
R^{21}	289.999	0.400014	116.003	242.601	0.925223	224.46
R^{22}	-103.134	0.267293	-27.5671	479.969	0.863639	414.519
R^{23}	2642.49	0.0012564	3.32002	1384.67	0.0366008	50.6799
R^{24}	-74.7544	0.288364	-21.5565	-508.034	0.143271	-72.7863
R^{25}	2157.13	0.0118589	25.5812	1137.36	0.165984	188.783
R^{26}	47.087	0.127253	5.99197	9.65178	0.49684	4.79539
R^{27}	-3681.77	0.00204187	-7.51769	1117.64	0.0513716	57.4151
R^{28}	948.506	0.288364	273.515	604.169	0.49684	300.175

Table 4.6: Rule output y^i and firing strength ω^i for each rules

j	p_j	w_j	sum of $y^i \cdot \omega^i$	sum of ω^i	S_j^f	S_j^d
1	1	7	6.48452	5.9529	1.0893	1
2	2	10	7.9529	3.9109	2.03352	2
3	2	9	14.3207	4.78682	2.99169	3
4	1	4	23.73	5.82837	4.07146	4
5	2	7	30.704	6.57874	4.66715	5
6	3	9	31.1602	5.21	5.98084	6
7	1	3	32.4625	5.20227	6.24008	7
8	3	8	43.5255	6.26752	6.94462	8
9	2	5	53.9406	6.9384	7.77422	9
10	3	7	66.1578	7.1158	9.29732	10
11	4	9	54.3639	5.65116	9.61996	11
12	3	6	79.9924	7.37739	10.8429	12
13	1	2	49.4193	4.5272	10.9161	13
14	5	9	76.3817	6.04609	12.6333	14
15	6	10	65.7821	5.19159	12.6709	15
16	3	5	103.11	7.54769	13.6611	16
17	5	8	103.686	7.15937	14.4826	17
18	6	9	97.323	6.44182	15.108	18
19	4	6	125.346	8.05111	15.5688	19
20	7	10	89.6518	5.46084	16.4172	20
21	5	7	138.89	8.13839	17.066	21
22	3	4	125.919	7.30334	17.2413	22
23	7	9	125.273	6.79511	18.4358	23
24	4	5	155.634	8.17066	19.0479	24
25	8	10	115.871	5.70662	20.3047	25
26	6	7	177.069	8.64582	20.4803	26
27	7	8	171.893	8.0214	21.4293	27
28	9	10	132.021	5.90027	22.3755	28
29	6	6	219.794	9.38944	23.4087	29
30	3	3	154.79	6.53091	23.7012	30
31	2	2	120.381	5.07665	23.7126	31
32	9	9	178.163	7.4508	23.9119	32
33	10	10	145.445	6.07872	23.9269	33
34	1	1	90.8026	3.74733	24.2313	34
35	5	5	217.546	8.82949	24.6386	35
36	4	4	199.12	8.07092	24.6713	36
37	11	10	160.663	6.2652	25.6438	37
38	10	9	203.392	7.73449	26.2968	38
39	8	7	260.273	9.5907	27.138	39
40	12	10	178.53	6.46193	27.6279	40

Table 4.7: A set of 40 jobs sequenced by using a Sugeno second order model shown in Table 4.5

Chapter 5

Clustering with scarce data

In this chapter, selective subtractive clustering algorithm and modified subtractive clustering algorithm are introduced. These two algorithms make a clustering based system identification method applicable to modeling highly nonlinear and spiral system based on scarce data.

5.1 Clustering analysis

The constraints of scheduling sequencing learning applications are:

- A sequence for each job has to be identified.
- The sequence of all jobs is maintained in a correct order.

After eliminating insignificant variables, clustering is required to identify all training data. The WSPT training surface is a nonlinear spiral shaped surface shown in Figure 4.1. Subtractive clustering algorithm has difficulty to model nonlinear and spiral systems based on the scarce data, when there is constraint that each data point has significant contribution to the system identification. The limitations are due to the following factors. The potential subtraction using remaining maximum potential P_k^* and a factor η could quickly lower the potential of the entire data space to a level lower than the lower threshold potential $\underline{\epsilon}$. Thus there are chances to ignore data points in subtractive clustering depending upon the lower rejection threshold and *squash factor*.

Cluster radius r_a is an approximation to the *support* of the exponential membership function. It indicates the association zone of a cluster center in the data space. Each data point in a cluster characterizes a system behavior with a degree associated to the cluster. The potential penalizing range r_b facilities emerging and positioning of new clusters. It is characterized by a constraint *squash factor* of greater than one. Because r_b is greater than r_a, the potential penalization is strong and is extending to the free zone. The free zone is a zone outside the *cluster radius* r_a. The data points in the free zone are not similar to any cluster centers. Thus, membership degree of data point is extremely low in a free zone. In order to identify the data points in a free zone, it is required to form cluster centers in the free zone or to form close cluster centers to them. The close cluster center is to make strong association to the data points in the free zone. The introduction of clusters in subtractive clustering algorithm is graphically displayed in Figures 5.1, 5.2 and 5.3.

The membership function and potential subtraction associated to a cluster center are demonstrated through a one dimensional case shown in Figure 5.1. The dashed line in the figure shows the initial potentials. The *scarce zone*, which is a zone with low potential is shown in the figure, and may occur in any cases. The *zone of penalty* endorses an efficient potential subtraction zone whenever a cluster center

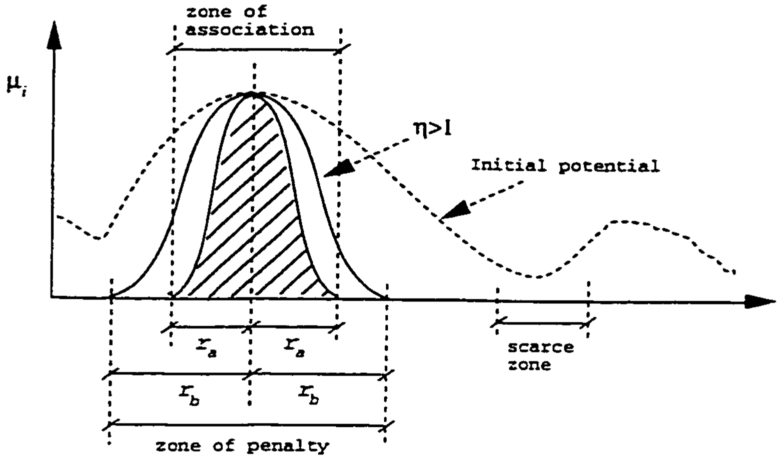


Figure 5.1: Degree of association and potential subtraction at $\eta > 1$

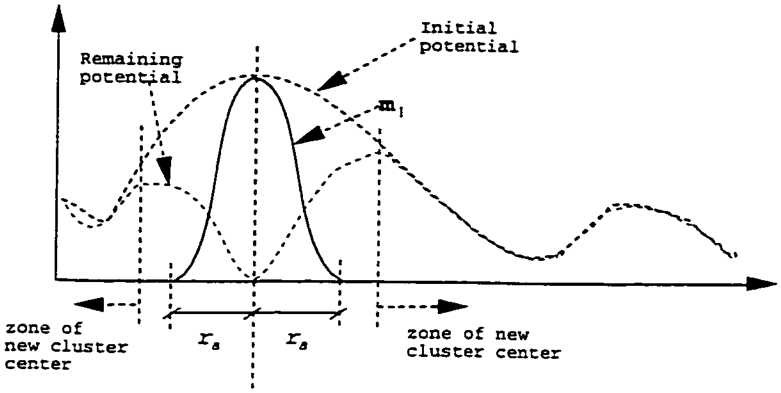


Figure 5.2: Effect of *squash factor* on potential subtraction

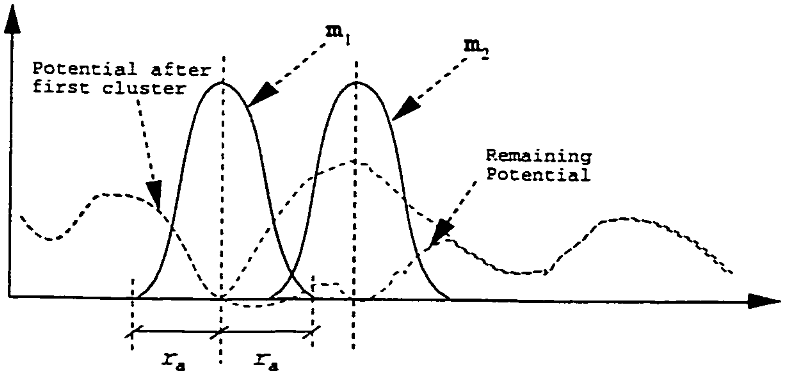


Figure 5.3: Effect of *squash factor* on positioning a new cluster center

is found. The *zone of association* is the zone with an efficient membership degree associated to a cluster center. Zone of penalty is larger than zone of association. A non-associated zone may have an associated potential penalty. This non-associated zone is between r_a and r_b.

The remaining potential after the introduction of the first cluster center is shown in Figure 5.2. The dashed line on the top is the initial potential. The one on the bottom is the remaining potential after the introduction of the first cluster center. The second cluster center is positioned based on the maximum remaining potential as shown in Figure 5.3. The dashed line on the top is the remaining potential after the introduction of the first cluster center. The one on the bottom is the remaining potential after the introduction of the first and the second cluster centers. Two membership functions m_1 and m_2 only have a little overlap which is controlled by *squash factor* η. The remaining low potential values between the first and second clusters can not form new clusters. Thus system information between these two clusters is at a low membership representation associated to either closest clusters.

It is shown in Figure 5.1 that a contradiction exists between zone of association r_a and zone of penalty r_b. The value of r_b is larger than the value of r_a. Zone of penalty depends on η and r_a which makes clusters far apart from each other. A large r_b can quickly squash potential in the entire data space, thus results in a small number of clusters. It reduces the overlap between the clusters in the entire data space. Selection of *squash factor* η depends on the complicated behavior of the system and the scarcity of the training information.

5.2 Selective subtractive clustering method

Sequence learning may involve complicated relationships between the job attributes and the job sequence. Lack of a complete set of training data might hinder the learning process. A complete set of training data is not always available. Representation of scarce set of training jobs strongly challenges clustering method. In the subtractive clustering, jobs in scarce zone have lower potentials. These jobs have no chance to form new clusters, even though they get less potential subtraction from the surrounding cluster centers. The exponential membership representation can only associate the jobs in *scarce zone* to some far with non-similar jobs. A scheduling model built on low represented or missed jobs may have problems capturing training job sequences or may lead to some wrong decisions when given new jobs.

To improve the representation in the scarce training zone, an auxiliary factor called *degree of association* δ is introduced. It is used to decide the closeness of a data point to a cluster center. The *strong members* of that cluster are the data points that have membership degree μ_i^j equal to or greater than the degree of association δ. The rest of the data points are the *weak members* of that cluster. The *strongest member* is the cluster center. It is defined as:

$$\mu_i^j \geq \delta \tag{5.1}$$

where j is the $j'th$ cluster center, and μ_i^j is the membership degree of $i'th$ data point associated to $j'th$ cluster.

A new strategy that subtracts selectively only from strong members of the cluster improves modeling of scarce data. This new method, which is introduced in this thesis, is called *selective subtractive clustering* method and is explained in this section.

The data points in the neighborhood of a cluster center are penalized selectively according to the degree of association they have with respect to that cluster center. This selective penalization enables these data points, which are not strongly associated with earlier clusters, to form new clusters. This process assures stronger representation of regions between clusters with fuzzy sets of high membership degree. This algorithm could be applied to both COSPI and CIOS modeling. A potential subtracting operation is carried out whenever a new cluster center is found. The selective subtractive clustering is different from the subtractive clustering, because the potential penalization is carried out selectively. The condition for ending the clustering process is that all data points are associated with at least one cluster with sufficient degree. The new clustering technique is explained in detail in the following paragraph.

Selective subtractive clustering method finds clusters from a set of input and output data. Each cluster center found by the clustering algorithm is a vector corresponding to the inputs and outputs. Inputs and outputs data are normalized so that the clustering is done inside a hypercube of p dimensions, where p is the sum of input and output dimensions. For example of the normalization of the q=*2*, where the ranges of the q is of 0 to 10, then the normalized result ρ_q is

$$\rho_q = \frac{2-0}{10-0} = 0.2 \tag{5.2}$$

The *clusters radius* r_a is chosen as an initial guess. In line with subtractive clustering method, each data point is assumed to be a candidate for cluster center. Potential P_i of each data point x_i is calculated using Equation (3.3). The data point with the highest potential is selected as the first cluster center. Each cluster forms an exponential membership function in each input dimension. Association or closeness of each data point with this cluster center is represented by membership degree, which can be calculated using Equation (3.7). As mentioned in this section, clusters should be formed by only stronger data points in order to facilitate the formation of new clusters from weakly associated data point close to the *cluster radius* r_a. *Degree of association*, δ, is introduced to determine the closeness of a data point from the cluster center. It can be adjusted according to the complexity of systems using available training data. When a data point's membership degree in a cluster is equal to or greater than the degree of association δ, then this data point is *identified* with that cluster. This process of association is called *identification*. The data points *identified* are *strong* members of that cluster. The *strong* members also include the cluster center itself. The data points not *identified* by any cluster are *free data*. Search for more clusters continues using selective subtraction as long as there are *free data* in data space. After a new cluster center and its *strong* members are identified, its *strong* members are penalized in proportion to the potential of that cluster center. The newly potential of the *identified* points is given by

$$P_i = P_i - P_k * e^{-\frac{4}{(r_a \cdot \nu)^2}\|x_i - x_k\|^2}, \tag{5.3}$$

where ν is the penalty factor. Even though penalty factor ν is analogous to the *squash factor* defined by Chiu [13], *squash factor* results in a penalty zone which is larger than r_a while the penalty factor results in severe penalty in the *identified* region that is smaller than r_a. With the newly potential distribution of data points in data space after penalizing only the recently *identified* data points, a data point with the largest potential will become the new cluster center.

