DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

GROUP TECHNOLOGY AND CELLULAR MANUFACTURING WITH ARTIFICIAL NEURAL NETWORKS

by

Ömer ÖZKAN

March, 2010

İZMİR

GROUP TECHNOLOGY AND CELLULAR MANUFACTURING WITH ARTIFICIAL NEURAL NETWORKS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science in
Industrial Engineering, Industrial Engineering Program

by
Ömer ÖZKAN

March, 2010
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"GROUP TECHNOLOGY AND CELLULAR MANUFACTURING WITH ARTIFICIAL NEURAL NETWORKS"** completed by **ÖMER ÖZKAN** under supervision of **ASSIST.PROF.DR. ÖZCAN KILINÇCI** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist.Prof.Dr. Özcan KILINÇCI

Supervisor

Prof.Dr. G.Miraç BAYHAN

(Jury Member)

Assist.Prof.Dr. Umay KOÇER

(Jury Member)

Prof.Dr. Mustafa SABUNCU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

This thesis is by far the most significant scientific accomplishment in my life and it would be impossible without people who supported me and believed in me.

Most of all I would like to thank to my research advisor, Assist.Prof.Dr. Özcan Kılınçcı for his valuable advice, encouragement and guidance of this thesis. His trust and scientific experience inspired me in making the right decisions and I am really glad to have worked with him.

Finally, I wish to express my love and thanks to all my family. Therefore, I dedicate this thesis to my dearest wife Gülşah and to my family; Gülsevim, Gülşen, Yunus, Mehmet, Işıl, Onur and Önder who have provided me constant support, endless love, patience and encouragement. I am particularly grateful to them.

Ömer ÖZKAN

GROUP TECHNOLOGY AND CELLULAR MANUFACTURING WITH ARTIFICIAL NEURAL NETWORKS

ABSTRACT

Group Technology (GT) / Cellular Manufacturing (CM) is a useful way of increasing the productivity in manufacturing high quality products, improving the flexibility of manufacturing systems and decreasing the costs. Cell Formation (CF) is the key step for GT. CF can identify part families and machine groups. Several kinds of methods can be used in CF. Artifial Neural Networks (ANNs) are very suitable for CF and have been widely applied in CF due to their robust and adaptive nature.

In the thesis, a review of different kinds of ANNs from the literature which are used in CF, is presented. An application of Self Organizing Map (SOM) and Competitive Neural Network (CNN) within a new methodology for grouping binary and nonbinary (fuzzy) problem sets simultaneously is made. 15 problem sets gathered from the literature are used as binary problem sets and 6 problem sets gathered from the literature are used as nonbinary problem sets. A performance measure which is created by taking the aritmetic average of five different well-known performance measures from the literature is proposed and used to evaluate and compare the cell solutions. Also, the performance measures in the articles the problem sets are taken from, are used once more to evaluate and compare the cell solutions. SOM and CNN results are compared with the results in the literature. In the last part of the application, different numbers of cells are tested to see whether there is a better cell configuration than the article has found. Matlab 7.5 is used to code the neural networks and find the best groupings.

Keywords: Group Technology, Cellular Manufacturing, Artificial Neural Networks, Cell Formation Problem.

YAPAY SİNİR AĞLARI İLE GRUP TEKNOLOJİ VE HÜCRESEL İMALAT

ÖZ

Grup Teknoloji (GT) / Hücresel İmalat (Hİ) yüksek kaliteli ürünlerin üretiminde verimliliği artırmak, üretim sistemlerinin esnekliğini geliştirmek ve maliyetleri düşürmek için yararlı bir yöntemdir. Hücrelerin Oluşturulması (HO), GT için anahtar bir adımdır. HO'nda parça aileleri ve makine grupları belirlenir. HO için değişik yöntemler kullanılabilir. Yapay Sinir Ağları (YSA) güçlü ve uyarlanabilir yapıları ile HO için oldukça uygundur.

Bu tezde, literatürde HO'nda kullanılan değişik türdeki YSA'nı içeren çalışmalar özetlenmiştir. Kendini Örgütleyen Ağlar (SOM) ve Rekabetçi Sinir Ağları (CNN)'un ikili ve ikili olmayan (bulanık) problem setlerini yeni bir metodoloji ile eşzamanlı olarak grupladığı uygulamalar gerçekleştirilmiştir. Literatürden seçilmiş 15 problem seti ikili problem setleri olarak ve literatürden bulunmuş 6 problem ikili olmayan problem setleri olarak ele alınmıştır. Tezde, hücre sonuçlarının değerlendirilmesi ve karşılaştırılması için literatürde yaygın olarak kullanılan 5 adet farklı performans ölçütünün aritmetik ortalamasından elde edilmiş bir performans ölçütü önerilmiş ve kullanılmıştır. Ayrıca, problem setlerinin alındığı makalelerde kullanılan performans ölçütleri de hücre sonuçlarının değerlendirilmesi ve karşılaştırılması için tekrar kullanılmıştır. Elde edilen SOM ve CNN sonuçları literatür sonuçları ile karşılaştırılmıştır. Uygulamanın son bölümünde, makalenin bulduğundan daha iyi bir hücre yapılanmasının olup olmadığını görmek için farklı hücre sayıları test edilmiştir. Matlab 7.5 programı sinir ağlarının kodlanması ve en iyi sonuçların bulunması için kullanılmıştır.

Anahtar Sözcükler: Grup Teknoloji, Hücresel İmalat, Yapay Sinir Ağları, Hücre Oluşturma Problemi.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

1.1 Cell Formation Problem

The manufacturing sector has become increasingly competitive, as markets have become more globalized. Producers of goods are under intense pressure to improve their operations by enhancing productivity, quality, customer responsiveness, and reducing manufacturing costs. Consequently, there have been major shifts in the design of manufacturing systems using innovative concepts (Hachicha, Masmoudi & Haddar, 2007).

The production process requires a variety of machines and often some complex procedures. Frequently, parts have to be moved from one place to another. This results not only in machine idle time but also wastes the manpower required for the physical movement of the parts. On the other hand, an increasing number of companies are encountering small to medium size production orders. In this situation, more setup changes and frequent part or machine movements occur (Yang & Yang, 2008).

The adoption of Group Technology (GT) has consistently formed a central element of many of these efforts and has received considerable interest from both practitioners and academicians (Hachicha, Masmoudi & Haddar, 2007). GT is a manufacturing philosophy that has attracted a lot of attention because of its positive impact in the batch-type production (Murugan & Selladurai, 2007). When GT is applied to the manufacturing field, it takes the form of Cellular Manufacturing System (CMS) (Lee, Yamakawa & Lee, 1997).

CMS has emerged in the last two decades as an innovative manufacturing strategy that collects the advantages of both product and process oriented systems for a high variety and medium volume product mix (Burbidge, 1992). Parts are grouped into part families based on the similarity in design and manufacturing and the machines which are needed to process the parts in a part family are put together to form a manufacturing cell. Unlike the job-shop system, machines in a manufacturing cell are dissimilar and cells are formed in a manner that all the parts in a family can be processed completely or nearly completely within a cell. CMS has the following benefits: reduction in working process inventory, setup time, throughput time and material handling cost, improvement in production quality (Lee, Yamakawa & Lee, 1997). One important advantage of Cellular Manufacturing (CM) is that production control is considerably simplified and a more realistic delivery quotation can be given to customers. That is because of the possibility of more accurately forecasting the time by which finished products will be dispatched after the job has been issued to the works (Hachicha, Masmoudi & Haddar, 2007).

GT has proven to be a useful way of addressing the difficulties of the manufacturing environment by creating a more flexible manufacturing process. It can be used to exploit similarities between components to achieve lower costs and increase productivity without loosing product quality. Cell Formation (CF) is a key step in GT. It is a tool for designing CMSs using the similarities between parts and machines to have part families and machine groups. The parts in the same machine group have similar requirements, reducing travel and setup time (Yang & Yang, 2008). The process of determining the part families and machine groups are referred to as the CF problem (Murugan & Selladurai, 2007).

The CF problem consists in grouping machines into cells and in determining part families such that parts of a family are entirely processed in one cell. Unfortunately, it is not always possible to ensure that a part is treated in one cell, because a machine of a cell may be required by parts from different families. Such parts or machines are called exceptional elements and are to be minimized when assigning parts and machines to cells (Shambu, Suresh & Pegels, 1996). An exceptional machine which also called bottleneck machines processes parts from two or more part families. An exceptional part can be viewed as parts that require processing on machines in two or more cells (Hachicha, Masmoudi & Haddar, 2007).

The CF problem is a large problem requiring a hierarchical procedure involving heuristic procedures and subjective inputs at several stages. Within this large problem context, most of the methods developed to date have addressed the initial part–machine grouping problem. This problem attempts to identify families of parts that require the same set of machines without considering the sequence in which they are required. This addresses, in effect, the creation of jobshop-like cells, or there is often a tacit assumption that material flows and minimization of backtracks within cells will be considered later in the overall CF problem (Park & Suresh, 2003). Considering the large number of parts and machines involved in the industrial design problem, efficient solution methods are highly desirable (Zolfaghari, 1997).

1.2 Thesis Motivation

In this thesis, GT and CMS are introduced. The CF problem is defined in detail. The CF methods are covered. A a review of different kinds of ANNs from literature which are used in CF, is presented. An application of Self Organizing Map (SOM) and Competitive Neural Network (CNN) within a new methodology for grouping the binary and nonbinary (fuzzy) problem sets simultaneously is covered. The new methodology used in both binary and nonbinary problems. Gathered 15 problem sets from literature are used as binary problem sets. Gathered 6 problem sets from literature are used as nonbinary problem sets. A performance measure which is proposed by aritmetic average of five different well-known performance measure found from literature is used to evaluate and compare the solutions for the cells. Also, the performance measures used in the articles the problem sets are taken from used again to evaluate and compare the solutions for the cells. The SOM and CNN results are compared with the literature results. In the last part of the application, different numbers of cells are tested to see whether there is a better cell configuration than the article has found. Matlab 7.5 is used to code the neural networks and find the best groupings.

The main aim of the present thesis is to implement SOM and CNN within the proposed methodology in chapter four to the CF problem using binary inputs, Fuzzy

SOM and Fuzzy CNN using nonbinary inputs. In the application section following, the aim of the thesis will be realized in two steps. The first step is to test if the proposed methodology works by using binary inputs. Solved binary problems by different kinds of methods within different methodologies are chosen from articles and solved again by SOM and CNN within the proposed methodology. For every problem, the results of the article the problem is taken from and the results of the present thesis are compared to give a decision about the use of the proposed methodology. So, the proposed methodology is tested for binary inputs, then the second step is to use it for nonbinary inputs. The procedure applied for the binary problems is implemented for nonbinary problems chosen from articles. Methods of Fuzzy SOM and Fuzzy CNN are used within the proposed methodology. The advantage of using SOM and CNN is that they are unsupervised neural networks. The thesis presents the CF methodology of unsupervised neural networks for grouping binary and nonbinary problem sets. Fuzzy SOM and Fuzzy CNN are used for the first time in literature for grouping nonbinary problem sets.

1.3 Thesis Outline

The thesis is organized as follows:

Chapter One contains introduction with a brief description of the CF problem and describes the motivation and scopes of the study.

Chapter Two concerns definition of GT, CMSs and CF problem in detail. Also it presents the traditional manufacturing systems, CMSs, the advantages/disanvantages of CM, CF methods and performance measures of cell groupings.

Chapter Three explains ANNs application in CM in detail. In the chapter, Artificial Intelligence (AI), the connection between human brain and ANNs, history of ANNs, the advantages/disanvantages of ANNs, ANNs applications, architecture of ANNs, learning types of ANNs, types of ANNs, and a detailed literature survey on CF with ANNs are covered.

Chapter Four suggests the methodologies of CF with binary and nonbinary (fuzzy) inputs. Also chapter explains the proposed performance measure of cell groupings.

Chapter Five includes the application of binary cases and nonbinary cases. In the chapter, problem sets, neural network variables, MATLAB codes, results, comparisons and discussions are presented.

Chapter Six summarizes the findings of this study and states the conclusions.

CHAPTER TWO
GROUP TECHNOLOGY AND CELLULAR MANUFACTURING

2.1 The Types of Manufacturing Systems

Shorter life-cycles, unpredictable demand and customized products have forced manufacturers to improve the efficiency and productivity of their production activities. Manufacturing systems must be able to produce items with low production costs and high quality as possible in order to meet the customers' demand on time. Moreover manufacturing systems have gone through major changes during recent years mainly due to advances in technology and new strategies to deal with the technology. Informational vagueness in parameter estimates is being recognized as a reality in most of the problems in manufacturing system design. Manufacturing systems, today, should be able to respond quickly to changes in product design, product demand, technology etc. Traditional manufacturing systems such as job shops and flow lines are not capable of satisfying such requirements. The concept of CM is one of the most effective strategies to the changing worldwide competitive environment (Eski, 2007).

2.1.1 Traditional Manufacturing Systems

Job shops and flow lines are the examples of the traditional manufacturing systems. In general, job shops are designed to achieve maximum flexibility such that a wide variety of products with small lot sizes can be manufactured. Products manufactured in job shops usually require different operations and have different operation sequences. Operating time for each operation could vary significantly. Products are released to the shops in batches (jobs). The requirements of the job shop - a variety of products and small lot sizes - dictate what types of machines are needed and how they are grouped and arranged. General-purpose machines are utilized in job shops because they are capable of performing many different types of operations. Machines are functionally grouped according to the general type of manufacturing process: lathes in one department, drill presses in another, and so forth. Figure 2.1 illustrates a job shop. A job shop layout can also be called a functional layout (Mungwattana, 2000).

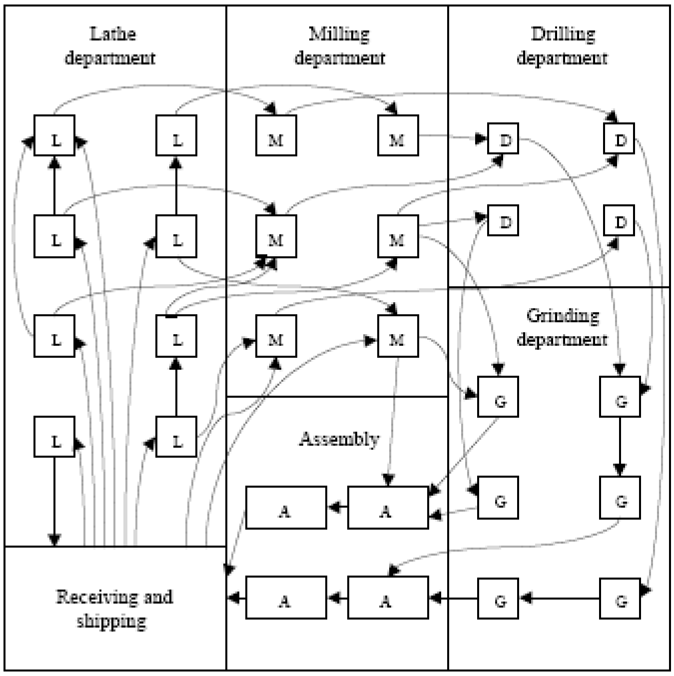

Figure 2.1 Job shop manufacturing (Mungwattana, 2000).

Such a job shops system involves about only 5% of the time being spent on a machine in productive activity with the remaining 95% being spent moving and waiting - nonproductive activity (Kioon, 2007). When the processing of a part in the job shop has been completed, it usually must be moved a relatively large distance to reach the next stage. It may have to travel the entire facility to complete all of the required processes, as shown in Figure 2.1. Therefore, to make processing more economical, parts are moved in batches. Each part in a batch must wait for the remaining parts in its batch to complete processing before it is moved to the next stage. This leads to longer production times, high levels of in-process inventory, high production costs and low production rates (Mungwattana, 2000). This extensive movement increases total material handling cost and decreases system productivity. These limitations are forcing traditional manufacturers to consider changing and improving their facilities to improve productivity (Abduelmola, 2000).

In contrast to job shops, flow lines are designed for high volume industries and require high capital commitments while retaining little production flexibility. A flow line is organized according to the processing sequence of a product. Specialized machines dedicated to the manufacture of utilized to achieve high production rates. Figure 2.2 shows an example of a flow line (Eski, 2007).

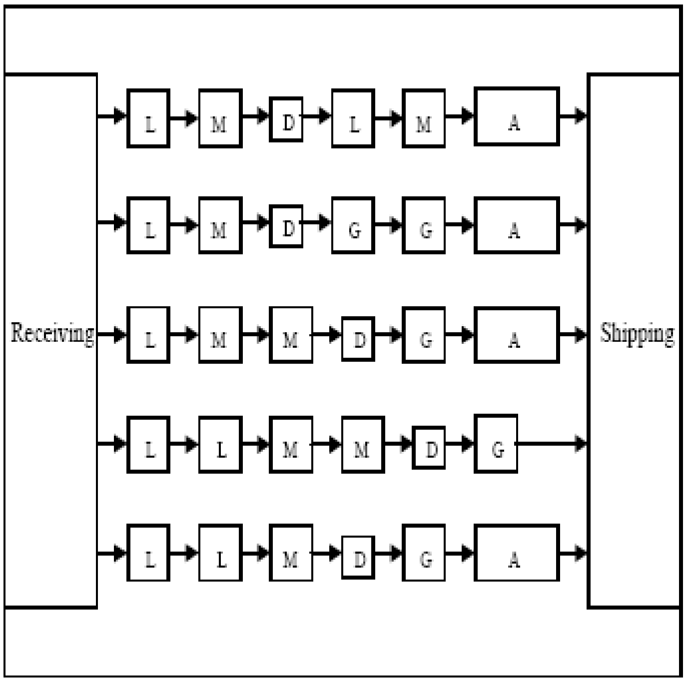

Figure 2.2 Flow line manufacturing (Mungwattana, 2000).

2.1.2 Cellular Manufacturing Systems (CMS)

As indicated above, job shops and flow lines cannot simultaneously provide the flexibility and efficiency requirements of today's production (Defersha, 2006). Within the manufacturing context, GT is defined as a manufacturing philosophy identifying similar parts and grouping them together into families to take advantage of their similarities in design and manufacturing (Selim, Askin & Vakharia, 1998). For other definitons; GT is a manufacturing philosophy that identifies and exploits the underlying sameness of parts and manufacturing processes (Ham, Hitomi & Yoshida, 1985). GT is an approach to manufacturing and engineering management that helps manage diversity by capitalizing on underlying similarities in products and activities (Selim, Askin & Vakharia, 1998).

The attractions for the pioneers of GT were, however based on cost directly but rather indirectly as a result of having more effective control over the manufacturing systems. GT can be one critical element in the rejuvenation of outdated and unproductive plant. GT adresses the following issues as a single coherent problem (Jaganathan, 2007):

- Components are aggregated into families with similar production requirements,
- Small groups of machines are matched to the component families,
- Groups of operatives are assigned to cells.

The basic idea of part families manufacture originally consisted of grouping parts with similar machining characteristics together to form so-called "additive batches" and routing them through the functional machine layout with the assistance of the production control. The basic idea of the GT cell is to split the manufacturing area into machine groups in which all the machining operations required for the manufacture of a certain parts spectrum can be accomplished. Within the GT cell itself all the forms of work can be employed with advantage that the task area is limited in such a way that the members of the group also have the feeling of belonging to a team. GT can be an effective tool in addressing large size facility layout problems (Jaganathan, 2007).

GT conceived during the 1940s in the USSR (Burbidge, 1963) for improving productivity in batch production systems. Batch manufacturing is estimated to be the most common form of production. There is a growing need to make batch manufacturing more efficient and productive. GT is best-suited to a batch-flow production system where many different parts, having relatively low annual volumes, are produced in small lot sizes (Carrie, 1973). GT was first proposed by Mitrofanov in 1966, and was propagated by Burbidge in 1971, who developed methods suitable for hand computation. Skinner (1974) was the first to propose the concept of a focused factory, in which small manufacturing systems operate independently within large production plants. The idea works best for medium-variety, medium-volume situations, that is, batch production. The focused factory is constructed using the notions of either Flexible Manufacturing Systems (FMS) or GT, which are based on the precept that certain activities should be dedicated to a family of related parts in a manufacturing cell. Later, Burbidge developed and popularized a systematic approach to this concept in 1975, which has subsequently seen widespread adoption in western industry (Foulds & Wilson, 2002). Among the well known methods of grouping based on binary data – Singh used the PMIM as the basic input data in 1993 (Mahdavi, Kaushal & Chandra, 2001).

One application of the GT philosophy is CM (Hachicha, Masmoudi & Haddar, 2007). CM is an application of the GT philosophy to designing manufacturing systems. (Mahdavi, Javadi, Fallah-Alipour & Slomp, 2007). The job shop in Figure 2.1 is converted into a CMS as shown in Figure 2.3. Obvious benefits gained from the conversion of the shop are less travel distance for parts, less space required, and fewer machines needed. Since similar part types are grouped, this could lead to a reduction in setup time and allow a quicker response to changing conditions. On the other hand, in the job shop, each part type may have to travel through the entire shop; hence scheduling and materials control are difficult. In addition, job priorities are complex to set and hence large inventories are needed so as to ensure that ample work is available (Mungwattana, 2000).

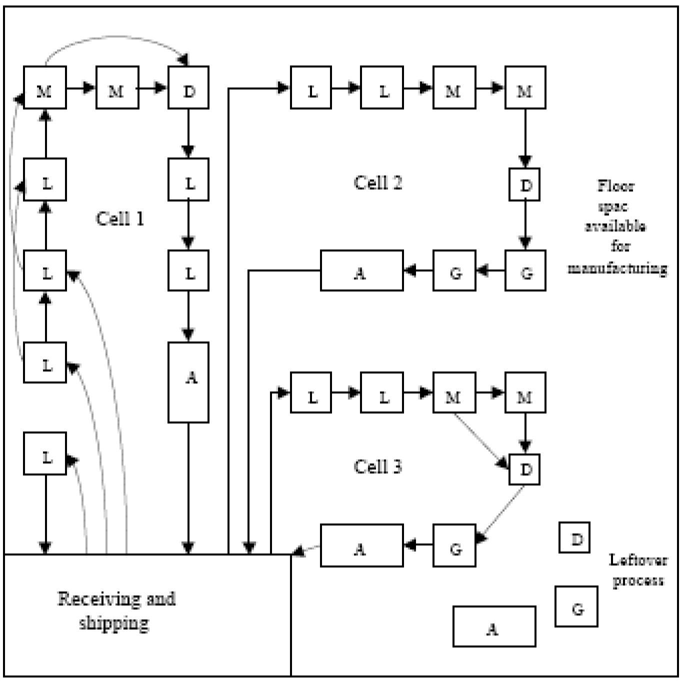

Figure 2.3 Cellular manufacturing (Mungwattana, 2000).

CM is a hybrid system linking the advantages of both job shops (flexibility in producing a wide variety of products) and flow lines (efficient flow and high production rate). In CM, machines are located in close proximity to one another and dedicated to a part family (Mungwattana, 2000). A part family is defined as a collection of parts that can be processed on the same group of machines because of geometric shape and size or similar processing steps required in their manufacture (Kioon, 2007). This provides the efficient flow and high production rate similar to a flow line. The use of general-purpose machines and equipment in CM allows machines to be changed in order to handle new product designs and product demand with little efforts in terms of cost and time. So it provides great flexibility in producing a variety of products (Mungwattana, 2000).

According to Hayret (2000) CM implementation for facilities in manufacturing can be considered as a hierarchical process involving the following principal stages:

- Determining families of parts based on part design and process similarities after then assigning part families to work cells (part classification approaches) or assigning parts to work cells directly (CF approaches),
- Selecting the type of cell layout,
- Laying out machines and auxiliary facilities in cells.

In order to introduce CM, it is necessary first to identify parts and machine types to be considered in the cellular configuration. This process differs with respect to whether cells are created by rearranging existing equipment on the factory floor or whether new equipment is acquired for the cells. Cells using existing equipment are typically manned and operators have major responsibilities for setup, processing, materials handling, and inspection. Cells may be designed to operate with completely new equipment often incorporating various forms of flexible automation (Selim, Askin & Vakharia, 1998).

Figure 2.4 shows the applicability of CM approach in terms of volume and variety of products. CM is a manufacturing system that can produce medium-volume/medium-variety part types more economically than other types of manufacturing systems. If volumes are very large, pure item flow lines are preferred; if volumes are small and part types are varied to the point of only slight similarities between jobs, there is less to be gained by CM.

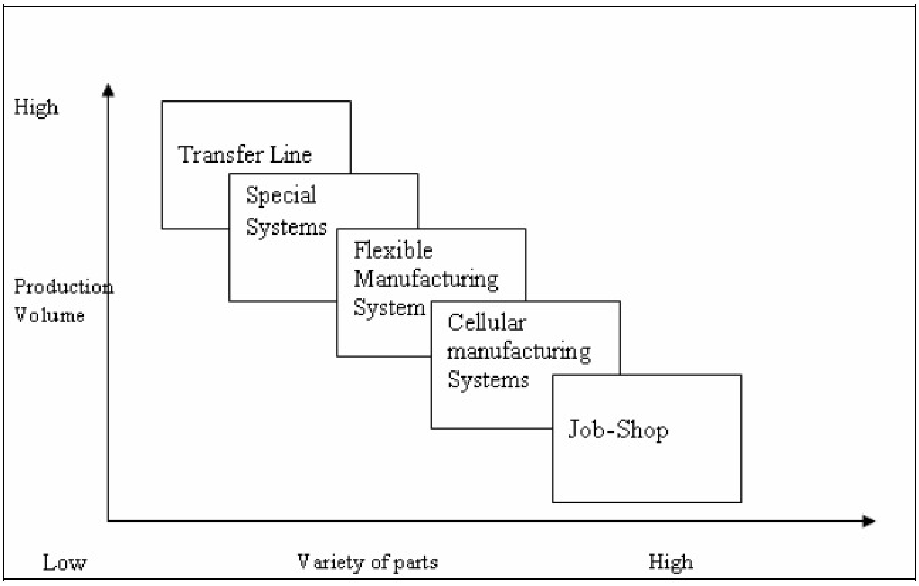

Figure 2.4 Applicability of CM (Eski, 2007).

CM provides an excellent production infrastructure that facilitates the incorporation of basic elements for successful implementation of modern manufacturing technologies, such as Just-in-Time manufacturing (JIT), Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Flexible Manufacturing Systems (FMS), Computer Integrated Manufacturing (CIM), etc (Soleymanpour, Vrat & Shankar, 2002). CM is considered as a prerequisite for JIT manufacturing (Singh & Rajamani, 1996). JIT requires manufacturing systems to have little or zero setup time, small lot sizes, and low inventory. Obviously, CM is well-suited for such requirements (Mungwattana, 2000). In addition to JIT, Total Quality Management (TQM) are greatly aided in their application in manufacturing cells, since the cells represent sociological units conductive to teamwork (Aljaber, 1999).

In conclusion, CM is a manufacturing strategy to global competition by reducing manufacturing costs, improving quality and by reducing the delivery lead times of products in a high variety, low demand environment. Hence CM has become popular among manufacturers in the last several decades (Eski, 2007). The survey by Wemmerlov & Johnson (1997) affirms that the greatest reported benefits from CM appear along the dimension of time (manufacturing lead time and customer response time). Thus, CM represents a logical choice for firms whose strategy is time-based competitive manufacturing (Stalk & Hout, 1990). During the five-year period ending in 1989, the estimated number of manufacturing cells operating in the U.S. has increased from 525 to over 8,000 and the trend continues to grow (Choi, 1992). The advantages and disadvantages of CM are presented below.

2.2 The Advantages/Disadvantages of Cellular Manufacturing

2.2.1 Advantages

The advantages derived from CM in comparison with traditional manufacturing systems have been discussed in Marsh (1993), Abduelmola (2000), Altınkılınç (2000), Hayret (2000), Mungwattana (2000), Defersha (2006), Kioon (2007), Eski (2007). These benefits have been established through simulation studies, analytical studies, surveys and actual implementations. They can be summarized as follows (Mungwattana, 2000):

- Setup time is reduced; Altınkılınç (2000), Mungwattana (2000), Defersha (2006), Kioon (2007), Eski (2007). A manufacturing cell is designed to handle parts having similar shapes and relatively similar sizes. For this reason, many of the parts can employ the same or similar holding devices (fixtures). Generic fixtures for the part family can be developed so that time required for changing fixtures and tools is decreased.
- Lot sizes are reduced; Altınkılınç (2000), Mungwattana (2000), Kioon (2007), Eski (2007). Once setup times are greatly reduced in CM, small lots are possible and economical. Small lots also smooth production flow.
- Work-in-process (WIP) and finished goods inventories are reduced; Hayret (2000), Altınkılınç (2000), Mungwattana (2000), Defersha (2006), Kioon (2007), Eski (2007). With smaller lot sizes and reduced setup times, the amount of WIP can be reduced. Askin & Standridge (1993) showed that the WIP can be reduced by 50%

when the setup time is cut in half. In addition to reduced setup times and WIP inventory, finished goods inventory is reduced. Instead of make-to-stock systems with parts either being run at long, fixed intervals or random intervals, the parts can be produced either JIT in small lots or at fixed, short intervals.

- Material handling costs and time are reduced; Hayret (2000), Altınkılınç (2000), Mungwattana (2000), Defersha (2006), Kioon (2007), Eski (2007). In CM, each part is processed completely within a single cell (where possible). Thus, part travel time and distance between cells is minimal.
- A reduction in flow time is obtained; Hayret (2000), Altınkılınç (2000), Mungwattana (2000). Reduced material handling time and reduced setup time greatly reduce flow time.
- Tool requirements are reduced; Hayret (2000), Altınkılınç (2000), Mungwattana (2000). Parts produced in a cell are of similar shape, size, and composition. Thus, they often have similar tooling requirements.
- A reduction in space required; Hayret (2000), Altınkılınç (2000), Mungwattana (2000), Kioon (2007), Eski (2007). Reductions in WIP, finished goods inventories and lot sizes lead to less space required.
- Throughput times are reduced; Altınkılınç (2000), Mungwattana (2000), Defersha (2006), Kioon (2007). In a job shop, parts are transferred between machines in batches. However, in CM each part is transferred immediately to the next machine after it has been processed. Thus, the waiting time is reduced substantially.
- Product quality is improved; Hayret (2000), Altınkılınç (2000), Mungwattana (2000), Defersha (2006), Kioon (2007), Eski (2007). Since parts travel from one station to another as single units, they are completely processed in a small area. The feedback is immediate and the process can be stopped when things go wrong.
- Better overall control of operations; Hayret (2000), Altınkılınç (2000), Mungwattana (2000), Eski (2007). In a job shop, parts may have to travel through the entire shop. Scheduling and material control are complicated. In CM, the manufacturing facility is broken down into manufacturing cells and each part travels with a single cell, resulting in easier scheduling and control.

- Increased output, reduced labor cost, increased job satisfaction, morale and communication, reduced scrap losses and rework, simplified process planning are the other advantages of CM (Altınkılınç, 2000).

These advantages are investigated in different implementations with different manufacturing conditions. The benefits gained from implementing CM also have been reported. Some of these implementations, their results and savings are covered below.

The relatively large autonomy within the manufacturing cells leads to extra motivation of the workers (who are responsible for “their products”), often resulting in higher productivity and product quality. These, and other advantages, have been also discussed by Hadley (1996).

Collet & Spicer (1995), in a case analysis of a small manufacturing company, found that CMS resulted in a number of performance improvements when compared to job shops. Reductions in operating time and less work space, due to less work in process, were achieved by CM. Setup cost was also reduced.

Northern Telecom, the leading supplier of digital communications systems applied CM to the DMS-100 Switching Division and gains more than $2 million in annual cost savings from the reduction of WIP inventory (by 82%), as well as improvement in throughput (by more than 50%). In an Indian engineering Company, the number of machines employed has been reduced from 120 to 94 and the shop floor space requirement is reduced by 21% (Eski, 2007).

In another case study at PMI Food Equipment Group, Howard & Newman (1993) reported the results of moving from a job shop to a CMS. Some of the benefits included doubling of capacity for part families due to cell configuration, $25,000 in labor saving from setup reductions, over $2 million decline in finished goods inventory, improved customer service and an improvement in quality of employee work life (Mungwattana, 2000).

Levasseur, Helms & Zink (1995) studied a case implementation of the CMS in Steward, Inc. The results were overwhelmingly in favor of the CMS. Every criteria in the case analysis showed dramatic improvement. These criteria included WIP, lead time, late orders, scrap, labor cost and manufacturing space. Table 2.1 summarizes the benefits gained from implementing CM.

Table 2.1 Benefits of CM after the first two months of operation in (Levasseur, Helms & Zink, 1995).

Criteria	**Job Shop**	**CMS**	**Resulting Improvement**
Work in process	$590,000	$116,336	$473,664 (80%)
Finished goods	$880,000	$353,167	$526,833 (60%)
Refractory supplies	$8,333/month	0	$8,333 (100%)
Lead time	14 days	2 days	12 days (86%)
Late orders	100	4	96%
Scraps	22%	14%	8%
Direct labor	198	145	53 employees (27%)
Mfg. Space (sq. ft.)	45,000	20,000	25,000 sq. ft. (56%)

Wemmerlov & Hyer (1989) reported the cost savings obtained by utilizing CM from a survey study of 32 U.S. firms. These 32 firms produced a wide variety of product lines such as machinery and machine tools, agricultural and construction equipment, hospital and medical equipment, defense products, piece parts and components, and engines. Table 2.2 shows the reported benefits from CM.

Wemmerlov & Johnson (1997) conducted another similar survey in implementation experiences and performance improvements of CM at 46 user plants. In the survey, products manufactured in these 46 plants are electrical/electronic products and components, fluid handling and flow control devices, machinery and machine tools, heating and cooling products and components, tools, engines, and bearings. Note that the surveyed firms in this publication are not the same firms in the previous survey by Wemmerlov & Hyer (1997). Table 2.3 displays the reported performance improvements.

Table 2.2 Reported benefits from CM in (Wemmerlov & Hyer, 1989).

Types of Benefit	Number of Responses	Average % Improvement	Minimum % Improvement	Maximum % Improvement
Reduction in throughput time	25	45.6	5.0	90.0
Reduction in WIP inventory	23	41.4	8.0	90.0
Reduction in material handling	26	39.3	10.0	83.0
Improvement of operator job satisfaction	16	34.4	15.0	50.0
Reduction in number of fixtures for cell parts	9	33.1	10.0	85.0
Reduction in setup time	23	32.0	2.0	95.0
Reduction in space needed	9	31.0	1.0	85.0
Improvement of part quality	26	29.6	5.0	90.0
Reduced in finished good inventory	14	29.2	10.0	75.0
Reduction in labor cost	15	26.2	5.0	75.0
Increase in utilization of equipment in the cells	6	23.3	10.0	40.0
Reduction in peaces of equipment required to manufacture cell parts	10	19.5	1.0	50.0

Table 2.3 Reported performance improvements in (Wemmerlov & Hyer, 1997).