Once a new cluster center is accepted, the selective subtraction continues to find all other possible clusters. The clustering technique involves *identification* and selective subtraction is used until every data point is identified. Once all the data points are clustered, a collection of clusters constitute the fuzzy rule base. The Algorithm for selective subtractive clustering is given below:

```
1. Initialize $\delta$ and a set radius $r_a$
2. Calculate potential $P_i$ using Equation (3.3) for all points
3. while free data is available
       Pick the point with the maximum potential value
       Assign it as a cluster center
       Compute a degree of membership to this cluster using
       Equation (3.7) for all data points
       Label identified data points by comparing membership
       degree to $\delta$
       Subtract potential selectively using Equation (5.3)
       for the identified data points
4. End clustering process
```

Selective subtractive increases the level of representation through a degree of association δ. The degree of association δ can be adjusted to reflect the scarcity level and the complexity of training information. It enables to form a cluster in weekly zone.

5.3 Modified subtractive clustering

It is observed that the selective subtractive is similar to penalizing with *squash factor* η smaller than one. When build a model from weakly represented data space, the contribution of each point to the system modeling will be made stronger by the introduction of *squash factor* η smaller than one, in addition to *squash factor* η greater than one as proposed by Chiu [13] in subtractive clustering technique. The optimum range of *squash factor* η depends on the relative availability of the data and complexity of the system behavior. The membership function and potential subtraction of *squash factor* η smaller than one is shown in Figure 5.4

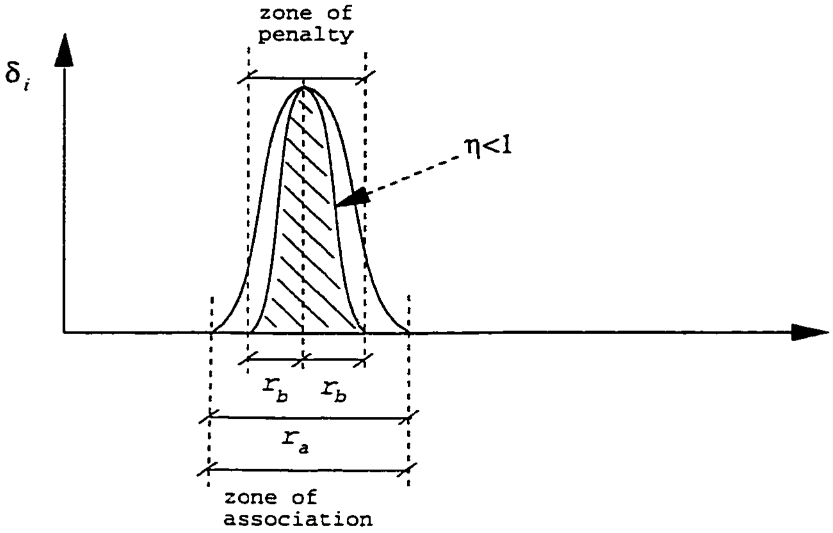


Figure 5.4: Degree of association and potential subtraction at $\eta < 1$

The clustering strategy of modified subtractive clustering algorithm is the similar to the selective subtractive clustering algorithm. Selecting *squash factor* η less than 1 includes different level of degree of association δ. Compare to selective subtractive clustering algorithm, better models are found by modified subtractive clustering algorithm. On the other hand, selective clustering algorithm uses less parameters (*cluster radius* r_a and degree association δ) than it is used in modified subtractive clustering algorithm (*cluster radius* r_a, *squash factor* η, *accept ratio* $\bar{\epsilon}$ and *reject ratio* $\underline{\epsilon}$). Fewer parameters lower the computation on parametric searching.

Modified subtractive clustering algorithm efficiently extends subtractive clustering algorithm to model the scarce set of training data. The model has higher generalization abilitedy than subtractive clustering algorithm. Thus it is more suitable for sequencing learning applications. The estimation of the optimum ranges of *accept ratio*, *reject ratio*, *squash factor* and *cluster radius* for a system behavior will be very useful for the selection of better models. The parametric search for optimal model is used for finding optimum ranges of parameters. The parameter ranges, step sizes and the heuristic control rule are the same as the parametric search of subtractive clustering method, except the *squash factor* η extends over smaller than 1. The test of the set of data using modified subtractive clustering algorithm is also presented in Chapter 6.

Chapter 6

Experiment results and analyses

6.1 Introduction

A fuzzy modeling method discussed in Chapter 4, includes significant variable identification which provides a set of training jobs (Section 4.2.3), fuzzy system identification which builds a fuzzy model on a set of training jobs (Section 4.2.4), and parametric optimization which finds the best model (Section 4.2.5). This method is applicable to scheduling decision mechanism learning applications. There are two cases to which this application is applicable. One is when a sequence is created by a human expert scheduler. The other is to collect sequence output from an optimization algorithm. Both require insignificant variable identification, because system identification of scheduling application needs a correct set of training jobs. Input and output of training have equal contributions to the system identification by using CIOS and LSE method. The modified subtractive clustering method is used to identify antecedent part of the Sugeno model. Although only the input dimension of each cluster is projected, and uses to form antecedent parts of the model, both inputs and outputs contribute to the form of clusters. LSE identifies consequent parameters optimally using training jobs and antecedents.

Experimental results of a scheduling sequence learning application are discussed in this chapter. The ability to extract knowledge using the modified subtractive clustering method is verified. Verification is carried out with three training sets of jobs. The three sets of training and validation jobs are *scarce set*, *moderate set* and *complete set* as defined in Section 4.2.2. The verification is done at different levels of availability of training jobs.

The experiments are designed to compare the modified subtractive clustering method with the conventional subtractive clustering. Comparisons were made on the ability of capturing training sequence of jobs; capturing global behavior of training sequence of jobs; the ideal order of Sugeno model for sequence learning; and the optimal parameters of the model.

In Section 6.2 of this work, the effect of the parameters on modeling training job sequences are presented. In Section 6.2.2, the effect of clustering radius and squash factor on zero modeling error is presented. Analyses and conclusions on modeling the training sequence are presented in Section 6.2.3. Effect of parameters on the optimal model are discussed on Section 6.3. A representation factor ϕ is defined in Section 6.3.2. The efficient representations of optimal models for scarce training jobs are explained in the same section. The relationship between the optimal cluster radius and the squash factor, to the system complexity and the data scarcity is reviewed in Section 6.3.2.

6.2 Modeling the sequencing mechanism

Correctly sequencing the training jobs shows that the model has captured sequencing mechanism. The sequencing mechanism is expressed as a fuzzy sequencing knowledge base. Parameters of both conventional subtractive clustering and the modified subtractive clustering methods have strong effects on capturing a training sequence of jobs. The effectiveness of both methods is analyzed and is discussed in this section.

6.2.1 Effect of squash factor and cluster radius on modeling

The experiment is a parametric search for optimal models for different sets of training jobs. The insignificant variable identification results is presented in Section 4.2.3. For a set of training jobs, an optimal model requires to fully capture sequencing mechanism from this set of jobs.

Identifying effect of the parameters on capturing the sequence of training jobs helps to improve the identification method. Identifying the parameter ranges helps to reduce the searching space of the optimal model.

Figure 6.1 shows models which are built with constant *accept ratio* of 0.3 and constant *reject ratio* of 0.1. This figure also shows various combinations of *cluster radius* r_a and *squash factor* η for *complete set* of training jobs in the Sugeno first order model (Section 3.1). This figure illustrates the influence of *cluster radius* r_a and *squash factor* η on modeling of the training jobs. The x axis in Figure 6.1 is *cluster radius*, the y axis is *squash factor* and the z axis is training sequential distance error σ_t. Each grid point corresponds to a parametric search point. The dark portions in the figure indicate the closeness of the search points. The same

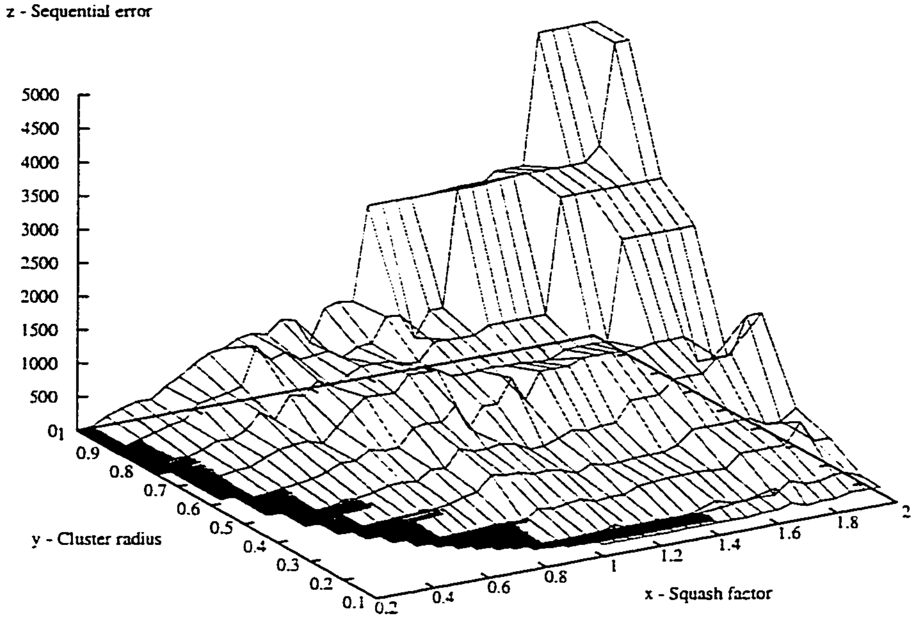


Figure 6.1: Variation of σ_t error versus r_a and η for *complete set* of jobs with $\bar{\epsilon} = 0.3$. $\underline{\epsilon} = 0.1$ for Sugeno first order model

properties as Figure 6.1 are also shown for Sugeno second order models for *complete set* is shown in Figure A.5, and Sugeno models for other sets of training jobs are shown in Figures A.1, A.2, A.3 and A.

In Figure 6.1, the part where *squash factor* greater than 1 denotes the results of using conventional subtractive clustering algorithm, and the part where the *squash factor* smaller than 1 denotes the results of using modified subtractive clustering algorithm. This remark also applies to other figures in this chapter and in the Appendix A and Appendix B.

Figure 6.1 shows that the training sequential error decreases drastically from $\eta = 2$ to $\eta = 1.0$. There is a trend that the decreasing of *squash factor* leads to the decreasing of training sequential error. There is small training sequential error σ_t when *cluster radius* r_a is large and *squash factor* η is small. Small sequential distance error also exists in the case of large η and small r_a. Dark portions indicate small training sequential errors. Most dark portions are in the zone of *squash factor* η less than 1. Even though, the set of training jobs is complete and uniformly distributed in entire training space, a small modeling sequential error model can not be found at *squash factor* η greater than 1.4.

There are more small sequential error models in the zone of *squash factor* less than 1, than in the zone of *squash factor* greater than 1. This indicates that a *complete set* of training jobs can easily capture the system behavior by using the modified subtractive cluster method rather than using a subtractive clustering.

Models that are built with various *cluster radius* r_a and *squash factor* η for *complete set* of training jobs are shown in Figure 6.2. Parameters in Figure 6.2 are the same as in Figure 6.1. The x axis is *cluster radius* r_a, the y axis is *squash factor*, and the z axis is the number of rules corresponding to *squash factor* and *cluster*

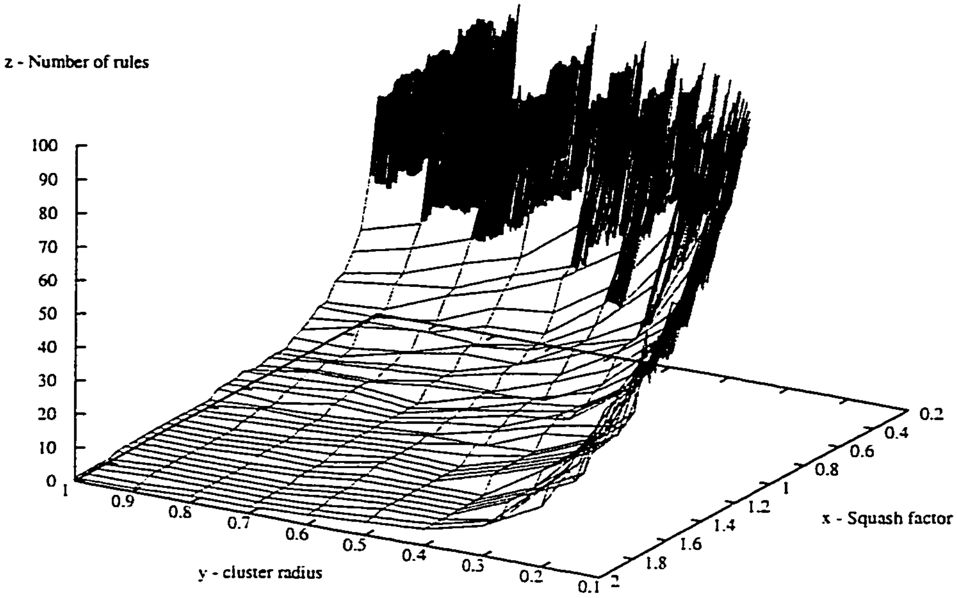


Figure 6.2: Variation of number of rules versus r_a and η for *complete set* of jobs with $\bar{\epsilon} = 0.3$, $\underline{\epsilon} = 0.1$ for the Sugeno first order model

radius. The dark portions in Figure 6.2 correspond to models with large number of rules which also corresponding to models with small training sequential errors in Figure 6.1.

The trend of the increasing of number of rules in Figure 6.2 and the decreasing of sequential error in Figure 6.1 indicates that small training sequential error models can be achieved by increasing the number of rules. Small value of *cluster radius* r_a and large value of *squash factor* η yield almost the same number of rules as large value of *cluster radius* r_a with small value of *squash factor* η. Similar result shown in Figure 6.2 can also be seen in Figure A.10 for Sugeno second order models for *complete set*, and Sugeno models for other sets of training jobs are illustrated in Figures A.6, A.7, A.8 and A.9 in Appendix A.

6.2.2 Effect of squash factor and clustering radius on obtaining models with zero training sequential error

Figure 6.3 through 6.8 are for *scarce set*, *moderate set* and *complete set* of training jobs of the Sugeno first and second order models. Each point in these figures represents a zero training sequential error model. In all those figures, the x axis is the *cluster radius*, and y is the *squash factor*. The *cluster radius* is varied from 1.0 to 0.15 while the *squash factor* is varied from 2.0 to 0.05. It can be observed that more zero training sequential error models can be achieved at *squash factor* less than 1 than at *squash factor* greater than 1. The ranges of zero training sequential error models of *scarce set* shown in Figures 6.3 and 6.4 are larger than that of *moderate set* shown in Figures 6.5 and 6.6, and they are also larger than that of *complete set* shown in Figures 6.7 and 6.8. So, more training data increases the difficulty of

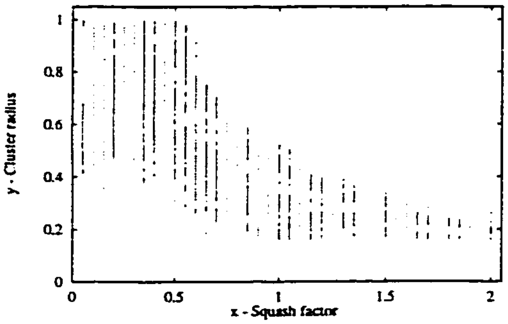


Figure 6.3: Range for models of the *scarce set* of training jobs with $\sigma_t = 0$ for the Sugeno first order model

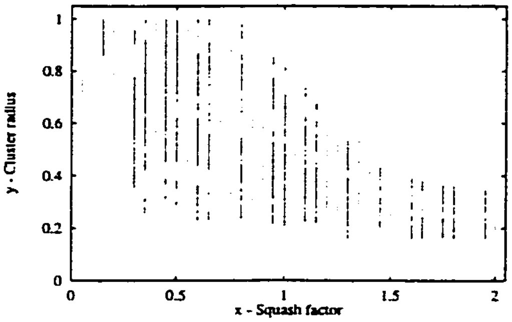


Figure 6.4: Range for models of the *scarce set* of training jobs with $\sigma_t = 0$ for the Sugeno second order model

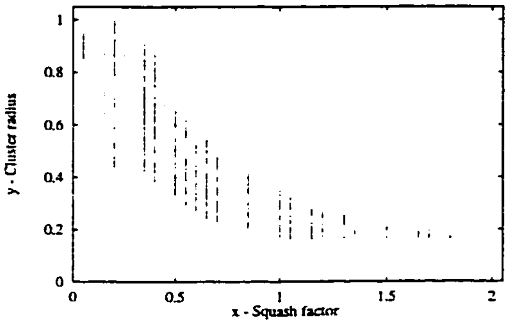


Figure 6.5: Range for models of the *moderate set* of training jobs with $\sigma_t = 0$ for the Sugeno first order model

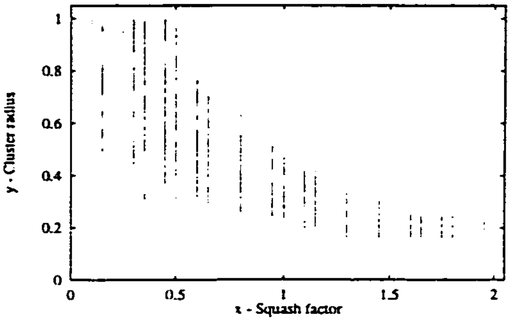


Figure 6.6: Range for models of the *moderate set* of training jobs with $\sigma_t = 0$ for the Sugeno second order model

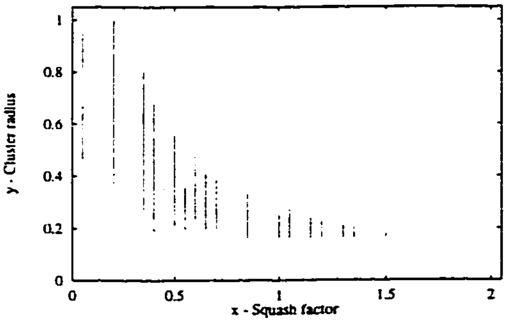


Figure 6.7: Range for models of the *complete set* of training jobs with $\sigma_t = 0$ for the Sugeno first order model

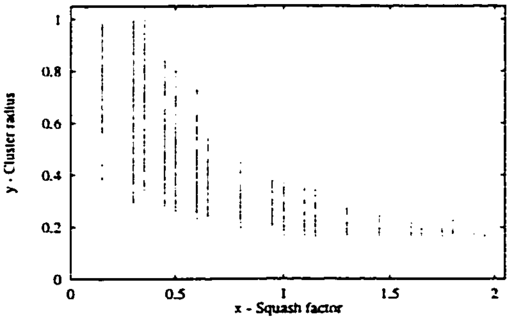


Figure 6.8: Range for models of the *complete set* of training jobs with $\sigma_t = 0$ for the Sugeno second order model

modeling. With Sugeno reasoning mechanism and clustering based system identification methods, training sequence can always be captured for three different sets of training jobs. As it is seen in Figure 6.3 through 6.8, compared to Sugeno first order model, the Sugeno second order model has lager range of zero training sequential error models. Therefore, the nonlinear relationship between inputs and output is better identified by the second order model.

6.2.3 Conclusions of modeling on training jobs

Subtractive clustering based system identification with Sugeno reasoning mechanism successfully captures the training sequencing mechanism as it shows in Figure 6.3 through 6.8. It can also be seen from these figures that modeling of system based on modified subtractive clustering algorithm provides more zero error models than that based on subtractive clustering algorithm. An increased number of training data requires more rules to model the system behavior as shown in Figures 6.1 and 6.2. *Squash factor* η greater than 1.5 proposed by Chiu [13] reduces the clustering performance as shown in Figure 6.1. It pushes clusters far away from each other, which not only leaves a low membership representation of training jobs between clusters, but also can cause a complete loss of training jobs of in the scarce zone. Modified subtractive clustering effectively represents training jobs by enlarging the association representation of each cluster center. It increases the overlap of clusters by placing clusters reasonable close to each other.

6.3 Effect of parameters on optimal model

For sequencing learning applications, validation is to verify a model's abilities to capture sequencing mechanism with new jobs. This sequencing mechanism is the method by which the training job sequence is generated. Each set of validation jobs contains a number of new jobs. The model that provides the best prediction of validation jobs (measured by σ_v) with reasonable small number of rules is the optimal one. The validation of a model is processed whenever a zero training sequential error model is constructed.

6.3.1 Effect of squash factor and cluster radius on optimal model

All the models shown in Figure 6.3 through 6.8 that have zero modeling sequential error are verified with the corresponding validation set of jobs. The predicted output from each model is used to calculating the validation sequential distance error, σ_v. The clustering parameters of models having the least validation sequential distance error σ_v at constant *squash factor* are plotted in Figure 6.9 through 6.14 for *scarce set* of jobs, and Figure 6.15 through 6.20 for *moderate set* of jobs. The first order models with the least validation error for each squash factor are shown in Figure 6.9 through 6.14 and second order model in 6.15 through 6.20. Number of rules, optimal cluster radius and least validation sequential error are plotted against squash factor.

For both, the first order and second order models, and for *scarce set* and *moderate set* of training jobs, number of rules of optimal model varies slightly over the entire range of *squash factor*. *Cluster radius* increases as the *squash factor* decreases dependently, this global trend is observed in the mid-range of the *squash*

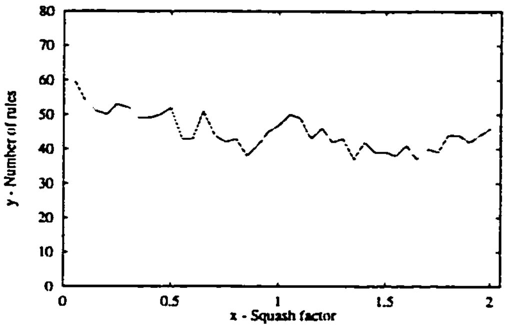


Figure 6.9: Variation of number of rules versus η for a *moderate set* for Sugeno first order model

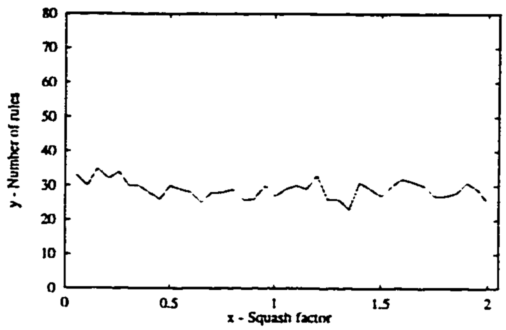


Figure 6.10: Variation of number of rules versus η for a *moderate set* for Sugeno second order model

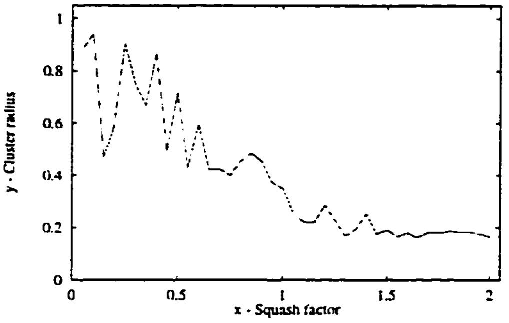


Figure 6.11: Variation of r_a versus η for a *moderate set* for Sugeno first order model

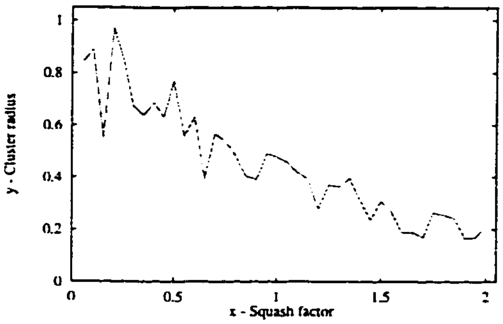


Figure 6.12: Variation of r_a versus η for a *moderate set* for Sugeno second order model

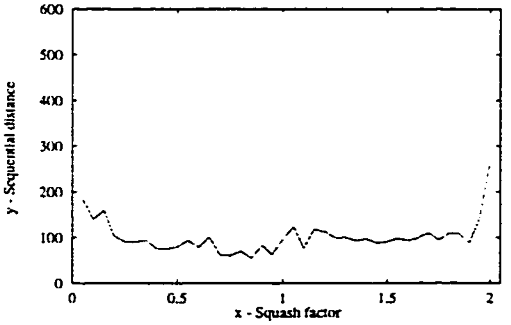


Figure 6.13: Variation of σ_v versus η for a *moderate set* for Sugeno first order model

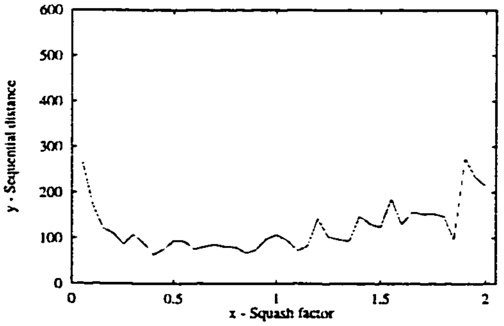


Figure 6.14: Variation of σ_v versus η for a *moderate set* for Sugeno second order model

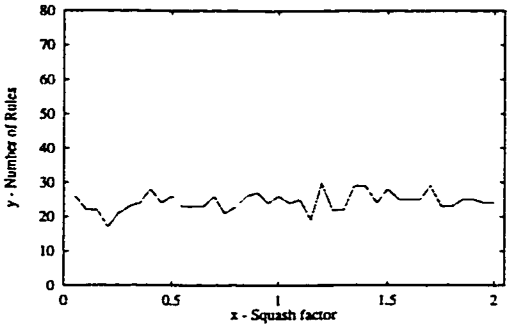


Figure 6.15: Variation of number of rules versus η for a *scarce set* for Sugeno first order model

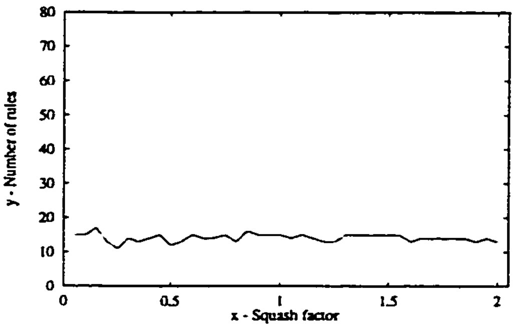


Figure 6.16: Variation of number of rules versus η for a *scarce set* for Sugeno second order model

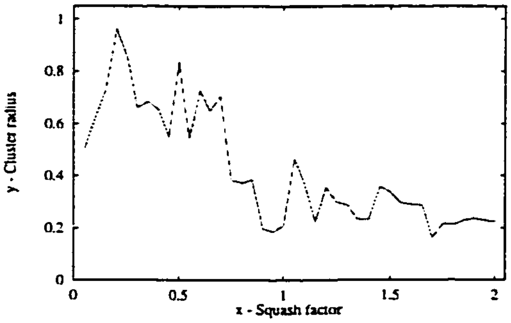


Figure 6.17: Variation of r_a versus η for a *scarce set* for Sugeno first order model