Performance Measure	Number of Responses	Average % Improvement	Minimum % Improvement	Maximum % Improvement
Reduction of move distance/time	37	61.3	15.0	99.0
Reduction in throughput time	40	61.2	12.5	99.5
Reduction of response time to orders	37	50.1	0.0	93.2
Reduction in WIP inventory	40	48.2	10.0	99.7
Reduction in setup times	33	44.2	0.0	96.6
Reduction in finished goods inventory	38	39.3	0.0	100.0
Improvement in part/product quality	39	28.4	0.0	62.5
Reduction in unit costs	38	16.0	0.0	60.0

Hyer collected data on 20 U.S. firms in 1984. A detailed questionnaire was employed to gather information on the costs and benefits of CM. A large majority of the respondents reported that the actual benefits from implementing CM met or exceeded their expectations. Specific savings generally occurred in reductions of lead times, throughput times, queuing times, setup times, work in process, labor costs, material handling costs, and in easier process plan preparation (Mungwattana, 2000).

Studies show that cells are now adopted by between 43 and 53 percent of firms in the USA and the UK (Johnson & Wemmerlov, 2004). In plants with more than 100 employees this share increases to 73 percent for all firms (Hyer & Wemmerlov, 2002). The presented advantages make CM a preferred manufacturing strategy.

2.2.2 Disadvantages

The advantages of CM are presented in previous section. However, CM have lots of advantages in implementation, there are also disadvantages in CMS, such as the relatively costly duplication of machines (Foulds & Wilson, 2002). According to Hayret (2000), the disadvantages of CM are;

- Implementation costs. There are some implementation costs that must be dealed with when forming a manufacturing system as a CMS. The system must be arranged according to CMS rules.
- Rate of change in product range and mix. As mentioned before, the CMS is more suitable while producing medium-volume/medium-variety part types. If the rate of change in product range and mix is high, then the changes in the system will effect the production.
- Diffuculties with out-of-cell operations. Sometimes, parts can be transported between the cells in CMS. These movements effect the production efficiency and cause some costs.
- Coexistence with non-cellular systems. Sometimes, the CMS can be implemented together with other types of manufacturing systems. This coexistence effects the production efficiency and cause some costs.

2.3 Cell Formation Problem : Grouping Machines and Parts

The implemantation of CMS begins with configuring the CF. CF is the most important step of the CMS. It is a tool for designing CMSs using the similarities between parts and machines to have part families and machine groups. The process of determining the part families and machine groups are referred to as the CF problem.

At the highest level, methods for part family/machine CF can be classified as design oriented or production oriented. Design oriented approaches group parts into families based on similar design features, whereas production oriented techniques aggregate parts requiring similar processing (Joines, 1996).

There are three basic CF strategies (Dobado, Lozano, Bueno & Larraneta, 2002):

- Some approaches group parts and machines simultaneously,
- Some others first form cells and then assigns parts,
- A third strategy is to form first part families and then assign machines.

The CF problems can be classified into binary, nonbinary (fuzzy) and comprehensive grouping problems according to their inputs. Binary problems consist of inputs with the values of 0 and 1. Nonbinary (fuzzy) problems consist of inputs such as operation sequences, processing times, work loads or demands/volumes of parts with the values between 0 and 1. Comprehensive grouping problems consist of inputs such as operation sequences, processing times, work loads, costs, images of parts, machine capacities or demands/volumes of parts with the real values. Using nonbinary PMIM provides processing more data from life which leads to get results more close to reality compared to binary PMIM. However this appears to be the main disadvantage of nonbinary PMIM when it is compared with comprehensive inputs, because nonbinary PMIM does not include data such as costs, constraints, times etc. Using binary PMIM provides reaching the CF results more quickly than the nonbinary PMIM. Using nonbinary PMIM provides reaching the CF results more quickly than the comprehensive inputs. Before giving a decision about using the type of inputs, conditions and data on designing process of CMS should be evaluated. The three types of cell groupings (binary, nonbinary (fuzzy) and comprehensive) according to input types are covered below in detail.

- **Binary Inputs :**

For binary grouping problems, the processing requirements of parts on machines can be represented in the form of a matrix (a_{ij}) called the Binary PMIM (shown in Figure 2.5). The matrix (a_{ij}) has has m rows representing machines and n columns representing parts. The element a_{ij} is 1 if part j requires an operation to be performed on machine i; otherwise a_{ij} is zero.

		MACHINES								
		1	2	3	4	5	6	7	8	9
PARTS	1	0	1	0	0	0	0	0	1	0
	2	0	0	1	0	0	1	1	0	0
	3	1	0	0	0	1	0	0	0	0
	4	0	0	1	0	0	1	1	0	0
	5	0	1	0	1	0	0	0	1	0
	6	1	0	0	0	1	0	0	0	0
	7	1	0	0	0	1	1	0	0	1
	8	0	1	0	1	0	0	0	1	0
	9	0	0	1	0	0	1	0	0	0

Figure 2.5 Binary PMIM.

After using several methods, the Binary PMIM can be transform to final CF matrix shown below (Figure 2.6).

		MACHINES								
		1	5	9	3	6	7	2	4	8
PARTS	3	1	1	0	0	0	0	0	0	0
	6	1	1	0	0	0	0	0	0	0
	7	1	1	1	0	1	0	0	0	0
	2	0	0	0	1	1	1	0	0	0
	4	0	0	0	1	1	1	0	0	0
	9	0	0	0	1	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	1
	5	0	0	0	0	0	0	1	1	1
	8	0	0	0	0	0	0	1	1	1

Figure 2.6 Final CF of Binary PMIM.

The existence of exceptional elements (1's outside of the diagonal block) and voids (0's inside of the diagonal block) are the major sources degrading efficiency of CMS (Won & Currie, 2007). An exceptional part can be also called an exceptional element or a bottleneck part (Mungwattana, 2000). In general, many authors seek to identify part families and machine cells, considering the trade-off between exceptional elements and voids so that the resulting block diagonal solution has minimum exceptional elements and voids, which mean minimum inter-cell part moves and maximum within-cell machine utilisation. However, the Part Machine Grouping approaches based on the conventional binary PMIM have the following unrealistic assumptions (Won & Currie, 2007):

- The operation sequences of parts including multiple visits to the same machine are not considered.
- Each part-type is assumed to make identical demands on each machine type it uses.

- **Nonbinary (Fuzzy) Inputs :**

Nonbinary inputs are the forms of Fuzzy Logic (FL) in the implementation of CMS. If the values of the PMIM is between 0 and 1 then the matrix has nonbinary (fuzzy) type inputs. The processing requirements of parts on machines can be represented in the form of a matrix (a_{ij}) called the Nonbinary PMIM (shown in Figure 2.7).

		MACHINES				
		1	2	3	4	5
PARTS	1	0	0.75	0	1	0.25
	2	1	0	1	0	0
	3	1	0	0.67	0	0.33
	4	0	1	0	1	0
	5	1	0.43	0	0	0.57

Figure 2.7 Nonbinary PMIM.

After using several methods, the Nonbinary PMIM can be transform to final CF matrix shown below (Figure 2.8). There are also exceptional elements and voids in this example.

		MACHINES				
		1	3	5	2	4
PARTS	2	1	1	0	0	0
	3	1	0.67	0.33	0	0
	5	1	0	0.57	0.43	0
	1	0	0	0.25	0.75	1
	4	0	0	0	1	1

Figure 2.8 Final CF of Nonbinary PMIM.

Nonbinary (fuzzy) problems consist of inputs such as operation sequences, processing time of each part, work load on each machine or demand/volume of each part with the values between 0 and 1. In the literature, processing time and work load values were fuzzy (between [0,1]), but demand/volume values were not between [0,1].

So, the demand/volume values are transformed into fuzzy values. The transformation process is explained in two examples. In the first exampla just the transformation process is covered. In the second example, the transformation process is widen with the machine dublication. The first example is explained below and is sourced by Won & Currie (2007).

With the following equation nonbinary values of each element of a part vector are calculated:

$$b_{ij} = \sum_{r \in R_{ij}} f_{ijr} d_i$$

where d_i is the production volume for part i, R_{ij} is the set of operation sequence number along which part i visits machine j, and

$$f_{ijr} = \begin{cases} 1 \text{ if the } r_{th} \text{ operation of part i on machine j is the first or last operation} \\ 2 \text{ if the } r_{th} \text{ operation of part i on machine j is the intermediate operation} \\ 0 \text{ otherwise.} \end{cases}$$

Table 2.4 shows the data of operation sequences and production volumes of five parts to be manufactured on five machines for the first example.

Table 2.4 The operation sequences and production volumes for the parts for the first example (Won & Currie, 2007).

Part number	Operation sequence	Production volume
1	2-4-2-4-5	20
2	1-3	10
3	1-3-1-5	50
4	4-2-4	40
5	2-1-5-1-2-1-5-1	30

Figure 2.9 shows the initial matrix obtained by applying equation mentioned above to the data given in Table 2.4.

VOLUME BASED		MACHINES				
		1	2	3	4	5
PARTS	1	0	60	0	80	20
	2	10	0	10	0	0
	3	150	0	100	0	50
	4	0	80	0	80	0
	5	210	90	0	0	120

Figure 2.9 Initial matrix for the first example (Won & Currie, 2007).

A simple sheme for normalisation of input paterns used by Won & Currie (2007) to transform the volume based values to fuzzy values. The input normalisation scheme is straightforward since each element b_{ij} of input pattern i is normalised with its maximum value in pattern i as follows:

$$\frac{b_{ij}}{\max(b_{ij} \mid j = 1,\ldots,n)}$$

When applying the above normalisation scheme to the input vector [0, 60, 0, 80, 20], the normalised input vector is found as [0, 0.75, 0, 1, 0.25]. Figure 2.10 shows the input matrix obtained by applying equation mentioned above to the data given in Figure 2.9. This matrix is now ready to use as a nonbinary input to CF methods.

VOLUME BASED		MACHINES					PARTS				
		1	2	3	4	5	1	2	3	4	5
PARTS	1	0	0.75	0	1	0.25	1	0	0	0	0
	2	1	0	1	0	0	0	1	0	0	0
	3	1	0	0.67	0	0.33	0	0	1	0	0
	4	0	1	0	1	0	0	0	0	1	0
	5	1	0.43	0	0	0.57	0	0	0	0	1
MACHINES	1	1	0	0	0	0	0	1	1	0	1
	2	0	1	0	0	0	0.75	0	0	1	0.43
	3	0	0	1	0	0	0	1	0.67	0	0
	4	0	0	0	1	0	1	0	0	1	0
	5	0	0	0	0	1	0.25	0	0.33	0	0.57

Figure 2.10 Input matrix for the first example (Won & Currie, 2007).

The second example is also volume based and has real values. In the article by Won & Currie, (2007) where the second example is taken from, machine 1 and 4 are duplicated. In the transformation, machine 14 and 15 columns are added to the input matrix. To provide the same volume/demand proportions of the parts with the original example, the values in the machine 1 column are divided by 2 and transferred to the machine 1 column in the input matrix. The same values are used in the machine 14 column in the input matrix as well. With the same process, machine 4 column values are divided by 2 and transferred to the machine 4 column in the input matrix. The same values are used in the machine 15 column in the input matrix. Figure 2.11 and Figure 2.12 interpret the process explained above.

VOLUME BASED		MACHINES							
		1	…	…	4	…	…	…	13
PARTS	1	0			0				
	2	310			0				
	3	0			0				
	4	0			0				
	5	180			0				
	6	0			240				
	7	0			0				
	8	2200			2200				
	9	430			860				
	10	280			560				
	11	0			0				
	12	0			0				
	13	90			0				

Figure 2.11 Initial matrix for the second example (Won & Currie, 2007).

VOLUME BASED		MACHINES									
		1	…	…	4	…	…	…	13	14	15
PARTS	1	0			0					0	0
	2	155			0					155	0
	3	0			0					0	0
	4	0			0					0	0
	5	90			0					90	0
	6	0			120					0	120
	7	0			0					0	0
	8	1100			1100					1100	1100
	9	215			430					215	430
	10	140			280					140	280
	11	0			0					0	0
	12	0			0					0	0
	13	45			0					45	0

Figure 2.12 Input matrix for the second example (Won & Currie, 2007).

The same normalisation process of first example is also applied for the second example. The second example is used in application chapter (numbered five) as a nonbinary problem set (numbered three (Won & Currie, 2007)). The input matrix is presented with the whole values in the next chapter.

- **Comprehensive Inputs :**

Comprehensive grouping problems consist of inputs such as operation sequences, processing times, work loads, costs, images of parts, machine capacities or demands/volumes of parts with the real values. The comprehensive grouping problem can be represented as a mathematical model in some examples. For an example of comprehensive inputs, the production volumes/demands of parts on machines can be represented in the form of a matrix (a_{ij}) shown in Figure 2.13.

VOLUME BASED		MACHINES				
		1	2	3	4	5
PARTS	1	0	60	0	80	20
	2	10	0	10	0	0
	3	150	0	100	0	50
	4	0	80	0	80	0
	5	210	90	0	0	120

Figure 2.13 Volumes of parts.

After using several methods, the volume based matrix can be transform to final CF matrix shown below (Figure 2.14). There are also out-of-cell volumes and voids in this example.

VOLUME BASED		MACHINES				
		1	3	5	2	4
PARTS	2	10	10	0	0	0
	3	150	100	50	0	0
	5	210	0	120	90	0
	1	0	0	20	60	80
	4	0	0	0	80	80

Figure 2.14 Final CF.

For comprehensive problem definitions, in the design of CMSs, design objective(s) must be specified. Minimizing intercell moves, distances, costs and the number of exceptional parts (parts that need more than one cell for processing) are common design objectives Typical costs used in the design objective for comprehensive problems are as follows (Mungwattana, 2000):

- Equipment cost.
- Intercell material handling cost.
- Inventory cost.
- Machine relocation cost.
- Operating cost.
- Setup cost.

In addition to the design objectives, a number of strategic issues such as machine flexibility, cell layout, machine types, etc., need to be considered as a part of the CM design problem. Further, any cell configuration should satisfy operational goals (constraints) such as desired machine utilization, production volume, number of manufacturing cells, cell sizes, etc. The followings are typical design constraints in the design of CMSs (Mungwattana, 2000):

- **Machine capacity.** It is obvious that, in the design of CMSs, one of the basic requirements is that there should be adequate capacity to process all the parts.
- **Cell size.** The size of a cell, as measured by the number of machines in the cell, needs to be controlled for several reasons. First, available space might impose limits on the number of machines in a cell. If a cell is run by operators, the size of the cell should not be so large that it hinders visible control of the cell. Ranges of cell sizes can be specified instead of a single value of cell size. This would allow more exibility in the design process.
- **Number of cells.** In practice, the number of cells would be set by organizational parameters such as the size of worker teams, span of supervisory authority, and group dynamics (Askin, Selim & Vakharia, 1997). Given a range of cell sizes, the number of cells are determined and the resultant solutions can be compared.
- **Utilization levels.** Two levels of machine utilization are normally used. Maximum utilization is specified to ensure that machines are not overloaded. Minimum utilization for a new machine ensures that it is economically justifiable to include the new machine in a cell.

2.4 Cell Formation Methods

In the previous section, three types of inputs of CF problem are explained in detail. CF methods are used to find out the best cell configuration using these types of inputs. In the last three decades, over 200 research papers and practical reports have been published in the field of CM, seeking effective methods for designing CMSs. Reviews of existing CM literature can be found in Selim, Askin & Vakharia (1998), Yin & Yasuda (2006). According to those reviews, the existing CM design methods in the CMSs can be classified into the following categories: Part coding analysis, cluster techniques, similarity coefficiency, graph partitioning, mathematical programming, heuristic search, and AI-based approaches (Mungwattana, 2000):

- Part Coding Analysis (PCA) uses a coding system to assign numerical weights to part characteristic and identifies part families using some classification scheme. It also provides a basis for the development of a data retrieval system for computer integrated manufacturing. In a classification and code system, parts are sorted by parameters such as geometric shape, dimension, type of material, shape of raw material and required accuracy. Each part is assigned a numerical and/or alphabetical code. Each digit of this code represents a feature of a part. There are many types of classification and code systems used around the world (ChunHung, 1990).
- Array-based clustering is the most commonly used clustering technique. In array based clustering, the processing requirements of parts on machines can be represented by an incidence matrix, referred to as PMIM. Clustering analysis approaches consider only one objective, the minimization of intercell moves. In the design process of clustering techniques, only part operations and the machines for processing those operations are considered. Other product data (such as operational sequences and processing times) and production requirements (such as production rate) are not incorporated into the design process. Thus, solutions obtained may be valid in limited situations. However, they are simple to implement and solutions can be obtained in reasonable amounts of time (Mungwattana, 2000). Direct Clustering Algorithm (DCA), Rank Order Clustering (ROC) are the examples of

array-based methods. Studies of array based algorithms can be found in King (1980), King & Nakornchai (1982).

- The similarity coefficiency approach requires identification of measures of similarity between machines, tools and design features. A large number of similarity coefficients have been proposed in the literature (Yin & Yasuda (2005), Yin (2006)). These similarity measures are used to form part families and machine groups based on some methods. Related studies can be found in Gupta & Saifoddini (1990), Mosier, Yelle & Walker (1997), Sarker & Xu (1998), Ravichandran & Rao (2001), Diaz, Lozano & Eguia (2005), Yin & Yasuda (2006), Oliveira, Ribeiro & Seok (2008).
- Graph partitioning approaches treat the machines and/or parts as nodes and the processing of parts as arcs connecting these nodes, studies are Askin & Chiu (1990), Rajagopalan & Barta (1975), Selim (2000). These models aim at obtaining disconnected subgraphs from a machine-machine or machine-part graph to identify manufacturing cells and allocate parts to cells.
- Mathematical programming approaches are widely employed in the design of CMSs, since they are capable of incorporating certain design requirements in the design procedure. They can be further classified into four categories based upon the type of formation: Linear Programming (LP), Linear and Quadratic Integer Programming (LQP), Dynamic Programming (DP), and Goal Programming (GP). Researches can be found in Chen (1998), Mansouri, Moattar Husseini & Newman (2000), Ravichandran & Rao (2001), Albadawia, Bashirb & Chen (2005), Defersha & Chen (2006), Mukattash & Al-Tahat (2006), Mahdavi, Javadi, Fallah-Alipour & Slomp (2007), Dasa, Lashkaria & Sengupta (2007), Kioon, Bulgak & Bektas (2009).
- Heuristic search approaches, such as simulated annealing; Abdelmola & Taboun (1999), Asokan, Prabhakaran & Satheesh Kumar (2001), Xambre & Vilarinho (2003), Ozturk, Ozturk & Islier (2006), Safaei, Mehrabad & Ameli (2008), Moghaddam, Vahed, Ghodratnama & Siadat (2009), genetic algorithms; Hsu & Su (1998), De Lit, Falkenauer & Delchambre (2000), Plaquin & Pierreval (2000), Asokan, Prabhakaran & Satheesh Kumar (2001), Onwubolu & Mutingi (2001), Meents (2001), Zolfaghari & Liang (2003), Goncalves & Resende (2004), Rogers

& Kulkarni (2005), Jeon & Leep (2006), Chan, Lau, Chan & Choy (2006), Boulif & Atif (2006), Car & Mikac (2006), Vosniakos, Tsifakis & Benardos (2006), Ozturk, Ozturk & Islier (2006), Wu, Chu, Wang & Yan (2007), Moghaddam, Aryanezhad, Safaei, Vasei & Azaron (2007), Sharif, El-Kilany & Helaly (2008), Mahdavi, Paydar, Solimanpur & Heidarzade (2009), Tariq, Hussain & Ghafoor (2009), and tabu search; Aljaber, Baek & Chen (1997), Diaz, Lozano, Racero & Guerrero (2001), Chen, Wu & Chen (2002), Cao & Chen (2004), Schaller (2005), Ozturk, Ozturk & Islier (2006), Nguema & Dao (2009) have been introduced in designing CMSs as alternatives to mathematical programming approaches when computational time is prohibitive and/or linear objectives cannot be formulated.

- AI-based approaches, such as expert systems; Basu, Hyer & Shtub (1995) and ANNs; Rao & Gu (1993), Liao (1994), Malakooti & Yang (1994), Venugopal & Narendran (1994), Chen & Cheng (1995), Rao & Gu, (1995), Liao, Chen, Chen & Coates (1996), Kamal & Burke (1996), Kusiak & Lee (1996), Chu (1997), Lee, Yamakawa & Lee (1997), Zolfaghari & Liang (1997), Christodoulou & Gaganis (1998), Liang & Zolfaghari (1999), Onwubolu (1999), Suresh, Slomp & Kaparthi, (1999), Kuo, Chi & Teng (2001), Lozano, Canca, Guerrero & Garcia (2001), Mahdavi, Kaushal & Chandra (2001), Rao, Rao, Srinivas & Krishna (2001), Chen, Wu & Chen (2002), Dobado, Lozano, Bueno & Larraneta (2002), Guerrero, Lozano, Smith, Canca & Kwok (2002), Soleymanpour, Vrat & Shankar (2002), Willow (2002), Park & Suresh (2003), Ampazis & Minis (2004), Peker & Kara (2004), Tateyama & Kawata (2004), Ozturk, Ozturk & Islier (2006), Mehrabad & Safaei, (2007), Won & Currie (2007), Yang & Yang (2008), Nguema & Dao (2009) have been employed for designing CMSs because of their attractiveness in terms of computational time and ability to capture and employ design knowledge. Both heuristic search and AI-based approaches are relatively new in this area.

Each design approach has its advantages and limitations. Some are simple to implement and to obtain solutions. Some capture the design problem more accurately by considering a number of objectives and constraints, but could require a substantial amount of time to obtain solutions (Mungwattana, 2000). Mathematical programming methods, heuristics and AI-based approaches are being used more often than the other methods recently.

Among the available design approaches, mathematical programming can capture the reality of the design problem better than others, since product data and production requirements can be incorporated. Product data includes processing times and costs, operational sequences, etc. Production requirements include product mix and demand in each period, available resources, machine cost, material handling cost, etc (Mungwattana, 2000). A major drawback of mathematical programming approaches is computational time required for large problems. Obtaining optimal solutions from mathematical programming approaches can be infeasible due to the combinatorial complexity of the CM design problem (Selim, Askin & Vakharia, 1998).

Heuristic approaches have been used as alternatives to obtain reasonably good solutions within acceptable amount of times. Heuristics can be classified into two categories. The first category is the problem-specific heuristic. This type of heuristic only works for one problem; it cannot be used to solve a different one. For instance, a specific heuristic developed to solve a traveling salesman problem is unlikely to be applied to solve the general assignment problem. The second category is the metaheuristics which are more general and can be used for different types of problems. Such heuristics include genetic algorithms, simulated annealing, tabu search, etc. With some adjustment, they can be used for a wide range of problems (Mungwattana, 2000).

2.5 Performance Measures of Cell Groupings

Performance measures are the various quantitative measures used for measuring the group efficiency of CF solutions. These measures and their effectiveness have an important role to find out the best cell configuration. A review study of group efficiency measures in CM is made by Sarker & Mondal (1999). The most used efficiency measures collected from literature are listed below:

- Grouping efficiency (Chandrasekharan & Rajagopalan (1986a, b)),
- Modified group efficiency (Kandiller (1994)),

- Efficiency measure pertaining to the inner-cell load (Kandiller (1994)),
- Measure of under-utilization of an individual machine (Kandiller (1994)),
- Grouping efficacy (Kumar & Chandrasekharan (1990)),
- Grouping index (Nair & Narendran (1996)),
- Grouping measure (Miltenburg & Zhang (1991)),
- Quality index (Seifoddini & Djassemi (1996)),
- Grouping Capability Index (Hsu (1990)),
- Clustering measure (Miltenburg & Zhang (1991)),
- Global efficiency (Harhalakis, Nagi & Proth (1990)),
- Group technology efficiency (Harhalakis, Nagi & Proth (1990)),
- Proportion of exceptional elements (Rogers & Shafer (1995)),
- Machine utilization (Chandrasekharan & Rajagopalan (1986b)),
- Number of exceptional elements,
- Sum of exceptional elements.

In this thesis, two types of ANNs (SOM and CNN) are used to solve CF problem. The SOM and CNN within a new methodology are used for grouping the binary and nonbinary (fuzzy) problem sets simultaneously. So in the next chapter, AI, the connection between human brain and ANNs, history of ANNs, the advantages/disanvantages of ANNs, ANNs applications, architecture of ANNs, learning types of ANNs, types of ANNs, and a detailed literature survey on CF with ANNs are covered.

CHAPTER THREE
ARTIFICIAL NEURAL NETWORKS APPLICATION IN CELLULAR MANUFACTURING

3.1 Artificial Intelligence (AI)

There are two complementary views of AI one as an engineering discipline concerned with the creation of intelligent machines, the other as an empirical science concerned with the computational modeling of human intelligence. When the field was young, these two views were seldom distinguished. Since then, a substantial divide has opened up, with the former view dominating modern AI and the latter view characterizing much of modern cognitive science (Jordan & Russell, 1999). According to Nilsson (1971) the goal of work in AI is build machines that perform tasks normally requiring human intelligence.

The phrase AI, which was coined by John McCarthy, three decades ago evades a concise and formal definition to date (Bender, 1996). The phrase “AI” thus can be defined as the simulation of human intelligence on a machine, so as to make the machine efficient to identify and use the right piece of “Knowledge” at a given step of solving a problem. A system capable of planning and executing the right task at the right time is generally called “rational” (Russel & Norvig, 1995).

A common question then naturally arises: Does rational thinking and acting include all possible characteristics of an intelligent system? If so, how does it represent behavioral intelligence such as machine learning, perception and planning? A little thinking, however, reveals that a system that can reason well must be a successful planner, as planning in many circumstances is part of a reasoning process. Further, a system can act rationally only after acquiring adequate knowledge from the real world. So, perception that stands for building up of knowledge from real world information is a prerequisite feature for rational actions. One step further thinking envisages that a machine without learning capability cannot possess perception. The rational action of an agent (actor), thus, calls for possession of all the elementary characteristics of intelligence. Relating AI with the computational models capable of thinking and acting rationally, therefore, has a pragmatic significance (Konar, 2000).

3.2 Human Brain and Neural Networks

The human nervous system consists of small cellular units, called neurons. These neurons when connected in tandem form nerve fiber. A biological neural net is a distributed collection of these nerve fibers. A neuron receives electrical signals from its neighboring neurons, processes those signals and generates signals for other neighboring neurons attached to it (Konar, 2000).

In human beings, the neural complex is embodied by the brain, the spinal cord, and a massively interconnected web of neurons. The neural complex has the function of processing a vast amount of information, brought to it in the form of electrical impulses gathered from the senses (perceptive responses) such as smell, sight, hearing and touch. This processing enables the neural complex to make decisions based on available information. The neural complex is made up of about 1010 individual cells known as neurons and about 1014 interconnections between them. The interconnections are known as synapses and can range in number between 1000 to 10000. Each biological neuron has one or more outputs called axons and many inputs called dendrites (Figure 3.1). The interconnections between the neurons can mediate excitatory and inhibitory effects. An excitatory connection induces the neuron to fire, while an inhibitory connection tells the neuron not to fire. The conglomeration of inputs to a neuron determine whether the neuron should fire or not (Neelakanta, 1999).

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Basically, a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result (Anderson & McNeill, 1992).

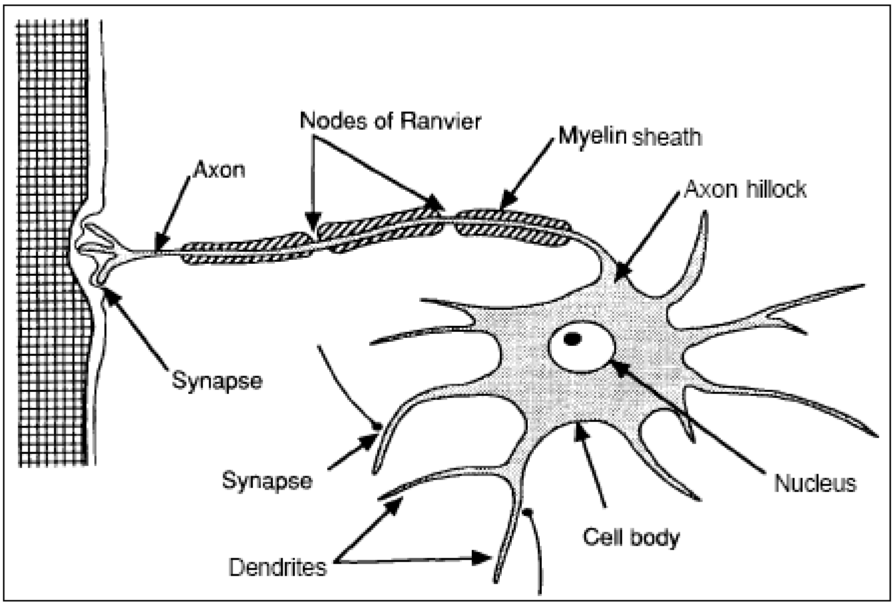

Figure 3.1 The major structures of a typical nerve cell (Freeman & Skapura, 1991).

Nerve cells come in different shapes, sizes, connections, and excitabilities. Therefore, the impression of uniformity of character which is often given for the cells is a vast oversimplification in almost all cases. However, certain properties such as excitability, development of an action potential, and synaptic linkage are considered as general characteristics of all nerve cells, and mathematical models of neurons are constructed based on these general features (Neelakanta & DeGroff, 1994).

3.3 Artificial Neural Networks (ANNs)

ANNs were a genuine improvement over the AI approach because their architecture is based, though very loosely, on real nervous systems. Instead of programming computers, NN researchers, also known as connectionists, were interested in learning what kinds of behaviors could be exhibited by hooking a bunch of neurons together. Brains are made of neurons; therefore, the brain is a NN (Hawkins & Blakeslee, 2004).

An ANN is a computational structure inspired by the study of biological neural processing (Rao, 1995). By a longer definition; an ANN is a biologically inspired computational model which consists of processing elements (called neurons) and connections between them with coefficients (weights) bound to the connections, which constitute the neuronal structure, and training and recall algorithms attached to the structure. ANNs are called connectionist models because of the main role of the connections in them. The connection weights are the "memory" of the system (Kasabov, 1998).

The neuron is the basic building block of the ANN. A neuron is a communication conduit that both accepts input and produces output. The neuron receives its input either from other neurons or the user program. Similarly the neuron sends its output to other neurons or the user program (Heaton, 2005).

The desirable information-processing characteristics are manifested in the brain such as learning, generalisation and error tolerance, and they are captured and mimicked in ANNs (Krishnamoorthy & Rajeev, 1996).

3.4 History of Artificial Neural Networks

McCulloch & Pitts, a neurobiologist and a statistician, published a seminal paper titled "A logical calculus of ideas imminent in nervous activity" in Bulletin of Mathematical Biophysics in 1943 (Hu & Hwang, 2002). Four years later the same authors explored the network paradigms for pattern recognition using a single layer perceptron. Along with the progress, psychologists were developing models of human learning. One such model, that has proved most fruitful, was due to D. O. Hebb, who, in 1949, proposed a learning law that became the starting point for ANNs training algorithm. Augmented by many other methods, it is now well recognized by scientists as indicative of how a network of artificial neurons could exhibit learning behavior. In the 1950s and 1960s, a group of researchers combined these biological and psychological insights to produce the first ANN (McClelland, Rumelhart & The PDP Research Group, 1986). Initially implemented as electronic circuits, they were later converted into a more flexible medium of computer simulation. However, from 1960 to 1980, due to certain severe limitations on what a NN could perform, as pointed out by Minsky in 1969, NN research went into near eclipse. The discovery of training methods for a multi-layer network of the 1980s has, more than any other factor, been responsible for the recent resurgence of NN (Zilouchian & Jamshidi, 2001).

3.5 The Advantages/Disadvantages of Artificial Neural Networks

3.5.1 Advantages

ANNs have good generalization capabilities. The learning and generalization capabilities of neural nets enable it to more effectively address nonlinear, time variant problems, even under noisy conditions. Thus, ANNs-can solve many problems that are either unsolved or inefficiently solved by existing techniques, including fuzzy logic. ANNs can develop solutions to meet a pre-specified accuracy (Jain & Martin, 1998).

Their attractiveness lies in the relative simplicity with which the networks can be designed for a specific problem, along with their ability to perform nonlinear data processing (Zahner & Micheli-Tzanakou, 2000).

Since ANNs are parallel distributed processing, they have the following advantages Liu (1999), Haykin (1994):

- They are adaptive and can learn from experience. They have neurobiological analogy.
- The network can be refined at any time with the addition of new training data.
- Various model architectures can be used.
- They can compute very quickly and thus they are very suitable for real-time applications.
- They can be used for analyzing large amounts of data to determine patterns that may predict certain types of behavior.
- They can capture the complexities of the process, including nonlinearities, even if the dynamics of the process is unknown.
- They can make decisions based upon incomplete and noisy information. They have uniformity in analysing and design so that they have evidental response.

- They degrade gracefully even when parts of the structure have been destroyed. They have fault tolerance.

3.5.2 Disadvantages

The major problem with neural nets is the “Black Box” nature, or rather, the relationships of the weight changes with the input-output behavior during training and use of trained system to generate correct outputs using the weights. Our understanding of the “Black Box” is incomplete compared to a fuzzy rule based system description. From an implementation point of view, ANNs may not provide the most cost effective solution - NN implementation is typically more costly than other technologies, in particular fuzzy logic (embedded control is a good example). A software solution generally takes a long time to process and a dedicated hardware implementation is more common for fuzzy logic than neural nets, due to cost. It is difficult, if not impossible, to determine the proper size and structure of a NN to solve a given problem. Also, ANNs do not scale well. Manipulating learning parameters for learning and convergence becomes increasingly difficult. ANNs are still far away from biological neural nets, but what we know today about ANNs is sufficient to solve many problems that were previously unsolvable or inefficiently solvable at best (Jain & Martin, 1998).

Another disadvantage is that the global search space for an agent is too big to start from zero with NN techniques. Much more initial structure must typically be encoded, which is sometimes difficult to express in network terms (Steels, 1998). Also analysing them in another disadvantage. That is, ANNs are rather complex systems to analyse. The reasons can be listed as: (i) the large number of interacting elements, (ii) the non-linear character of the operation of the individual elements, (iii) the interactions between the elements are not identical, or at least regular in space, but usually different in strength for each individual pair of elements, (iv) two given neurons can operate on one another in a different way (there is not even pairwise symmetry), and (v) the interactions and firing thresholds change all the time. In response to these hurdles, two distinct strategies have largely been followed in order to simplify analysis. The first is to look at layered networks, where no interaction loops are present, so that the states of the neurons can be calculated iteratively, layer by layer. The second is to describe the system statistically at a macroscopic level of global quantities, and to forget about the microscopic details at the level of the behaviour of individual neurons (Coolen, Kühn & Sollich, 2005).