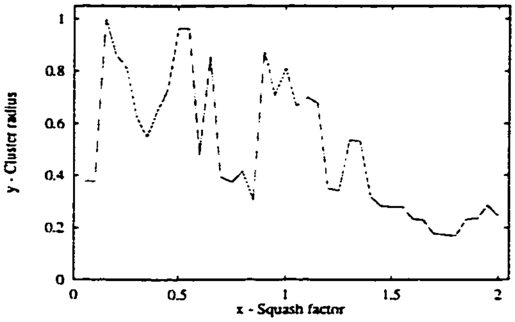


Figure 6.18: Variation of r_a versus η for a *scarce set* for Sugeno second order model

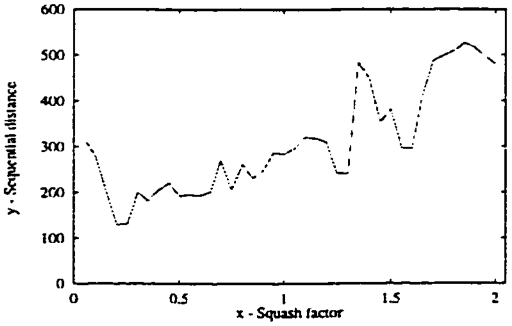


Figure 6.19: Variation of σ_v versus η for a *scarce set* for Sugeno first order model

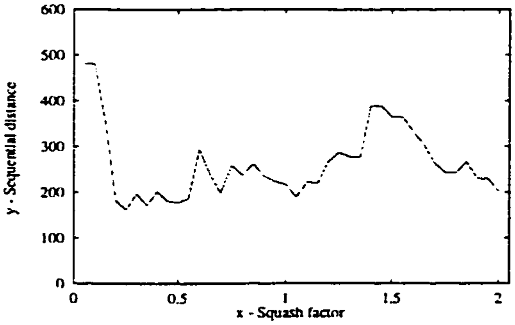


Figure 6.20: Variation of σ_v versus η for a *scarce set* for Sugeno second order model

factor. The least validation sequential error is achieved with *squash factor* $\eta \leq 1$. Reduced validation sequential error is noticed with small values of squash factors for the *scarce set.* The number of rules remains almost constant over the entire range of *squash factors.* Validation sequential error of model is small for a large range of *squash factor* for the *moderate set.* Therefore the cluster is insensitive to the validation sequential error σ_v in a large range of squash factor between 0.3 and 1.9.

6.3.2 Efficient representations by modified subtractive clustering algorithm

The optimal predicted variation of sequences for the validation jobs are shown in Figures 6.21 and 6.22 with the *squash factor* η of 1.75 and 0.55, respectively. These two figures demonstrate the effects of *squash factor* in sequence learning with the scarce set of training jobs. These values of *squash factor* are selected for the comparison with 23 rules for both cases. It can be seen that the modeling with squash factor of 1.75 dose not capture the monotonocity of the sequencing in the approximate *processing time* range of between 5 and 15, and the priority range of between 1 and 10. However, the model with *squash factor* of 0.55 captures the monotonocity over the entire input space as shown in Figure 6.22.

In order to visualize different representation capability of the models, the fuzzy sets of the antecedents are shown in Figures 6.23 and 6.24 for *squash factor* 1.75 and 0.55, respectively. The narrow fuzzy sets in Figure 6.23 show non-uniform representation of processing time at the *squash factor* of 1.75. These fuzzy sets with a small support defined by small cluster radius are not suitable for modeling

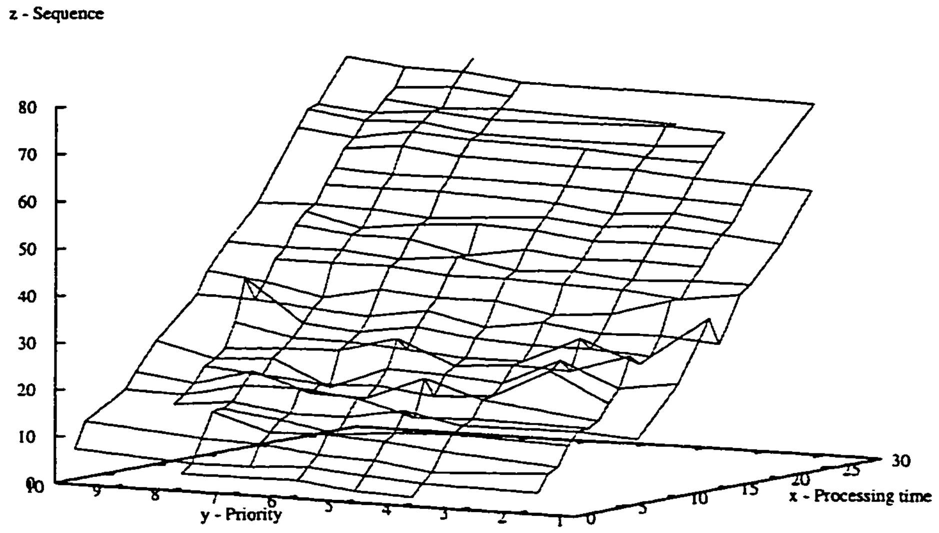


Figure 6.21: Validation surface of sequence in relation to p_j and w_j at η of 1.75 for *scarce set*

scarce set of jobs. The wide fuzzy sets with *squash factor* of 0.55 result in a uniform representation of data space. Wide fuzzy set is defined by large cluster radius. Wide fuzzy set and uniform representation helps sharing of the information over a large space and the behavior of the scarce points are shared to the farther points. Thus the modeling with small *squash factor* helps extracting the global behavior-sequencing mechanism out of *scarce set* of training jobs. In order to measure the representation capability of the fuzzy sets, a *representation factor* ϕ introduced by Demirli and Muthukumaran [29] is used. It is defined as follows:

$$\phi_j = \sum_{i=1}^{n} \omega_j^i \tag{6.1}$$

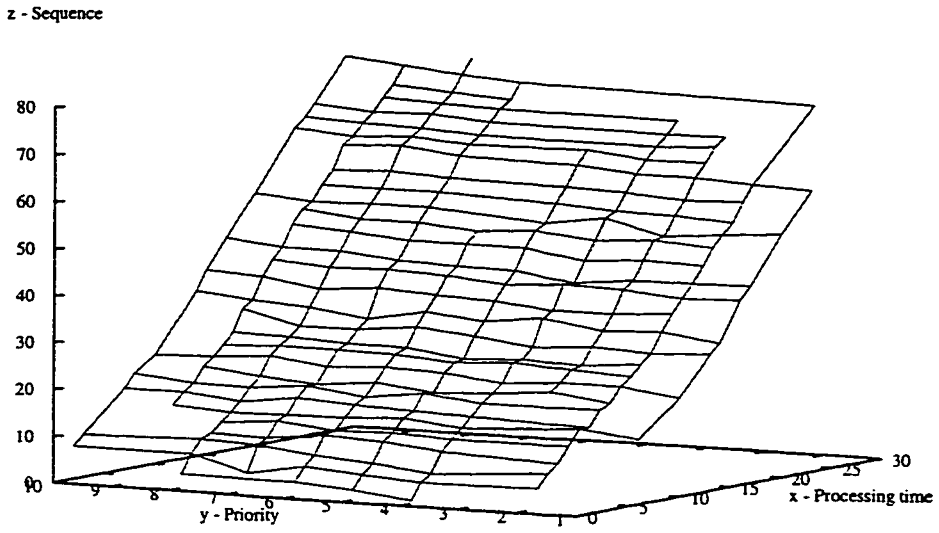


Figure 6.22: Validation surface of sequence in relation to p_j and w_j at η of 0.55 of *scarce set*

where ϕ_j is the representation factor of the $j'th$ data point, n is the total number of rules, ω_j^i is the firing strength for input x_j of $i'th$ rule.

The representation factors for the *squash factor* of 1.75 and 0.55 are shown in Figures 6.25 and 6.26, respectively. In the case of *squash factor* of 1.75, the large values of representation factor can be seen only for the zone with dense training jobs and it is very small for the zone with scarce training jobs. The failed representation of sequencing mechanism of scarce zone of *processing time* from 5 to 15 and *priority* from 3 to 7 produces large validation sequential error as shown in Figure 6.21. This indicates that generalization ability of conventional subtractive cluster is very low, because of its local *representation character.* The local representation is because

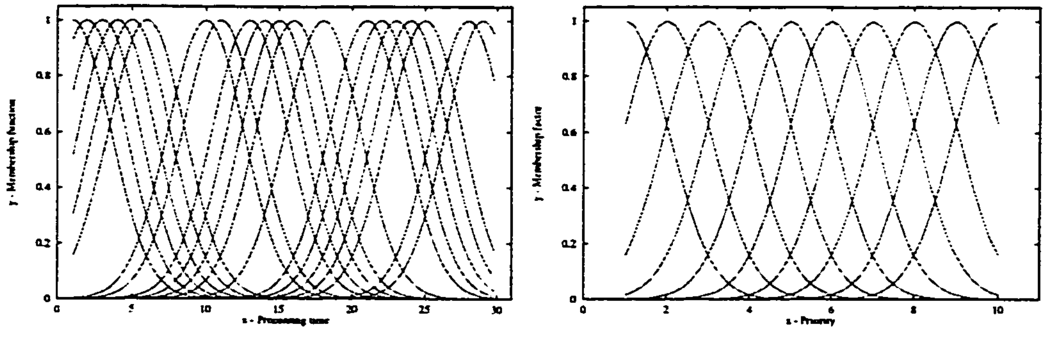


Figure 6.23: Fuzzy sets of a model at η of 1.75 for *scarce set* of jobs

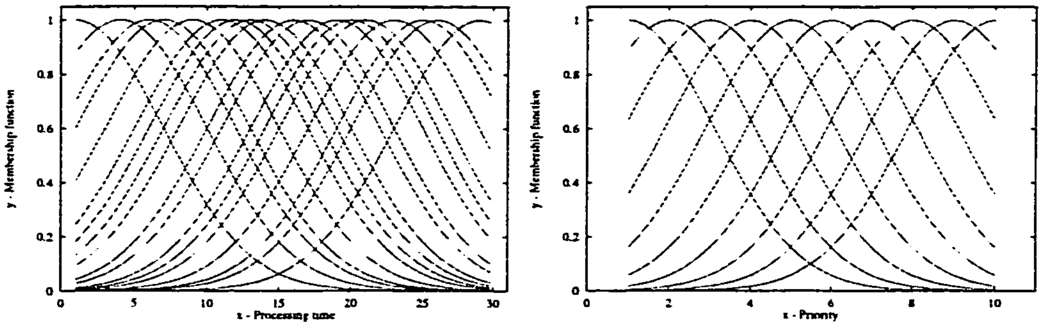


Figure 6.24: Fuzzy sets of a model at η of 0.55 for *scarce set* of jobs

of the pushes of *squash factor*. *Squash factor* greater than 1 contradicts the large cluster radius.

Modified subtractive clustering is more suitable for these kinds of applications that require capturing system behaviors with only scarce training data. The representation factors of modified subtractive clustering method shown in Figure 6.26 are smoothly increase from the boundary zone towards the more intensive zone in the middle. System behavior is well predicted in the entire training space. The values of representation factor of the scarce zone are large. The explanations of this phenomenon are that modified subtractive clustering encourages data points in the

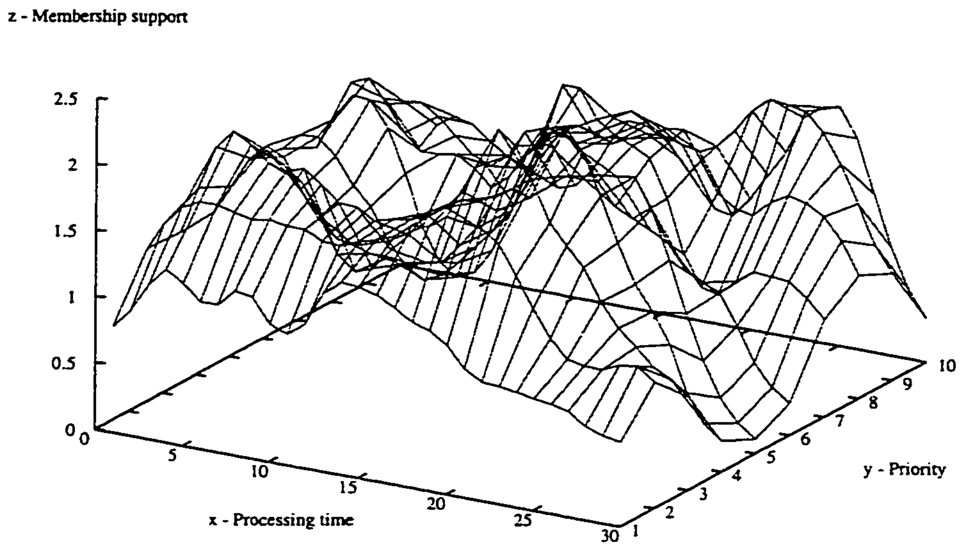


Figure 6.25: ϕ at η of 1.75 for *scarce set* for Sugeno first order model

scarce zone to form new cluster. When the *squash factor* is less than 1, the potential subtraction is only effective on the strong members of a cluster.

The advantage of modeling scarce set of training jobs by small values of *squash factor* is also shown in Figures 6.27 and 6.28. The average validation sequential error which is defined as the ratio of σ_v over the total number of validation jobs is shown in Figures 6.27 and 6.28 for *scarce set* and *moderate set* of jobs, respectively. The smaller values of *squash factor* result in better models for scarce set of jobs. The model is insensitive to *squash factor* for *moderate set* of jobs. As the number of training data increased the performance of conventional subtractive clustering algorithm becomes closer to the modified subtractive clustering algorithm.