3.6 Artificial Neural Networks Applications

The overall applications in manufacturing can be classified as condition monitoring, cost estimation, fault diagnosis, parameter selection, production scheduling, manufacturing CF, quality control, and others (Kamruzzaman, Sarker & Begg, 2006). A literature survey about ANNs applications in business between 1988-1995 is made by Wong, Bodnovich & Selvi (1997). Table 3.1 summaries the ANNs applications. CF with ANNs problem is a classification problem. The classification procedure of ANNs is same for CF problem so that the procedure is explained below in detail.

Pattern recognition and classification is one of the most common ANNs applications (Lisboa, 1992). The task of classification occurs in a wide range of human activity. At its broadest, the term could cover any context in which some decision or forecast is made on the basis of currently available information, and a classification procedure is then some formal method for repeatedly making such judgments in new situations (Michie, Spiegelhalter & Taylor, 1994).

Contexts in which a classification task is fundamental include, for example, mechanical procedures for sorting letters on the basis of machine-read postcodes, assigning individuals to credit status on the basis of financial and other personal information, and the preliminary diagnosis of a patient's disease in order to select immediate treatment while awaiting definitive test results. In fact, some of the most urgent problems arising in science, industry and commerce can be regarded as classification or decision problems using complex and often very extensive data (Michie, Spiegelhalter & Taylor, 1994).

Table 3.1 Commercial NN applications (Bigus, 1996).

Application	Industry	Function
Database marketing	All	Clustering, Classification, Modeling
Customer relationship management	All	Clustering, Classification, Modeling
Fraud detection	Finance, Insurance, Health	Classification, Modeling
Optical character recognition	Finance, Retail	Classification
Handwriting recognition	Computer, Finance	Clustering, Classification
Sales forecasting, inventory control	Manufacturing, Wholesale, Retail, Distribution	Clustering, Time-Series Forecasting
Stock portfolio management	Finance	Classification, Time-Series Forecasting
Bankruptcy prediction	Finance	Modeling
Job shop sheduling	Manufacturing/Process	Constraint Satisfaction
Process control	Manufacturing/Process	Modeling
Bond rating	Finance	Classification
Mortgage underwriting	Finance	Modeling, Time-Series Forecasting
Mineral exploration	Energy	Clustering, Classification
Medical (lab) diagnosis	Health	Classification, Modeling
Power demand prediction	Utility/Manufacturing	Time-Series Forecasting
Computer virus detection	Computer	Classification
Speech recognition	Computer	Clustering, Classification
Market price estimation	Real Estate, Finance	Modeling

3.7 The Architecture of Artificial Neural Networks

In the architecture of an ANN, there are input signals (x_i), their weights (w_{ji}), output signals (y_j) and activation function (f). Figure 3.2 illustrates a typical model of a neuron. Output signal y_j is a function f of the sum of weighted input signals x_i. The activation function f can be a linear, simple threshold, sigmoidal, hyberbolic tangent or radial basis function. Instead of being deterministic, f can be a probabilistic function, in which case y_j will be a binary quantity, for example, +1 or –1. The net input to such a stochastic neuron — that is, the sum of weighted input signals x_i — will then give the probability of y_j being +1 or –1. How the interneuron connections are arranged and the nature of the connections determine the structure of a network. How the strengths of the connections are adjusted or trained to achieve a desired overall behaviour of the network is governed by its learning algorithm. ANNs can be classified according to their structures and learning algorithms (Pham & Pham, 2001).

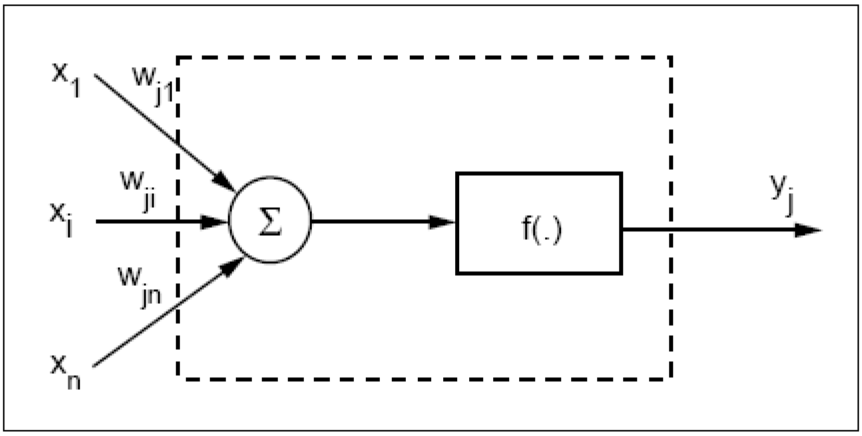

Figure 3.2 Model of a neuron. (Pham & Pham, 2001).

The individual neurons of ANNs are placed in layers. The most popular configuration is to have a three-layer network. The neurons in the layers are interconnected by weighted synapses. Figure 3.3 shows a schematic of a typical backpropagation network consisting of three layers: (1) an input layer, (2) a hidden or middle layer, and (3) an output layer. The input layer neurons receive an activation signal from the physical environment (data from the system to be modeled). These input layer neurons then sum the input signals and compute an activation based on their activation or sigmoidal functions. The resulting activation, or output, is passed to each of the neurons in the hidden layer (for a fully connected network each neuron in the input layer is connected to each neuron in the hidden layer, and each neuron in the hidden layer is connected to each neuron in the output layer). Along the way, the activation signals are multiplied by the weights associated with each of the synaptic connections along which the signals are passed. This is equivalent to taking a weighted average of the activations from the contributing neurons. These weighted activation signals serve as inputs to the hidden layer neurons. And, selecting the weights in the network such that it gives the correct answers is the whole problem (Karr, 1999).

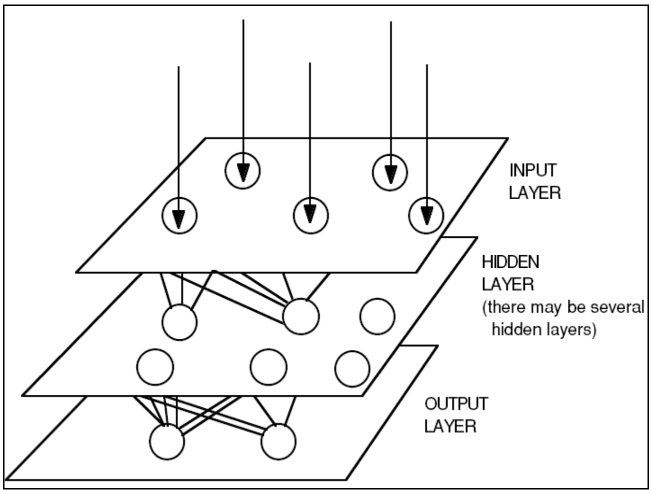

Figure 3.3 A typical ANN diagram (Anderson & McNeill, 1992).

Basically, all ANNs have a similar structure or topology as shown in Figure 3.3. In that structure some of the neurons interfaces to the real world to receive its inputs. Other neurons provide the real world with the network's outputs. This output might be the particular character that the network thinks that it has scanned or the particular image it thinks is being viewed. All the rest of the neurons are hidden from view (Anderson & McNeill, 1992).

3.8 Learning Types

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins (Anderson & McNeill, 1992). Like inductive learning programs, ANNs can capture domain knowledge from examples (Pham & Pham, 2001). ANNs differ from most computer algorithms in that they are not “programmed” rather they are “trained.” (Karr, 1999).

Learning can be defined as any change in the weights to produce some desirable state, and the learning method is a rule that adjusts the weights to the desirable state (Adeli & Park, 1998). Training the network involves moving from the training set to a set of weights which correctly classifies the training set vectors at least to within some defined error limit. In effect the network learns what the training set has to teach it. If the training set is good and the training algorithm is effective, the network should then be able to correctly classify inputs not belonging to the training set. This phenomenon is sometimes termed generalization (Pandya & Macy, 1995).

The learning situations can categorise in two distinct sorts. These are (Kröse & Smagt, 1996):

- Supervised learning or Associative learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs can be provided by an external teacher, or by the system which contains the network (self-supervised).
- Unsupervised learning or Self-organisation in which an (output) unit is trained to respond to clusters of pattern within the input. In this paradigm the system is supposed to discover statistically salient features of the input population. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli.

3.9 Artificial Neural Networks Types

Types of ANNs are specified by the net topology, node characteristics and training or learning rules. From the perspective of connection patterns, ANNs can be grouped into two categories: feedforward networks, in which graphs have no loops, and recurrent networks, where loops occur because of feedback connections (Mandic & Chambers, 2001). The ANN types according to their training or learning rules are presented in Table 3.2.

In this thesis, two types of ANNs (SOM and CNN) are used to solve CF problem. The SOM and CNN within a new methodology are used for grouping the binary and nonbinary (fuzzy) problem sets simultaneously. So, SOM and CNN are explained below in detail.

Table 3.2 ANN Types.

Supervised ANNs	Unsupervised ANNs
Perceptron	Self Organizing Maps (SOM)
Multilayered Percepton	Competitive Neural Network (CNN)
Back-propagation Network	Adaptive Resonance Theory (ART)
Adaline	Hopfield Network
Feed-forward Networks	Hamming Network
Probabilistic Neural Network	Potts Mean Field Annealing
Boltzman Machine	Interactive Activation and Competition (IAC)
High Order Neural Network (HONN)	Transiently Chaotic Neural Network (TCNN)

3.9.1 Self Organizing Map (SOM):

The term self-organizing refers to the ability to learn and organize information without being given correct answers for input patters (Fu, 1994). SOMs are unsupervised learning NNs which were introduced by T. Kohonen in 1989. The SOM is a neural network model and algorithm that implements a characteristic nonlinear projection from the high-dimensional space of sensory or other input signals onto a low-dimensional array of neurons. The SOM is able to map a structured, high-dimensional signal manifold onto a much lower-dimensional network in an olderly fashion. The mapping tends to preserve the topological relationships of the signal domains. Due to this order, the image of signal space tends to manifest clusters of input information and their relationships on the map (Kohonen & Simula, 1996).

Accordingly, the most important applications of the SOM are in the visualization of high-dimensional systems and processes and discovery of categories and abstractions from raw data. The letter operation is called the exploratory data analysis or data mining (Kohonen & Simula, 1996)

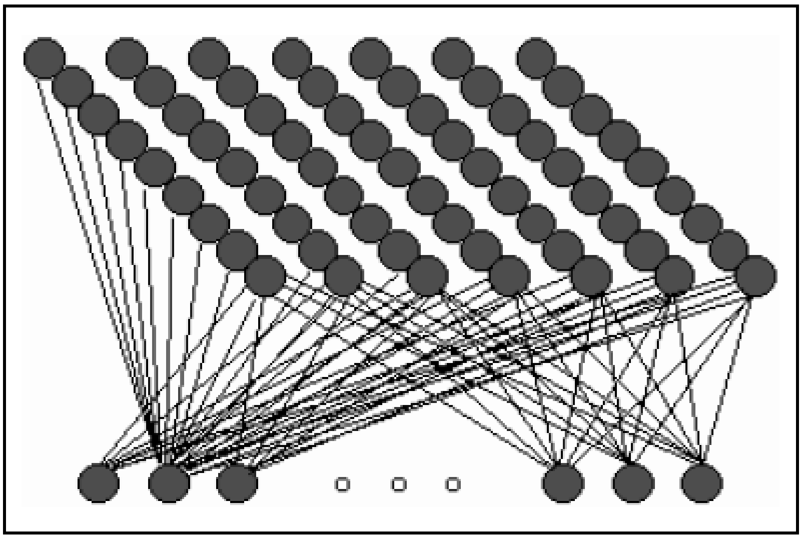

Figure 3.4 A two-dimensional SOM (Ampazis & Minis, 2004).

This type of NN is usually a two-dimensional lattice of neurons all of which have a reference model weight vector (Figure 3.4). The input layer of SOM are fully connected to a two-dimensional Kohonen layer (Onwubolu, 1999). The Kohonen layer behaves similar to the biological systems because it can compress the as a result of the SOM training algorithm, these reference vectors (otherwise known as codebook vectors) are fitted to a set of input vectors by approximating the model of the data distribution in the high-dimensional document feature space. Therefore the model vectors of neighboring units gradually learn to represent similar input data vectors (Ampazis & Minis, 2004).

The training of the SOM is achieved through a competitive learning process which consists of two steps that are applied iteratively. In the first step each input vector is compared to all the neurons' codebook vectors. The neuron "s" that has its codebook vector at the shortest geometric distance to an input vector, becomes the winner for that input vector. In the second step, each winning neuron and its surrounding neurons, i.e., neurons within a neighbourhood N_s gradually change the value of their codebook vectors in an attempt to match the input vector for which it has won. This cycle of competition and learning processes is repeated. At each cycle the size of the neighborhood of the winning neuron is decreased. The whole process terminates when each codebook vector has reached a satisfactory approximation of their corresponding input vector (Ampazis & Minis, 2004).

The SOM has some parameters that has to be determined according to its application type, such as; epoch, goal, dimension of maps, topology function, distance function, ordering phase learning rate, ordering phase steps, tuning phase learning rate and tuning phase neighborhood distance.

3.9.2 Competitive Neural Network (CNN):

In competitive learning, as the name implies, the output neurons of a neural network compete among themselves for being the one to be active (fired). Thus, whereas in a neural network based on hebbian learning several output neurons may be active simultaneously, in the case of competitive learning only a single output neuron is active at any one time. It is this feature that makes competitive leraning highly suited to discover those statistically salient features that may be used to classify a set of input patterns (Haykin, 1994). Unsupervised CNN have emerged over the past years as an important technique. CNNs implement the winner–take–all (WTA) paradigm which enforces based on lateral inhibition a localized representation of a single active neuron. When used for unsupervised learning, they yield data representations similar to those obtained based on vector quantization. They perform for each input pattern a global search for the ”winner neuron”. The proposed CNN represents a nonlinear dynamical system which includes the mutual interference between neuron and learning dynamics. It is based on the standard competitive learning law introduced by Kohonen to determine the best–matching representant among all neurons for a given input (Baese, Thummler & Theis, 2006). Recently, some articles have discussed neural systems with time–varying weights based on the competitive learning law. The winner–take–all competition between group of neurons was studied in (Xie, Hahnloser & Seung, 2000).

Competitive networks are two layer and fully connected. Usually the connects are inter-layer, not intra-layer. The weights on the connections are normally set to random positive values in the range 0 to 1 (There may be other conditions on the initial weights too). The architecture of the CNN is shown below Figure 3.5.

The || dist || box in the Figure 3.5 accepts the input vector p and the input weight matrix $IW^{1,1}$ and produces a vector having S^1 elements. The elements are the negative of the distances between the input vector and vectors $i^{IW1,1}$ formed from the rows of the input weight matrix. Compute the net input n^1 of a competitive layer by finding the negative distance between input vector p and the weight vectors and adding the biases b. If all biases are zero, the maximum net input a neuron can have is 0. This occurs when the input vector p equals that neuron's weight vector. The competitive transfer function accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the winner, the neuron associated with the most positive element of net input n^1. The winner's output is 1. If all biases are 0, then the neuron whose weight vector is closest to the input vector has the least negative net input and, therefore, wins the competition to output a 1.

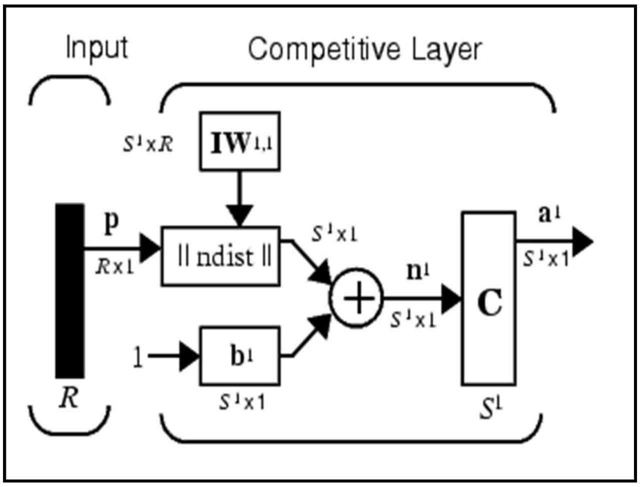

Figure 3.5 Architecture of the CNN.

The CNN has some parameters that has to be determined according to its application type, such as; epoch, goal, number of neurons, Kohonen learning rate and conscience learning rate.

3.10 Literature Survey of Cell Formation with Artificial Neural Networks

Recognizing the potential of ANNs for pattern recognition, researchers first began to apply ANNs for GT applications in the late 1980s and early 1990s. After a decade of effort, ANNs have emerged as an important and viable means for pattern classification for the application of GT and design of CMS (Suresh, 2001).

A review study for performance evaluation of CMS is made by Shambu, Suresh & Pegels (1996). Two review studies for human related issues in manufacturing cell design is made by Bidanda, Ariyawongrat, Needy, Norman & Tharmmaphornphilas (2005) and Fraser, Harris & Luong (2007a). Also a review study for virtual manufacturing cells is made by Nomden, Slomp & Suresh (2006). And a paper about CM implementation is prepared by Fraser, Harris & Luong (2007b).

Venugopal (1999)'s paper presents a state-of-the-art review synthesizing the literature on the use of soft-computing-based approaches, e.g. ANNs and fuzzy models to the CF problem. In 10 years after the study, different kinds of ANNs types and the studies are used for finding final CF. So, recent studies are listed below. The literature review is based on learning types of ANNs (supervised and unsupervised) and their input types (binary, comprehensive and nonbinary (fuzzy)).

3.10.1 Grouping with Supervised Artificial Neural Networks

3.10.1.1 Grouping with Binary Inputs

Feed-Forward Multi-layer Neural Network and Perceptron are used for CF problem with binary inputs. A neuro computing approach for integrating design and manufacturing engineering developed by Kusiak & Lee (1996). Products and components have been traditionally designed without considering constraints imposed by a manufacturing system. With the introduction of concurrent engineering, design and manufacturing engineering are viewed as an integrated area. A three layer feed-forward NN that integrates several manufacturing functions is constructed for designing a CMS. The neuro computing system proposed provides a designer with the desired features that meet the current manufacturing constraints for a design of a new component. The proposed methodology overcomes the typical limitations of ANNs such as the internal representation and training problem, and proves to be appropriate for concurrent engineering.

3.10.1.2 Grouping with Comprehensive Inputs

Feed-Forward Multi-layer Neural Network and Perceptron are used for CF problem with also comprehensive inputs. Willow (2002) introduces a linear classifier with a classical feedforward NN in forming machine cells or groups for CIM. The proposed method, through experiment, has been proven to outperform conventional mathods such as Part Family Analysis (PFA) and BLOCPLAN, among others. A single-layer perceptron, along with multi-layer feedforward network where applicable, have been employed in forming the part families. The underlying philosophy is the GT. The developed models and algorithms are illustrated with a numerical example.

Another study covers Backpropagation, High Order Neural Network and Interpolar Training Algorithm to solve CF problem with comprehensive inputs. Christodoulou & Gaganis (1998) presents an ANN approach in determining the appropriate manufacturing cell configuration that meets the required performance measures. Simulation experiments were conducted with many possible combinations of design changes to calculate cell performance measures, and thus generate training pairs for a NN. Three different static NN structures (Backpropagation NN / High Order NN / Interpolar Training Algorithm) have been trained using the above data. Comparison of NN efficiency and computational effort required is made through a case study, for every NN architecture.

3.10.2 Grouping with Unsupervised Artificial Neural Networks

3.10.2.1 Grouping with Binary Inputs

Adaptive Resonance Theory (ART) NNs offer many attractive properties for applications to engineering and manufacturing problems (Smith & Escobedo, 1994). It is most widely used NN in CF with binary inputs.

ART-1 approach's weakness is the quality of a grouping solution is highly dependent on the initial disposition of the PMIM especially in the presence of bottleneck machines and/or bottleneck parts. Chen & Cheng (1995)'s paper efforts to remove this weakness by the introdution of a set of supplementary procedures. The advantages of the supplementary procedures are demonstrated by 40 examples from the literature. The results clearly demonstrate that their algorithm is more reliable and efficient in cases of ill-structured data.

Liao, Chen, Chen & Coates (1996) compared two approaches : ART-1 and Fuzzy Rank Order Clustering (Fuzzy ROC) in CF. The study uses binary, nonbinary and comprehensive inputs. ART-1 requires the processing of batches of products in only one manufacturing cell, while Fuzzy ROC allows batches of product to be split and made simultaneously in different cells. Both design approaches consist of three stages. The first stage determines the best part routings among alternate routings to minimize the operating cost. At the second stage, a specific number of cells is obtained by using an ART1 NN-based CF module in ART-1 and a fuzzy ROC. At the third stage, production sequence is considered to find the best layout with lowest material handling cost. An example demonstrates that both approaches are effective in designing production line CMS. Fuzzy ROC gives a lower operating cost but a higher material handling cost than ART-1. Both approaches analysed and compared, since the best approach depends on the operating and material-handling costs for the application.

Tateyama & Kawata (2004)'s study aims to divide machines in a factory into any number of groups so that the machines in each group can process a similar set of parts to increase productivity. In their method, ART-1 is used to divide machines roughly. After that, adjusting algorithms are executed to satisfy specified grouping conditions (the number of groups, maximum and minimum number of machines in a group). Some experimental results show that their new algorithm is more effective than the grouping algorithm using SOM proposed by the authors in past times.

Dagli & Huggahalli (1995) adopted the ART-1 network with an application in machine-part CF, there are still several drawbacks to this approach. To address these concerns, Yang & Yang (2008) proposed a modified ART-1 neural learning algorithm. In their modified ART-1, the vigilance parameter can be simply estimated by the data so that it is more efficient and reliable than Dagli and Huggahalli's method for selecting a vigilance value. They then apply the proposed algorithm to machine-part CF in GT. Several examples are presented to illustrate its efficiency in their study. In comparison with Dagli and Huggahalli's method based on the performance measure called grouping efficiency, their modified ART-1 neural learning algorithm provides better results. Overall, the proposed algorithm is vigilance parameter-free and very efficient to use in CF with a wide variety of machine/part matrices.

The second ANN type in CF problem with binary inputs is SOM. In Malakooti & Yang (1994)'s study, they develop an unsupervised learning clustering neural network (a SOM) method for designing machine-part cells in CM. Their approach is based on the well known competitive learning algorithm. They use the generalized Euclidean distance as similarity measurement, and add a momentum term in the weight vector updating equations. The cluster in the generalized Euclidean distance. They also develop a NN clustering system which can be used to cluster a 0-1 matrix into diagonal blocks. The developed NN clustering system is independent of the initial matrix and gives clear final clustering results which specify the machines and parts in each group. They use the developed NN clustering system to solve an example, in which the PMIM is to be clustered into diagonal block structure. The computarional results are compared with those from the well-known ROC and DCA methods.

CNN is another ANN to solve CF problem. Venugopal & Narendran (1994) compared Competitive learning model, ART model and SOM. Then, Ozturk, Ozturk & Islier (2006)'s paper a simple but effective Competitive Neural Network (CNN) algorithm is applied and compared with genetic algorithms, tabu search, simulated annealing and ant systems by making use of some well known data sets from literature. As a result at 14 out of 15 cases, better results are obtained by CNN.

Hopfield ANN is the third ANN type to solve CF with binary inputs. Zolfaghari & Liang (1997) reported an Ortho-Synapse Hopfield Network (OSHN) for solving CF problem. An objective-guided search scheme is proposed to lead the search process. The OSHN structure and the objective-guided search scheme have been implemented in an algorithm. Unlike the back propagation or ART1 NNs, the proposed approach does not require training process and is not affected by the initial arrangement of the PMIM. As compared with the original Hopfield NN, the computational efficiency and the solution quality can be considerably improved due to the reduced synapses in the OSHN and the objective-guided search process. The performance of the proposed algorithm has been tested using 28 notable problems from the literature and compared favourably with the solutions obtained in the literature.

Lee, Yamakawa & Lee (1997) proposes a new machine CF method based on the Adaptive Hamming Net (AHN) which is also a NN model. To see the applicability of the method, they show some experiment results and compare the proposed method with other CF methods. From the experiments in the paper, it can be seen that the proposed method can produce good cells for the machine CF problem.

An unsupervised NN model, based upon the interactive activation and competition (IAC) learning paradigm, is proposed as a good alternative decision-support tool to solve the CF problem of CM by Chu (1997). The proposed implementation is easy to use and can simultaneously form part families and machine cells, which is very difficult or impossible to achieve by conventional methods. His computational experience shows that the procedure is fairly efficient and robust, and it can consistently produce good clustering results.

The TCNN is a recent methodology in intelligent computation that has the advantages of both the chaotic NN and the Hopfield NN. The paper of Solimanpur, Vrat & Shankar (2004) investigates the dynamics of the TCNN and studies the feasibility and robustness of final solutions of TCNN when applied to the CF problem. The paper provides insight into the feasibility and robustness of TCNN for CF problems. It also discusses how to set the initial values of the TCNN parameters in the case of well-structured and ill-structured CF problems.

3.10.2.2 Grouping with Comprehensive Inputs

ART is also used for CF problem with comprehensive inputs. Rao, Rao, Srinivas & Krishna (2001) proposes a new methodology for CF utilizing a syntactic pattern recognition approach. The selection of an appropriate cell for a new part is based on the operational information of the part. With the use of ART-2 self-organizing neural networks, the machine cells are identified. Results are presented with a numerical example.

Two studies are explained below which uses SOM to solve CF problem with comprehensive inputs. Rao & Gu (1993) presents a multi-layered SOM which can deal with practical constraints and objectives. These constraints and objectives are embedded within the network as transfer functions which help impose the practical constraints and guide the cell design process. A case study presented illustrates the efficacy of the network to deal with multiple constraints and come up with practical cell designs. The network is also capable of generating different cell configurations as specified by the user. The approach is comprehensive and can be easily expanded to include other constraints and objectives as needed.

In Guerrero, Lozano, Smith, Canca & Kwok (2002)'s study, groupings parts into families and machines into cells is done in two steps : first, part families are formed and then machines are assigned. In phase one, weighted similarity coefficients are computed and parts are clustered using a new SOM. In phase two, a linear network flow model is used to assign machines to families. To test this approach, different problems from literature have been solved. As benchmarks they have used a Maximum Spanning Tree Heuristic.

Three studies are presented below that uses Hopfield NN to solve CF problem with comprehensive inputs. Liang & Zolfaghari (1999) proposed a new NN approach

OSHNg to solve the comprehensive grouping problems. The proposed approach has been tested on 28 test problems. The results show that the OSHNg method is very efficient and its solution quality is comparable to that of a simulated annealing approach.

Mehrabad & Safaei (2007) proposes a nonlinear integer model of CF under dynamic conditions. The CF problem is a portion of a CM strategy, in which the parts and machines are clustered with the aim of minimizing the material handling cost. In most previous research the CF problem has always been under static conditions in which cells are formed for a single-period planning horizon where product mix and demand are constant. In contrast, in dynamic conditions, a multi-period planning horizon is considered, where the product mix and demand in each period is different. This occurs in seasonally or monthly production. As a result, the best cell design for one period may not be efficient for subsequent periods. To verify the presented model, different problems have been solved and results are reported. Where the CF problem belongs to NP class, the use of a novel approach is necessary. In this research, they apply a neural approach (Hopfield) based on mean filed theory for solving the proposed model. In this approach, the network weights are updated by an interaction procedure. The proposed model is solved by LINGO software and an optimum solution is obtained. Comparison of optimum and neural approach solutions shows the efficiency of the presented NN approach.

Lozano, Canca, Guerrero & Garcia (2001) investigate two sequence-based NN (Hopfield and Potts Mean Field Annealing) approaches for CF. The objective function considered is the minimization of transportation costs (including both intracellular and intercellular movements). Constraints on the minimum and maximum number of machines per cell can be imposed. The problem is formulated mathematically and shown to be equivalent to a quadratic programming integer program that uses symmetric, sequence-based similarity coefficients between each pair of machines. Of the two energy-based NN approaches investigated, namely Hopfield model and Potts Mean Field Annealing, the latter seems to give better and faster solutions, although not as good as a Tabu Search algorithm used for benchmarking.

As the last ANN type, Transiently Chaotic Neural Network (TCNN) is used. The standard version of CF problem is formulated and a "TCNN" with supplementary procedures is introduced as a powerful rival by Soleymanpour, Vrat & Shankar (2002). A simplified network is constructed. After developing the related equations the approach is tested using the proposed algorithm with 18 problems selected from literature. The results are compared with various other approaches including ART-1, Extended ART-1, OSHN, etc. The main advantages of their proposed method are: the ability to avoid the local optima trap, the ability to solve problems of different sizes with the same set of values for parameters and the less computation time. The results also indicate considerable improvement in grouping effciency through the proposed approach.

3.10.2.3 Grouping with Nonbinary (Fuzzy) Inputs

Fuzzy ART NNs is the most widely used NN in CF with nonbinary (fuzzy) inputs. Suresh & Kaparthi (1994)'s study investigates the performance of Fuzzy ART NN for grouping parts and machines, as part of the design of CMS. Fuzzy ART is compared with ART-1 NN and a modification to ART-1, along with DCA and ROC2 algorithms. A series of replicated clustering experiments were performed, and the efficiency and consistency with clusters were identified were examined, using large data sets of differing sizes and degrees of imperfection. The performance measures included the recovery ratio of bond energy and execution times. It is shown that Fuzzy ART NN results in better and more consistent identification of block diagonal structures than ART-1, a recent modification to ART-1, as well as DCA and ROC2. The execution times were found to be more than those of ART-1 and modified ART-1, but they were stil superior to traditional algorithms for large data sets.

The Fuzzy ART with Add Clustering Technique (FACT) algorithm is introduced by Kamal & Burke (1996) which is a new NN-based clustering technique. FACT can be trained to cluster machines and parts for CM under a multiple objective enviroment. The existing GT clustering techniques are mainly concerned with grouping parts and machines based on only one criterion which is the parts' processing routes. The FACT algortihm is able to consider several similarity criteria such as parts' processing routes, design requirements of parts, processing time on each machine and demand for each part. The FACT algorithm, which is based on the fuzzy ART NN, is powerful enough to solve problems of real-world sized complexity.

A pattern recognition approach based on ANN is proposed by Suresh, Slomp & Kaparthi (1999) and it is shown that the Fuzzy ART NN can be effectively utilized for this application. First, a representation scheme for operation sequences is developed, followed by an illustrative example. A more comprehensive experimental verification, based on the mixture-model approach is then performed to evaluate its performance. The experimental factors include size of the part-machine matrix, proportion of voids, proportion of exceptional elements, and vigilance threshold. It is shown that this NN is effective in identifying good clustering solutions, consistently and with relatively fast execution times.

Park & Suresh (2003) develops an experimental procedure to compare the performance of a Fuzzy ART NN, a relatively recent NN method, with the performance of traditional hierarchical clustering methods. For large, industry-type data sets, the Fuzzy ART network, with the modifications proposed here, is capable of performance levels equal or superior to those of the widely used hierarchical clustering methods. However, like other ART networks, Fuzzy ART also results in category proliferation problems, an aspect that continues to require attention for ART networks. However, low execution times and superior solution quality make Fuzzy ART a useful addition to the set of tools and techniques now available for GT and design of CMS.

An efficient methodology adopting Fuzzy ART NN is presented by Won & Currie (2007) to solve the comprehensive CF problem in CM. The Fuzzy ART/RRR-RSS (Fuzzy ART/ReaRRangement-ReaSSignment) algorithm can effectively handle the real-world manufacturing factors such as the operation sequences with multiple visits to the same machine, production volumes of parts, and multiple copies of machines. Their approach is based on the non-binary production data-based PMIM where the operation sequences with multiple visits to the same machine, production volumes of parts, and multiple identical machines are incorporated simultaneously. A new measure to evaluate the goodness of the non-binary block diagonal solution is proposed and compared with conventional performance measures. The comparison result shows that their performance measure has more powerful discriminating capability than conventional ones. The Fuzzy ART/RRR-RSS algorithm adopts two phase approach to find the proper block diagonal solution in which all the parts and machines are assigned to their most preferred part families and machine cells for minimisation of inter-cell part moves and maximisation of within-cell machine utilisation. Phase 1 (clustering phase) attempts to find part families and machines cells quickly with Fuzzy ART NN algorithm which is implemented with an ancillary procedure to enhance the block diagonal solution by rearranging the order of input presentation. Phase 2 (reassignment phase) seeks to find the best proper block diagonal solution by reassigning exceptional parts and machines and duplicating multiple identical machines to cells with the purpose of minimising inter-cell part moves and maximising within-cell machine utilisation. To show the robustness and recoverability of the Fuzzy ART/RRR-RSS algorithm to large size data sets, a modified procedure of replicated clustering which starts with the near-best solution and rigorous qualifications on the number of cells and duplicated machines has been developed. Experimental results from the modified replicated clustering show that the proposed Fuzzy ART/RRR-RSS algorithm has robustness and recoverability to large-size ill-structured data sets by producing highly independent block diagonal solution close to the near-best one.

Kuo, Chi & Teng (2001) proposed Fuzzy SOM for CF. Their study is dedicated to developing a novel Fuzzy Neural Network (FNN) for clustering the parts into several families based on image captured from the vision sensor. The proposed network, which possesses the fuzzy inputs as well the fuzzy weights, integrates the SOM NN and fuzzy set theory. The model evaluation results showed that the proposed FNN can provide a more accurate decision compared to the Fuzzy c-means algorithm.

Dobado, Lozano, Bueno & Larraneta (2002)'s paper proposes the application of a Fuzzy Min-Max NN for CF in a CM environment. Once part families have been formed, a minimum cost flow model is used to form the corresponding machine cells. For simplicity, the input data are in the form of a binary PMIM, although the algorithm can work with an incidence matrix with continuous values. The application of Fuzzy Min-Max is interpreted in physical terms and compared with a related NN applied previously for CF, the Fuzzy ART network (Fuzzy ART-CC NN). Both NNs have similarities and differences that are outlined. The algorithms have been programmed and applied to a large set of problems from the literature. Fuzzy Min-Max generally outperforms Fuzzy ART, and the computational times are small and similar in both algorithms.

These studies are summarized in Table 3.3. In this thesis, SOM and CNN are chosen to group part and machines simultaneously. They are suggested with a new methodology to solve CF problems with binary and nonbinary inputs. The new methodology and the proposed performance measure are covered in the next chapter in detail.

Table 3.3 Literature studies.