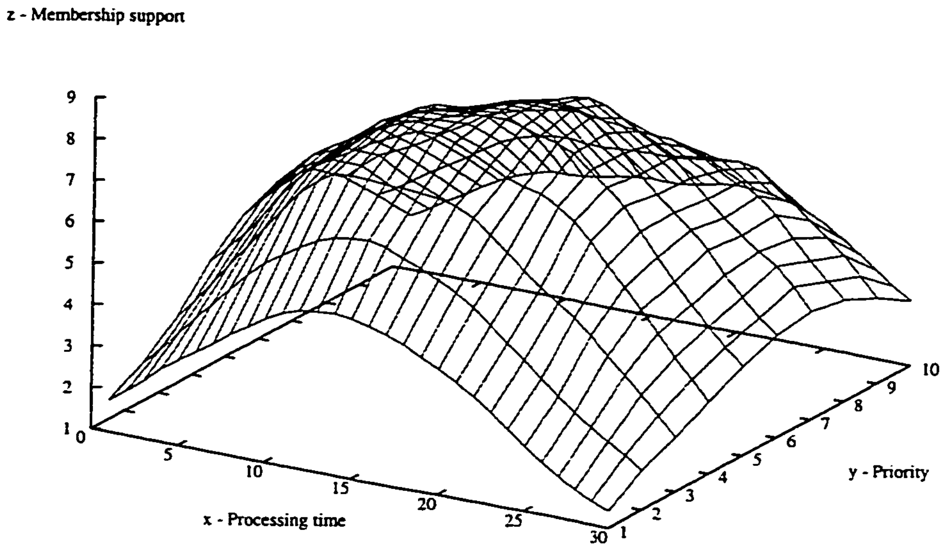


Figure 6.26: ϕ at η of 0.55 for *scarce set* for Sugeno first order model

6.4 Parameters for optimal model

Optimal models of different set of training jobs of the Sugeno first and the second order models are shown in Tables 6.1 and 6.2, respectively. The models with *squash factor* $\eta > 1$ refer to the subtractive clustering, and the models with *squash factor* $\eta < 1$ refer to the modified subtractive clustering.

Optimal models of *scarce set* and *moderate set* of jobs have least validation sequential distance error σ_v. For the *complete set* of training jobs, the same set of jobs also constitutes the validation set, so the optimal models are obtained with the minimum number of rules.

As the number of training jobs increase, i.e., as the set of training jobs moves

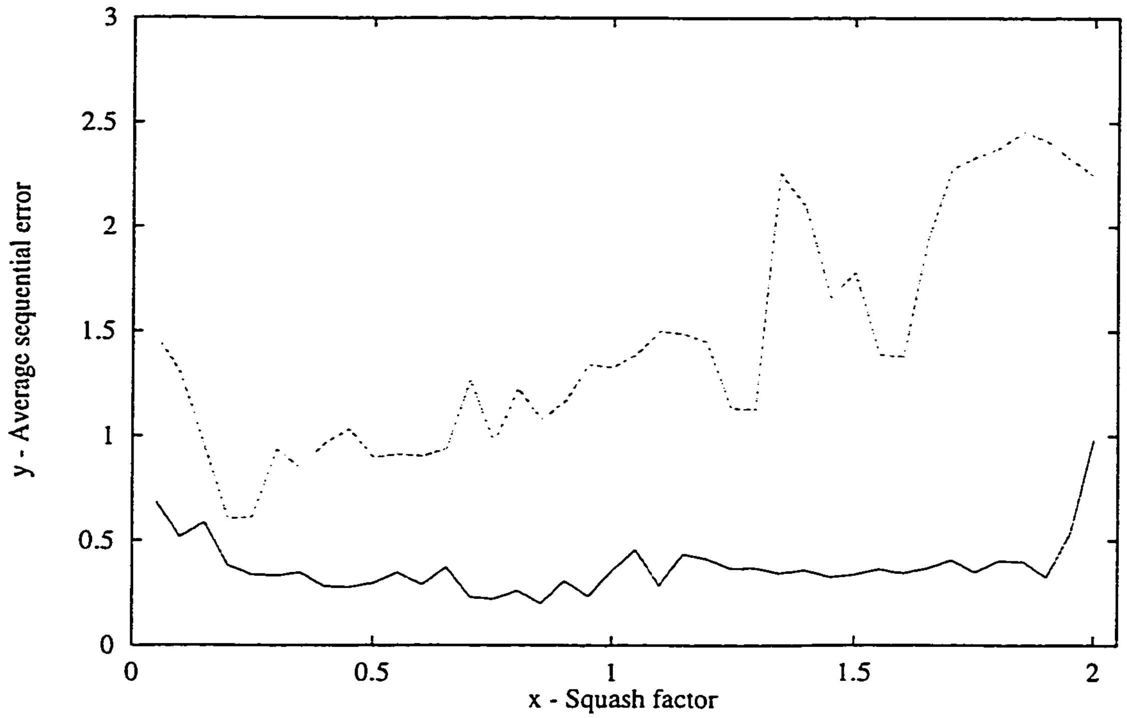


Figure 6.27: Variation of average sequential error versus η form both *scarce* and *moderate set* for Sugeno first order model

from the *scarce set* to the *complete set* of jobs, for $\eta < 1$, optimal *squash factor* increases. More training jobs increases the association among training jobs, thus increase *squash factor* at $\eta < 1$, which reduce the association zone. But for *squash factor* $\eta > 1$ it stays between 1.05 to 1.30. Once *squash factor* is greater than 1.3, the squashing of potential is too strong. More local and separated clusters make it difficult for the system to capture the sequencing mechanism.

The best models found by using conventional subtractive clustering, modified subtractive clustering and selective subtractive clustering algorithms are listed in Tables 6.3 and 6.4. The selective subtractive clustering algorithm improves conventional subtractive clustering algorithm in modeling scarce set of training jobs. For

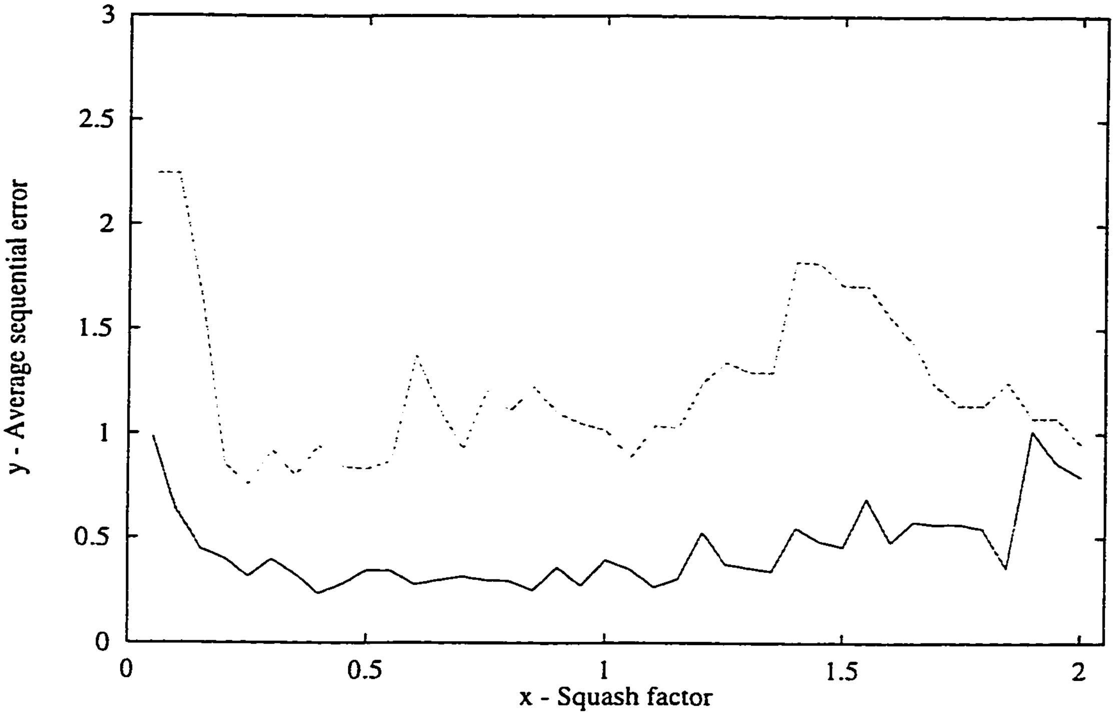


Figure 6.28: Variation of average sequential error versus η form both *scarce* and *moderate set* for Sugeno second order model

the moderate set of training jobs, the best Sugeno first and second models built by selective subtractive clustering algorithm are the same as the best models of the conventional subtractive clustering. For the scarce set of training jobs, the first order model of the selective subtractive clustering algorithm with $\sigma_v = 156$ is better than the model of conventional subtractive clustering with $\sigma_v = 241$. Compared to the conventional subtractive clustering and modified subtractive clustering, except for the first order model with *complete set* of jobs, the overall performances of these first order and second order models show that $\eta < 1$ dominants $\eta > 1$ in both number of rules and validation sequential error. The optimal combination of *cluster radius* and *squash factor* leads to a similar number of rules. But, using large clusters yields fewer

		η	$\bar{\epsilon}$	$\underline{\epsilon}$	r_a	N	σ_v
scarce set	$\eta > 1$	1.30	0.20	0.10	0.286	22	241
	$\eta < 1$	0.20	0.70	0.60	0.964	17	129
moderate set	$\eta > 1$	1.10	0.20	0.00	0.222	49	77
	$\eta < 1$	0.85	0.00	0.00	0.486	38	54
complete set	$\eta > 1$	1.15	0.10	0.00	0.276	42	0
	$\eta < 1$	0.95	0.00	0.00	0.344	52	0

Table 6.1: Optimal model with the least σ_v for both subtractive $\eta \geq 1$ and modified subtractive $\eta \leq 1$ algorithm of the Sugeno first order model

		η	$\bar{\epsilon}$	$\underline{\epsilon}$	r_a	N	σ_v
scarce set	$\eta > 1$	1.05	0.00	0.00	0.666	14	189
	$\eta < 1$	0.25	1.00	0.70	0.812	11	162
moderate set	$\eta > 1$	1.10	0.00	0.00	0.416	30	72
	$\eta < 1$	0.40	0.40	0.00	0.686	28	64
complete set	$\eta > 1$	1.25	0.20	0.00	0.252	31	0
	$\eta < 1$	0.40	0.30	0.00	0.762	29	0

Table 6.2: Optimal model with the least σ_v for both subtractive $\eta \geq 1$ and modified subtractive $\eta \leq 1$ algorithm of the Sugeno first order model

rules due to their strong representation ability. The Sugeno second order models are more suitable for weighted flowtime based sequence learning applications. For the example of moderate set of jobs, optimal second order model results in $\sigma = 64$ with 28 rules, while for the first order with 38 rules the error is $\sigma = 54$. The similar result of the first order model can be achieved by a second order model with 10 less rules. From the Tables 6.3 and 6.4, both Sugeno first order and second order models of modified subtractive clustering algorithm are better than the models of selective subtractive clustering algorithm.

scarce set	conventional		modified		-	Selective	
Sugeno	*1st*	*2nd*	*1st*	*2nd*	*Sugeno*	*1st*	*2nd*
η	1.30	1.05	0.20	0.25	δ	0.9	0.75
$\bar{\epsilon}$	0.20	0.00	0.70	1.00	-	-	-
$\underline{\epsilon}$	0.10	0.00	0.60	0.70	-	-	-
r_a	0.286	0.666	0.964	0.812	r_a	0.836	0.996
N	22	14	17	11	N	26	11
σ_v	241	189	129	162	σ_v	156	200

Table 6.3: The best models found by using conventional subtractive clustering method, selective subtractive and modified subtractive clustering method, with *scarce set* of jobs

moderate set	conventional		modified			Selective	
Sugeno	*1st*	*2nd*	*1st*	*2nd*	*Sugeno*	*1st*	*2nd*
η	1.10	1.10	0.85	0.40	δ	0.754	0.540
$\bar{\epsilon}$	0.20	0.00	0.00	0.40	-	-	-
$\underline{\epsilon}$	0.00	0.00	0.00	0.00	-	-	-
r_a	0.222	0.416	0.486	0.686	r_a	0.836	0.996
N	49	30	38	28	N	40	30
σ_v	77	72	54	64	σ_v	79	80

Table 6.4: The best models found by using conventional subtractive clustering method, selective subtractive and modified subtractive clustering method, with *moderate set* of jobs

Chapter 7

Conclusions and future work

Subtractive clustering based system identification method is discussed in this thesis. Firstly, we use the COSPI method to identify significant input variables. Secondly, we employ CIOS and LSE methods to build a fuzzy model. Finally, we utilize parametric search method to find the best model. A new scheduling decision learning approach is developed to solve scheduling problems. To learn scheduling decision mechanism from an existing schedule, a subtractive clustering based system identification method is introduced and utilized. It is also used to build a fuzzy expert scheduler. Then the fuzzy expert scheduler generates new schedules following the decision mechanism it has learned from the existing schedule. The existing schedule can be an optimal schedule produced by an optimization method. It can also be a training schedule generated by human expert scheduler. A decision objective function, which is sequential distance error, is introduced for the scheduling sequencing learning application. WSPT sequencing rule optimizes weighted flowtime problem in single machine model. The model built through learning the WSPT sequencing mechanism from the input and output of a schedule can be used to solve single machine weighted flowtime problem. The faulty representation of scarce data zone using conventional subtractive clustering algorithm is identified. Selective subtractive clustering and modified subtractive clustering algorithms are introduced to improve the performance of the conventional subtractive clustering algorithm.

Fuzzy expert model built by the human expert knowledge is flexible in dealing with real world scheduling problems. The limitation is that human expert scheduler may not always be available, or may not be able to explain their knowledge explicitly. In these situations, scheduling decision learning approach is more suitable. It learns scheduling decision mechanism from a schedule and builds a fuzzy expert model to generate new schedules. The experimental results of a WSPT sequence learning application in Section 6.2.2 show that the models built by this modeling methodology are successfully capturing the job sequence of *complete set*, *moderate set*, and *scarce set* of training jobs.