NO	AUTHOR	YEAR	NN TYPE	INPUT TYPE	PERF. MEA.
1	Kusiak & Lee	1996	Feed-Forward Multi-layer NN & Perceptron	Binary	-
2	Willow	2002	Feed-Forward Multi-layer NN & Perceptron	Comprehensive	-
3	Christodouluo & Gaganis	1998	Backpropagation, High Order NN & Interpolar Training	Comprehensive	-
4	Chen & Cheng	1995	ART-1	Binary	Measure of Effectiveness, Grouping Efficiency
5	Liao, Chen, Chen & Coates	1996	ART-1	Binary, Nonbinary, Comprehensive	Cost values
6	Tateyama & Kawata	2004	ART-1	Binary	Exceptional Elements
7	Yang & Yang	2008	Modified ART-1	Binary	Grouping Efficiency
8	Malakooti & Yang	1994	SOM	Binary	Percentage of Exceptional Elements, Machine Utilization, Grouping Efficiency
9	Venugopal & Narendran	1994	CNN	Binary	Number of Exceptional Elements, Grouping Efficacy
10	Ozturk, Ozturk & Islier	2006	CNN	Binary	Ultimate Efficacy
11	Zolfaghari & Liang	1997	OSHN	Binary	Grouping Efficiency
12	Lee, Yamakawa & Lee	1997	AHN	Binary	Machine Utilization, Grouping Efficiency
13	Chu	1997	IAC	Binary	Number of Exceptional Elements, Percentage of Exceptional Elements, Grouping Efficiency
14	Solimanpur, Vrat & Shankar	2004	TCNN	Binary	Grouping Efficiency

15	Rao, Rao, Srinivas & Krishna	2001	ART-2	Comprehensive	-
16	Rao & Gu	1993	SOM	Comprehensive	-
17	Guerrero, Lozano, Smith, Canca & Kwok	2002	SOM	Comprehensive	Objective Function, CPU Time
18	Liang & Zolfaghari	1999	OSHNg	Comprehensive	Grouping Efficiency, CPU Time
19	Mehrabad & Safaei	2007	Hopfield	Comprehensive	CPU Time, Cost values
20	Lozano, Canca, Guerrero & Garcia	2001	Potts Mean Field Annealing	Comprehensive	Objective Function Ratio
21	Soleymanpour, Vrat & Shankar	2002	TCNN	Comprehensive	Grouping Efficiency, Global Efficiency, CPU Time, Group Efficieny
22	Suresh & Kaparthi	1994	Fuzzy ART	Nonbinary	Bond Energy Recovery Ratio, Coefficient of Variation, Execution Time
23	Kamal & Burke	1996	FACT	Nonbinary	Number of Shared Machines, Inter-cellular Movements, Grouping Efficiency, Controlled Cluster Seperation Measure, Fuzzy Similarity Measure
24	Suresh, Slomp & Kaparthi	1999	Fuzzy ART	Nonbinary	Proportion of Voids, Proportion of Exceptional Elements, Clustering Effectiveness
25	Park & Suresh	2003	Fuzzy ART	Nonbinary	Cohesion Measure
26	Won & Currie	2007	Fuzzy ART/RRR-RSS	Nonbinary	Grouping Efficiency, Cohesion Measure, Grouping Capability Index, Weighted Grouping Capability Index
27	Kuo, Chi & Teng	2001	Fuzzy SOM	Nonbinary	Accurate Rate
28	Dobado, Lozano, Bueno & Larraneta	2002	Fuzzy Min-Max NN	Nonbinary	Number of Voids, Number of Exceptional Elements

CHAPTER FOUR
CELL FORMATION WITH BINARY AND NONBINARY (FUZZY) INPUTS

4.1 Introduction

In the previous chapters, GT, CMS, CF problem and solving methods, ANNs, ANNs applications to solve CF problem were explained in detail. The CF problems are classified into binary, nonbinary (fuzzy) and comprehensive grouping problems according to their inputs. In the current chapter, CF with binary and nonbinary methodologies used by SOM and CNN will be explained in detail. These methodologies make it possible for SOM and CNN to group machines and parts simultaneously. SOM and CNN are chosen regarding some criteria. These criteria are;

- Learning. The selected ANNs are unsupervised, they do not need training.
- Appropriateness. The selected ANNs are appropriate for the CF problem. SOM and CNN are used for the CF problem in literature.
- Improvement. Fuzzy SOM and Fuzzy CNN are used in literature for grouping nonbinary problems for the first time.

4.2 Cell Formation Methodology with Binary Inputs

Binary problems consist of inputs with the values of 0 and 1. Binary PMIM and its usage explained in chapter two. The methodology steps for the binary CF problem which cover Binary PMIM as problem set matrix are explained below. The methodology steps of CF with binary inputs will be covered. SOM and CNN followed the same methodology for binary inputs. Binary CF problem methodology has six steps. Methodology will be explained with an example which has 9 parts and 9 machines.

Step 1. Determine the binary problem set matrix. An example of binary problem set matrix is given in Figure 4.1.

PROBLEM SET MATRIX		MACHINES								
		1	2	3	4	5	6	7	8	9
PARTS	1	0	1	0	0	0	0	0	1	0
	2	0	0	1	0	0	1	1	0	0
	3	1	0	0	0	1	0	0	0	0
	4	0	0	1	0	0	1	1	0	0
	5	0	1	0	1	0	0	0	1	0
	6	1	0	0	0	1	0	0	0	0
	7	1	0	0	0	1	1	0	0	1
	8	0	1	0	1	0	0	0	1	0
	9	0	0	1	0	0	1	0	0	0

Figure 4.1 A binary problem set matrix example.

Step 2. Transform the binary problem set matrix into a binary input matrix. This transformation provides determining the cells by grouping machines and parts simultaneously. The binary input matrix includes the reverse matrix of the binary problem set matrix and the identity matrices. The binary input matrix of the binary problem set matrix example is shown in Figure 4.2.

Step 3. Generate SOM (or CNN) for CF problem. SOM (or CNN) is used to transform the binary input matrix into an output matrix. The topology of the SOM (or CNN) for CF problem is given in Figure 4.3. According to the topology, SOM (or CNN) classifies the machines and parts into cell groupings by evaluating each column of the input matrix.

INPUT MATRIX		MACHINES									PARTS								
		1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
PARTS	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
	2	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0
	3	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	4	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
	5	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0
	6	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
	7	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0
	8	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	9	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
MACHINES	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0
	2	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0
	3	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1
	4	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0
	5	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0
	6	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1
	7	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1	0
	9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0

REVERSE MATRIX

Figure 4.2 Input matrix of the binary problem set matrix example.

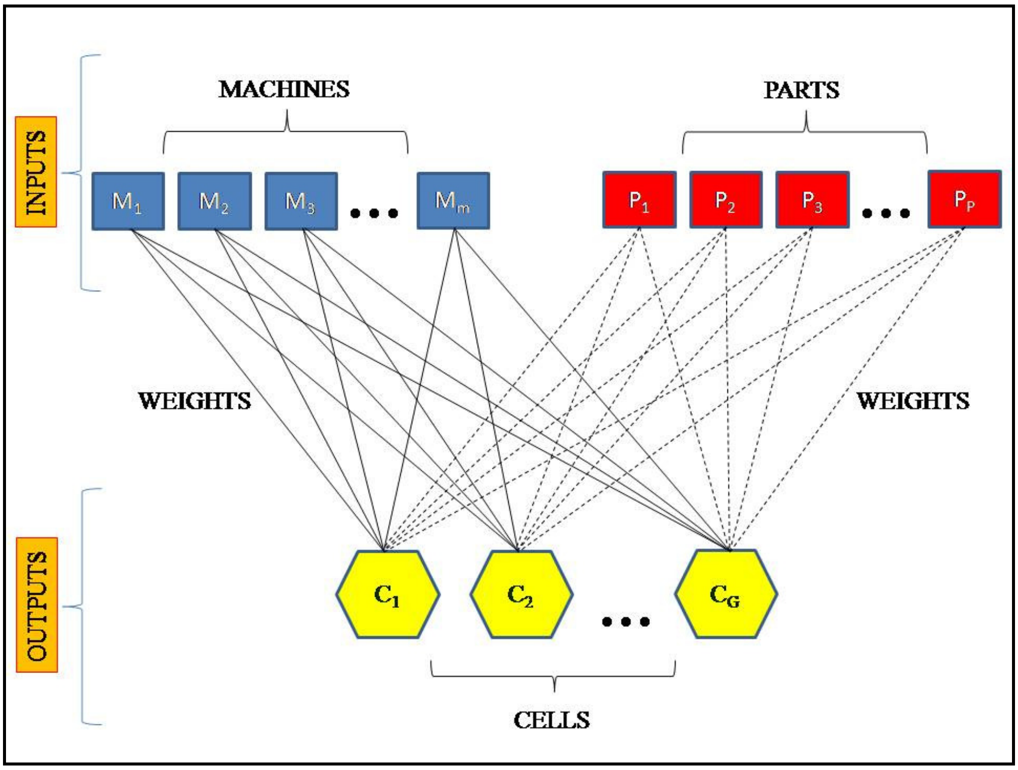

Figure 4.3 Topology of the SOM (or CNN) for CF problem.

Step 4. Use binary input matrix to train the SOM (or CNN). As mentioned in the previous chapters, SOM (or CNN) is an unsupervised network. However, it needs to adjust its weights by training to solve CF problem. The binary input matrix is also used to train the SOM (or CNN).

Step 5. Transform the binary input matrix into an output matrix by SOM (or CNN). After adjusting the weights, SOM (or CNN) is used to transform the binary input matrix into an output matrix. For a classification problem in ANNs; often the output vector from a NN is used to represent one of a set of known possible outcomes, i.e., the network acts as a classifier (Hopgood, 2001). So, cell number (G) is determined by the parameters in MATLAB. SOM (or CNN) uses each column of the binary input matrix to classify the machines and the parts into predetermined cell groupings. The output matrix of the binary problem set matrix example is presented in Figure 4.4. According to Figure 4.4, the values in the column of machine 1 are [1 0 0], which means machine 1 is classified into cell 1. The values in the column of machine 3 are [0 1 0], which means machine 3 is classified into cell 2. The values in the column of machine 2 are [0 0 1], which means machine 2 is classified into cell 3.

OUTPUT MATRIX	MACHINES									PARTS								
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
CELLS	1	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0
	0	0	1	0	0	1	1	0	0	0	1	0	1	0	0	0	0	1
	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	1	0

Figure 4.4 Output matrix of the binary problem set matrix example.

Step 6. Transform the output matrix into a result matrix. The result matrix presents the cells that provide a solution for the problem. The result matrix of the binary problem set matrix example is interpreted in Figure 4.5. According to Figure 4.5, the cells are illustrated in yellow squares: machines 1,5,9 and parts 3,6,7 are in cell 1, machines 3,6,7 and parts 2,4,9 are in cell 2, machines 2,4,8 and parts 1,5,8 are in cell 3. All the steps of the binary problems methodology are shown in Figure 4.6.

RESULT MATRIX		MACHINES								
		1	5	9	3	6	7	2	4	8
PARTS	3	1	1	0	0	0	0	0	0	0
	6	1	1	0	0	0	0	0	0	0
	7	1	1	1	0	1	0	0	0	0
	2	0	0	0	1	1	1	0	0	0
	4	0	0	0	1	1	1	0	0	0
	9	0	0	0	1	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	1
	5	0	0	0	0	0	0	1	1	1
	8	0	0	0	0	0	0	1	1	1

Figure 4.5 Result matrix of the binary problem set matrix example.

4.3 Cell Formation Methodology with Nonbinary Inputs

Nonbinary (fuzzy) problems consist of the values between 0 and 1. Nonbinary PMIM and its usage explained in chapter two. In this section, the methodology steps for the Nonbinary CF problem which cover Nonbinary PMIM as problem set matrix are explained below. Nonbinary CF problem methodology has six steps. For the nonbinary examples SOM and CNN are named as Fuzzy SOM and Fuzzy CNN. Fuzzy SOM and Fuzzy CNN followed the same methodology for nonbinary inputs. Methodology will be explained with an example which has 9 parts and 9 machines.

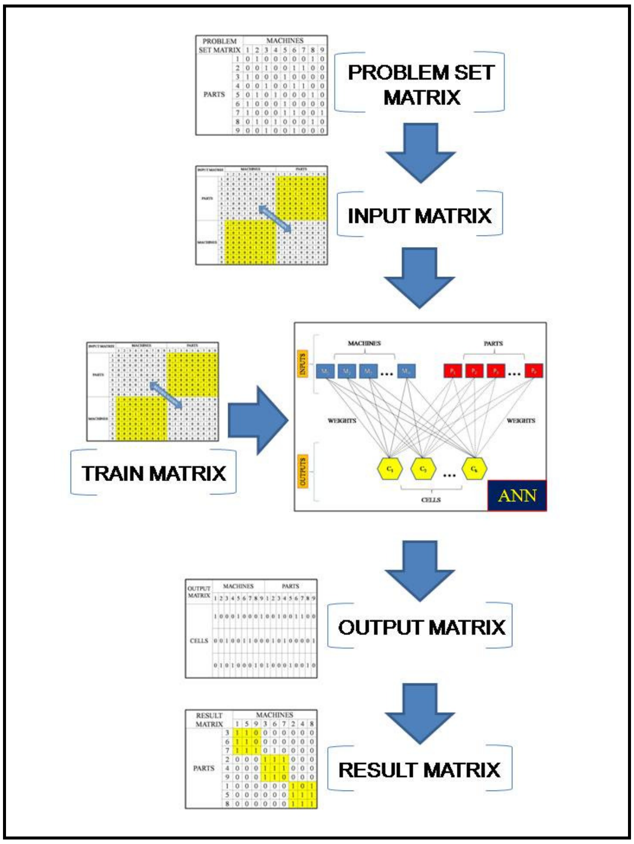

Figure 4.6 Illustrative steps of the binary problems methodology.

Step 1. Determine the nonbinary problem set matrix. An example of nonbinary problem set matrix is given in Figure 4.7.

Step 2. Transform the nonbinary problem set matrix into a nonbinary input matrix. This transformation provides determining the cells by grouping machines and parts simultaneously. The nonbinary input matrix includes the reverse matrix of the nonbinary problem set matrix and the identity matrices. The nonbinary input matrix of the nonbinary problem set matrix example is shown in Figure 4.8.

Step 3. Generate Fuzzy SOM (or Fuzzy CNN) for CF problem. Fuzzy SOM (or Fuzzy CNN) is used to transform the nonbinary input matrix into an output matrix. The topology of the Fuzzy SOM (or Fuzzy CNN) for CF problem is given in Figure 4.9. According to the topology, Fuzzy SOM (or Fuzzy CNN) classifies the machines and parts into cell groupings by evaluating each column of the nonbinary input matrix.

WORK LOAD		MACHINES								
		1	2	3	4	5	6	7	8	9
PARTS	1	1	0	0	0	0	0	1	0	0
	2	0	1	0	0	1	0	0	0	0
	3	0	0.95	0.05	0	0	0	0	0	1
	4	1	0	0	1	0	0	1	0	0
	5	0	0	0	0.1	0.5	0.4	0	1	0
	6	0	0	1	0	0	1	0	0	1
	7	0.7	0	0.3	0	0	0	1	0	0
	8	0	1	0	0	1	0	0	1	0
	9	0	0.45	0.55	0	0	1	0	0	0

Figure 4.7 A nonbinary (fuzzy) problem set matrix example.

WORK LOAD		MACHINES									PARTS								
		1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
PARTS	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
	2	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
	3	0	0.95	0.05	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
	4	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0
	5	0	0	0	0.1	0.5	0.4	0	1	0	0	0	0	0	1	0	0	0	0
	6	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0
	7	0.7	0	0.3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
	8	0	1	0	0	1	0	0	[illegible]	0	0	0	0	0	0	0	0	1	0
	9	0	0.45	0.55	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
MACHINES	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0.7	0	0
	2	0	1	0	0	0	0	0	0	0	0	1	0.95	0	0	0	0	1	0.45
	3	0	0	1	0	0	0	0	0	0	0	0	0.05	0	0	1	0.3	0	0.55
	4	0	0	0	1	0	0	0	0	0	0	0	0	1	0.1	0	0	0	0
	5	0	0	0	0	1	0	0	0	0	0	1	0	0	0.5	0	0	1	0
	6	0	0	0	0	0	1	0	0	0	0	0	0	0	0.4	1	0	0	1
	7	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0
	8	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
	9	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0

REVERSE MATRIX

Figure 4.8 Input matrix of the nonbinary problem set matrix example.

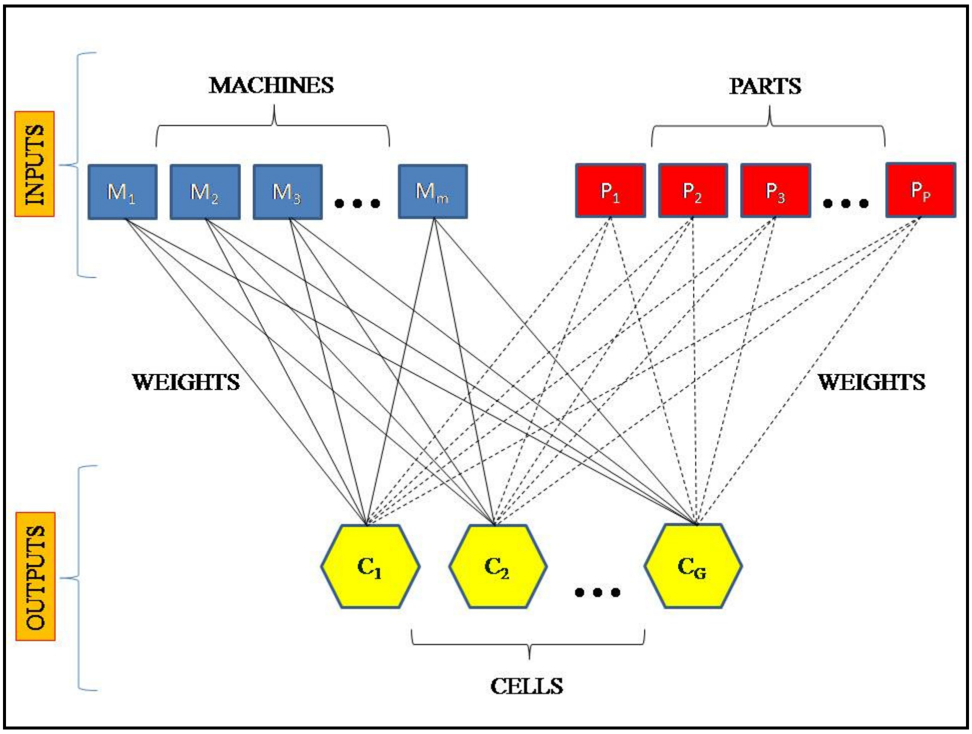

Figure 4.9 Topology of the Fuzzy SOM (or Fuzzy CNN) for CF problem.

Step 4. Use nonbinary input matrix to train the Fuzzy SOM (or Fuzzy CNN). As mentioned in the previous chapters, Fuzzy SOM (or Fuzzy CNN) is an unsupervised network. However, it needs to adjust its weights by training to solve CF problem. The nonbinary input matrix is also used to train the Fuzzy SOM (or Fuzzy CNN).

Step 5. Transform the nonbinary input matrix into an output matrix by Fuzzy SOM (or Fuzzy CNN). After adjusting the weights, Fuzzy SOM (or Fuzzy CNN) is used to transform the nonbinary input matrix into an output matrix. Cell number (G) is determined by the parameters in MATLAB. Fuzzy SOM (or Fuzzy CNN) uses each column of the nonbinary input matrix to classify the machines and the parts into predetermined cell groupings. The output matrix of the nonbinary problem set matrix example is presented in Figure 4.10. According to Figure 4.10, the values in the column of machine 2 are [1 0 0], which means machine 2 is classified into cell 1. The values in the column of machine 3 are [0 1 0], which means machine 3 is classified into cell 2. The values in the column of machine 1 are [0 0 1], which means machine 1 is classified into cell 3.

OUTPUT MATRIX	MACHINES									PARTS								
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
CELLS	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0

Figure 4.10 Output matrix of the nonbinary problem set matrix example.

Step 6. Transform the output matrix into a result matrix. The result matrix presents the cells that provide a solution for the problem. The result matrix of the nonbinary problem set matrix example is interpreted in Figure 4.11. According to Figure 4.11, the cells are illustrated in yellow squares: machines 2,5,8 and parts 2,5,8 are in cell 1, machines 3,6,9 and parts 3,6,9 are in cell 2, machines 1,4,7 and parts 1,4,7 are in cell 3. All the steps of the nonbinary problems methodology are shown in Figure 4.12.

WORK LOAD		MACHINES								
		2	5	8	3	6	9	1	4	7
PARTS	2	1	1	0	0	0	0	0	0	0
	5	0	0.5	1	0	0.4	0	0	0.1	0
	8	1	1	1	0	0	0	0	0	0
	3	0.95	0	0	0.05	0	1	0	0	0
	6	0	0	0	1	1	1	0	0	0
	9	0.45	0	0	0.55	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	1
	4	0	0	0	0	0	0	1	1	1
	7	0	0	0	0.3	0	0	0.7	0	1

Figure 4.11 Result matrix of the nonbinary problem set matrix example.

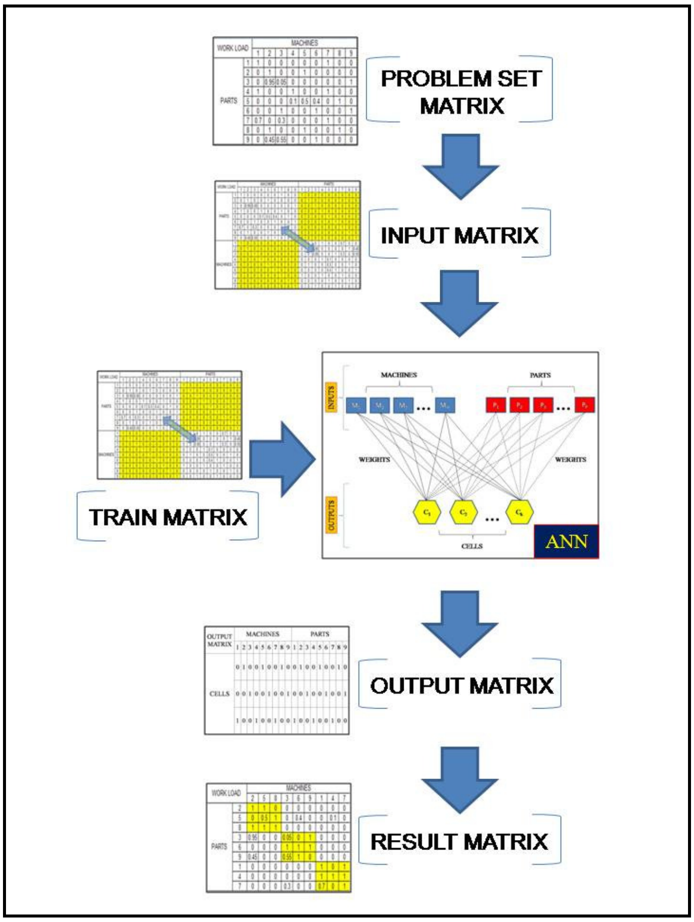

Figure 4.12 Illustrative steps of the nonbinary problems methodology.

4.4 Proposed Performance Measure

The performance measures and their important role to find out the best cell configuration are explained in chapter two. Different types of performance measures are seen in the literature review. In this study, a new performance measure is proposed. The proposed performance measure is composed by five well-known performance measures which aggregates the effectiveness of its dimensions. In the following part, these five of the performance measures will be explained. The selected performance measures are; Grouping Efficiency, Grouping Efficacy, Grouping Measure, Grouping Capability Index and Machine Utilization. The choosing process is made regarding some criteria. These criteria are;

- Frequency. The more used performance measures in the literature are chosen.

- Diversity. Performance measures evaluating more variant parameters of cell groupings such as exceptional elements, voids, etc. are chosen.
- Appropriateness. Performance measures appropriate for both binary and nonbinary input matrices are chosen.

Parameters :

m = number of machines,

p = number of parts,

o = number of 1s in the part/machine matrix, total number of operations,

e = number of 1s outside the diagonal block, number of exceptional elements,

v = number of 0s in the diagonal block, number of voids,

w = weight,

u = number of exceptional elements (e) / total number of operations (o),

q = number of voids (v) / total number of operations (o),

e_1 = number of 1s in the cells (o-e),

G = number of cells,

j_k = number of machines in the kth cell,

i_k = number of parts in the kth cell.

- **Grouping Efficiency (ή):**

Grouping efficiency (ή) is an aggregatemeasure, which takes both the number of exceptional elements and machine utilization into consideration. ή is defined by Chandrasekharan & Rajagopalan (1986) as:

$$\eta = w\left(\frac{o-e}{o-e+v}\right)+(1-w)\left(\frac{mp-o-v}{mp-o-v+e}\right)$$

The weight w is assigned to reveal the relative importance of each term, though a value of 0.5 is commonly used.

- **Grouping Efficacy (Γ):**

To overcome the low discriminating power of group efficiency and to avoid ignoring the importance of exceptional elements and other related deficiencies of ή in the measure, Kumar & Chandrasekharan (1990) introduced the concept of group efficacy (Γ) which is defined by

$$\Gamma = \frac{1-u}{1+q}$$

where q is the ratio of the number of exceptional elements to the total number of operations and u is the ratio of the number of voids in the diagonal blocks to the total number of operations. This expression has the requisite properties like non-negativity and zero to one range. Moreover, q and u are only the ratios and are not affected by the size of the matrix.

- **Grouping Measure ($\acute{\eta}_g$):**

Based on the work of Chandrasekharan & Rajagopalan (1986a, b), Miltenburg & Zhang (1991) proposed this index to measure machine utilization in a cell. A higher index value indicates utilization of a higher number of machines (having fewer voids) and fewer parts require processing on machines in more than one cell (fewer bottleneck parts or exceptional elements). This grouping measure $\acute{\eta}_g$ is given by:

$$\acute{\eta}_g = \acute{\eta}_u - \acute{\eta}_m \qquad -1 \leq \acute{\eta}_g \leq 1$$

where $\acute{\eta}_u = e_1 / (e_1 + v)$ with $0 \leq \acute{\eta}_u \leq 1$ and $\acute{\eta}_m = 1 - (e_1 / e)$ with $0 \leq \acute{\eta}_m \leq 1$. $\acute{\eta}_u$ indicates the measure of usage of parts in a cell. The higher values of $\acute{\eta}_u$ indicates the higher usage of parts, i.e each part of the machine-part cell gets processed in most of the machines in a cell. $\acute{\eta}_m$ indicates the measure of the movement of the parts of a cell to other cell, i.e basically the measure of inter-cellular movement of the parts. Naturally, the smaller value of $\acute{\eta}_m$ indicates fewer inter-cellular movements of the parts which is desired to obtain a greater value of the machine-utilization index ($\acute{\eta}_g$). $\acute{\eta}_g$ would be at maximum when $\acute{\eta}_u$ is large and $\acute{\eta}_m$ is small, preferably zero.

- **Grouping Capability Index (GCI):**

In both the group efficiency and group efficacy measures, only the requirements of the machining of parts are considered. No other factors like production volume and processing times of operations are taken into consideration in those measures. Hsu (1990) considered these factors and proposed a measure called the Grouping Capability Index (GCI):

$$GCI = 1 - e / o$$

where e is the number of exceptional elements in the machine-component matrix and o is the total number of `one' entries in the PMIM. Hsu (1990) claimed that this measure is more consistent in predicting the suitability of a manufacturing system for cellular manufacturing.

- **Machine Utilization (MU):**

Machine utilization (MU) indicates the percentage of time the machines within the clusters are used in production. MU is defined by Chandrasekharan & Rajagopalan (1986) as

$$MU = \frac{e_1}{\sum_{k=1}^{G} j_k i_k}$$

where e_1 is the total number of ones within the part family-machine cells, G is the number of cells, j_k is the number of machines in the kth cell, and i_k is the number of parts in the kth cell. Generally, the higher the value, the better the machines are utilized.

As mentioned above, four of the selected performance measure values are between 0 and 1. However, η_g values are between -1 and 1. So, η_g measure is normalised to fit with the other measures. It is normalised with the following equation:

$$\eta_g(norm) = \frac{1+\eta_g}{2}$$

As mentioned before, different types of performance measures are seen in the literature review. The studies in the literature use limited number of performance measures. Each of these performance measures evaluates the cell configurations from different aspects. So, a new performance measure is proposed to evaluate the results more objectively. The proposed performance measure used in the thesis is generated by giving 20% weigth to each of the five performance measures shown in Figure 4.13. The proposed performance measure is used for both binary and nonbinary data sets.

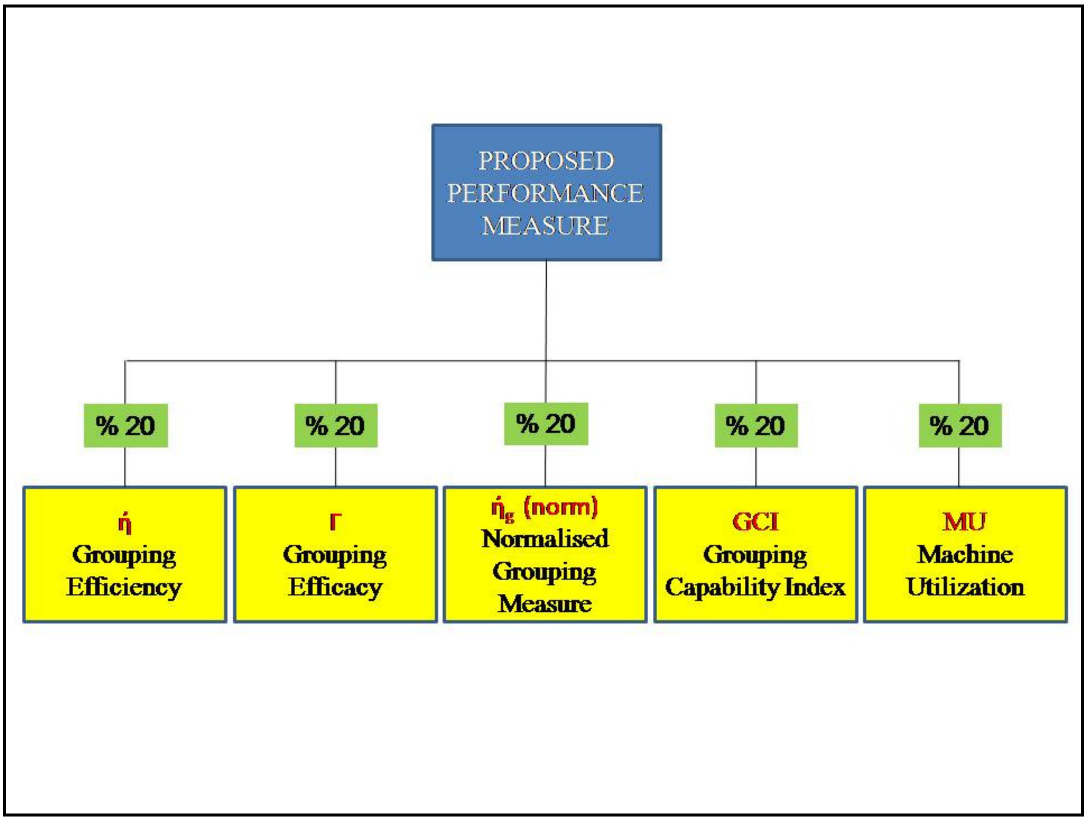

Figure 4.13 Proposed performance measure.

CHAPTER FIVE
APPLICATION OF CELL FORMATION PROBLEM WITH ANNs

5.1 Introduction

In the previous chapters GT, CMS, the CF problem and its solving methods; ANNs, ANNs applications for solving the CF problem; proposed binary and nonbinary (fuzzy) CF methodologies based on SOM and CNN were explained in detail.

In the current chapter, the proposed two types of ANNs (SOM and CNN) are applied to solve the CF problem with both binary and nonbinary (fuzzy) inputs. SOM is named Fuzzy SOM and CNN is named Fuzzy CNN when they are applied with nonbinary (fuzzy) inputs.

The main aim of the present thesis is to implement SOM and CNN within the proposed methodology in chapter four to the CF problem using binary inputs, Fuzzy SOM and Fuzzy CNN using nonbinary (fuzzy) inputs. In the application section following, the aim of the thesis will be realized in two steps. The first step is to test if the proposed methodology works by using binary inputs. Solved binary problems by different kinds of methods within different methodologies are chosen from the literature and solved again by SOM and CNN within the proposed methodology. For every problem, the results of the article's problem is taken from and the results of the proposed method are compared to determine performance of the proposed methodology. So, the proposed methodology is tested for binary inputs, then the second step is to use it for nonbinary (fuzzy) inputs. The procedure applied for the binary problems is implemented for nonbinary problems chosen from the literature. Methods of Fuzzy SOM and Fuzzy CNN are used within the proposed methodology.

Related with the aim of the thesis, in the next section, binary problem sets which are collected from the literature will be explained. The variables of SOM and CNN will be interpreted. The results and discussion for the binary problems will be presented.

In the last section, the explanations for nonbinary (fuzzy) problem sets collected from the literature will be presented. The variables of Fuzzy SOM and the variables of Fuzzy CNN will be covered. An example of MATLAB codes for a nonbinary problem will be interpreted. Finally, the results and discussion for the nonbinary problems will be presented.

5.2 Binary Cases

Binary cases in the present study are the problem sets collected from literature which have inputs that are Binary PMIMes. Binary PMIM is explained in chapter two and the methodology of CF with binary inputs is explained in chapter four. In the literature, SOM and CNN are used to solve binary cases within different methodologies. In this thesis, SOM and CNN are used to solve binary cases chosen from articles once more following the proposed methodology in chapter four. This is done to see whether the proposed methodology gets better results than the previous trials in the literature.

5.2.1 The Binary Problem Sets

15 binary problem sets are collected from 8 articles (published between 1994 and 2008) and used as inputs for SOM and CNN within the proposed methodology. The binary problem sets are presented below (in Table 5.1). Number of parts in the problems differs within a range of 5 to 35 and the number of machines differs within a range of 4 to 28. In the Table 5.1, problem number 5, 6 and 7 sourced by Malakooti & Yang (1994) have the same number of machines, parts and cells (15, 10, 3) and problem number 10, 11, 12 and 13 sourced by Yang & Yang (2008) also have the same number of machines, parts and cells (15, 15, 4). However their structure of matrices are different. The binary problem set matrices are given in Appendix A1.

Table 5.1 The binary problem sets.