When validating the models which are built on the *scarce set* of training jobs, the generalization ability of the models built by conventional subtractive clustering algorithm is very low. The validation result shown in Figure 6.21 indicates that the model has faulty representation in the certain zone. The local representation behavior of subtractive clustering algorithm is shown in Figure 6.25. The constraint which is squash factor η greater than 1 lower the clustering performance when the scarcity of training data is high.

Selective subtractive clustering and modified subtractive clustering algorithms are introduced in Chapter 5 to remedy this situation. Using *degree of association* δ and selecting *squash factor* η less than 1 improve the identification of scarce data zone of subtractive clustering algorithm. With the same number of rules for the model shown in Figure 6.21, a model identified by modified subtractive clustering algorithm is able to capture the global behavior of the system as shown in Figure 6.22. The global representation of the training data space can be seen in Figure 6.25.

Sugeno first order and second order models are investigated. For the WSPT sequence learning application, Sugeno second order model has the same generalization ability as the first order model with between 8 and 21 less rules as shown in Table 6.3.

The WSPT surface in Figure 4.1 shows nonlinear relationship between input and output. The WSPT surface is a spiral shaped surface. Successful modeling of single machine weighted flowtime sequencing problem shows that the selective subtractive clustering and modified subtractive clustering algorithms are applicable to modeling systems with this kind of complex behavior.

Future research is suggested to focus on solving problems of multiple machines and multiple decisions using the subtractive clustering based system identification method. Investigating more complicated performance measures, which can be used to quantify the capabilities of the models in capturing training schedules, can also be an interesting work. Applying the selective subtractive clustering and modified subtractive clustering algorithm in solving the function approximation applications and pattern classification applications can also be part of future work.

Bibliography

[1] L. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Process," *IEEE Trans. on Systems, Man and Cybernetic*, vol. 1, pp. 28–44, 1973.

[2] R. M. Tong, "The Construction and Evalution of Fuzzy Models," in *In Advances in Fuzzy Set Theory and Applications* (M. M. Gopta, R. K. Rewgade, and R. R. Yager, eds.), pp. 559–576, Amsterdam, 1979.

[3] W. Pedrycz, "Identification in Fuzzy Systems," *IEEE Trans. on Fuzzy Syst.*, vol. 14, pp. 361–366, 1984.

[4] G. J. Trojan, J. B. Kiszka, M. M. Gupta, and P. N. Nikiforuk, "Solution of Multivariable Fuzzy Equation," *Fuzzy Sets and Syst.*, vol. 22, pp. 271–279, 1987.

[5] M. Sugeno and T. Yasukawa, "A Fuzzy-Logic-Based Approach to Qualitative Modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 7–31, Aug. 1993.

[6] R. R. Yager and D. P. Filev, "Approximate Clustering Via Mountain Method," *Tech. Report # MII1305, Machine Intelligence Institute, Iona College, New Rochelle, NY*, 1992.

[7] R. R. Yager and D. P. Filev, "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent and Fuzzy Systems*, vol. 2, pp. 267–278, 1994.

[8] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control.* New York: Wiley, 1994.

[9] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *Int. J. Man-Mach, Studies*, vol. 7, pp. 1–3, 1975.

[10] J. Bezdek, "Cluster Validity with Fuzzy Sets," *Journal of Cyberbetics*, vol. 3, pp. 58–71, 1974.

[11] J. Bezdek, R. Hathaway, M. Sabin, and W. Tucker, "Convergence Theory for Fuzzy c-means: Counter Examples and Repairs," in *The Analysis of Fuzzy Information* (J. Bezdek, ed.), vol. 3, ch. 8, CRC Press, 1987.

[12] M. R. Emami, I. B. Turksen, and A. A. Goldenberg, "Development of A Systematic Methodology of Fuzzy Logic Modeling," *IEEE Trans. on Fuzzy Syst.*, vol. 6, pp. 346–361, Aug. 1998.

[13] S. L. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, vol. 2, pp. 267–278, 1994.

[14] S. L. Chiu, "Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification," in *Fuzzy Information Engineering: a guide tour of applications* (D. Dubois, H. Prade, and R. R. Yager, eds.), ch. 9, pp. 149–162, John Wiley & Sons, INC., 1997.

[15] I. B. Turksen, T. Yurtsever, and K. Demirli, "Fuzzy Expert System Scheduler," *Proceedings, SPIE*, pp. 308–319, 1993.

[16] R. Caprihan, S. Kumar, and S. Wadhwa, "Fuzzy Systems for Sontrol of Flexible Machines Operation under Information Delays," *INT. J. PROD. RES.*, vol. 35, no. 5, pp. 1331–1348, 1997.

[17] D. Ben-Arieh and E. S. Lee, "Fuzzy Logic Controller for Part Routing," in *Design and implementation of intelligent manufacturing systems: from expert systems, neural network, to fuzzy logic* (H. R. Parsaei and M. Jamshidi, eds.), ch. 5, pp. 81–106, Prentice Hall PTR, 1995.

[18] R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling.* Addsion Wesley, 1967.

[19] K. R. Baker, *Introduction to Sequencing and Scheduling.* John Wiley & Sons, INC., 1974.

[20] S. French, *Sequencing and Scheduling: An Introduction to Mathematics of Job-shop.* Ellis Horwood Limited, 1982.

[21] W. Slany, "Fuzzy Scheduling," *CD-Technical Report, 66*, 1994.

[22] K. G. Murty, *Operations Research: Deterministic Optimization Models.* Prentice Hall, 1995.

[23] I. B. Turksen, "Scheduling System Design: Three Fuzzy Theory Approaches," in *Fuzzy Information Engineering* (D. Dubios, H. Prade, and R. R. Yager, eds.), John Wiley & Sons, INC., 1997.

[24] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and its Application to Modeling and Control," *IEEE Trans. on Systems, Man and Cybernetic*, vol. 15, pp. 116–132, 1985.

[25] Z. Chi, H. Yan, and T. Pham, *Fuzzy Algorithm: with Applications to Image Processing and Pattern Recognition.* Singapore: World Scientific Pub., 1996.

[26] J. Dunn, "A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well-Separated Clusters," *Journal of Cyberbetics*, vol. 3, pp. 32–57, 1974.

[27] H. Imai, A. Tanaka, and M. Miyakoshi, "A Method of Identifying Influential Data in Fuzzy Clustering," *IEEE Trans. on Fuzzy Syst.*, vol. 6, pp. 90–101, Febuary 1998.

[28] R. krishnapuram and J. M. Keller, "A Possiblistic Approach to Clustering," *IEEE Trans. on Fuzzy Syst.*, vol. 1, pp. 98–110, 1993.

[29] K. Demirli and P. Muthukumaran, "Fuzzy System Identification with Higher Order Subtractive Clustering," *To be Submitted to the journal of Intelligent and Fuzzy Systems*, 1999.

[30] K. Demirli, S. X. Cheng, and P. Muthukumaran, "Fuzzy Modeling of Job Sequenceing with Subtractive Clustering," *To be submitted to the special issue of the journal of Robotics and Autonomous Systems*, 1999.

Appendix A

Effect of cluster radius and squash factor on modeling of the training jobs

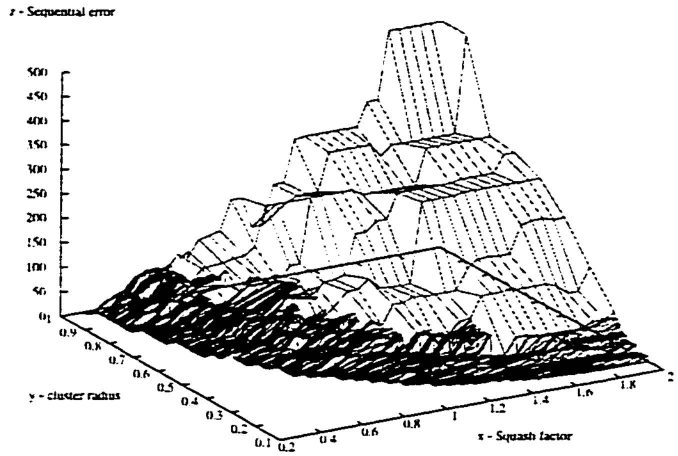


Figure A.1: Variation of σ_t in relation to r_a and η for *scarce set* at $\bar{\epsilon} = 0.2$, $\underline{\epsilon} = 0.1$ for Sugeno first order model

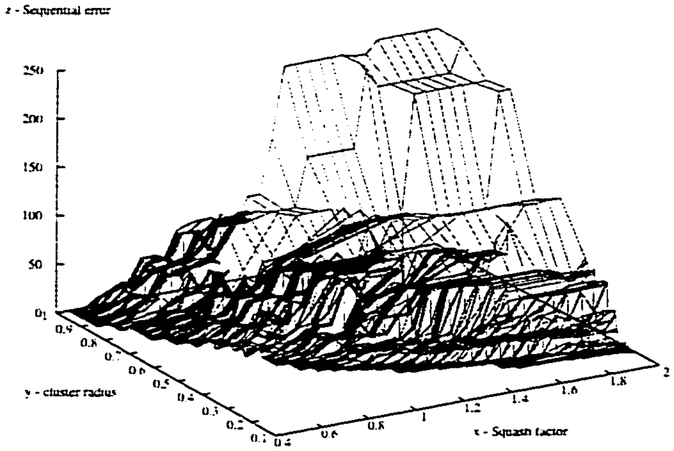


Figure A.2: Variation of σ_t in relation to r_a and η for *scarce set* at $\bar{\epsilon} = 0.2$, $\underline{\epsilon} = 0.1$ for Sugeno second order model

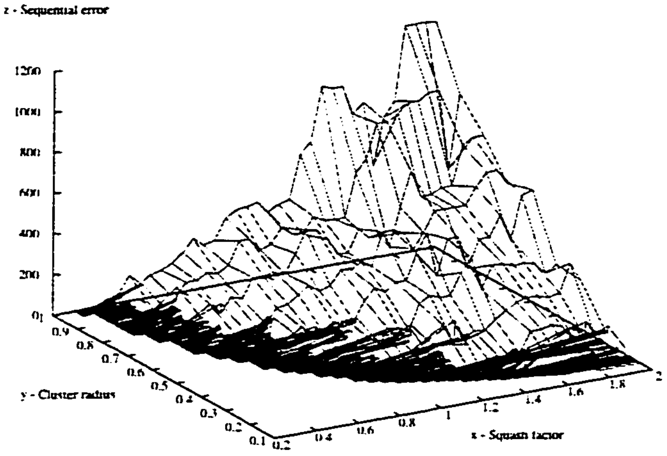


Figure A.3: Variation of σ_t in relation to r_a and η for *moderate set* at $\bar{\epsilon} = 0.0$, $\underline{\epsilon} = 0.0$ for Sugeno first order model

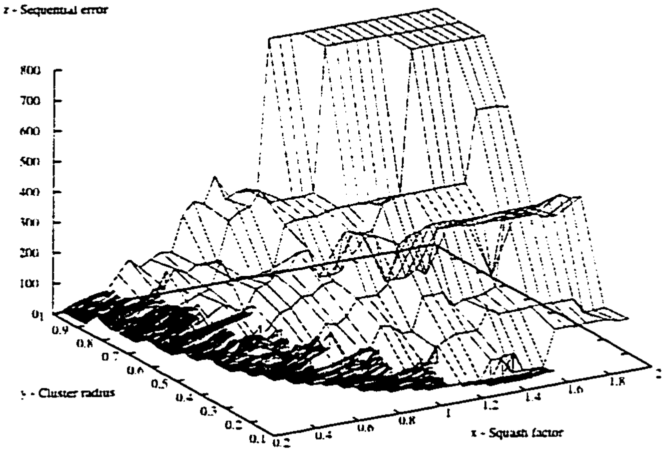


Figure A.4: Variation of σ_t in relation to r_a and η for *moderate set* at $\bar{\epsilon} = 0.0$, $\underline{\epsilon} = 0.0$ for Sugeno second order model

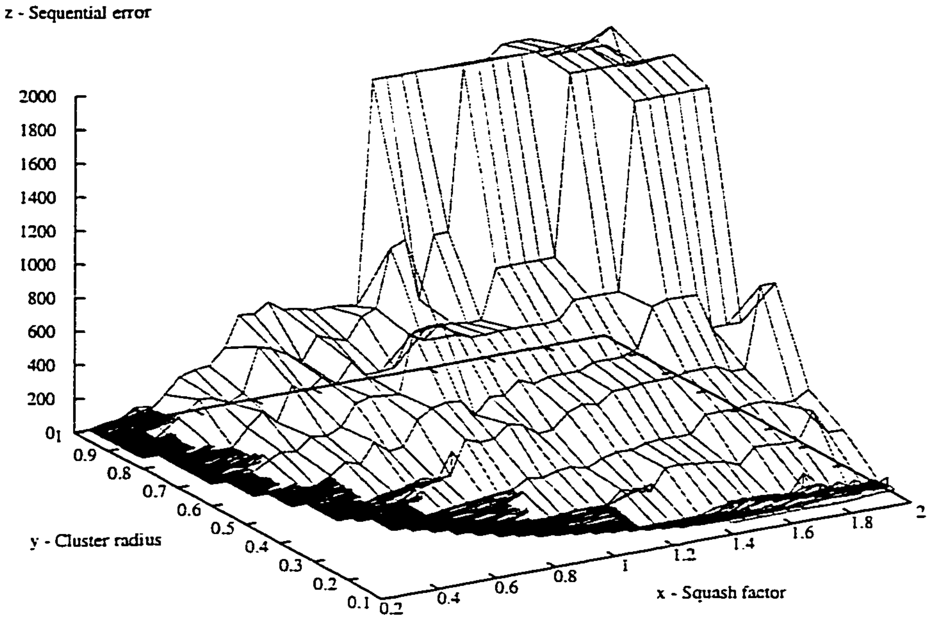


Figure A.5: Variation of σ_t in relation to r_a and η for *complete set* at $\bar{\epsilon}$ = 0.3. $\underline{\epsilon}$ = 0.1 for Sugeno second order model