NO	AUTHOR	# OF PARTS	# OF MACHINES	# OF CELLS IN ARTICLE
1	Lee, Yamakawa & Lee (1997)	5	4	2
2	Ozturk, Ozturk & Islier (2006)	7	5	2
3	Yang & Yang (2008)	9	9	2,3
4	Chen & Cheng (1995)	10	10	3
5	Malakooti & Yang (1994)	15	10	3
6	Malakooti & Yang (1994)	15	10	3
7	Malakooti & Yang (1994)	15	10	3
8	Kusiak & Lee (1996)	15	10	3
9	Onwubolu (1999)	15	10	3
10	Yang & Yang (2008)	15	15	4
11	Yang & Yang (2008)	15	15	4
12	Yang & Yang (2008)	15	15	4
13	Yang & Yang (2008)	15	15	4
14	Smith & Escobedo (1994)	13	17	3
15	Yang & Yang (2008)	35	28	6

5.2.2 SOM Solutions

5.2.2.1 SOM Variables

The binary problem set solutions by SOM are provided by using MATLAB 7.5. The variable values or functions chosen for SOM are coded in MATLAB and presented below. Being experienced on ANNs is very important to decide on the best value or function for a variable in ANN applications. For the present thesis, epoch and goal values are determined by testing the possible values or by investigating the literature for the most used epoch and goal values used in previous ANN applications. The rest of the variables coded are the default values or functions of MATLAB.

- **SOM Variables :**

Epoch = 100 (for all cell groupings)

Goal = 1e-5 (for all cell groupings)

Dimension of maps = [2 1] (for 2 cell groupings)

Dimension of maps = [3 1] (for 3 cell groupings)

Dimension of maps = [4 1] (for 4 cell groupings)

Dimension of maps = [5 1] (for 5 cell groupings)

Dimension of maps = [6 1] (for 6 cell groupings)

Topology function = hextop (for all cell groupings default function in MATLAB)

Distance function = linkdist (for all cell groupings default function in MATLAB)

Ordering phase learning rate = 0.9 (for all cell groupings default value in MATLAB)

Ordering phase steps = 1000 (for all cell groupings default value in MATLAB)

Tuning phase learning rate = 0.02 (for all cell groupings default value in MATLAB)

Tuning phase neighborhood distance = 1 (for all cell groupings default value in MATLAB)

The MATLAB codes for SOM is generated by following the proposed methodology mentioned in chapter four. The codes are presented in Appendix A2. For all of the binary problem sets, each MATLAB code is runned 10 times. The best result value among the results is selected to be compared with the other results in the articles. The best results are presented below and the matrices for the best results of SOM for binary problems are presented in Appendix A3. If the code can not classify at least one machine and one part into each cell grouping after 20 runs, the result is nominated as "no classification after 20 runs".

5.2.2.2 Problem Solutions According to Proposed Performance Measure

The best result values of SOM provided according to the proposed performance measure for the binary problem sets are presented in Table 5.2 – Table 5.6. The results show that SOM grouped parts and machines into cells simultaneously by using the proposed methodology. For each binary problem set, the result found by SOM within the proposed methodology is compared with the result/results found by the method/methods in the article the problem is taken from. All comparison results are analysed using the proposed performance measure and the five performance measures (Grouping Efficiency (GE), Grouping Efficacy (GEA), Normalised Grouping Measure (NGM), Grouping Capability Index (GCI) and Machine Utilization (MU)) creating the proposed performance measure. The comparisons are divided into five tables (Table 5.2 – Table 5.6) according to the methods (expected, CNN, ART-1/Modified ART-1, heuristics and SOM with article methodologies) used in the articles and are explained below.

The best result value for each of the binary problems (numbered 1 (Lee, Yamakawa & Lee, 1997), 5 and 6 (Malakooti & Yang, 1994), 8 (Kusiak & Lee, 1996), 14 (Smith & Escobedo, 1994)) provided by SOM are compared with the expected solution (the best solution found in the literature according to the article) and presented in Table 5.2. For all data sets, SOM found the same results with expected results according to all performance measures which means the result matrices of SOM and the expected solution are the same for each data set.

Table 5.2 Problem solutions by SOM according to proposed performance measure compared with expected solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
1	Lee. Yamakawa & Lee (1997)	5	4	9	Expected	2	0.9	0.8182	0.9	0.9	0.9	0.8836
					SOM							
5	Malakooti & Yang (1994)	15	10	25	Expected	3	0.95	0.9	0.95	1	0.9	0.94
					SOM							
6	Malakooti & Yang (1994)	15	10	25	Expected	3	0.915	0.7895	0.8827	0.8654	0.9	0.8705
					SOM							
8	Kusiak & Lee (1996)	15	10	25	Expected	3	0.92	0.8148	0.8983	0.9167	0.88	0.886
					SOM							
14	Smith & Escobedo (1994)	13	17	30	Expected	3	0.92	0.84	0.92	1	0.84	0.904
					SOM							

In Table 5.3, SOM solution and CNN solution for binary problem number 2 (Ozturk, Ozturk & Islier, 2006) are compared. For data set 2, SOM found the same result values with CNN according to the 6 performance measures.

Table 5.3 Problem solution by SOM according to proposed performance measure compared with CNN solution.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
2	Ozturk, Ozturk & Islier (2006)	7	5	12	CNN	2	0.8562	0.7368	0.8492	0.875	0.8235	0.8282
					SOM							

For the binary problem sets (numbered 3, 10, 11, 12, 13 and 15 (Yang & Yang, 2008), 4 (Chen & Cheng, 1995)) the best result values of SOM are compared with ART-1/ Modified ART-1 solutions and are presented in Table 5.4. For data set 3 (Yang & Yang, 2008), SOM found the same result values with ART-1 when the number of cells is 2 and SOM found the same result values also with Modified ART-1 when the number of cells is 3. For data set 4 (Chen & Cheng, 1995), SOM reached better result values than ART-1 according to each of the 6 performance measures. For data set 15 (Yang & Yang, 2008), SOM found better result values than Modified ART-1 according to all performance measures. For the rest of the sets, the SOM result values were the same with the article result values. Among these sets, data set 10 (Yang & Yang, 2008) is exceptional because SOM found "1" as the result value for the performance measures. This means a "perfect diagonal matrix" is found.

Table 5.4 Problem solutions by SOM according to proposed performance measure compared with ART-1/Modified ART-1 solutions.

<table>
<tr><th>PROB. NO</th><th>SOURCE</th><th># OF PARTS</th><th># OF MACH.</th><th>TOTAL #</th><th>METHOD</th><th># OF CELLS</th><th>GE</th><th>GEA</th><th>NGM</th><th>GCI</th><th>MU</th><th>PROP. PERF. MEA.</th></tr>
<tr><td rowspan="4">3</td><td rowspan="4">Yang & Yang (2008)</td><td rowspan="4">9</td><td rowspan="4">9</td><td rowspan="4">18</td><td>ART-1</td><td rowspan="2">2</td><td rowspan="2">0.8162</td><td rowspan="2">0.6591</td><td rowspan="2">0.8068</td><td rowspan="2">0.9063</td><td rowspan="2">0.7073</td><td rowspan="2">0.7791</td></tr>
<tr><td>SOM</td></tr>
<tr><td>Modified ART-1</td><td rowspan="2">3</td><td rowspan="2">0.8906</td><td rowspan="2">0.7429</td><td rowspan="2">0.8545</td><td rowspan="2">0.8125</td><td rowspan="2">0.8966</td><td rowspan="2">0.8394</td></tr>
<tr><td>SOM</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Chen & Cheng (1995)</td><td rowspan="2">10</td><td rowspan="2">10</td><td rowspan="2">20</td><td>ART-1</td><td rowspan="2">3</td><td>0.807</td><td>0.5952</td><td>0.7464</td><td>0.7576</td><td>0.7353</td><td>0.7283</td></tr>
<tr><td>SOM</td><td>0.8643</td><td>0.6923</td><td>0.8182</td><td>0.8182</td><td>0.8182</td><td>0.8022</td></tr>
<tr><td rowspan="2">10</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td rowspan="2">4</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">1</td></tr>
<tr><td>SOM</td></tr>
<tr><td rowspan="2">11</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td rowspan="2">4</td><td rowspan="2">0.9766</td><td rowspan="2">0.871</td><td rowspan="2">0.9355</td><td rowspan="2">0.871</td><td rowspan="2">1</td><td rowspan="2">0.9308</td></tr>
<tr><td>SOM</td></tr>
<tr><td rowspan="2">12</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td rowspan="2">4</td><td rowspan="2">0.9167</td><td rowspan="2">0.8333</td><td rowspan="2">0.9166</td><td rowspan="2">1</td><td rowspan="2">0.8333</td><td rowspan="2">0.9</td></tr>
<tr><td>SOM</td></tr>
<tr><td rowspan="2">13</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td rowspan="2">4</td><td rowspan="2">0.8933</td><td rowspan="2">0.7258</td><td rowspan="2">0.8412</td><td rowspan="2">0.8491</td><td rowspan="2">0.8333</td><td rowspan="2">0.8285</td></tr>
<tr><td>SOM</td></tr>
<tr><td rowspan="2">15</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">35</td><td rowspan="2">28</td><td rowspan="2">63</td><td>Modified ART-1</td><td rowspan="2">6</td><td>0.9074</td><td>0.6682</td><td>0.8072</td><td>0.7371</td><td>0.8773</td><td>0.7994</td></tr>
<tr><td>SOM</td><td>0.9105</td><td>0.6729</td><td>0.8109</td><td>0.7385</td><td>0.8834</td><td>0.8032</td></tr>
</table>

SOM solution for the binary problem set number 7 (Malakooti & Yang, 1994) is compared with heuristics (ROC and DCA) solutions and the comparison is presented in Table 5.5. For data set 7, SOM within the proposed methodology found better result values than ROC and DCA.

Table 5.5 Problem solutions by SOM according to proposed performance measure compared with heuristics solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
7	Malakooti & Yang (1994)	15	10	25	ROC	3	0.855	0.678	0.8081	0.8163	0.8	0.7914
					DCA		0.9091	0.7818	0.8775	0.8776	0.8776	0.8647
					SOM		0.915	0.8	0.889	0.898	0.88	0.8764

The best result values of proposed SOM for the problem sets 7 (Malakooti & Yang, 1994) and 9 (Onwubolu, 1999) are compared with SOM within the article methodology and the results are presented in Table 5.6. For the mentioned data sets, our SOM found the same results with SOM in the article.

Table 5.6 Problem solutions by SOM according to proposed performance measure compared with Article SOM solutions.

<table>
<tr><th>PROB. NO</th><th>SOURCE</th><th># OF PARTS</th><th># OF MACH.</th><th>TOTAL #</th><th>METHOD</th><th># OF CELLS</th><th>GE</th><th>GEA</th><th>NGM</th><th>GCI</th><th>MU</th><th>PROP. PERF. MEA.</th></tr>
<tr><td rowspan="2">7</td><td rowspan="2">Malakooti & Yang (1994)</td><td rowspan="2">15</td><td rowspan="2">10</td><td rowspan="2">25</td><td>SOM (Article)</td><td rowspan="2">3</td><td rowspan="2">0.915</td><td rowspan="2">0.8</td><td rowspan="2">0.889</td><td rowspan="2">0.898</td><td rowspan="2">0.88</td><td rowspan="2">0.8764</td></tr>
<tr><td>SOM *</td></tr>
<tr><td rowspan="2">9</td><td rowspan="2">Onwubolu (1999)</td><td rowspan="2">15</td><td rowspan="2">10</td><td rowspan="2">25</td><td>SOM (Article)</td><td rowspan="2">3</td><td rowspan="2">0.925</td><td rowspan="2">0.8302</td><td rowspan="2">0.9081</td><td rowspan="2">0.9362</td><td rowspan="2">0.88</td><td rowspan="2">0.8959</td></tr>
<tr><td>SOM *</td></tr>
<tr><td colspan="13">* The results of SOM with proposed methodology</td></tr>
</table>

Totally, 15 binary problem sets are solved by our SOM. Because some of the problems are solved with more than one method in the articles, 18 different solutions are compared according to 6 performance measures with the proposed SOM solutions. The comparisons reveal that SOM found the same results with the method used in the article for 14 of the 18 solutions. For the rest 4 solutions, our SOM found better results according to each of the 6 performance measures than the method used in the article which are ART-1 (in problem 4), ROC and DCA (in problem 7) and Modified ART-1 (in problem 15). SOM never found worse results according to any of the performance measures in any of the 18 problem solutions. As a result, SOM within the proposed methodology can be used as an effective ANN method to solve the CF problem according to 6 performance measures.

5.2.2.3 Problem Solutions According to Article Performance Measures

In this section, each binary problem is solved by SOM according to the performance measure in the article the problem is taken from. The best result values of SOM within the proposed methodology are compared with the result/results found by the method/methods used in the article. Performance measures used in the articles that the 15 binary problems are chosen from are GE, GEA, Exceptional Elements and MU. For 5 of the data sets (numbered 1 (Lee, Yamakawa & Lee, 1997), 5 and 6 (Malakooti & Yang, 1994), 8 (Kusiak & Lee, 1996) and 14 (Smith & Escobedo, 1994)) there is no article performance measure. The comparisons are divided into four tables (Table 5.7 – Table 5.10) according to the performance measures in the articles (GE, GEA, Exceptional Elements and MU) and are presented below.

For each binary problem set, the best result values of SOM within the proposed methodology provided according to GE are presented in Table 5.7. According to the table, for 4 binary problem sets (numbered 10, 11, 12 and 13 (Yang & Yang, 2008)) SOM found the same results with the methods in the article. For data set 3 (Yang & Yang, 2008), SOM found the same result values with ART-1 when the number of cells is 2 and SOM found the same result values also with Modified ART-1 when the number of cells is 3. For data set 7 (Malakooti & Yang, 1994), the SOM within the proposed methodology and SOM within the methodology in the article found the same result which is better than the results found by ROC and DCA. For data set 15 (Yang & Yang, 2008), SOM found a better result value than Modified ART-1 according to GE.

The best result value of SOM and the result value found by CNN in the article (Ozturk, Ozturk & Islier, 2006) for data set 2 is compared according to GEA and presented in Table 5.8. According to the table, the result found by SOM is the same with the result by CNN.

Table 5.7 Article problem solutions by SOM according to GE.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
3	Yang & Yang (2008)	9	9	18	ART-1	2	GE	0.8162
					SOM *			
					Modified ART-1	3		0.8906
					SOM *			
7	Malakooti & Yang (1994)	15	10	25	ROC	3	GE	0.855
					DCA			0.9091
					SOM (Article)			0.9150
					SOM *			0.9150
10	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	1
					SOM *			
11	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	0.9766
					SOM *			
12	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	0.9167
					SOM *			
13	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	0.8933
					SOM *			
15	Yang & Yang (2008)	35	28	63	Modified ART-1	6	GE	0.9074
					SOM *			0.9105
* The results of SOM with proposed methodology								

Table 5.8 Article problem solutions by SOM according to GEA.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
2	Ozturk, Ozturk & Islier (2006)	7	5	12	CNN	2	GEA	0.7368
					SOM			

In Table 5.9, the best result values of SOM according to the Exceptional Elements are presented in comparison with the results of the articles. As mentioned before, the lower number of exceptional elements means that a better result is found. For data set 4 (Chen & Cheng, 1995), SOM found a better result than ART-1. For data set 7 (Malakooti & Yang, 1994), SOM within the proposed methodology and SOM within the methodology in the article found the same result which is better than the results found by ROC and DCA. For data set 9 (Onwubolu, 1999), proposed SOM found the same result with SOM within the methodology in the article.

Table 5.9 Article problem solutions by SOM according to Exceptional Elements.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
4	Chen & Cheng (1995)	10	10	20	ART-1	3	Exceptional Elements	8
					SOM *			6
7	Malakooti & Yang (1994)	15	10	25	ROC	3	Percentage of Exceptional Elements	0.183
					DCA			0.1224
					SOM (Article)			0.102
					SOM *			0.102
9	Onwubolu (1999)	15	10	25	SOM (Article)	3	Exceptional Elements	3
					SOM *			
* The results of SOM with proposed methodology								

In Table 5.10, solutions by the methods in the article (Malakooti & Yang, 1994) and SOM solution for data set 7 are compared. Our SOM and SOM within the methodology in the article found the same result which is better than the results found by ROC and DCA.

Table 5.10 Article problem solutions by SOM according to MU.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
7	Malakooti & Yang (1994)	15	10	25	ROC	3	MU	0.80
					DCA			0.8776
					SOM (Article)			0.88
					SOM *			0.88
* The results of SOM with proposed methodology								

In this section, 10 binary data sets among 15 data sets are solved by our SOM according to the performance measures in the articles because there is no performance meausure in 5 binary problem sets. For the 10 data sets, 19 different solutions from the articles are compared according to performance measures in articles. In the article by Malakooti & Yang (1994) problem set 7 is solved by 3 different methods (ROC, DCA and SOM) according to 3 of the performance measures (GE, Percentage of Exceptional Elements and MU). So, 12 of the 19 solutions are from the article by Malakooti & Yang (1994).

The comparisons show that SOM found the same results with the methods used in the articles for 11 of the 19 solutions. For the rest 8 solutions, SOM found better results than the articles. ART-1 according to Exceptional Elements (in problem 4), ROC and DCA according to GE, Percentage of Exceptional Elements and MU (in problem 7), and Modified ART-1 according to GE (in problem 15) are the methods and performance measures used in the problems that SOM found better results than the article. SOM never found worse results according to any of the performance measures in any of the 19 problem solutions. As a result, SOM within the proposed methodology is an effective ANN method to solve the CF problem according to 4 performance measures in the articles as it was according to 6 performance measures mentioned in the previous section.

5.2.2.4 Problem Solutions for Different Numbers of Cells

In the literature, some of the methods and methodologies for solving the CF problem use predefined number of cells to reach results. However, some other methods and methodologies determine number of cells by their algorithms. So that, for the binary problem sets chosen for the present study, different numbers of cells are tested to see whether there is a better cell configuration than the article has found. Number of exceptional elements, number of voids, total number of operations, number of "1"s in the cells, number of machines and parts in the cells are the basic parameters used in performance measures. Related to these parameters and the structure of input matrices, different numbers of cells can effect the solution. That is because, using more numbers of cells means there is more chance for an exceptional element to be included in a cell grouping which can lead to a better performance result. However there can be some cases where the success of the performance result is not related to the number of cells used. In these cases, the critical point is to search for the optimal number of cells that fits to the structure of the input matrix the best.

SOM is coded to group the machines and the parts into 2, 3, 4, 5 and/or 6 cells. Cell numbers are chosen to be coherent with the part and machine numbers. For data set 1 (Lee, Yamakawa & Lee, 1997) and 2 (Ozturk, Ozturk & Islier, 2006), no different numbers of cells were tried because the number of parts and machines are not suitable for more number of cells to be formed.

The best result values of SOM provided according to the proposed performance measure with different numbers of cells are presented in Table 5.11 – Table 5.14. The comparisons are divided into the mentioned four tables according to the methods (expected, ART-1/Modified ART-1, heuristics and SOM with article methodologies) used in the articles and are explained below. According to the tables, our SOM could not find a better result in any of the trials redefining the numbers of cells. This result underlines the fact that optimal number of cells that fits to the structure of the input matrix finds the best performance results.

Table 5.11 Problem solutions by SOM with different numbers of cells according to proposed performance measure compared with expected solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
5	Malakooti & Yang (1994)	15	10	25	Expected	-	0.94	-	-	-
					SOM	0.7375		NC	NC	-
6	Malakooti & Yang (1994)	15	10	25	Expected	-	0.8705	-	-	-
					SOM	0.753		0.8371	NC	-
8	Kusiak & Lee (1996)	15	10	25	Expected	-	0.886	-	-	-
					SOM	0.7556		0.825	NC	-
14	Smith & Escobedo (1994)	13	17	30	Expected	-	0.904	-	-	-
					SOM	0.7203		NC	NC	-

NC : No classification after 20 runs

5.2.3 CNN Solutions

5.2.3.1 CNN Variables

Like it was done in SOM implementations, the binary problem set solutions by CNN are provided by using MATLAB 7.5. All the procedure applied while solving the binary problems by SOM were also applied for CNN. So, the variable values or functions chosen for CNN are the same with the ones chosen for SOM and they are presented below.

Table 5.12 Problem solutions by SOM with different numbers of cells according to proposed performance measure compared with ART-1/Modified ART-1 solutions.

<table>
<tr><th rowspan="2">PROB. NO</th><th rowspan="2">SOURCE</th><th rowspan="2"># OF PARTS</th><th rowspan="2"># OF MACH.</th><th rowspan="2">TOTAL #</th><th rowspan="2">METHOD</th><th colspan="5">PROPOSED PERFORMANCE MEASURE</th></tr>
<tr><th>2 CELLS</th><th>3 CELLS</th><th>4 CELLS</th><th>5 CELLS</th><th>6 CELLS</th></tr>
<tr><td rowspan="3">3</td><td rowspan="3">Yang & Yang (2008)</td><td rowspan="3">9</td><td rowspan="3">9</td><td rowspan="3">18</td><td>ART-1</td><td>0.7791</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Modified ART-1</td><td>-</td><td rowspan="2">0.8394</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.7791</td><td>0.8226</td><td>-</td><td>-</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Chen & Cheng (1995)</td><td rowspan="2">10</td><td rowspan="2">10</td><td rowspan="2">20</td><td>ART-1</td><td>-</td><td>0.7283</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.7178</td><td>0.8022</td><td>0.7629</td><td>NC</td><td>-</td></tr>
<tr><td rowspan="2">10</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td>-</td><td>-</td><td rowspan="2">1</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.6945</td><td>0.832</td><td>NC</td><td>-</td></tr>
<tr><td rowspan="2">11</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td>-</td><td>-</td><td rowspan="2">0.9308</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.6779</td><td>0.7889</td><td>NC</td><td>-</td></tr>
<tr><td rowspan="2">12</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td>-</td><td>-</td><td rowspan="2">0.9</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.6455</td><td>0.76</td><td>NC</td><td>-</td></tr>
<tr><td rowspan="2">13</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">15</td><td rowspan="2">15</td><td rowspan="2">30</td><td>Modified ART-1</td><td>-</td><td>-</td><td rowspan="2">0.8285</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.6254</td><td>0.7221</td><td>0.8143</td><td>-</td></tr>
<tr><td rowspan="2">15</td><td rowspan="2">Yang & Yang (2008)</td><td rowspan="2">35</td><td rowspan="2">28</td><td rowspan="2">63</td><td>Modified ART-1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0.7994</td></tr>
<tr><td>SOM</td><td>0.5532</td><td>0.6204</td><td>0.6442</td><td>0.7325</td><td>0.8032</td></tr>
<tr><td colspan="11">NC : No classification after 20 runs</td></tr>
</table>

Table 5.13 Problem solutions by SOM with different numbers of cells according to proposed performance measure compared with heuristics solutions.

<table>
<tr><th rowspan="2">PROB. NO</th><th rowspan="2">SOURCE</th><th rowspan="2"># OF PARTS</th><th rowspan="2"># OF MACH.</th><th rowspan="2">TOTAL #</th><th rowspan="2">METHOD</th><th colspan="5">PROPOSED PERFORMANCE MEASURE</th></tr>
<tr><th>2 CELLS</th><th>3 CELLS</th><th>4 CELLS</th><th>5 CELLS</th><th>6 CELLS</th></tr>
<tr><td rowspan="3">7</td><td rowspan="3">Malakooti & Yang (1994)</td><td rowspan="3">15</td><td rowspan="3">10</td><td rowspan="3">25</td><td>ROC</td><td>-</td><td>0.7914</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>DCA</td><td>-</td><td>0.8647</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SOM</td><td>0.763</td><td>0.8764</td><td>0.7692</td><td>NC</td><td>-</td></tr>
<tr><td colspan="11">NC : No classification after 20 runs</td></tr>
</table>

Table 5.14 Problem solutions by our SOM with different number of cells according to proposed performance measure compared with Article SOM solutions.

<table>
<tr><th rowspan="2">PROB. NO</th><th rowspan="2">SOURCE</th><th rowspan="2"># OF PARTS</th><th rowspan="2"># OF MACH.</th><th rowspan="2">TOTAL #</th><th rowspan="2">METHOD</th><th colspan="5">PROPOSED PERFORMANCE MEASURE</th></tr>
<tr><th>2 CELLS</th><th>3 CELLS</th><th>4 CELLS</th><th>5 CELLS</th><th>6 CELLS</th></tr>
<tr><td rowspan="2">7</td><td rowspan="2">Malakooti & Yang (1994)</td><td rowspan="2">15</td><td rowspan="2">10</td><td rowspan="2">25</td><td>SOM (Article)</td><td>-</td><td rowspan="2">0.8764</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SOM *</td><td>0.763</td><td>0.7692</td><td>NC</td><td>-</td></tr>
<tr><td rowspan="2">9</td><td rowspan="2">Onwubolu (1999)</td><td rowspan="2">15</td><td rowspan="2">10</td><td rowspan="2">25</td><td>SOM (Article)</td><td>-</td><td rowspan="2">0.8959</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>SOM *</td><td>0.7481</td><td>0.8338</td><td>NC</td><td>-</td></tr>
<tr><td colspan="11">NC : No classification after 20 runs
* The results of SOM with proposed methodology</td></tr>
</table>

- **CNN Variables :**

Epoch = 100 (for all cell groupings)

Goal = 1e-5 (for all cell groupings)

Dimension of maps = [2 1] (for 2 cell groupings)

Dimension of maps = [3 1] (for 3 cell groupings)

Dimension of maps = [4 1] (for 4 cell groupings)

Dimension of maps = [5 1] (for 5 cell groupings)

Dimension of maps = [6 1] (for 6 cell groupings)

Topology function = hextop (for all cell groupings default function in MATLAB)

Distance function = linkdist (for all cell groupings default function in MATLAB)

Ordering phase learning rate = 0.9 (for all cell groupings default value in MATLAB)

Ordering phase steps = 1000 (for all cell groupings default value in MATLAB)

Tuning phase learning rate = 0.02 (for all cell groupings default value in MATLAB)

Tuning phase neighborhood distance = 1 (for all cell groupings default value in MATLAB)

The MATLAB codes for CNN is generated by following the proposed methodology mentioned in chapter four. For all of the binary problem sets, each MATLAB code is runned 10 times. The best result value among the results derived after 10 runs is selected to be compared with the other results in the articles. The best results are presented below and the matrices for the best results of CNN for binary problems are presented in Appendix A4. If the code can not classify at least one machine and one part into each cell grouping after 20 runs, the result is nominated as “no classification after 20 runs”.

5.2.3.2 Problem Solutions According to Proposed Performance Measure

The best result values provided by proposed CNN according to the proposed performance measure for the binary problem sets are presented in Table 5.15 – Table 5.19. For each binary problem set, the result found by CNN within the proposed methodology is compared with the result/results found by the method/methods used in the article the problem is taken from. All the results reached by CNN or any other method used in the articles are provided according to 6 different performance measures including the proposed performance measure and its dimensions. The comparisons are divided into five tables (Table 5.15 – Table 5.19) according to the methods (expected, CNN within the article methodologies, ART-1/Modified ART-1, heuristics and SOM within the article methodologies) used in the articles and are covered below.

The best result value for each of the binary problems (numbered 1 (Lee, Yamakawa & Lee, 1997), 5 and 6 (Malakooti & Yang, 1994), 8 (Kusiak & Lee, 1996), 14 (Smith & Escobedo, 1994)) provided by CNN are compared with the expected solution (the best solution found in the literature according to the article) and presented in Table 5.15. For all data sets, CNN found the same results with the expected results according to the 6 performance measures.

Table 5.15 Problem solutions by CNN according to proposed performance measure compared with expected solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
1	Lee. Yamakawa & Lee (1997)	5	4	9	Expected	2	0.9	0.8182	0.9	0.9	0.9	0.8836
					CNN							
5	Malakooti & Yang (1994)	15	10	25	Expected	3	0.95	0.9	0.95	1	0.9	0.94
					CNN							
6	Malakooti & Yang (1994)	15	10	25	Expected	3	0.915	0.7895	0.8827	0.8654	0.9	0.8705
					CNN							
8	Kusiak & Lee (1996)	15	10	25	Expected	3	0.92	0.8148	0.8983	0.9167	0.88	0.886
					CNN							
14	Smith & Escobedo (1994)	13	17	30	Expected	3	0.92	0.84	0.92	1	0.84	0.904
					CNN							

In Table 5.16, CNN solution within the article methodology and proposed CNN solution for binary problem number 2 (Ozturk, Ozturk & Islier, 2006) are compared. For data set 2, our CNN found the same result values with CNN within the article methodology according to the 6 performance measures.

Table 5.16 Problem solution by our CNN according to proposed performance measure compared with CNN solution within the article methodology.

<table>
<tr><th>PROB. NO</th><th>SOURCE</th><th># OF PARTS</th><th># OF MACH.</th><th>TOTAL #</th><th>METHOD</th><th># OF CELLS</th><th>GE</th><th>GEA</th><th>NGM</th><th>GCI</th><th>MU</th><th>PROP. PERF. MEA.</th></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Ozturk, Ozturk & Islier (2006)</td><td rowspan="2">7</td><td rowspan="2">5</td><td rowspan="2">12</td><td>CNN (Article)</td><td rowspan="2">2</td><td rowspan="2">0.8562</td><td rowspan="2">0.7368</td><td rowspan="2">0.8492</td><td rowspan="2">0.875</td><td rowspan="2">0.8235</td><td rowspan="2">0.8282</td></tr>
<tr><td>CNN *</td></tr>
<tr><td colspan="13">* The results of CNN with proposed methodology</td></tr>
</table>

For the binary problem sets (numbered 3, 10, 11, 12, 13 and 15 (Yang & Yang, 2008), 4 (Chen & Cheng, 1995)) the best result values of CNN are compared with ART-1/ Modified ART-1 solutions and are presented in Table 5.17. For data set 15 (Yang & Yang, 2008), CNN found worse result values than Modified ART-1 according to each of the 6 performance measures. For data set 3 (Yang & Yang, 2008), CNN found the same result values with ART-1 when the number of cells is 2. When results by CNN are compared with the results by Modified ART-1 for the same data set, CNN is found to get better result values according to GE and MU, and worse result values according to GEA, NGM, GCI and proposed performance measure in the condition G=3. For data set 4 (Chen & Cheng, 1995), CNN reached better result values than ART-1 according to each of the 6 performance measures. For the rest of the sets, the result values by CNN were the same with the result values by the article methods.

CNN solution for the binary problem set number 7 (Malakooti & Yang, 1994) is compared with heuristics (ROC and DCA) solutions and the comparison is presented in Table 5.18. For data set 7, CNN within the proposed methodology found better result values than ROC and DCA according to each of the 6 performance measures.

The best result values of proposed CNN for the problem sets 7 (Malakooti & Yang, 1994) and 9 (Onwubolu, 1999) are compared with the results by SOM within the article methodology and the comparisons are presented in Table 5.19. For the mentioned data sets, our CNN found the same results with SOM in the article.

Table 5.17 Problem solutions by CNN according to proposed performance measure compared with ART-1/Modified ART-1 solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
3	Yang & Yang (2008)	9	9	18	ART-1	2	0.8162	0.6591	0.8068	0.9063	0.7073	0.7791
					CNN							
					Modified ART-1	3	0.8906	0.7429	0.8545	0.8125	0.8966	0.8394
					CNN		0.8981	0.7353	0.8536	0.7813	0.9259	0.8388
4	Chen & Cheng (1995)	10	10	20	ART-1	3	0.807	0.5952	0.7464	0.7576	0.7353	0.7283
					CNN		0.8643	0.6923	0.8182	0.8182	0.8182	0.8022
10	Yang & Yang (2008)	15	15	30	Modified ART-1	4	1	1	1	1	1	1
					CNN							
11	Yang & Yang (2008)	15	15	30	Modified ART-1	4	0.9766	0.871	0.9355	0.871	1	0.9308
					CNN							
12	Yang & Yang (2008)	15	15	30	Modified ART-1	4	0.9167	0.8333	0.9166	1	0.8333	0.9
					CNN							
13	Yang & Yang (2008)	15	15	30	Modified ART-1	4	0.8933	0.7258	0.8412	0.8491	0.8333	0.8285
					CNN							
15	Yang & Yang (2008)	35	28	63	Modified ART-1	6	0.9074	0.6682	0.8072	0.7371	0.8773	0.7994
					CNN		0.8938	0.633	0.7824	0.7077	0.8571	0.7748

Table 5.18 Problem solutions by CNN according to proposed performance measure compared with heuristics solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
7	Malakooti & Yang (1994)	15	10	25	ROC	3	0.855	0.678	0.8081	0.8163	0.8	0.7914
					DCA		0.9091	0.7818	0.8775	0.8776	0.8776	0.8647
					CNN		0.915	0.8	0.889	0.898	0.88	0.8764

Table 5.19 Problem solutions by CNN according to proposed performance measure compared with Article SOM solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
7	Malakooti & Yang (1994)	15	10	25	SOM	3	0.915	0.8	0.889	0.898	0.88	0.8764
					CNN							
9	Onwubolu (1999)	15	10	25	SOM	3	0.925	0.8302	0.9081	0.9362	0.88	0.8959
					CNN							

Finally, 15 binary problem sets are solved by CNN within the proposed methodology. Because the same problems in the SOM application are used in the CNN application, 18 different solutions are compared according to 6 performance measures with CNN solutions derived within the proposed methodology. The comparisons reveal that CNN found the same results with the methods used in the articles for 13 of the 18 solutions. For 3 solutions from 2 data sets, CNN found better results according to each of the 6 performance measures than the methods used in the articles. ART-1 (in problem 4), ROC and DCA (in problem 7) are the methods used in the 2 problems that CNN found better results than the articles. CNN found worse results than Modified ART-1 according to GEA, NGM, GCI and proposed performance measure in problem 4. CNN found worse results than Modified ART-1 also in problem 15 according to each of the 6 performance measures. As a result, CNN within the proposed methodology can be used as a good ANN method to solve the CF problem according to all of the performance measures.

5.2.3.3 Problem Solutions According to Article Performance Measures

In this section, each binary problem is solved by SOM according to the performance measure in the article the problem is taken from. The best result values of SOM within the proposed methodology are compared with the result/results found by the method/methods used in the article. Performance measures used in the articles that the 15 binary problems are chosen from are GE, GEA, Exceptional Elements and MU. For 5 of the data sets (numbered 1 (Lee, Yamakawa & Lee, 1997), 5 and 6 (Malakooti & Yang, 1994), 8 (Kusiak & Lee, 1996) and 14 (Smith & Escobedo, 1994)) there is no article performance measure. The comparisons are divided into four tables (Table 5.20 – Table 5.23) according to the performance measures in the articles (GE, GEA, Exceptional Elements and MU) and are presented below.

For each binary problem set, the best result values of proposed CNN provided according to GE are presented in Table 5.20. According to the table, for 4 binary problem sets (numbered 10, 11, 12 and 13 (Yang & Yang, 2008)) CNN found the same results with the methods in the article. For data set 3 (Yang & Yang, 2008), CNN found the same result value with ART-1 when G=2 and CNN found a better result value than Modified ART-1 when G=3. For data set 7 (Malakooti & Yang, 1994), our CNN and CNN within the methodology in the article found the same result which is better than the results found by ROC and DCA. For data set 15 (Yang & Yang, 2008), CNN found a worse result value than Modified ART-1 according to GE.

Table 5.20 Article problem solutions by CNN according to GE.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
3	Yang & Yang (2008)	9	9	18	ART-1	2	GE	0.8162
					CNN *			
					Modified ART-1	3		0.8906
					CNN *			0.8981
7	Malakooti & Yang (1994)	15	10	25	ROC	3	GE	0.855
					DCA			0.9091
					CNN (Article)			0.915
					CNN *			0.915
10	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	1
					CNN *			
11	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	0.9766
					CNN *			
12	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	0.9167
					CNN *			
13	Yang & Yang (2008)	15	15	30	Modified ART-1	4	GE	0.8933
					CNN *			
15	Yang & Yang (2008)	35	28	63	Modified ART-1	6	GE	0.9074
					CNN *			0.8938

* The results of CNN with proposed methodology

For data set 2, the best result value by CNN within the proposed methodology and the result found by CNN in the article (Ozturk, Ozturk & Islier, 2006) is compared according to GEA and presented in Table 5.21. According to the table, the result found by the proposed CNN is the same with the result by CNN in the article.

Table 5.21 Article problem solutions by CNN according to GEA.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
2	Ozturk, Ozturk & Islier (2006)	7	5	12	CNN (Article)	2	GEA	0.7368
					CNN *			

* CNN result is the result of CNN with proposed methodology

In Table 5.22, the best result values of CNN according to the Exceptional Elements are presented in comparison with the results of the articles. CNN found a better result than ART-1 for data set 4 (Chen & Cheng, 1995). For data set 7 (Malakooti & Yang, 1994), our CNN and CNN within the methodology in the article found the same result which is better than the results found by ROC and DCA. Proposed CNN found the same result with CNN within the methodology in the article for data set 9 (Onwubolu, 1999).

Table 5.22 Article problem solutions by CNN according to Exceptional Elements.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
4	Chen & Cheng (1995)	10	10	20	ART-1	3	Exceptional Elements	8
					CNN *			6
7	Malakooti & Yang (1994)	15	10	25	ROC	3	Percentage of Exceptional Elements	0.183
					DCA			0.1224
					CNN (Article)			0.102
					CNN *			0.102
9	Onwubolu (1999)	15	10	25	CNN (Article)	3	Exceptional Elements	3
					CNN *			
* The results of CNN with proposed methodology								

In Table 5.23, solutions by the methods in the article (Malakooti & Yang, 1994) and CNN solution for data set 7 are compared. Our CNN and SOM within the methodology in the article found the same result which is better than the results found by ROC and DCA.

Table 5.23 Article problem solutions by CNN according to MU.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
7	Malakooti & Yang (1994)	15	10	25	ROC	3	MU	0.8
					DCA			0.8776
					SOM			0.88
					CNN			0.88

In this section, 10 binary data sets among 15 data sets are solved by our CNN according to the performance measures in the articles because there is no performance meausure in 5 binary problem sets as it was in the SOM application. For the 10 data sets, 19 different solutions from the articles are compared according to performance measures in articles. In the article by Malakooti & Yang (1994) problem set 7 is solved by 3 different methods (ROC, DCA and SOM) according to 3 of the performance measures (GE, Percentage of Exceptional Elements and MU). So, 12 of the 19 solutions are from the article by Malakooti & Yang (1994).

The comparisons show that our CNN found the same results with the methods used in the articles for 10 of the 19 solutions. For the rest 8 solutions, CNN found better results than the articles. ART-1 according to Exceptional Elements (in problem 4), ROC and DCA according to GE, Percentage of Exceptional Elements and MU (in problem 7), and Modified ART-1 according to GE (in problem 3) are the methods and performance measures used in the problems that CNN found better results than the article. CNN found just one worse result than Modified ART-1 according to GE in problem 15. As a result, CNN within the proposed methodology is a good ANN method to solve the CF problem according to 4 performance measures in the articles as it was according to 6 performance measures mentioned in the previous section.

5.2.3.4 Problem Solutions for Different Numbers of Cells

Problems are solved by CNN with different numbers of cells to search for better performance results as it was done while working with SOM. For data set 1 (Lee, Yamakawa & Lee, 1997) and 2 (Ozturk, Ozturk & Islier, 2006), no different numbers of cells were tried because the number of parts and machines are not suitable for more number of cells to be formed.

The best result values of CNN provided according to the proposed performance measure with different numbers of cells are presented in Table 5.24 – Table 5.27. The comparisons are grouped into four according to the methods (expected, ART-1/Modified ART-1, heuristics and SOM with article methodologies) used in the articles and are presented below. According to tables, for all the binary problem sets used in the present study, changing the numbers of the cells did not give better results by our CNN than the results found by the articles.

Table 5.24 Problem solutions by CNN with different numbers of cells according to proposed performance measure compared with expected solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
5	Malakooti & Yang (1994)	15	10	25	Expected	-	0.94	-	-	-
					CNN	0.7375		NC	NC	-
6	Malakooti & Yang (1994)	15	10	25	Expected	-	0.8705	-	-	-
					CNN	0.7185		NC	NC	-
8	Kusiak & Lee (1996)	15	10	25	Expected	-	0.886	-	-	-
					CNN	0.7024		NC	NC	-
14	Smith & Escobedo (1994)	13	17	30	Expected	-	0.904	-	-	-
					CNN	0.7203		NC	NC	-
NC : No classification after 20 runs										

Table 5.25 Problem solutions by CNN with different numbers of cells according to proposed performance measure compared with ART-1/Modified ART-1 solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
3	Yang & Yang (2008)	9	9	18	ART-1	0.7791	-	-	-	-
					Modified ART-1	-	0.8394	-	-	-
					CNN	0.7791	0.8388	0.7759	-	-
4	Chen & Cheng (1995)	10	10	20	ART-1	-	0.7283	-	-	-
					CNN	0.6704	0.8022	0.7418	0.6956	-
10	Yang & Yang (2008)	15	15	30	Modified ART-1	-	-	1	-	-
					CNN	0.6945	0.832		NC	-
11	Yang & Yang (2008)	15	15	30	Modified ART-1	-	-	0.9308	-	-
					CNN	0.6779	0.7877		NC	-
12	Yang & Yang (2008)	15	15	30	Modified ART-1	-	-	0.9	-	-
					CNN	0.6455	0.76		NC	-
13	Yang & Yang (2008)	15	15	30	Modified ART-1	-	-	0.8285	-	-
					CNN	0.623	0.717		0.7416	-
15	Yang & Yang (2008)	35	28	63	Modified ART-1	-	-	-	-	0.7994
					CNN	0.5532	0.6121	0.6406	0.7181	0.7748
NC : No classification after 20 runs										

Table 5.26 Problem solutions by CNN with different numbers of cells according to proposed performance measure compared with heuristics solutions.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
7	Malakooti & Yang (1994)	15	10	25	ROC	-	0.7914	-	-	-
					DCA	-	0.8647	-	-	-
					CNN	0.7315	0.8764	0.8394	NC	-
NC : No classification after 20 runs										

Table 5.27 Problem solutions by CNN with different number of cells according to proposed performance measure compared with Article SOM solutions.

<table>
<tr><th rowspan="2">PROB. NO</th><th rowspan="2">SOURCE</th><th rowspan="2"># OF PARTS</th><th rowspan="2"># OF MACH.</th><th rowspan="2">TOTAL #</th><th rowspan="2">METHOD</th><th colspan="5">PROPOSED PERFORMANCE MEASURE</th></tr>
<tr><th>2 CELLS</th><th>3 CELLS</th><th>4 CELLS</th><th>5 CELLS</th><th>6 CELLS</th></tr>
<tr><td rowspan="2">7</td><td rowspan="2">Malakooti & Yang (1994)</td><td rowspan="2">15</td><td rowspan="2">10</td><td rowspan="2">25</td><td>SOM</td><td>-</td><td rowspan="2">0.8764</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>CNN</td><td>0.7315</td><td>0.8394</td><td>NC</td><td>-</td></tr>
<tr><td rowspan="2">9</td><td rowspan="2">Onwubolu (1999)</td><td rowspan="2">15</td><td rowspan="2">10</td><td rowspan="2">25</td><td>SOM</td><td>-</td><td rowspan="2">0.8959</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>CNN</td><td>0.7481</td><td>NC</td><td>NC</td><td>-</td></tr>
<tr><td colspan="11">NC : No classification after 20 runs</td></tr>
</table>

5.2.4 Discussion

In the binary cases section, 15 binary problem sets are solved by SOM and CNN within the proposed methodology. According to the results and the comparisons, there can be five subjects to discuss.

The first subject is that “Can SOM solve binary CF problems effectively?” This question is investigated by testing if SOM is superior to the other CF methods. The comparisons of 15 binary problem solutions showed that SOM found better results than 4 CF methods in 3 data sets regardless of the type of the performance measure used. Furthermore, SOM did not find any worse solutions in any data set. By these findings, it can be claimed that SOM within the proposed methodology can be used as an effective ANN method to solve small and medium sized (according to number of parts and machines) CF problems.

The second subject is that “Can CNN solve binary CF problems effectively?” Like it was done for SOM, it is tested if CNN is superior to the other CF methods to get an answer to the question. The comparisons of 15 binary problem solutions showed that CNN found better results than 3 CF methods in 2 data sets regardless of the type of the performance measure used. CNN found worse results than Modified ART-1 in one data set. Also, CNN found worse results than Modified ART-1 in another data set according to GEA, NGM, GCI and proposed performance measure. For the rest 4 data sets, CNN found the same results with Modified ART-1 regardless of the type of the performance measure used. According to these comparisons, it can be said that our CNN solves small and medium sized (according to number of parts and machines) CF problems well.

The third subject is that “Is SOM a better CF method compared to CNN in binary cases?” When the results by SOM within the proposed methodology is compared to the results by CNN within the proposed methodology in solving binary problems, SOM is found to get better results than CNN in data set 3 and data set 15. For the rest 13 sets, SOM got the same results with CNN. So, SOM has a better performance in solving small and medium sized binary problem sets.

The fourth subject is that “Is the proposed performance measure worth being used by itself instead of performance measures in the literature including the five performance measures composing the proposed performance measure?” Three (GE, GEA and MU) of the five performance measures were common in some of the articles used in the present study. Percentage of Exceptional Elements/Exceptional Elements also used in the articles is already the basic parameter of the five performance measures. So that, to investigate the effectiveness of the proposed performance measure for binary cases, it is decided to compare the solutions according to the five performance measures. When the five performance measures are compared with each other in their values of results, it is found that they measure different values almost in each binary problem set for the same result matrix. This is thought to result from the formulas of five performance measures that include different basic parameters. When a method found a better result than another method according to the proposed performance measure, it found better results also according to each of the five performance measures composing the proposed performance measure consistently in 37 of 38 solutions. As a conclusion, the variety of the result values found by the five performance measures and the consistency of the results by the proposed performance measure and the five performance measures in finding better values support the claim of the present study about the superiority of the proposed performance measure to its dimensions.

The last subject of discussion is that “Does redefining number of cells when solving CF problems have any advantages in getting better results?” As mentioned before, optimal number of cells that fits to the structure of input matrix finds the best performance results. Redefining numbers of cells turns into an advantage when a problem has not been solved with the optimal number of cells. Since the articles used in the present thesis studied the optimal numbers of cells for their binary problems, increasing or decreasing the number of the cells did not provide better results. However, it can be useful to redefine numbers of cells by a number that is not optimal for the best solution. It may not always be possible to implement optimal number of cells to a CMS in practice. In these cases, the decision on number of cells to be used can be made by analyzing and comparing the result values by different number of cells other than the optimal number.

5.3 Nonbinary Cases

Nonbinary (fuzzy) cases in the present study are the problem sets collected from literature which have inputs that are Nonbinary PMIMes. In this thesis, our Fuzzy SOM and Fuzzy CNN are used for solving nonbinary cases in the literature. The proposed methodology is used for nonbinary cases after testing it in binary cases.

5.3.1 The Nonbinary Formation

Nonbinary (fuzzy) problem sets in CF problems can be produced by various data sources such as processing time of each part on different machines, demand/volume of each part, work load on each machine (Kamal & Burke, 1996). In this thesis, 6 nonbinary problem sets are selected from literature; 3 sets were produced by processing time of each part on different machines (Peker & Kara (2004), Kamal & Burke (1996)), 1 set was produced by demand/volume of each part (Won & Currie, 2007), 2 sets were produced by work loads on each machine (Liao, Chen, Chen & Coates (1996), Kamal & Burke (1996)). Processing time and work load values were fuzzy (between [0,1]), but demand/volume values (shown in the 3^{rd} example of the nonbinary problem sets) were not between [0,1]. So, the demand/volume values are transformed into fuzzy values.

5.3.2 The Nonbinary Problem Sets

6 nonbinary problem sets are collected from 4 articles (published between 1996 and 2007) and used as inputs for Fuzzy SOM and Fuzzy CNN within the proposed methodology. The nonbinary problem sets are presented below (in Table 5.28). Number of parts in the problems differs within a range of 7 to 24 and the number of machines differs within a range of 7 to 30. The nonbinary problem set matrices are given in Appendix A5. The nonbinary (fuzzy) problem sets are divided into three parts; volume based, work load based and processing time based. Each part is divided into three subtitles; proposed performance measure, article performance measure and different numbers of cells. Before the presentation of problem sets and the solutions Fuzzy SOM variables used in MATLAB are given below.

Table 5.28 The nonbinary problem sets.

NO	AUTHOR	# OF PARTS	# OF MACHINES	# OF CELLS IN ARTICLE
1	Peker & Kara (2004)	7	7	2
2	Liao, Chen, Chen & Coates (1996)	9	9	3
3	Won & Currie (2007)	13	15	3
4	Kamal & Burke (1996)	24	14	4
5	Kamal & Burke (1996)	20	24	3
6	Kamal & Burke (1996)	15	30	3

5.3.3 Fuzzy SOM Solutions

5.3.3.1 Fuzzy SOM Variables

Like it was done in binary cases, the nonbinary problem set solutions by Fuzzy SOM are provided by using MATLAB 7.5. All the procedure applied while solving the binary problems by SOM and CNN were also applied for Fuzzy SOM. So, the variable values or functions chosen for Fuzzy SOM are the same with the ones chosen for SOM and CNN. The variable values or functions are presented below.

- **Fuzzy SOM Variables :**

Epoch = 100 (for all cell groupings)

Goal = 1e-5 (for all cell groupings)
Dimension of maps = [2 1] (for 2 cell groupings)
Dimension of maps = [3 1] (for 3 cell groupings)
Dimension of maps = [4 1] (for 4 cell groupings)
Dimension of maps = [5 1] (for 5 cell groupings)
Dimension of maps = [6 1] (for 6 cell groupings)
Topology function = hextop (for all cell groupings default function in MATLAB)
Distance function = linkdist (for all cell groupings default function in MATLAB)
Ordering phase learning rate = 0.9 (for all cell groupings default value in MATLAB)
Ordering phase steps = 1000 (for all cell groupings default value in MATLAB)
Tuning phase learning rate = 0.02 (for all cell groupings default value in MATLAB)
Tuning phase neighborhood distance = 1 (for all cell groupings default value in MATLAB)

The MATLAB codes for Fuzzy SOM is generated by following the proposed methodology mentioned in chapter four. For all of the nonbinary problem sets, each MATLAB code is runned 10 times. The best result value among the results selected to be compared with the other results in the articles. The best results are presented below and the matrices for the best results of Fuzzy SOM for nonbinary problems are presented in Appendix A6. If the code can not classify at least one machine and one part into each cell grouping after 20 runs, the result is nominated as “no classification after 20 runs”.

5.3.3.2 Volume Based Problem Solutions

Volume based nonbinary problem set is taken from Won & Currie (2007). The set is solved by our Fuzzy SOM and compared with Fuzzy ART/RRR-RSS. The comparisons are presented in Table 5.29 – Table 5.31.

5.3.3.2.1 Problem Solutions According to Proposed Performance Measure. Our Fuzzy SOM result provided according to six performance measures is presented in

Table 5.29. Proposed Fuzzy SOM found a better result according to MU and found worse results according to rest performance measures than Fuzzy ART/RRR-RSS.

Table 5.29 Volume based problem solutions with Fuzzy SOM according to proposed performance measure.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
3	Won & Currie (2007)	13	15	28	Fuzzy ART/RRR-RSS	3	0.808	0.6051	0.7829	0.9334	0.5028	0.7264
					Fuzzy SOM		0.8054	0.5920	0.7621	0.8803	0.5285	0.7137

5.3.3.2.2 Problem Solutions According to Article Performance Measures. Proposed Fuzzy SOM result values provided according to the article performance measure are presented in Table 5.30. No better result is found by our Fuzzy SOM than Fuzzy ART/RRR-RSS.

Table 5.30 Volume based problem solutions with Fuzzy SOM according to article performance measures.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
3	Won & Currie (2007)	13	15	28	Fuzzy ART/RRR-RSS	3	Weighted Group Capability Index	0.9771 (Volume based)
					Fuzzy SOM			0.8803

5.3.3.2.3 Problem Solutions for Different Numbers of Cells. The best Fuzzy SOM result values with different numbers of cells are presented in Table 5.31. For data set 3, our Fuzzy SOM with 4 cells and 5 cells found better results than Fuzzy ART/RRR-RSS and proposed Fuzzy SOM with 3 cells.

Table 5.31 Volume based problem solutions with Fuzzy SOM according to proposed performance measure for different numbers of cells.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE RESULTS				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
3	Won & Currie (2007)	13	15	28	Fuzzy ART/RRR-RSS	-	0.7264	-	-	-
					Fuzzy SOM	-	0.7137	0.7362	0.7575	-
NC : No classification after 20 runs										
All Fuzzy SOM results are the results of Fuzzy SOM with proposed methodology										

5.3.3.3 Processing Time Based Problem Solutions

3 nonbinary processing time based problem sets are collected from literature. Our Fuzzy SOM results for the processing time based nonbinary problem sets (numbered 1 (Peker & Kara, 2004), 4 and 6 (Kamal & Burke, 1996)) are presented in Table 5.32 – Table 5.34. The results found by proposed Fuzzy SOM is compared with the Fuzzy ART, FACT and a genetic algorithm used in the articles.

5.3.3.3.1 Problem Solutions According to Proposed Performance Measure. Fuzzy SOM result values provided according to the proposed performance measure by this thesis are presented in Table 5.32. For data set 1 (Peker & Kara, 2004), our Fuzzy SOM found the same results with Fuzzy ART according to all performance measures. For data set 6 (Kamal & Burke, 1996), proposed Fuzzy SOM found the same results with Venugopal & Narendran (1992) - Genetic Algorithm and FACT. For data set 4 (Kamal & Burke, 1996), our Fuzzy SOM found better results according to GE, GEA and MU, and worse results according to NGM, GCI and proposed performance measure.

Table 5.32 Processing time based problem solutions with Fuzzy SOM according to proposed performance measure.

<table>
<tr><th>PROB. NO</th><th>SOURCE</th><th># OF PARTS</th><th># OF MACH.</th><th>TOTAL #</th><th>METHOD</th><th># OF CELLS</th><th>GE</th><th>GEA</th><th>NGM</th><th>GCI</th><th>MU</th><th>PROP. PERF. MEA.</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Peker & Kara (2004)</td><td rowspan="2">7</td><td rowspan="2">7</td><td rowspan="2">14</td><td>Fuzzy ART</td><td rowspan="2">2</td><td rowspan="2">0.9076</td><td rowspan="2">0.7963</td><td rowspan="2">0.8867</td><td rowspan="2">0.8776</td><td rowspan="2">0.7167</td><td rowspan="2">0.837</td></tr>
<tr><td>Fuzzy SOM</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Kamal & Burke (1996)</td><td rowspan="2">24</td><td rowspan="2">14</td><td rowspan="2">38</td><td>FACT</td><td rowspan="2">4</td><td>0.7894</td><td>0.573</td><td>0.7758</td><td>0.9674</td><td>0.4902</td><td>0.7192</td></tr>
<tr><td>Fuzzy SOM</td><td>0.7984</td><td>0.5732</td><td>0.7552</td><td>0.8966</td><td>0.5092</td><td>0.7065</td></tr>
<tr><td rowspan="3">6</td><td rowspan="3">Kamal & Burke (1996)</td><td rowspan="3">15</td><td rowspan="3">30</td><td rowspan="3">45</td><td>Venugopal, & Narendran (1992) - Genetic Algorithm</td><td rowspan="3">3</td><td rowspan="3">0.9612</td><td rowspan="3">0.9045</td><td rowspan="3">0.9504</td><td rowspan="3">0.9725</td><td rowspan="3">0.5180</td><td rowspan="3">0.8613</td></tr>
<tr><td>FACT</td></tr>
<tr><td>Fuzzy SOM</td></tr>
</table>

5.3.3.3.2 Problem Solutions According to Article Performance Measures. Fuzzy SOM within the proposed methodology results provided according to the calculatable performance measures in the article are presented in Table 5.33. For data set 1 (Peker

& Kara, 2004), there is no article performance measure so it is not included in the table. For data set 6 (Kamal & Burke, 1996), Fuzzy SOM found the same results with Venugopal & Narendran (1992) - Genetic Algorithm and FACT according to number of shared machines and inter-cellular movements. For data set 4 (Kamal & Burke, 1996), our Fuzzy SOM found worse results according to number of shared machines and inter-cellular movements.

Table 5.33 Processing time based problem solutions with Fuzzy SOM according to article performance measures.

<table>
<tr><th>PROB. NO</th><th>SOURCE</th><th># OF PARTS</th><th># OF MACH.</th><th>TOTAL #</th><th>METHOD</th><th># OF CELLS</th><th>ARTICLE PERF. MEA.</th><th>PERF. RESULT</th></tr>
<tr><td rowspan="4">4</td><td rowspan="4">Kamal & Burke (1996)</td><td rowspan="4">24</td><td rowspan="4">14</td><td rowspan="4">38</td><td>FACT</td><td rowspan="4">4</td><td rowspan="2">Number of Shared Machines</td><td>3</td></tr>
<tr><td>Fuzzy SOM</td><td>5</td></tr>
<tr><td>FACT</td><td rowspan="2">Inter-cellular Movements</td><td>4</td></tr>
<tr><td>Fuzzy SOM</td><td>7</td></tr>
<tr><td rowspan="6">6</td><td rowspan="6">Kamal & Burke (1996)</td><td rowspan="6">15</td><td rowspan="6">30</td><td rowspan="6">45</td><td>Venugopal, & Narendran (1992) - Genetic Algorithm</td><td rowspan="6">3</td><td rowspan="3">Number of Shared Machines</td><td rowspan="3">6</td></tr>
<tr><td>FACT</td></tr>
<tr><td>Fuzzy SOM</td></tr>
<tr><td>Venugopal, & Narendran (1992) - Genetic Algorithm</td><td rowspan="3">Inter-cellular Movements</td><td rowspan="3">6</td></tr>
<tr><td>FACT</td></tr>
<tr><td>Fuzzy SOM</td></tr>
</table>

5.3.3.3.3 Problem Solutions for Different Numbers of Cells. Proposed Fuzzy SOM result values with different numbers of cells are presented in Table 5.34. For data set 4 (Kamal & Burke, 1996), Fuzzy SOM with G=5 found the same result with FACT with G=4 and Fuzzy SOM with G=6 found a better result than FACT with G=4.

Table 5.34 Processing time based problem solutions with Fuzzy SOM according to proposed performance measure for different numbers of cells.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE RESULTS				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
1	Peker & Kara (2004)	7	7	14	Fuzzy ART	0.837	-	-	-	-
					Fuzzy SOM		NC	-	-	-
4	Kamal & Burke (1996)	24	14	38	FACT	-	-	0.7192	-	-
					Fuzzy SOM	-	0.6435	0.7065	0.7192	0.7324
6	Kamal & Burke (1996)	15	30	45	Venugopal, & Narendran (1992) - Genetic Algorithm	-	0.8613	-	-	-
					FACT	-		-	-	-
					Fuzzy SOM	-		NC	NC	NC

NC : No classification after 20 runs

5.3.3.4 Work Load Based Problem Solutions

2 work load based nonbinary problem sets are collected from literature. Our Fuzzy SOM results for the work load based nonbinary problem sets (numbered 2 (Liao, Chen, Chen & Coates (1996)) and 5 (Kamal & Burke (1996)) are presented in Table 5.35 – Table 5.37. Proposed Fuzzy SOM is compared with the Fuzzy ROC, FACT and Jacobs (1985) solutions. The comparisons are explained below.

5.3.3.4.1 Problem Solutions According to Proposed Performance Measure. In Table 5.35, solutions by the methods in the articles (Liao, Chen, Chen & Coates (1996) and Kamal & Burke (1996)) and Fuzzy SOM solution for data sets 2 and 5 are compared. For data set 2, proposed Fuzzy SOM is found to get the same result with Fuzzy ROC. For data set 5, Fuzzy SOM is found to get a better result than the results by Jacobs (1985) and the results by FACT according to all performance measures.

Table 5.35 Work load based problem solutions with Fuzzy SOM according to proposed performance measure.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
2	Liao, Chen, Chen & Coates (1996)	9	9	18	Fuzzy ROC	3	0.8595	0.6963	0.8267	0.8952	0.6963	0.7948
					Fuzzy SOM							
5	Kamal & Burke (1996)	20	24	44	Jacobs (1985)	3	0.6469	0.3142	0.5285	0.6915	0.3151	0.4992
					FACT		0.6621	0.3362	0.5482	0.7051	0.3372	0.5177
					Fuzzy SOM		0.6771	0.3572	0.5634	0.7077	0.3607	0.5332

5.3.3.4.2 Problem Solutions According to Article Performance Measures. Our Fuzzy SOM result values according to the article performance measures are presented in Table 5.36. For data set 2, there is no article performance measure to compare so it is not included in the table. For data set 5, Fuzzy SOM is found to get a better result than Jacobs (1985) and FACT according to number of shared machines and inter-cellular movements.

Table 5.36 Work load based problem solutions with Fuzzy SOM according to article performance measures.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
5	Kamal & Burke (1996)	20	24	44	Jacobs (1985)	3	Number of Shared Machines	21
					FACT			21
					Fuzzy SOM			17
					Jacobs (1985)		Inter-cellular Movements	37
					FACT			37
					Fuzzy SOM			35

5.3.3.4.3 Problem Solutions for Different Numbers of Cells. Problem 2 (Liao, Chen, Chen & Coates, 1996) and problem 5 (Kamal & Burke, 1996) are solved by the methods and the number of cells (G=3) in the articles. They are solved again by our Fuzzy SOM by redefining the number of cells. The results are compared and presented in Table 5.37. For data set 2 (Liao, Chen, Chen & Coates, 1996), Fuzzy SOM with 4 cells found a better result than Fuzzy SOM and Fuzzy ROC with 3 cells. For data set 5 (Kamal & Burke, 1996), Fuzzy SOM with 3, 4, 5 and 6 cells found better results than the method of Jacobs (1985) with 3 cells and FACT with 3 cells.

Table 5.37 Work load based problem solutions with Fuzzy SOM according to proposed performance measure for different numbers of cells.

PROB. NO	**SOURCE**	**# OF PARTS**	**# OF MACH.**	**TOTAL #**	**METHOD**	**PROPOSED PERFORMANCE MEASURE RESULTS**				
						2 CELLS	**3 CELLS**	**4 CELLS**	**5 CELLS**	**6 CELLS**
2	Liao, Chen, Chen & Coates (1996)	9	9	18	Fuzzy ROC	-	0.7948	-	-	-
					Fuzzy SOM	0.682		0.8077	-	-
5	Kamal & Burke (1996)	20	24	44	Jacobs (1985)	-	0.4992	-	-	-
					FACT	-	0.5177	-	-	-
					Fuzzy SOM	-	0.5332	0.5537	0.5507	0.6089

Totally, 6 nonbinary problem sets are solved by our Fuzzy SOM. Because some of the problems are solved with more than one method in the articles, 8 nonbinary solutions are collected from the articles. They are compared with our Fuzzy SOM solutions derived according to proposed performance measure, its 5 dimensions and the performance measures (number of shared machines and inter-cellular movements (for data set 4, 5 and 6 (Kamal & Burke, 1996)) or weighted GCI (for data set 3 (Won & Currie, 2007)) in the articles. The comparisons reveal that Fuzzy SOM found the same results with the methods used in the data set 1-Fuzzy ART (Peker & Kara, 2004), the data set 2- Fuzzy ROC (Liao, Chen, Chen & Coates, 1996) and the data set 6-FACT and Genetic Algorithm (Kamal & Burke, 1996). For one solution, Fuzzy SOM found better results than the methods (FACT and Jacobs (1985)) used in the article (Kamal & Burke, 1996) according to each of the 6 performance measures and article performance measures (number of shared machines and inter-cellular movements). In data set 3 (Won & Currie, 2007), when Fuzzy SOM solutions are compared with FuzzyART/RRR-RSS solutions, Fuzzy SOM is found to get worse results according to GE, GEA, NGM, GCI, proposed performance measure, weighted GCI (the article performance measure) and a better result according to MU. In data set 4 (Kamal & Burke, 1996), Fuzzy SOM found worse results than FACT results according to NGM, GCI, proposed performance measure, number of shared machines and inter-cellular movements (the article performance measures) and better results according to GE, GEA and MU. When the 6 nonbinary problems are solved with trying different numbers of cells, better results than the results found in the articles are provided for 4 of the problems. As a result, proposed Fuzzy SOM can be used as a good ANN method to solve the nonbinary CF problems according to the performance measures.

5.3.4 Fuzzy CNN Solutions

5.3.4.1 Fuzzy CNN Variables

Like it was done in binary cases and the Fuzzy SOM application, the nonbinary problem set solutions by Fuzzy CNN are provided by using MATLAB 7.5. All the procedure applied while solving the binary and nonbinary problems were also applied for Fuzzy CNN. So, the variable values or functions chosen for Fuzzy CNN are the same with the ones chosen for SOM, CNN and Fuzzy SOM. The variable values or functions are presented below.

- **Fuzzy CNN Variables :**

Epoch = 100 (for all cell groupings)
Goal = 1e-5 (for all cell groupings)
Dimension of maps = [2 1] (for 2 cell groupings)
Dimension of maps = [3 1] (for 3 cell groupings)
Dimension of maps = [4 1] (for 4 cell groupings)
Dimension of maps = [5 1] (for 5 cell groupings)
Dimension of maps = [6 1] (for 6 cell groupings)
Topology function = hextop (for all cell groupings default function in MATLAB)
Distance function = linkdist (for all cell groupings default function in MATLAB)
Ordering phase learning rate = 0.9 (for all cell groupings default value in MATLAB)
Ordering phase steps = 1000 (for all cell groupings default value in MATLAB)
Tuning phase learning rate = 0.02 (for all cell groupings default value in MATLAB)
Tuning phase neighborhood distance = 1 (for all cell groupings default value in MATLAB)

The MATLAB codes for Fuzzy CNN is generated by following the proposed methodology mentioned in chapter four. For all of the nonbinary problem sets, each

MATLAB code is runned 10 times. The best result value among the results is selected to be compared with the other results in the articles. The best results are presented below and the matrices for the best results of Fuzzy CNN for nonbinary problems are presented in Appendix A7. If the code can not classify at least one machine and one part into each cell grouping after 20 runs, the result is nominated as “no classification after 20 runs”.

5.3.4.2 Volume Based Problem Solutions

Proposed Fuzzy CNN results for the volume based nonbinary problem set (numbered 3 (Won & Currie, 2007)) are presented in Table 5.38 – Table 5.40. For nonbinary problem set 3, the result found by our Fuzzy CNN is compared with the Fuzzy ART/RRR-RSS. The comparisons are presented below.

5.3.4.2.1 Problem Solutions According to Proposed Performance Measure. In Table 5.38, our Fuzzy CNN results provided according to six performance measures are presented. Proposed Fuzzy CNN found a better result according to MU and found worse results according to rest of the performance measures.

Table 5.38 Volume based problem solutions with Fuzzy CNN according to proposed performance measure.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
3	Won & Currie (2007)	13	15	28	Fuzzy ART/RRR-RSS	3	0.808	0.6051	0.7829	0.9334	0.5028	0.7264
					Fuzzy CNN		0.8054	0.5920	0.7621	0.8803	0.5285	0.7137

5.3.4.2.2 Problem Solutions According to Article Performance Measures. For data set 3 (Won & Currie, 2007), our Fuzzy CNN result values provided according to the performance measure in the article are showed in Table 5.39. Fuzzy CNN within the proposed methodology could not find any better results than Fuzzy ART/RRR-RSS.

Table 5.39 Volume based problem solutions with Fuzzy CNN according to article performance measures.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
3	Won & Currie (2007)	13	15	28	Fuzzy ART/RRR-RSS	3	Weighted Group Capability Index	0.9771 (Volume based)
					Fuzzy CNN			0.8803

5.3.4.2.3 Problem Solutions for Different Numbers of Cells. Our Fuzzy CNN results for data set 3 with different numbers of cells are presented in Table 5.40. Our Fuzzy CNN with 4 cells and 5 cells found better results than Fuzzy ART/RRR-RSS and our Fuzzy CNN with 3 cells according to proposed performance measure.

Table 5.40 Volume based problem solutions with Fuzzy CNN according to proposed performance measure for different numbers of cells.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE RESULTS				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
3	Won & Currie (2007)	13	15	28	Fuzzy ART/RRR-RSS	-	0.7264	-	-	-
					Fuzzy CNN	-	0.7137	0.7416	0.7446	-

5.3.4.3 Processing Time Based Problem Solutions

The best result values of Fuzzy CNN within the proposed methodology for the processing time based nonbinary problem sets (numbered 1 (Peker & Kara, 2004), 4 and 6 (Kamal & Burke, 1996)) are presented in Table 5.41 – Table 5.43. The result found by our Fuzzy CNN is compared with Fuzzy ART, FACT and a genetic algorithm.

5.3.4.3.1 Problem Solutions According to Proposed Performance Measure. In Table 5.41, our Fuzzy CNN result values provided according to the proposed performance measure are presented. For data set 1 (Peker & Kara, 2004), proposed Fuzzy CNN found the same result with Fuzzy ART according to all performance measures. For data set 6 (Kamal & Burke, 1996), Fuzzy CNN within the proposed methodology found the same results with Genetic Algorithm (Venugopal & Narendran,

1992) and FACT according to all performance measures. For data set 4 (Kamal & Burke, 1996), proposed Fuzzy CNN found better results according to GE, GEA and MU and worse results according to NGM, GCI and proposed performance measure.