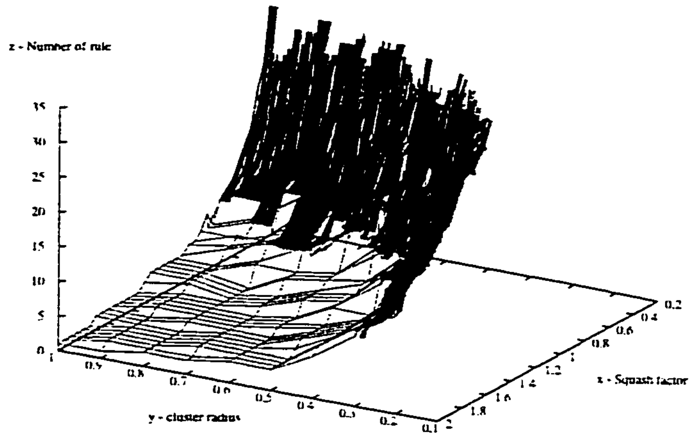


Figure A.6: Variation of number of rules in relation to r_a and η for *scarce set* at $\bar{\epsilon} = 0.2$, $\underline{\epsilon} = 0.1$ for Sugeno first order model

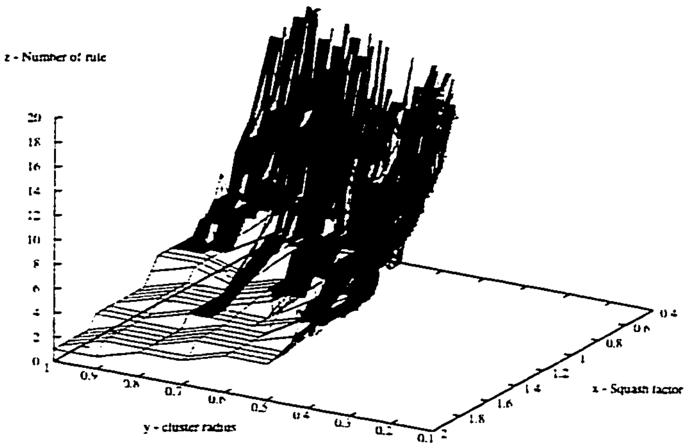


Figure A.7: Variation of number of rules in relation to r_a and η for *scarce set* at $\bar{\epsilon} = 0.2$, $\underline{\epsilon} = 0.1$ for Sugeno second order model

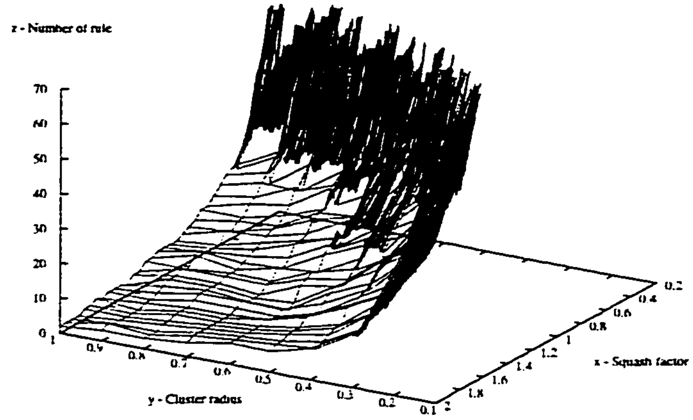


Figure A.8: Variation of number of rules in relation to r_a and η for *moderate set* at $\bar{\epsilon} = 0.0$, $\underline{\epsilon} = 0.0$ for Sugeno first order model

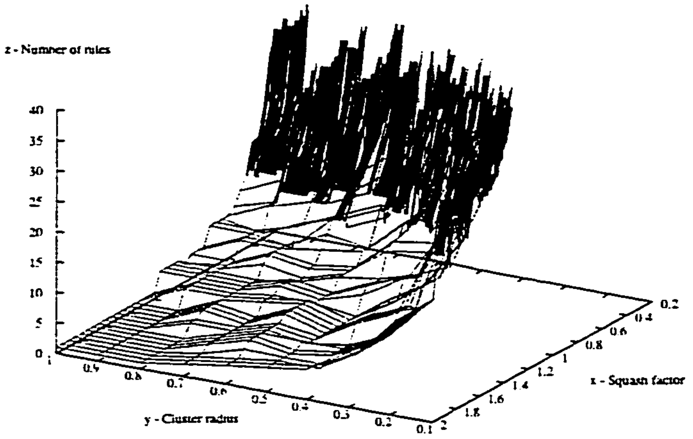


Figure A.9: Variation of number of rules in relation to r_a and η for *moderate set* at $\bar{\epsilon} = 0.0$, $\underline{\epsilon} = 0.0$ for Sugeno second order model

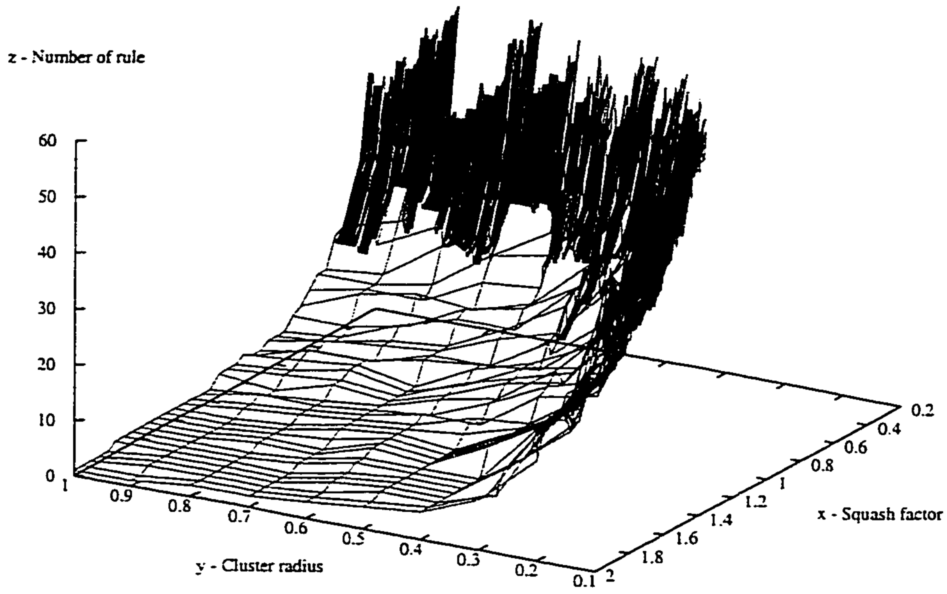


Figure A.10: Variation of number of rules in relation to r_a and η for *complete set* at $\bar{\epsilon} = 0.3$, $\underline{\epsilon} = 0.1$ for Sugeno second order model