Table 5.41 Processing time based problem solutions with Fuzzy CNN according to proposed performance measure.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
1	Peker & Kara (2004)	7	7	14	Fuzzy ART	2	0.9076	0.7963	0.8867	0.8776	0.7167	0.837
					Fuzzy CNN							
4	Kamal & Burke (1996)	24	14	38	FACT	4	0.7894	0.573	0.7758	0.9674	0.4902	0.7192
					Fuzzy CNN		0.7953	0.5696	0.7551	0.9038	0.5005	0.7048
6	Kamal & Burke (1996)	15	30	45	Venugopal, & Narendran (1992) - Genetic Algorithm	3	0.9612	0.9045	0.9504	0.9725	0.5180	0.8613
					FACT							
					Fuzzy CNN							

5.3.4.3.2 Problem Solutions According to Article Performance Measures. Solutions by Fuzzy CNN within the proposed methodology for the processing time based nonbinary problem sets provided according to the article performance measures are presented in Table 5.42. For data set 1 (Peker & Kara, 2004), there is no article performance measure so it is not included in the table. For data set 6 (Kamal & Burke, 1996), our Fuzzy CNN found the same results with Genetic Algorithm (Venugopal & Narendran, 1992) and FACT according to number of shared machines and inter-cellular movements. For data set 4 (Kamal & Burke, 1996), proposed Fuzzy CNN found worse results according to number of shared machines and inter-cellular movements.

5.3.4.3.3 Problem Solutions for Different Numbers of Cells. Our Fuzzy CNN result values with different numbers of cells provided according to the proposed performance measure are presented in Table 5.43. For data set 4 (Kamal & Burke, 1996), Fuzzy CNN with G=5 and G=6 found better results than FACT with G=4.

Table 5.42 Processing time based problem solutions with Fuzzy CNN according to article performance measures.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	ARTICLE PERF. MEA.	PERF. RESULT
4	Kamal & Burke (1996)	24	14	38	FACT	4	Number of Shared Machines	3
					Fuzzy CNN			4
					FACT		Inter-cellular Movements	4
					Fuzzy CNN			6
6	Kamal & Burke (1996)	15	30	45	Venugopal, & Narendran (1992) - Genetic Algorithm	3	Number of Shared Machines	6
					FACT			
					Fuzzy CNN			
					Venugopal, & Narendran (1992) - Genetic Algorithm		Inter-cellular Movements	6
					FACT			
					Fuzzy CNN			

Table 5.43 Processing time based problem solutions with Fuzzy CNN according to proposed performance measure for different numbers of cells.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	PROPOSED PERFORMANCE MEASURE RESULTS				
						2 CELLS	3 CELLS	4 CELLS	5 CELLS	6 CELLS
1	Peker & Kara (2004)	7	7	14	Fuzzy ART	0.837	-	-	-	-
					Fuzzy CNN		0.7544	-	-	-
4	Kamal, & Burke (1996)	24	14	38	FACT	-	-	0.7192	-	-
					Fuzzy CNN	-	0.6435	0.7048	0.7222	0.7407
6	Kamal & Burke (1996)	15	30	45	Venugopal, & Narendran (1992) - Genetic Algorithm	-	0.8613	-	-	-
					FACT	-		-	-	-
					Fuzzy CNN	-		0.7678	NC	NC
NC : No classification after 20 runs										

5.3.4.4 Work Load Based Problem Solutions

Proposed Fuzzy CNN results for the work load based nonbinary problem sets (numbered 2 (Liao, Chen, Chen & Coates (1996)) and 5 (Kamal & Burke (1996)) are showed in Table 5.44 – Table 5.46. For nonbinary problem set 2, the result found by our Fuzzy CNN is compared with the Fuzzy ROC. For data set 5, solutions by our Fuzzy CNN is compared with FACT and Jacobs (1985).

5.3.4.4.1 Problem Solutions According to Proposed Performance Measure. In Table 5.44, solutions by the methods in the articles (Liao, Chen, Chen & Coates (1996) and Kamal & Burke (1996)) and Fuzzy CNN solution for data sets 2 and 5 are compared. For data set 2, our Fuzzy CNN is found to get the same result with Fuzzy ROC. For data set 5, proposed Fuzzy CNN is found to get a better result than Jacobs (1985) and FACT according to all performance measures.

Table 5.44 Work load based problem solutions with Fuzzy CNN according to proposed performance measure.

PROB. NO	SOURCE	# OF PARTS	# OF MACH.	TOTAL #	METHOD	# OF CELLS	GE	GEA	NGM	GCI	MU	PROP. PERF. MEA.
2	Liao, Chen, Chen & Coates (1996)	9	9	18	Fuzzy ROC	3	0.8595	0.6963	0.8267	0.8952	0.6963	0.7948
					Fuzzy CNN							
5	Kamal & Burke (1996)	20	24	44	Jacobs (1985)	3	0.6469	0.3142	0.5285	0.6915	0.3151	0.4992
					FACT		0.6621	0.3362	0.5482	0.7051	0.3372	0.5177
					Fuzzy CNN		0.6762	0.3577	0.5685	0.7224	0.3497	0.5349

5.3.4.4.2 Problem Solutions According to Article Performance Measures. Fuzzy CNN result values provided according to the article performance measures are presented in Table 5.45. For data set 2, there is no article performance measure to compare, so it is not included in the table. For data set 5, our Fuzzy CNN is found to get a better result than Jacobs (1985) and FACT according to number of shared machines and inter-cellular movements.

5.3.4.4.3 Problem Solutions for Different Numbers of Cells. Problem 2 (Liao, Chen, Chen & Coates, 1996) and problem 5 (Kamal & Burke, 1996) are solved by the methods and the number of cells (G=3) used in the articles. They are solved once more by Fuzzy CNN within the proposed methodology by changing the number of cells in the articles. The results are compared and presented in Table 5.46. For data set 2 (Liao, Chen, Chen & Coates, 1996), Fuzzy CNN with 4 cells found a better result than Fuzzy CNN and Fuzzy ROC with 3 cells. For data set 5 (Kamal & Burke, 1996), Fuzzy CNN with 3, 4, 5 and 6 cells found better results than Jacobs (1985) with 3 cells and FACT with 3 cells.

Table 5.45 Work load based problem solutions with Fuzzy CNN according to article performance measures.

<table>
<tr><th>PROB. NO</th><th>SOURCE</th><th># OF PARTS</th><th># OF MACH.</th><th>TOTAL #</th><th>METHOD</th><th># OF CELLS</th><th>ARTICLE PERF. MEA.</th><th>PERF. RESULT</th></tr>
<tr><td rowspan="6">5</td><td rowspan="6">Kamal & Burke (1996)</td><td rowspan="6">20</td><td rowspan="6">24</td><td rowspan="6">44</td><td>Jacobs (1985)</td><td rowspan="6">3</td><td rowspan="3">Number of Shared Machines</td><td>21</td></tr>
<tr><td>FACT</td><td>21</td></tr>
<tr><td>Fuzzy CNN</td><td>15</td></tr>
<tr><td>Jacobs (1985)</td><td rowspan="3">Inter-cellular Movements</td><td>37</td></tr>
<tr><td>FACT</td><td>37</td></tr>
<tr><td>Fuzzy CNN</td><td>32</td></tr>
</table>

Table 5.46 Work load based problem solutions with Fuzzy CNN according to proposed performance measure for different numbers of cells.

<table>
<tr><th rowspan="2">PROB. NO</th><th rowspan="2">SOURCE</th><th rowspan="2"># OF PARTS</th><th rowspan="2"># OF MACH.</th><th rowspan="2">TOTAL #</th><th rowspan="2">METHOD</th><th colspan="5">PROPOSED PERFORMANCE MEASURE RESULTS</th></tr>
<tr><th>2 CELLS</th><th>3 CELLS</th><th>4 CELLS</th><th>5 CELLS</th><th>6 CELLS</th></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Liao, Chen, Chen & Coates (1996)</td><td rowspan="2">9</td><td rowspan="2">9</td><td rowspan="2">18</td><td>Fuzzy ROC</td><td>-</td><td rowspan="2">0.7948</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Fuzzy CNN</td><td>0.6559</td><td>0.8082</td><td>-</td><td>-</td></tr>
<tr><td rowspan="3">5</td><td rowspan="3">Kamal & Burke (1996)</td><td rowspan="3">20</td><td rowspan="3">24</td><td rowspan="3">44</td><td>Jacobs (1985)</td><td>-</td><td>0.4992</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>FACT</td><td>-</td><td>0.5177</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Fuzzy CNN</td><td>-</td><td>0.5349</td><td>0.5514</td><td>0.5667</td><td>0.5684</td></tr>
</table>

Overall, 6 nonbinary problem sets are solved by proposed Fuzzy CNN. Because the same problem sets solved by Fuzzy SOM are used in the Fuzzy CNN application, totally 8 solutions were studied. The solutions are compared with the solutions by Fuzzy CNN according to the performance measures used for Fuzzy SOM comparisons. The comparisons showed that for one solution, Fuzzy CNN found better results according to all performance measures and article performance measures (number of shared machines and inter-cellular movements) than the methods (FACT and Jacobs (1985)) used in the article (Kamal & Burke, 1996). Fuzzy CNN found the same results with the methods used in the data set 1-Fuzzy ART (Peker & Kara, 2004), data set 2-Fuzzy ROC (Liao, Chen, Chen & Coates, 1996) and data set 6- FACT and Genetic Algorithm (Kamal & Burke, 1996). When Fuzzy CNN is compared with FuzzyART/RRR-RSS for data set 3 (Won & Currie, 2007), it is found to get worse results than according to GE, GEA, NGM, GCI, proposed performance measure, weighted GCI (the article performance measure) and a better result according to MU. For data set 4 (Kamal & Burke, 1996), the results by Fuzzy CNN is compared with the results by FACT, and it is seen that Fuzzy CNN found worse results according to GEA, NGM, GCI, proposed performance measure, number of shared machines and inter-cellular movements (the article performance measures) and better results according to GE and MU. For 4 of 6 the nonbinary problems, Fuzzy CNN found better results than the results in the articles with using different numbers of cells. As a conclusion, Fuzzy CNN within the proposed methodology can be used as a good ANN method to solve the small and medium sized nonbinary CF problems according to all performance measures.

5.3.5 Discussion

In the nonbinary cases section, 6 nonbinary problem sets are solved by Fuzzy SOM and Fuzzy CNN within the proposed methodology. According to the results and the comparisons, there can be five subjects to discuss.

The first subject is that "Can Fuzzy SOM solve nonbinary CF problems effectively?" This question is investigated by testing if Fuzzy SOM is superior to the other CF methods. The comparisons of 6 nonbinary problem sets and 8 different solutions by the methods showed that Fuzzy SOM found the same results with 4 CF methods in 3 data sets and better results than 2 CF methods in one data set regardless of the type of the performance measure used. However, Fuzzy SOM found worse results than 2 CF methods in 2 data sets. These findings leads to the result that our Fuzzy SOM can be used as well as other CF methods to solve small and medium sized (according to number of parts and machines) CF problems.

The second subject is that "Can Fuzzy CNN solve nonbinary CF problems effectively?" When the question is tested, the comparisons revealed that Fuzzy CNN found the same results with 4 CF methods in 3 data sets. Fuzzy CNN reached better results than 2 CF methods in one data set regardless of the type of the performance measure used and it found worse results than 2 CF methods in 2 data sets. These findings supports the idea that Fuzzy CNN within the proposed methodology is a good method for solving small and medium sized (according to number of parts and machines) CF problems.

The third subject is that “Is Fuzzy SOM a better CF method compared to Fuzzy CNN in nonbinary cases?” When the results by Fuzzy SOM within the proposed methodology compared to the results by Fuzzy CNN within the proposed methodology for nonbinary problems, Fuzzy SOM is found to get better results than Fuzzy CNN in data set 4 and Fuzzy CNN is found to get better results than Fuzzy SOM in data set 5. For the rest 4 sets, Fuzzy SOM got the same results with Fuzzy CNN. Both methods have almost the same effectiveness in solving small and medium sized nonbinary problems according to these data.

The fourth subject is that “Is the proposed performance measure worth being used by itself instead of performance measures in the literature including the five performance measures composing the proposed performance measure?” With the new data provided from the nonbinary solutions, this question should be discussed again in this section. When the five performance measures are compared with each other in terms of their result values, it is found that they measure different values almost in each nonbinary problem set for the same result matrix. The inconsistency of the performance measures suggests that they should not be used by themselves in studies because they measure different values and evaluate different methods as better performing in the same result matrices. This is thought to result from the formulas of five performance measures that include different basic parameters. As a final conclusion reached by evaluating the data from both binary and nonbinary problem sets, the variety of the result values found by the five performance measures and the consistency of the results by the proposed performance measure and the five performance measures in finding better values support the claim of the present study about the superiority of the proposed performance measure to its dimensions.

As the last subject of discussion, the question "Does redefining number of cells when solving CF problems have any advantages in getting better results?" asked in the discussion of binary cases, is asked again to investigate for additional information. As mentioned in the discussion of binary cases, redefining numbers of cells turns into an advantage when a problem has not been solved with the optimal number of cells. When the 6 nonbinary problems are solved with trying different numbers of cells, better results (better cell configurations) than the results found in the articles are provided for 4 of the problems with increasing the numbers of the cells. By this data, it can be claimed that redefining number of cells when solving CF problems can be more efficient and provide better cell configurations.

CHAPTER SIX
CONCLUSION

GT is a manufacturing principle which brings the change in a factory from process organization to product organization and also from process layout to product layout. These changes significantly simplify material flows in a factory. GT promises many advantages. They include reductions in production lead time, work in process, labor, tooling rework and scrapt materials, setup time, order time delivery and paper work. Therefore, GT is seen as a way to solve many problems faced by today's manufacturing industry.

The initial step in applying GT to manufacturing is the identification of part families and/or the formation of machine cells. Parts of similar design features and/or machining operations are grouped into families. To each family, machines of different types are allocated to ensure efficient operations. This is referred CF problem.

Several kinds of methods can be used in solving CF problem. ANNs are very suitable in CF and have been widely applied in CF due to their robust and adaptive nature. The main aim of the present thesis is to implement two ANNs (SOM and CNN) to the CF problem using binary and nonbinary (fuzzy) inputs. The study to realize this aim is summarized below:

Chapter One consists of an introduction with a brief description of the CF problem and described the motivation of the thesis and scopes of the study. Chapter Two detailly defines GT, CMSs and CF problem. In addition, it presents the traditional manufacturing systems, CMSs, the advantages/disanvantages of CM, CF methods and performance measures of cell groupings used in the literature. Chapter Three explains ANNs application in CM. In the chapter, AI, the connection between human brain and ANNs, history of ANNs, the advantages/disanvantages of ANNs, ANNs applications, architecture of ANNs, learning types of ANNs, types of ANNs, and a detailed literature survey on CF with ANNs are covered.

Chapter Four suggests two methodologies for CF with binary and nonbinary (fuzzy) inputs. The methodologies include 6 steps: The binary problem set matrix is determined; it is transformed into a binary input matrix determining the cells by grouping machines and parts simultaneously; ANNs (SOM or CNN) are generated with codes; binary input matrix is used to train the networks; networks transformed the binary input matrix into an output matrix by using MATLAB; the output matrix is transformed into a result matrix presenting the cells that provide a solution for the problem. In the chapter, all of the steps are presented with examples. A proposed performance measure composed by five well-known performance measures aggregating the effectiveness of its dimensions is also presented in this chapter.

Chapter Five includes applications of binary and nonbinary cases. The application is done in two steps. The first step is testing the performance of the proposed methodology by using solved binary inputs from the litarature. The previous result found for each problem and the result by the proposed method are compared. The second step is using the new methodology for nonbinary (fuzzy) inputs. The same procedure applied for the binary problems is implemented for nonbinary problems from the literature. The results are presented and discussed.

According to results, the comparisons of SOM, CNN, Fuzzy SOM and Fuzzy CNN with other methods, SOM with CNN, and Fuzzy SOM with Fuzzy CNN are summarized in tables (Table 6.1 – Table 6.6) below.

Table 6.1 The comparison of SOM solutions.

PERF. MEA. TYPE	**# OF SETS**	**# OF SOLUTIONS**	**SOM > OTHER METHODS**	**SOM = OTHER METHODS**	**SOM < OTHER METHODS**
Prop.	15	18	4	14	0
Article	10	19	8	11	0

Table 6.2 The comparison of CNN solutions.

PERF. MEA. TYPE	# OF SETS	# OF SOLUTIONS	CNN > OTHER METHODS	CNN = OTHER METHODS	CNN < OTHER METHODS
Prop.	15	18	3	13	2
Article	10	19	8	10	1

Table 6.3 The comparison of SOM with CNN solutions.

PERF. MEA. TYPE	# OF SETS	# OF SOLUTIONS	SOM > CNN	SOM = CNN	SOM < CNN
Prop.	15	17	2	15	0
Article	10	13	1	11	1

Table 6.4 The comparison of Fuzzy SOM solutions.

PERF. MEA. TYPE	# OF SETS	# OF SOLUTIONS	FUZZY SOM > OTHER METHODS	FUZZY SOM = OTHER METHODS	FUZZY SOM < OTHER METHODS
Prop.	6	8	2	4	2
Article	6	11	4	4	3

Table 6.5 The comparison of Fuzzy CNN solutions.

PERF. MEA. TYPE	# OF SETS	# OF SOLUTIONS	FUZZY CNN > OTHER METHODS	FUZZY CNN = OTHER METHODS	FUZZY CNN < OTHER METHODS
Prop.	6	8	2	4	2
Article	6	11	4	4	3

Table 6.6 The comparison of Fuzzy SOM with Fuzzy CNN solutions.

PERF. MEA. TYPE	# OF SETS	# OF SOLUTIONS	FUZZY SOM > FUZZY CNN	FUZZY SOM = FUZZY CNN	FUZZY SOM < FUZZY CNN
Prop.	6	6	1	4	1
Article	6	7	0	3	4

By these findings, it can be claimed that SOM within the proposed methodology can be used as an effective ANN method to solve small and medium sized CF problems. CNN is also a good ANN method to solve CF problem. However, SOM has a better performance in solving small and medium sized binary problem sets. Fuzzy SOM and Fuzzy CNN can be used to solve CF problems as well as other CF methods.

There are two more conclusions to be reached in addition to the information given above. Conclusion one is the proposed performance measure is superior to its dimensions. That is because performance measures used in the literature give inconsistent results measuring different values and evaluating different methods as better performing for the same result matrices. Conclusion two is redefining number of cells when solving CF problems can be more efficient and provide better cell configurations.

For future research, the effectiveness of SOM, CNN, Fuzzy SOM and Fuzzy CNN within the proposed methodologies in solving large sized CF problems still remains as a subject to study. In the literature, hybrid models which are more close to real world conditions are suggested for problems in all sizes and in all input types.

REFERENCES

Abduelmola, A. I. (2000). *Modeling of cellular manufacturlng systems with productivity consideration: A simulated annealing algorithm.* Ph.D. Thesis, University of Windsor, The Department of Industrial & Manufacturing Systems Engineering.

Abdelmola, I. & Taboun, S. M. (1999). Productivity model for the cell formation problem: A simulated annealing algorithm. *Computers & Industrial Engineering,* 37, 327-330.

Adeli, H. & Park, H.S. (1998). *Neurocomputing for design automation.* Florida: CRC Press.

Albadawia, Z., Bashirb, H. A. & Chen, M. (2005). A mathematical approach for the formation of manufacturing cells. *Computers & Industrial Engineering,* 48, 3–21.

Aljaber, N., Baek, W. & Chen, C. L. (1997). A tabu search approach to the cell formation problem. *Computers & Industrial Engineering,* 32 (1), 169-185.

Aljaber, N. J. (1999). *New approaches to cell formation problems in cellular manufacturing systems.* Ph.D. Thesis, Mississippi State University, The Department of Industrial Engineering.

Altınkılınç, M. (2000). *A simulation based comparison of functional and cellular layouts in a manufacturing plant.* Master's Degree Thesis, Dokuz Eylul University, The Department of Industrial Engineering.

Ampazis, N. & Minis, I. (2004). Design of cellular manufacturing systems using latent semantic indexing and self organizing maps. *Computational Management Science,* 1, 275-292.

Anderson, D. & McNeill, G. (1992). Artificial Neural Networks technology. *Data & Analysis Center for Software (DACS) Report, ELIN: A011,* 10.

Askin, R. & Chiu, K. (1990). A graph partitioning procedure for machine assignment and cell formation in group technology. *International Journal of Production Research,* 28 (8), 1555-1572.

Askin, R. & Standridge, C. (1993). *Modeling and analysis of manufacturing systems.* New York: John Wiley & Sons.

Askin, R., Selim, H. & Vakharia, A. (1997). A methodology for designing flexible cellular manufacturing systems. *IIE Transactions,* 29 (7), 599-610.

Asokan, P., Prabhakaran, G. & Satheesh Kumar, G. (2001). Machine-cell grouping in cellular manufacturing systems using non-traditional optimisation techniques – A comparative study. *International Journal of Advanced Manufacturing Technology,* 18, 140–147.

Baese, A. M., Thummler, V. & Theis, F. (2006). Stability analysis of an unsupervised competitive neural network. *International Joint Conference on Neural Networks,* July 16-21, Canada.

Basu, A., Hyer, N. & Shtub, A. (1995). An expert system based approach to manufacturing cell design. *International Journal of Production Research,* 33 (10), 2739-2755.

Bender, E. A. (1996). *Mathematical methods in artificial intelligence*, Los Alamitos: IEEE Computer Society Press.

Bidanda, B., Ariyawongrat, P., Needy, K. L., Norman, B. A. & Tharmmaphornphilas, W. (2005). Human related issues in manufacturing cell design, implementation, and operation: a review and survey. *Computers & Industrial Engineering,* 48, 507–523.

Bigus, J. P. (1996). *Data mining with neural networks.* New York: McGraw-Hill Companies.

Boulif, M. & Atif, K. (2006). A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. *Computers & Operations Research,* 33, 2219–2245.

Burbidge, J. L. (1963). Production flow analysis. *Production Engineer,* 42, 472.

Burbidge, J. L. (1992). Change to group technology: Process organization is obsolete, *International Journal of Production Research,* 30 (5), 1209–1219.

Cao, D. & Chen, M. (2004). Using penalty function and tabu search to solve cell formation problems with fixed cell cost. *Computers & Operations Research,* 31, 21–37.

Car, Z. & Mikac, T. (2006). Evolutionary approach for solving cell-formation problem in cell manufacturing. *Advanced Engineering Informatics,* 20, 227–232.

Carrie, A. S. (1973). Numerical taxonomy applied to group technology and plant layout. *International Journal of Production Research,* 11 (4), 399-416.

Chan, F. T. S., Lau, K. W., Chan, P. L. Y. & Choy, K. L. (2006). Two-stage approach for machine-part grouping and cell layout problems. Robotics and Computer-Integrated Manufacturing, 22, 217–238.

Chandrasekharan, M. P. & Rajagopalan, R. (1986a). An ideal seed non-hierarchical clustering algorithm for cellular manufacturing. *International Journal of Production Research,* 24, 451-464.

Chandrasekharan, M. P. & Rajagopalan, R. (1986b). MODROC: An extension of rank order clustering for group technology. *International Journal of Production Research,* 24, 1221-1233.

Chen, S. J. & Cheng, C. S. (1995). A neural network-based cell formation algorithm in cellular manufacturing. *International Journal of Production Research,* 33 (2), 293-318.

Chen, M. (1998). A mathematical programming model for system reconfiguration in a dynamic cellular manufacturing environment. *Annals of Operations Research,* 77, 109-128.

Chen, M. L., Wu, C. M. & Chen, C. L. (2002). An integrated approach of ART1 and tabu search to solve cell formation problems. *Journal of the Chinese Institute of Industrial Engineers,* 19 (3), 62-74.

Choi, M. (1992). Manufacturing cell design. *Production and Inventory Management Journal,* 33 (2), 66-69.

Christodoulou, M. & Gaganis, V. (1998). Neural networks in manufacturing cell design. *Computers in Industry,* 36, 133–138.

Chu, C. H. (1997). An improved neural network for manufacturing cell formation. *Decision Support Systems,* 20, 279-295.

ChunHung, C. (1990). *An investigation of algorithms for solving the group technology problem.* Ph.D. Thesis, The University of Iowa, The Department of Business Administration.

Collet, S. & Spicer, R. (1995). Improving productivity through cellular manufacturing. *Production and Inventory Management Journal,* 36 (1), 71-75.

Coolen, A. C. C., Kühn, R. & Sollich, P. (2005). *Theory of neural information processing systems.* New York: Oxford University Press.

Dagli, C. & Huggahalli, R. (1995). Machine-part family formation with the adaptive resonance theory paradigm. *International Journal of Production Research,* 33, 893–913.

Dasa, K., Lashkaria, R. S. & Sengupta, S. (2007). Reliability consideration in the design and analysis of cellular manufacturing systems. *International Journal of Production Economics,* 105, 243–262.

De Lit, P., Falkenauer, E. & Delchambre, A. (2000). Grouping genetic algorithms: an efficient method to solve the cell formation problem. Mathematics and Computers in Simulation, 51, 257–271.

Defersha, F. M. & Chen, M. (2006). A comprehensive mathematical model for the design of cellular manufacturing systems. *International Journal of Production Economics,* 103, 767–783.

Defersha, F. M. (2006). *An integrated approach to the design of cellular manufacturing systems for dynamic production requirements.* Ph.D. Thesis, Concordia University, The Department of Mechanical and Industrial Engineering.

Diaz, B. A., Lozano, S., Racero, J. & Guerrero, F. (2001). Machine cell formation in generalized group technology. *Computers & Industrial Engineering,* 41, 227-240.

Diaz, B. A., Lozano, S. & Eguia, I. (2005). Part-machine grouping using weighted similarity coefficients. *Computers & Industrial Engineering,* 48, 553–570.

Dobado, D., Lozano, S., Bueno, J. M. & Larraneta, J. (2002). Cell formation using a fuzzy min-max neural network. *International Journal of Production Research,* 40 (1), 93-107.

Eski, Ö. (2007). *Design of manufacturing cells for uncertain production requirements.* Ph.D. Thesis, Dokuz Eylul University, The Department of Industrial Engineering.

Foulds, L. R. & Wilson, J. M. (2002). Approaches to the general cell formation problem. *Loughborough University Business School Research Series,* 2002 (2), 3-4.

Fraser, K., Harris, H. & Luong, L. (2007a). Team-based cellular manufacturing: A review and survey to identify important social factors. *Journal of Manufacturing Technology Management,* 18 (6), 714-730.

Fraser, K., Harris, H. & Luong, L. (2007b). Improving the implementation effectiveness of cellular manufacturing: A comprehensive framework for practitioners. *Journal of Manufacturing Technology Management,* 45 (24), 5835-5856.

Freeman, J. A. & Skapura, D. M. (1991). *Neural networks : Algorithms, applications and programming techniques.* Teksas: Addison-Wesley Publishing Company.

Fu, L. (1994). *Neural networks in computer intelligence.* Singapore: McGraw-Hill.

Goncalves, J. F. & Resende, M. G. C. (2004). An evolutionary algorithm for manufacturing cell formation. *Computers & Industrial Engineering,* 47, 247–273.

Guerrero, F., Lozano, S., Smith, K. A., Canca, D. & Kwok, T. (2002). Manufacturing cell formation using a new self organizing neural network. *Computers & Industrial Engineering,* 42, 377-382.

Gupta, T. & Saifoddini, H. (1990). Production data based similarity coefficient for machinecomponent grouping decisions in the design of a cellular manufacturing system, *International Journal of Production Research,* 28, 1247-1269.

Hachicha, W., Masmoudi, F. & Haddar, M. (2007). An improvement of a cellular manufacturing system design using simulation analysis. *Unit of mechanic, modelling and production, engineering school of Sfax Tunisia,* 8922, 1-2.

Hadley, S. W. (1996). Finding part-machine families using graph partitioning techniques. *International Journal of Production Research,* 34, 1821-1839.

Ham, I., Hitomi, K. & Yoshida, T. (1985). *Group Technology : Applications to production management.* Dordrecht: Kluwer-Nijhoff Publishing.

Harhalakis, J., Nagi, R. & Proth, J. M. (1990). An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research*, 28, 185-198.

Hawkins, J. & Blakeslee, S. (2004). *On intelligence.* New York: Henry Holt and Company.

Haykin, S. (1994). *Neural networks: A comprehensive foundation.* New Jersey: Macmillan Publishing.

Hayret, T. (2000). *A compasion study of cellular manufacturing and job-shop manufacturing systems using simulation.* Master's Degree Thesis, Dokuz Eylul University, The Department of Industrial Engineering.

Heaton, J. (2005). *Introduction to neural networks with java.* (1st ed.) Missouri: Heaton Research.

Hopgood, A. A. (2001). *Intelligent systems for engineers and scientists.* (2nd ed.) Florida: CRC Press.

Howard, M. & Newman, R. (1993). From job shop to just-in-time-a successful conversion. *Production and Inventory Management Journal,* 34 (3), 70-74.

Hsu, C. P. (1990). *Similarity coefficient approaches to machine-component cell formation in cellular manufacturing: A comparative study.* Ph.D. Thesis, University of Wisconsin, Department of Industrial and Systems Engineering.

Hsu, C. M. & Su, C. T. (1998). Multi-objective machine-component grouping in cellular manufacturing: a genetic algorithm. *Production Planning & Control,* 9 (2), 155-166.

Hu, Y. H. & Hwang, J. (2002). Introduction to neural networks for signal processing. Y. H. Hu & J. Hwang, (Ed.). *Handbook of neural network signal processing.* Florida: CRC Press.

Hyer, N. L. & Wemmerlov, U. (2002). *Reorganizing the factory: Competing through cellular manufacturing*. Portland: Productivity Press.

Jain, L. C. & Martin, N. M. (1998). *Fusion of neural networks, fuzzy systems and genetic algorithms: Industrial applications.* Florida: CRC Press.

Jaganathan, J. K. J. (2007). *Solution to large facility layout problems using group technology.* Master's Degree Thesis, Wichita State University, The Department of Industrial and Manufacturing Engineering.

Jeon, G. & Leep, H. R. (2006). Forming part families by using genetic algorithm and designing machine cells under demand changes. *Computers & Operations Research,* 33, 263–283.

Johnson, D. J. & Wemmerlov, U. (2004). Why does cell implementation stop? Factors influencing cell penetration in manufacturing plants. *Production and Operations Management,* 13 (3), 272-89.

Joines, J. A. (1996). *Hybrid genetic search for manufacturing cell design.* Ph.D. Thesis, North Carolina State University, The Department of Industrial Engineering.

Jordan, M. I. & Russell, S. (1999). Computational intelligence. R. A. Wilson & F. C. Keil, (Ed.) *The MIT encyclopedia of the cognitive sciences.* London: The MIT Press.

Kamal, S. & Burke, L. I. (1996). FACT : A new neural network-based clustering algorithm for group technology. *International Journal of Production Research,* 34 (4), 919-946.

Kamruzzaman, J., Sarker, R. A. & Begg, R. (2006). Artificial neural networks: Applications in finance and manufacturing. J. Kamruzzaman, R. A. Sarker & R. Begg, (Ed.). *Artificial neural networks: Applications in finance and manufacturing.* London: Idea Group Publishing.

Kandiller, L. (1994). A comparative study of cell formation in cellular manufacturing systems. *International Journal of Production Research,* 32, 2395-2429.

Karr, C. (1999). *Practical applications of computational intelligence for adaptive control.* Florida: CRC Press.

Kasabov, N. K. (1998). *Foundations of neural networks, fuzzy systems, and knowledge engineering.* (2nd ed.) London: The MIT Press.

King, J. (1980). Machine-component grouping in production flow analysis: An approach using a rank order clustering algorithm. *International Journal of Production Research,* 18 (2), 213-232.

King, J. & Nakornchai, V. (1982). Machine-component group formation in group technology: Review and extension. *International Journal of Production Research,* 20 (2), 117-133.

Kioon, S. A. (2007). *Comprehensive models and solution procedures for integrated cellular manufacturing systems design.* Ph.D. Thesis, Concordia University, The Department of Mechanical and Industrial Engineering.

Kioon, S. A., Bulgak, A. A. & Bektas, T. (2009). Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration. *European Journal of Operational Research,* 192, 414–428.

Kohonen, T. & Simula, O. (1996). Engineering applications of the Self Organizing Map. *Proceedings of The Institute of Electrical and Electronics Engineers,* 84 (10), 1358-1384.

Konar, A. (2000). *Artificial intelligence and soft computing : Behavioral and cognitive modeling of the human brain.* Florida: CRC Press.

Krishnamoorthy, C. S. & Rajeev, S. (1996). *Artificial intelligence and expert systems for engineers.* Florida: CRC Press.

Kröse, B. & Smagt, P.v.d. (1996). *An introduction to neural networks.* (8th ed.) Amsterdam: The University of Amsterdam.

Kumar, C. S. & Chandrasekharan, M. P. (1990). Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology. *International Journal of Production Research,* 28 (2), 233-243.

Kuo, R. J., Chi, S. C. & Teng, P. W. (2001). Generalized part family formation through fuzzy self organizing feature map neural network. *Computers & Industrial Engineering,* 40, 79-100.

Kusiak, A. & Lee, H. (1996). Neural computing-based design of components for cellular manufacturing. *International Journal of Production Research,* 34 (7), 1777-1790.

Lee, K. M., Yamakawa, T. & Lee, K. M. (1997). Machine-part grouping for cellular manufacturing systems : A neural network approach. *Institute of Electrical and Electronics Engineers,* 575-580.

Levasseur, G., Helms, M. & Zink, A. (1995). A conversion from a functional to a cellular manufacturing layout at Steward, Inc. *Production and Inventory Management Journal,* 36 (3), 37-42.

Liang, M. & Zolfaghari, S. (1999). Machine cell formation considering processing times and machine capacities: An ortho-synapse hopfield neural network approach. *Journal of Intelligent Manufacturing,* 10, 437-447.

Liao, T. W. (1994). Design of line-type cellular manufacturing systems for minimum operating and material-handling costs. *International Journal of Production Research,* 32 (2), 387-397.

Liao, T. W., Chen, L. J., Chen, Z. H. & Coates, E. R. (1996). A comparison of two approaches for designing line type cellular manufacturing systems. *Integrated Manufacturing Systems,* 7 (1), 6–15.

Lisboa, P. G. J. (1992). *Neural networks: Current applications.* (1st ed.). London: Chapman & Hall.