Appendix B

Effect of parameters on optimal model

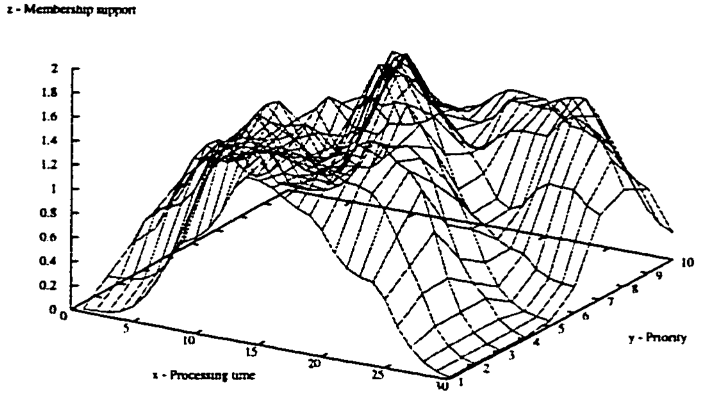


Figure B.1: ϕ at η of 1.60 for *scarce set* for Sugeno second order model

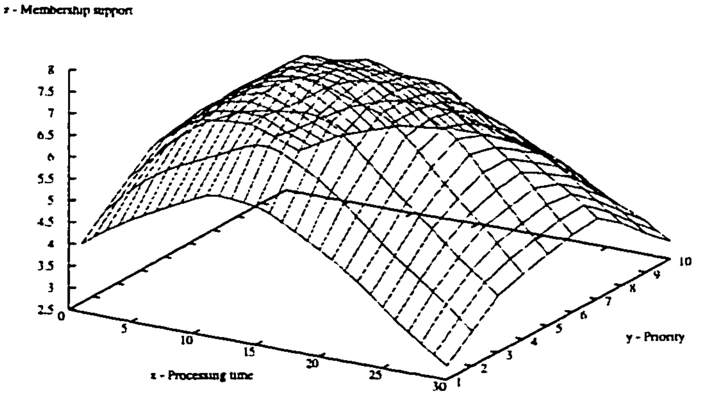


Figure B.2: ϕ at η of 0.55 for *scarce set* for Sugeno second order model

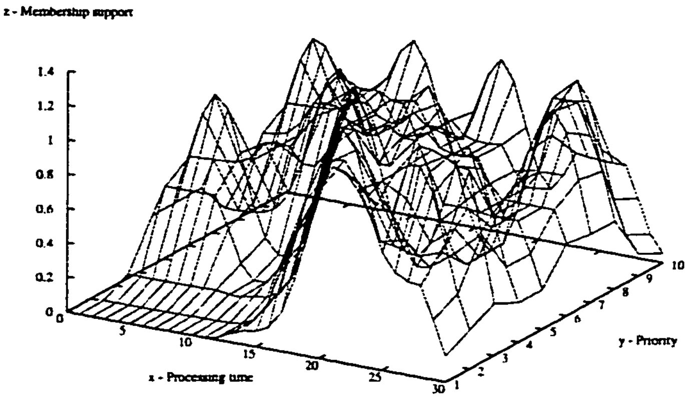


Figure B.3: ϕ at η of 1.5 for *moderate set* for Sugeno first order model

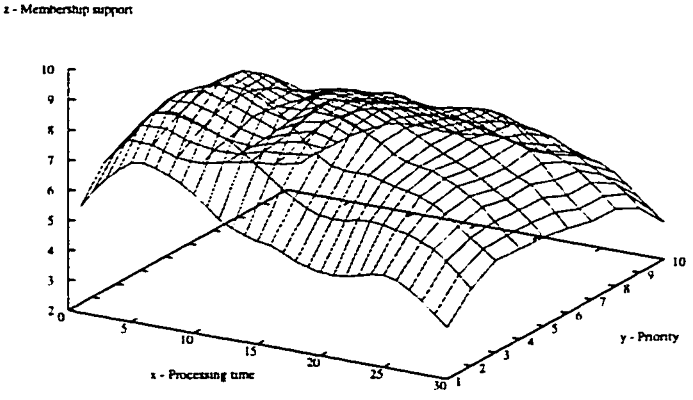


Figure B.4: ϕ at η of 0.85 for *moderate set* for Sugeno first order model

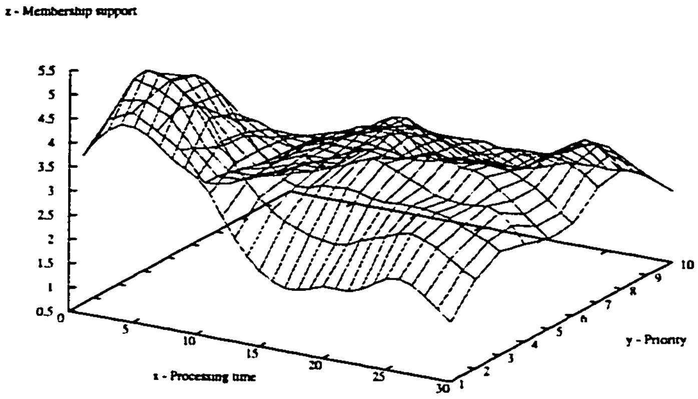


Figure B.5: ϕ at η of 1.25 for *moderate set* for Sugeno second order model

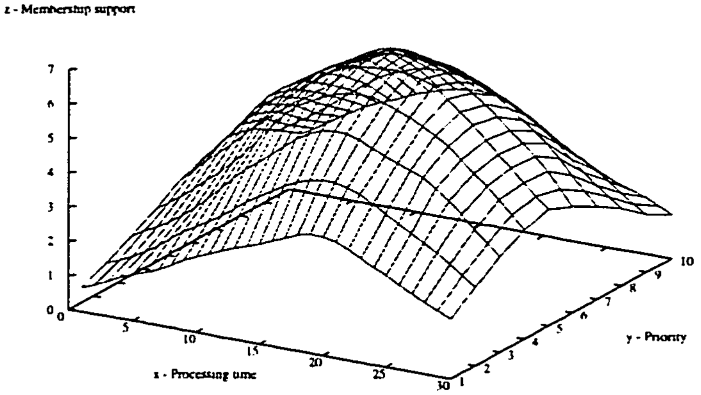


Figure B.6: ϕ at η of 0.85 for *moderate set* for Sugeno second order model

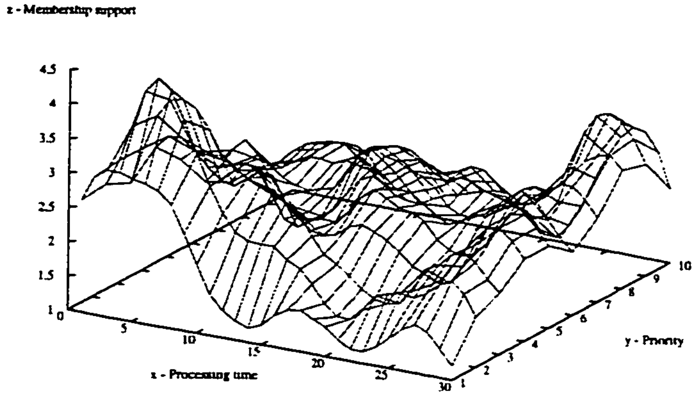


Figure B.7: ϕ at η of 1.75 for *complete set* for Sugeno first order model

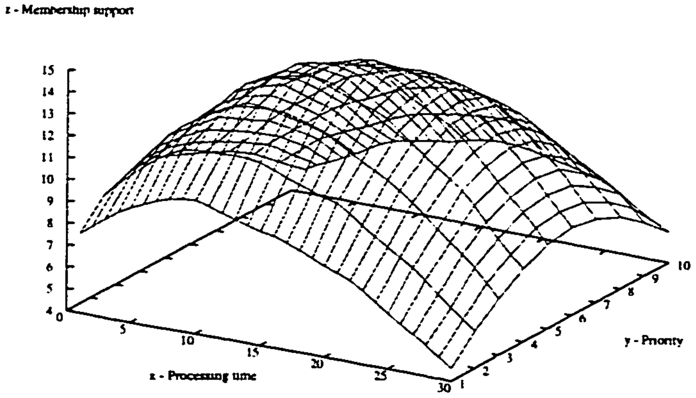


Figure B.8: ϕ at η of 0.55 for *complete set* for Sugeno first order model

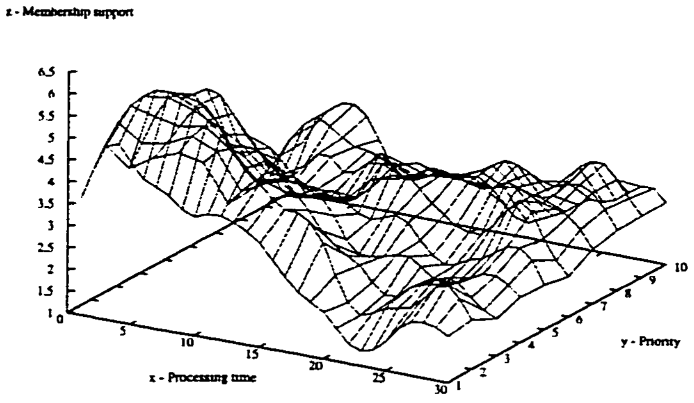


Figure B.9: ϕ at η of 1.75 for *complete set* for Sugeno second order model

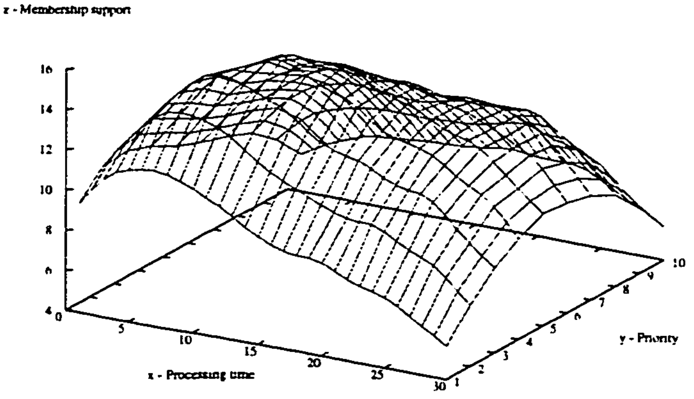


Figure B.10: ϕ at η of 0.55 for *complete set* for Sugeno second order model

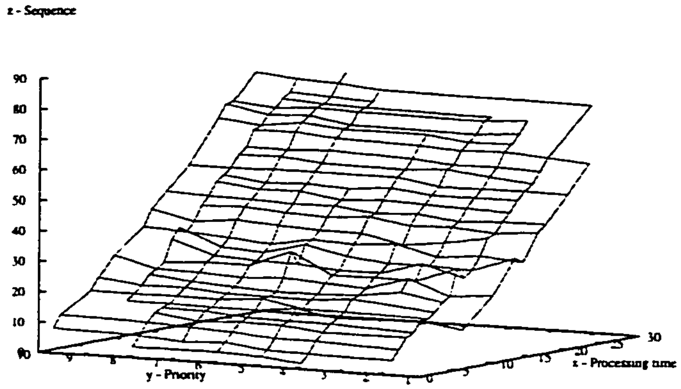


Figure B.11: Validation surface of sequence in relation to p_j and w_j at η of 1.60 for *scarce set* for Sugeno second order model

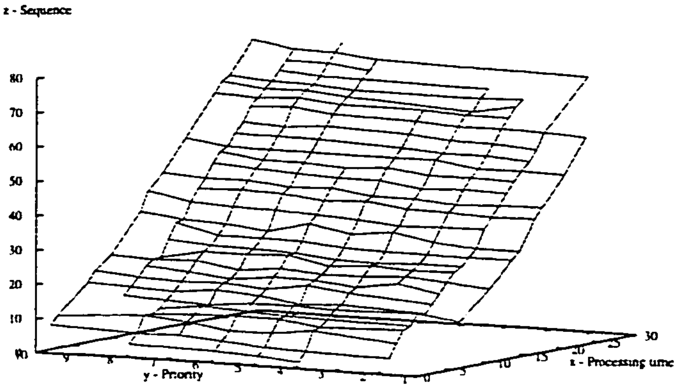


Figure B.12: Validation surface of sequence in relation to p_j and w_j at η of 0.55 for *scarce set* for Sugeno second order model

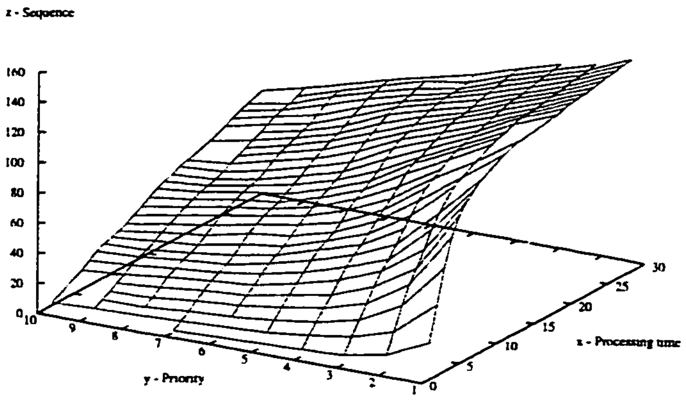


Figure B.13: Validation surface of sequence in relation to p_j and w_j at η of 1.55 for *moderate set* for Sugeno first order model

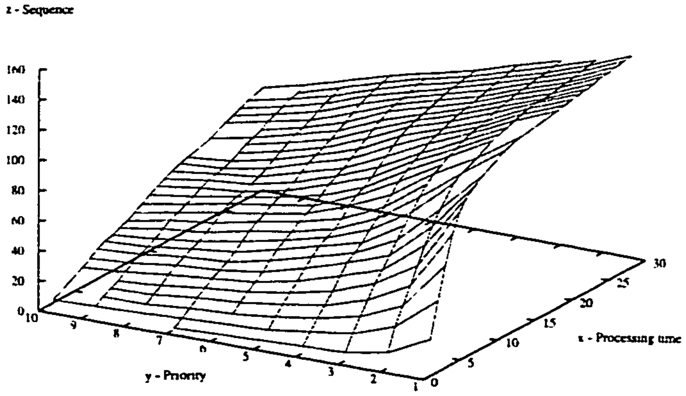


Figure B.14: Validation surface of sequence in relation to p_j and w_j at η of 0.85 for *moderate set* for Sugeno first order model

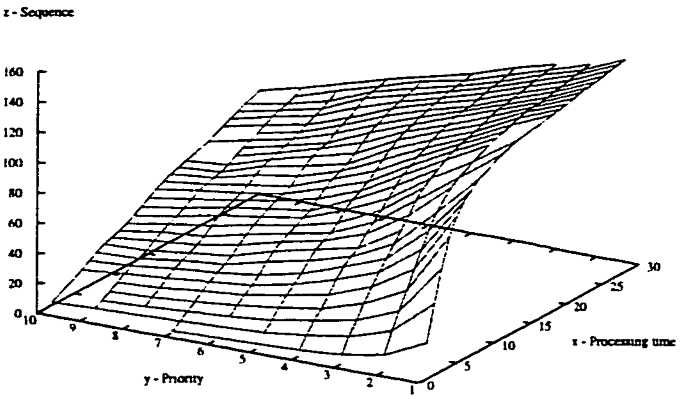


Figure B.15: Validation surface of sequence in relation to p_j and w_j at η of 1.25 for *moderate set* for Sugeno second order model

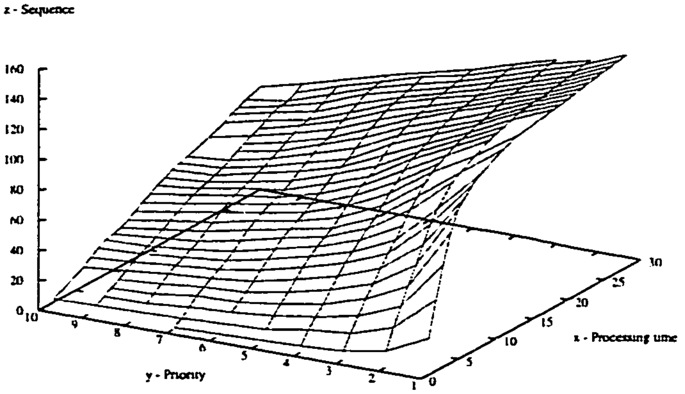


Figure B.16: Validation surface of sequence in relation to p_j and w_j at η of 0.45 for *moderate set* for Sugeno second order model

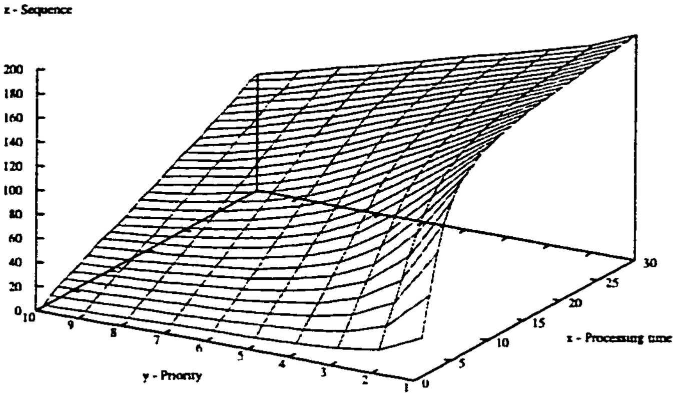


Figure B.17: Validation surface of sequence in relation to p_j and w_j at η of 1.50 for *complete set* for Sugeno first order model

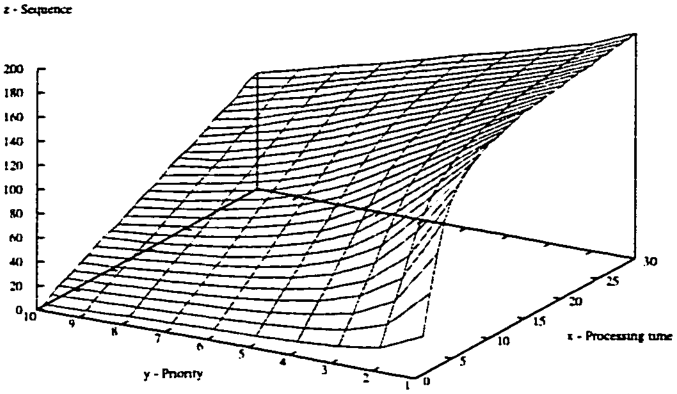


Figure B.18: Validation surface of sequence in relation to p_j and w_j at η of 0.60 for *complete set* for Sugeno first order model

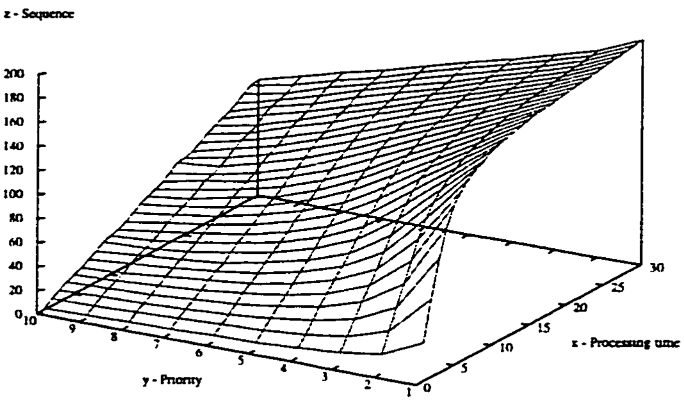


Figure B.19: Validation surface of sequence in relation to p_j and w_j at η of 1.70 for *complete set* for Sugeno second order model

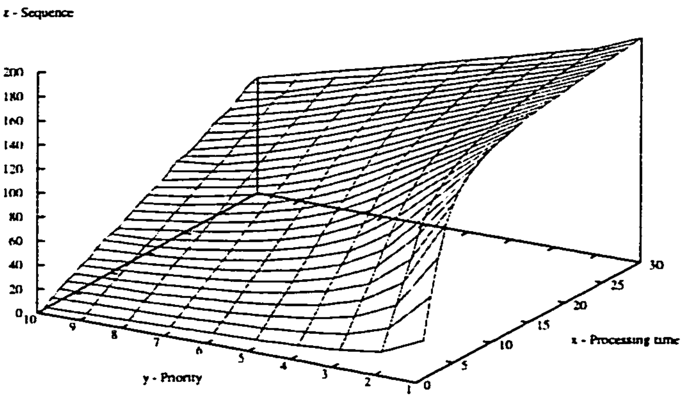


Figure B.20: Validation surface of sequence in relation to p_j and w_j at η of 0.50 for *complete set* for Sugeno second order model