Liu, T. (1999). Tools for intelligent manufacturing processes and systems: Neural networks, fuzzy logic, and expert systems. F. Kreith, (Ed.). *Mechanical engineering handbook.* Florida: CRC Press.

Lozano, S., Canca, D., Guerrero, F. & Garcia, J. M. (2001). Machine grouping using sequence-based similarity coefficients and neural networks. *Robotics and Computer Integrated Manufacturing,* 17, 399-404.

Mahdavi, I., Kaushal, O. P. & Chandra, M. (2001). Graph-neural network approach in cellular manufacturing on the basis of a binary system. *International Journal of Production Research,* 39 (13), 2913-2922.

Mahdavi, I., Javadi, B., Fallah-Alipour, K. & Slomp, J. (2007). Designing a new mathematical model for cellular manufacturing system based on cell utilization. *Applied Mathematics and Computation,* 190, 662–670.

Mahdavi, I., Paydar, M. M., Solimanpur, M. & Heidarzade, A. (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems with Applications,* 36, 6598–6604.

Malakooti, B. & Yang, Z. (1994). An unsupervised neural network approach for machine-part cell design. *Institute of Electrical and Electronics Engineers,* 665-670.

Mandic, D. P. & Chambers, J. A. (2001). *Recurrent neural networks for prediction.* West Sussex: John Wiley & Sons Ltd.

Mansouri, S. A., Moattar Husseini, S. M. & Newman, S. T. (2000). A review of the modern approaches to multi-criteria cell design. *International Journal of Production Research,* 38 (5), 1201-1218.

Marsh, R. F. (1993). *The lif cycle of manufacturing cells.* Ph.D. Thesis, University of Cincinnati, The Department of Operations Management.

McClelland, J. L., Rumelhart, D. E. & The PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 2: Psychological and Biological Models,* London: MIT Press.

Meents, I. (2001). A genetic algorithm for the group technology problem. EvoWorkshop 2001, 2037, 90-99.

Mehrabad, M. S. & Safaei, N. (2007). A new model of dynamic cell formation by a neural approach. *International Journal of Advanced Manufacturing Technology,* 33, 1001–1009.

Michie, D., Spiegelhalter, D. J. & Taylor, C. C. (1994). Introduction. D. Michie, D. J. Spiegelhalter & C. C. Taylor, (Ed.). *Machine learning, neural and statistical classification.* West Sussex: Horwood Publishing.

Miltenburg, J. & Zhang, W. (1991). A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology. *Journal of Operation Management*, 10 (1), 44-72.

Moghaddam, R. T., Aryanezhad, M. B., Safaei, N., Vasei, M. & Azaron, A. (2007). A new approach for the cellular manufacturing problem in fuzzy dynamic conditions by a genetic algorithm. *Journal of Intelligent & Fuzzy Systems*, 18, 363–376.

Moghaddam, R. T., Vahed, A. R. R., Ghodratnama, A. & Siadat, A. (2009). A simulated annealing method for solving a new mathematical model of a multi-criteria cell formation problem with capital constraints. *Advances in Engineering Software,* 40, 268–273.

Mosier, C. T., Yelle, J. & Walker, G. (1997). Survey of similarity coefficient based methods as applied to the group technology configuration problem. *Internatiomal Journal of Management Sciences,* 25 (1), 65-79.

Mukattash, A. M. & Al-Tahat, M. D. (2006). A developed mathematical computerized approach for designing cellular maanufacturing systems. *Journal of Applied Sciences.* 6 (9), 1915-1923.

Mungwattana, A. (2000). *Design of cellular manufacturing systems for dynamic and uncertain production requirements with presence of routing flexibility.* Ph.D. Thesis, Virginia Polytechnic Institute and State University, The Department of Industrial and Systems Engineering.

Murugan, M. & Selladurai, V. (2007). Optimization and implementation of cellular manufacturing system in a pump industry using three cell formation algorithms. *International Journal of Advanced Manufacturing Technology,* 35, 135–149.

Nair, G. J. K. & Narendran, T. T. (1996). Grouping index: A new quantitative criterion for goodness of block diagonal forms in group technology. *International Journal of Production Research*, 34 (10), 2767-2782.

Neelakanta, P. S. (1999). *Information - Theoretic aspects of neural networks.* Florida: CRC Press.

Neelakanta, P. S. & DeGroff, D. (1994). *Neural network modeling: Statistical mechanics and cybernetic perspectives.* Florida: CRC Press.

Nguema, B. A. & Dao, T. M. (2009). Quantized hopfield networks and tabu search for manufacturing cell formation problems. International Journal of Production Economics, 121, 88–98.

Nilsson, N. J. (1971). *Problem solving methods in artificial intelligence.* New York: McGraw-Hill Companies.

Nomden, G., Slomp, J. & Suresh, N. C. (2006). Virtual manufacturing cells: A taxonomy of past research and identification of future research issues. *International Journal of Flexible Manufacturing Systems,* 17, 71–92.

Oliveira, S., Ribeiro, J. F .F. & Seok, S. C. (2008). A comparative study of similarity measures for manufacturing cell formation. *Journal of Manufacturing Systems,* 27, 19-25.

Onwubolu, G. C. (1999). Design of parts for cellular manufacturing using neural network-based approach. *Journal of Intelligent Manufacturing,* 10, 251-265.

Onwubolu, G. C. & Mutingi, M. (2001). A genetic algorithm approach to cellular manufacturing. *Computers & Industrial Engineering,* 39, 125-144.

Ozturk, G., Ozturk, Z. K. & Islier, A. A. (2006). A comparison of competitive neural network with other AI techniques in manufacturing cell formation. *International Conference on Neurocybernetics,* 575–583.

Pandya, A. S. & Macy, R. B. (1995). *Pattern recognition with neural networks in C++.* Florida: CRC Press.

Park, S. & Suresh, N. C. (2003). Performance of fuzzy ART neural network and hierarchical clustering for part–machine grouping based on operation sequences. *International Journal of Production Research,* 41 (14), 3185–3216.

Peker, A. & Kara, Y. (2004). Parameter setting of the fuzzy ART neural network to part–machine cell formation problem. *International Journal of Production Research,* 42 (6), 1257–1278.

Pham, P. T. N. & Pham, D. T. (2001). Computational intelligence for manufacturing. J. Wang & A. Kusiak, (Ed.). *Computational intelligence in manufacturing handbook.* Florida: CRC Press.

Plaquin, M. F. & Pierreval, H. (2000). Cell formation using evolutionary algorithms with certain constraints. *International Journal of Production Economics,* 64, 267-278.

Rajagopalan, R. & Barta, J. (1975). Design of cellular production system: A graphic-theoretic approach. *International Journal of Production Research,* 13 (6), 567-579.

Rao, H. A. & Gu, P. (1993). Multi-layered self organizing neural network for machine clustering. *Institute of Electrical and Electronics Engineers,* 738-741.

Rao, H. A. & Gu, P. (1995). A multi-constraint neural network for the pragmatic design of cellular manufacturing systems. *International Journal of Production Research,* 33 (4), 1049-1070.

Rao, V. B. & Rao, H. (1995). *C++ Neural networks and fuzzy logic.* California: M&T Books.

Rao, M. A., Rao, M. S., Srinivas, J. & Krishna, P. V. G. (2001). Recognition of machine cells in a group technology layout using self-organizing neural networks. *Proceedings of the Institution of Mechanical Engineers,* 215 (Part B), 897-901.

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

Rogers, D. F. & Shafer, S. M. (1995). Measuring cellular manufacturing performance. A. K. Kamrani, H. R. Parsaei & D. H. Liles (Ed.) *Planning, design, and analysis of cellular manufacturing systems,* The Netherlands: Elsevier Publishing.

Rogers, D. F. & Kulkarni, S. S. (2005). Optimal bivariate clustering and a genetic algorithm with an application in cellular manufacturing. *European Journal of Operational Research,* 160, 423–444.

Russel, S. & Norvig, P. (1995). *Artificial intelligence: A modern approach*, New Jersey: Prentice-Hall.

Safaei, N., Mehrabad, M. S. & Ameli, M. S. J. (2008). A hybrid simulated annealing for solving an extended model of dynamic cellular manufacturing system. *European Journal of Operational Research.* 185, 563–592.

Sarker, B. R. & Mondal, S. (1999). Grouping efficiency measures in cellular manufacturing: A survey and critical review. *International Journal of Production Research,* 37 (2), 285–314.

Sarker, B. R. & Xu, Y. (1998). Operation sequences-based cell formation methods: A critical survey. *Production Planning & Control,* 9 (8), 771-783.

Schaller, J. (2005). Tabu search procedures for the cell formation problem with intra-cell transfer costs as a function of cell size. Computers & Industrial Engineering, 49, 449–462.

Seifoddini, H. & Djassemi, M. (1996). The threshold value of a quality index for formation of cellular manufacturing systems. *International Journal of Production Research*, 34 (12), 3401-3416.

Selim, H. M. (2000). Manufacturing cell formation problem: A graph partitioning approach. *Seventh Cairo University International MDP Conference,* Cairo, February 15-17.

Selim, H. M., Askin, R. & Vakharia, A. (1998). Cell formation in group technology: Review, evaluation and directions for future research. *Computers and Industrial Engineering.* 34 (l), 3-20.

Shambu, G., Suresh, N. C. & Pegels, C. C. (1996). Performance evaluation of cellular manufacturing systems: A taxonomy and review of research. *International Journal of Operations & Production Management,* 16 (8), 81-103.

Sharif, H. H., El-Kilany, K. S. & Helaly, M. A. (2008). A genetic algorithm approach to the group technology problem. Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 19-21 March.

Skinner, W. (1974). The focused factory. *Harvard Business Review,* 113-121.

Smith, S. D. G. & Escobedo, R. A. (1994). Engineering and manufacturing applications of ART-1 neural networks. *Institute of Electrical and Electronics Engineers,* 3780-3785.

Soleymanpour, M., Vrat, P. & Shankar, R. (2002). A transiently chaotic neural network approach to the design of cellular manufacturing. *International Journal of Production Research,* 40 (10), 2225-2244.

Solimanpur, M., Vrat, P. & Shankar, R. (2004). Feasibility and robustness of transiently chaotic neural networks applied to the cell formation problem. *International Journal of Production Research,* 42 (6), 1065–1082.

Stalk, N. & Hout, T. (1990). *Competing against time.* New York: The Free Press.

Steels, L. (1998). The artificial life roots of artificial intelligence. C. G. Langton, (Ed.). *Artificial life.* London: The MIT Press.

Suresh, N. C. (2001). Neural network applications for group technology and cellular manufacturing. J. Wang & A. Kusiak, (Ed.). *Computational intelligence in manufacturing handbook.* Florida: CRC Press.

Suresh, N. C. (2001). Neural network applications for group technology and cellular manufacturing. J. Wang & A. Kusiak, (Ed.). *Computational intelligence in manufacturing handbook.* Florida: CRC Press.

Suresh, N. C. & Kaparthi, S. (1994). Performance of fuzzy ART neural network for group technology cell formation. *International Journal of Production Research,* 32 (7), 1693-1713.

Suresh, N. C., Slomp, J. & Kaparthi, S. (1999). Sequence-dependent clustering of parts and machines: a fuzzy ART neural network approach. *International Journal of Production Research,* 37 (12), 2793-2816.

Tariq, A., Hussain, I. & Ghafoor, A. (2009). A hybrid genetic algorithm for machine-part grouping. *Computers & Industrial Engineering,* 56, 347–356.

Tateyama, T. & Kawata, S. (2004). Plant layout planning using ART neural networks. *SICE Annual Conference in Sapporo,* Japan, August 4-6, 1863-1868.

Venugopal, V. (1999). Soft-computing-based approaches to the group technology problem: a state-of-the-art review. *International Journal of Production Research,* 37 (14), 3335-3357.

Venugopal, V. & Narendran, T. T. (1994). Machine cell formation through neural network models. *International Journal of Production Research,* 32 (9), 2105-2116.

Vosniakos, G. C., Tsifakis, A. & Benardos, P. (2006). Neural network simulation metamodels and genetic algorithms in analysis and design of manufacturing cells. *International Journal of Advanced Manufacturing Technology*, 29, 541–550.

Wemmerlov, U. & Hyer, N. (1989). Cellular manufacturing in the US industry: A survey of users. *International Journal of Production Research,* 27 (9), 1511-1530.

Wemmerlov, U. & Johnson, D. (1997). Cellular Manufacturing at 46 user plants: Implementation experiences and performance improvements. *International Journal of Production Research,* 35 (1), 29-49.

Willow, C. C. (2002). A feed-forward multilayer neural network for machine cell information in computer integrated manufacturing. *Journal of Intelligent Manufacturing,* 13, 75-87.

Won, Y. & Currie, K. R. (2007). Fuzzy ART/RRR-RSS: a two-phase neural network algorithm for part-machine grouping in cellular manufacturing. *International Journal of Production Research,* 45 (9), 2073–2104.

Wong, B. K., Bodnovich, T. A. & Selvi, Y. (1997). Neural network applications in business: A review and analysis of the literature (1988-95). *Decision Support Systems,* 19, 301-320.

Wu, X., Chu, C. H., Wang, Y. & Yan, W. (2007). A genetic algorithm for cellular manufacturing design and layout. *European Journal of Operational Research.* 181, 156–167.

Xambre, A. R. & Vilarinho, P. M. (2003). A simulated annealing approach for manufacturing cell formation with multiple identical machines. *European Journal of Operational Research.* 151, 434–446.

Xie, X., Hahnloser, R. & Seung, S. (2000). Learning winner-take-all competition between groups of neurons in lateral inhibitory networks. T. K. Leen, T. G. Dietterich & V. Tresp (Ed.). *Advances in neural information processing systems.* Cambridge: MIT Press.

Yang, M. S. & Yang, J. H. (2008). Machine-part cell formation in group technology using a modified ART1 method. *European Journal of Operational Research,* 188, 140–152.

Yin, Y. & Yasuda, K. (2005). Similarity coefficient methods applied to the cell formation problem: A comparative investigation. *Computers & Industrial Engineering,* 48, 471–489.

Yin, Y. & Yasuda, K. (2006). Similarity coefficient methods applied to the cell formation problem: A taxonomy and review. *International Journal of Production Economics,* 101, 329–352.

Yin, Y. (2006). Application similarity coefficient method to cellular manufacturing. V. Kordic, A. Lazinica & M. Merdan, (Ed.). *Manufacturing the future: Concepts, technologies & visions.* Mammendorf: Pro Literatur Verlag.

Zahner, D. A. & Micheli-Tzanakou, E. (2000). Artificial neural networks: Definitions, methods, applications. E. Micheli-Tzanakou, (Ed.). *Supervised and unsupervised pattern recognition: Feature extraction and computational intelligence.* Florida: CRC Press.

Zilouchian, A. & Jamshidi, M. (2001). *Intelligent control systems using soft computing methodologies.* Florida: CRC Press.

Zolfaghari, S. & Liang, M. (1997). An objective-guided ortho-synapse hopfield network approach to machine grouping problems. *International Journal of Production Research,* 35 (10), 2773-2792.

Zolfaghari, S. (1997). *Design and planning for cellular manufacturing: Application of neural networks and advanced search techniques.* Ph.D. Thesis, University of Ottawa, The Department of Mechanical Engineering.

Zolfaghari, S. & Liang, M. (2003). A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes. *Computers & Industrial Engineering,* 45, 713–731.

APPENDIX A1

THE BINARY PROBLEM SETS

		MACHINES			
		1	2	3	4
PARTS	1	1	0	1	0
	2	0	1	0	1
	3	1	1	1	0
	4	1	0	0	0
	5	0	1	0	1

Binary problem set 1.

		MACHINES				
		1	2	3	4	5
PARTS	1	0	0	1	1	0
	2	1	0	1	0	0
	3	0	1	0	1	1
	4	1	0	1	0	1
	5	0	1	0	0	1
	6	0	0	0	1	1
	7	1	0	1	0	0

Binary problem set 2.

		MACHINES								
		1	2	3	4	5	6	7	8	9
PARTS	1	1	1	0	0	1	0	0	0	0
	2	1	1	0	1	0	1	0	0	1
	3	0	0	1	1	0	0	1	1	0
	4	0	0	0	1	1	0	0	1	0
	5	1	0	0	0	1	0	0	1	0
	6	0	1	0	0	0	1	0	0	1
	7	0	0	1	0	0	0	1	1	0
	8	0	0	1	1	1	0	1	1	0
	9	0	1	0	0	0	1	0	0	1

Binary problem set 3.

		MACHINES									
		1	2	3	4	5	6	7	8	9	10
	1	1	1	1	0	0	0	0	0	1	1
	2	1	0	0	0	0	0	0	0	1	1
	3	0	0	0	1	1	0	0	0	1	1
	4	0	0	0	0	0	1	1	1	1	0
PARTS	5	0	0	0	0	0	0	1	1	0	0
	6	1	0	1	0	0	0	0	0	1	1
	7	0	0	0	1	0	0	0	0	1	0
	8	0	0	0	1	1	0	0	0	0	1
	9	0	0	0	0	0	1	1	1	1	0
	10	0	0	0	0	0	0	1	1	0	0

Binary problem set 4.

		MACHINES									
		1	2	3	4	5	6	7	8	9	10
	1	0	0	1	1	0	1	0	0	0	0
	2	1	0	0	0	0	0	1	0	0	1
	3	0	1	0	0	1	0	0	1	0	0
	4	0	0	0	1	0	1	0	0	1	0
	5	0	1	0	0	1	0	0	1	0	0
	6	0	0	1	0	0	1	0	0	1	0
	7	1	0	0	0	0	0	1	0	0	1
PARTS	8	0	0	0	0	1	0	0	1	0	0
	9	0	0	1	1	0	1	0	0	1	0
	10	1	0	0	0	0	0	1	0	0	0
	11	1	0	0	0	0	0	1	0	0	1
	12	1	0	0	0	0	0	1	0	0	1
	13	0	1	0	0	1	0	0	1	0	0
	14	0	0	1	1	0	1	0	0	1	0
	15	0	1	0	0	1	0	0	1	0	0

Binary problem set 5.

		MACHINES									
		1	2	3	4	5	6	7	8	9	10
PARTS	1	1	1	1	0	0	0	0	0	0	1
	2	1	1	0	0	0	0	1	0	0	0
	3	1	1	1	0	0	0	0	0	0	0
	4	1	1	1	0	0	0	0	0	0	1
	5	0	0	0	1	0	1	0	0	0	0
	6	0	0	0	0	1	1	1	0	0	0
	7	0	0	0	1	1	1	0	0	0	0
	8	0	0	0	1	1	1	0	0	0	0
	9	0	0	0	1	1	1	1	0	0	0
	10	0	0	0	1	1	1	0	0	0	0
	11	0	0	0	0	0	0	1	1	1	0
	12	0	1	0	0	0	0	1	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	14	0	1	0	0	0	0	1	1	1	1
	15	0	0	0	0	0	0	1	1	1	1

Binary problem set 6.

		MACHINES									
		1	2	3	4	5	6	7	8	9	10
PARTS	1	0	0	1	1	0	0	0	1	0	0
	2	1	0	0	0	0	1	1	0	0	0
	3	0	0	1	1	0	0	0	1	1	0
	4	0	0	0	1	0	0	0	1	1	0
	5	0	1	0	0	0	0	0	0	0	0
	6	1	0	1	1	0	1	1	1	1	0
	7	1	0	0	0	0	1	0	0	1	0
	8	0	0	0	0	1	0	0	0	0	1
	9	0	1	0	0	1	0	0	0	0	1
	10	0	1	0	0	1	0	0	0	0	1
	11	1	0	0	0	0	1	1	1	0	0
	12	1	0	0	0	0	1	1	0	0	0
	13	0	1	0	0	1	0	0	0	0	1
	14	0	0	1	1	0	0	0	1	1	0
	15	0	1	0	0	1	0	0	0	0	1

Binary problem set 7.

		MACHINES									
		1	2	3	4	5	6	7	8	9	10
PARTS	1	0	0	1	1	0	0	0	1	0	0
	2	1	0	0	0	0	1	1	0	0	0
	3	0	0	1	1	0	0	0	1	1	0
	4	0	0	0	1	0	0	0	1	1	0
	5	0	1	0	0	0	0	0	0	0	0
	6	0	0	1	1	0	1	1	1	1	0
	7	1	0	0	0	0	1	0	0	1	0
	8	0	0	0	0	1	0	0	0	0	1
	9	0	1	0	0	1	0	0	0	0	1
	10	0	1	0	0	1	0	0	0	0	1
	11	1	0	0	0	0	1	1	1	0	0
	12	1	0	0	0	0	1	1	0	0	0
	13	0	1	0	0	1	0	0	0	0	1
	14	0	0	1	1	0	0	0	1	1	0
	15	0	1	0	0	1	0	0	0	0	1

Binary problem set 8.

		MACHINES									
		1	2	3	4	5	6	7	8	9	10
PARTS	1	0	0	1	1	0	0	0	1	0	0
	2	1	0	0	0	0	1	1	0	0	0
	3	0	0	1	1	0	0	0	1	1	0
	4	0	0	0	1	0	0	0	1	1	0
	5	0	1	0	0	0	0	0	0	0	0
	6	0	0	1	1	0	1	1	1	1	0
	7	1	0	0	0	0	1	0	0	0	0
	8	0	0	0	0	1	0	0	0	0	1
	9	0	1	0	0	1	0	0	0	0	1
	10	0	1	0	0	1	0	0	0	0	1
	11	1	0	0	0	0	1	1	1	0	0
	12	1	0	0	0	0	1	1	0	0	0
	13	0	1	0	0	1	0	0	0	0	1
	14	0	0	1	1	0	0	0	1	1	0
	15	0	1	0	0	1	0	0	0	0	1

Binary problem set 9.

		MACHINES														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PARTS	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	2	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	3	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	4	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	5	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1
	6	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1
	7	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	8	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	9	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	10	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	11	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	12	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	13	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	14	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1
	15	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0

Binary problem set 10.

		MACHINES														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PARTS	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	2	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	3	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0
	4	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1
	5	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1
	6	0	0	0	0	1	1	1	0	0	1	1	0	0	0	1
	7	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	8	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	9	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	10	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	11	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
	12	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0
	13	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	14	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1
	15	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0

Binary problem set 11.

		MACHINES														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	2	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0
	3	0	1	0	1	0	0	0	1	1	0	0	1	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	5	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1
	6	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
	7	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
PARTS	8	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	9	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0
	10	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	11	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	12	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	13	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
	14	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1
	15	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Binary problem set 12.

		MACHINES														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	2	0	1	0	1	0	0	0	1	1	0	0	0	0	0	0
	3	0	1	0	1	1	0	0	1	1	0	0	1	0	0	0
	4	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
	5	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1
	6	0	0	0	0	1	1	1	0	0	1	0	0	0	0	1
	7	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
PARTS	8	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
	9	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0
	10	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	11	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
	12	1	0	0	0	0	0	0	1	0	1	0	0	0	1	0
	13	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0
	14	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1
	15	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Binary problem set 13.

		MACHINES																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
PARTS	1	1	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0
	2	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0
	3	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
	4	0	1	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0
	5	0	1	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1
	6	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0
	7	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0
	8	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0
	9	1	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0
	10	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1
	11	0	1	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0
	12	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0
	13	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0

Binary problem set 14.

		MACHINES																										
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
PARTS	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
	3	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
	4	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0
	5	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
	6	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
	7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
	8	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0
	9	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
	10	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
	11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
	12	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	1	0	0
	13	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0
	14	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
	15	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0
	16	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	17	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0
	18	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
	19	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
	20	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	1
	21	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0
	22	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
	23	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	1
	24	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0
	25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
	26	0	1	0	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
	27	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
	28	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	29	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	1	0	1	0	0	0	1
	30	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
	31	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	1	1	0
	32	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1
	33	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
	34	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
	35	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1

Binary problem set 15.

APPENDIX A2

SOM MATLAB CODES

```
% BINARY PROBLEM SET NUMBER 2
% DIVIDES TO TWO CELLS
% SOM NETWORK
clear
input=[0    0   1   1   0   1   0   0   0   0   0   0;
1   0   1   0   0   0   1   0   0   0   0   0;
0   1   0   1   1   0   0   1   0   0   0   0;
1   0   1   0   1   0   0   0   1   0   0   0;
0   1   0   0   1   0   0   0   0   1   0   0;
0   0   0   1   1   0   0   0   0   0   1   0;
1   0   1   0   0   0   0   0   0   0   0   1;
1   0   0   0   0   0   1   0   1   0   0   1;
0   1   0   0   0   0   0   1   0   1   0   0;
0   0   1   0   0   1   1   0   1   0   0   1;
0   0   0   1   0   1   0   1   0   0   1   0;
0   0   0   0   1   0   0   1   1   1   1   0]; % input of the
network
net=newsom(minmax(input),[2 1]); % create network
net.trainParam.epochs=100; % train parameters
net.trainParam.goal=1e-5;
net=train(net,input); % train network
m=5; % number of machines = first 5 columns = rest of rows
p=7; % number of parts = rest of columns = first 7 rows
total=m+p; % total columns of input matrix
a=2; % number of cells
output=zeros(a,total); % output of the network
output=sim(net,input);
% cell groupings
Group1machines=[];
Group2machines=[];
Group1parts=[];
Group2parts=[];
c=1;
for i=1:a
    if i==1
        for j=1:m
            if output(i,j)==max(output(:,j))
                Group1machines(c)=j;
                c=c+1;
            end
        end
    end
    c=1;
    if i==2
        for j=1:m
            if output(i,j)==max(output(:,j))
                Group2machines(c)=j;
                c=c+1;
            end
        end
    end
end
c=1;
for i=1:a
    if i==1
```

```
        for j=m+1:total
            if output(i,j)==max(output(:,j))
                Group1parts(c)=j-m;
                c=c+1;
            end
        end
    end
    c=1;
    if i==2
        for j=m+1:total
            if output(i,j)==max(output(:,j))
                Group2parts(c)=j-m;
                c=c+1;
            end
        end
    end
end
% jk = number of machines in the kth cell
% ik = number of parts in the kth cell
[f,h]=size(Group1machines);
j1=f*h;
[f,h]=size(Group2machines);
j2=f*h;
[f,h]=size(Group1parts);
i1=f*h;
[f,h]=size(Group2parts);
i2=f*h;
% result matrix
k=zeros(p,m);
result=zeros(p,m); % result of the input
c=1;
for i=1:a
    for j=1:m
        if output(i,j)==max(output(:,j))
            k(:,c)=input(1:p,j);
            c=c+1;
        end
    end
end
c=1;
for i=1:a
    for j=m+1:total
        if output(i,j)==max(output(:,j))
            result(c,:)=k(j-m,:);
            c=c+1;
        end
    end
end
% exceptional elements, voids
e=sum(sum(result(1:i1,j1+1:m)))+sum(sum(result(i1+1:p,1:j1))); % e =
number of 1s outside the diagonal block (exceptions)
v=(i1*j1-sum(sum(result(1:i1,1:j1))))+(i2*j2-
sum(sum(result(i1+1:p,j1+1:m)))); % v = number of 0s in the diagonal
block (voids)
% operations
count0=0;
count1=0;
for i=1:p
    for j=1:m
```

```
        if result(i,j)==0
            count0=count0+1;
        end
        if result(i,j)==1
            count1=count1+1;
        end
    end
end
o=count1; % o = number of 1s in the part-machine matrix
w=0.5; % w = weight
e1=o-e; % e1 = number of 1s in the cells
n=[w*((o-e)/(o-e+v))]+[(1-w)*((m*p-o-v)/(m*p-o-v+e))]; % n = Grouping
Efficiency
GF=(1-(e/o))/(1+(v/o)); % GF = Grouping Efficacy
ngnorm=[1+[[e1/(e1+v)]-(e/o)]]/2; % ng = Normalised Grouping Measure
GCI=1-(e/o); % GCI = Grouping Capability Index
PE=e/o; % PE = Proportion of Exceptional Elements
MU=e1/[(j1*i1)+(j2*i2)]; % MU = Machine Utilization
PERFORMANCE=(n/5)+(GF/5)+(ngnorm/5)+(GCI/5)+(MU/5);
disp('Machine Number : '), disp(m)
disp('Part Number : '), disp(p)
disp('Input Matrix : '), disp(input)
disp('Result Matrix : '), disp(result)
disp('1st Group Machines : '), disp(Group1machines)
disp('2nd Group Machines : '), disp(Group2machines)
disp('1st Group Parts : '), disp(Group1parts)
disp('2nd Group Parts : '), disp(Group2parts)
disp('Exceptions : '), disp(e)
disp('Voids : '), disp(v)
disp('Grouping Efficiency : '), disp(n)
disp('Grouping Efficacy : '), disp(GF)
disp('Normalised Grouping Measure : '), disp(ngnorm)
disp('Grouping Capability Index : '), disp(GCI)
disp('Machine Utilization : '), disp(MU)
disp('Performance Measure : '), disp(PERFORMANCE)
```

APPENDIX A3

BINARY PROBLEM SET RESULT MATRICES WITH SOM

		MACHINES			
		1	3	2	4
PARTS	1	1	1	0	0
	3	1	1	1	0
	4	1	0	0	0
	2	0	0	1	1
	5	0	0	1	1

Binary problem set 1 SOM result
(Perf.Mea.= 0.8836).

		MACHINES				
		1	3	2	4	5
PARTS	1	0	1	0	1	0
	2	1	1	0	0	0
	4	1	1	0	0	1
	7	1	1	0	0	0
	3	0	0	1	1	1
	5	0	0	1	0	1
	6	0	0	0	1	1

Binary problem set 2 SOM result
(Perf.Mea.= 0.8282).

		MACHINES								
		2	6	9	1	5	3	4	7	8
PARTS	2	1	1	1	1	0	0	1	0	0
	6	1	1	1	0	0	0	0	0	0
	9	1	1	1	0	0	0	0	0	0
	1	1	0	0	1	1	0	0	0	0
	5	0	0	0	1	1	0	0	0	1
	3	0	0	0	0	0	1	1	1	1
	4	0	0	0	0	1	0	1	0	1
	7	0	0	0	0	0	1	0	1	1
	8	0	0	0	0	1	1	1	1	1

Binary problem set 3 SOM result
(Perf.Mea.=0.8394).

		MACHINES									
		6	7	8	4	5	1	2	3	9	10
PARTS	4	1	1	1	0	0	0	0	0	1	0
	5	0	1	1	0	0	0	0	0	0	0
	9	1	1	1	0	0	0	0	0	1	0
	10	0	1	1	0	0	0	0	0	0	0
	3	0	0	0	1	1	0	0	0	1	1
	7	0	0	0	1	0	0	0	0	1	0
	8	0	0	0	1	1	0	0	0	0	1
	1	0	0	0	0	0	1	1	1	1	1
	2	0	0	0	0	0	1	0	0	1	1
	6	0	0	0	0	0	1	0	1	1	1

Binary problem set 4 SOM result
(Perf.Mea.=0.8022).

		MACHINES									
		1	7	10	2	5	8	3	4	6	9
PARTS	2	1	1	1	0	0	0	0	0	0	0
	7	1	1	1	0	0	0	0	0	0	0
	10	1	1	0	0	0	0	0	0	0	0
	11	1	1	1	0	0	0	0	0	0	0
	12	1	1	1	0	0	0	0	0	0	0
	3	0	0	0	1	1	1	0	0	0	0
	5	0	0	0	1	1	1	0	0	0	0
	8	0	0	0	0	1	1	0	0	0	0
	13	0	0	0	1	1	1	0	0	0	0
	15	0	0	0	1	1	1	0	0	0	0
	1	0	0	0	0	0	0	1	1	1	0
	4	0	0	0	0	0	0	0	1	1	1
	6	0	0	0	0	0	0	1	0	1	1
	9	0	0	0	0	0	0	1	1	1	1
	14	0	0	0	0	0	0	1	1	1	1

Binary problem set 5 SOM result
(Perf.Mea.=0.94).

		MACHINES									
		4	5	6	1	2	3	7	8	9	10
PARTS	5	1	0	1	0	0	0	0	0	0	0
	6	0	1	1	0	0	0	1	0	0	0
	7	1	1	1	0	0	0	0	0	0	0
	8	1	1	1	0	0	0	0	0	0	0
	9	1	1	1	0	0	0	1	0	0	0
	10	1	1	1	0	0	0	0	0	0	0
	1	0	0	0	1	1	1	0	0	0	1
	2	0	0	0	1	1	0	1	0	0	0
	3	0	0	0	1	1	1	0	0	0	0
	4	0	0	0	1	1	1	0	0	0	1
	11	0	0	0	0	0	0	1	1	1	0
	12	0	0	0	0	1	0	1	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	14	0	0	0	0	1	0	1	1	1	1
	15	0	0	0	0	0	0	1	1	1	1

Binary problem set 6 SOM result
(Perf.Mea.=0.8705).

		MACHINES									
		2	5	10	1	6	7	3	4	8	9
PARTS	5	1	0	0	0	0	0	0	0	0	0
	8	0	1	1	0	0	0	0	0	0	0
	9	1	1	1	0	0	0	0	0	0	0
	10	1	1	1	0	0	0	0	0	0	0
	13	1	1	1	0	0	0	0	0	0	0
	15	1	1	1	0	0	0	0	0	0	0
	2	0	0	0	1	1	1	0	0	0	0
	7	0	0	0	1	1	0	0	0	0	1
	11	0	0	0	1	1	1	0	0	1	0
	12	0	0	0	1	1	1	0	0	0	0
	1	0	0	0	0	0	0	1	1	1	0
	3	0	0	0	0	0	0	1	1	1	1
	4	0	0	0	0	0	0	0	1	1	1
	6	0	0	0	1	1	1	1	1	1	1
	14	0	0	0	0	0	0	1	1	1	1

Binary problem set 7 SOM result
(Perf.Mea.=0.8764).

		MACHINES									
		3	4	8	9	1	6	7	2	5	10
PARTS	1	1	1	1	0	0	0	0	0	0	0
	3	1	1	1	1	0	0	0	0	0	0
	4	0	1	1	1	0	0	0	0	0	0
	6	1	1	1	1	0	1	1	0	0	0
	14	1	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	1	1	1	0	0	0
	7	0	0	0	1	1	1	0	0	0	0
	11	0	0	1	0	1	1	1	0	0	0
	12	0	0	0	0	1	1	1	0	0	0
	5	0	0	0	0	0	0	0	1	0	0
	8	0	0	0	0	0	0	0	0	1	1
	9	0	0	0	0	0	0	0	1	1	1
	10	0	0	0	0	0	0	0	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	15	0	0	0	0	0	0	0	1	1	1

Binary problem set 8 SOM result
(Perf.Mea.=0.886).

		MACHINES									
		3	4	8	9	1	6	7	2	5	10
PARTS	1	1	1	1	0	0	0	0	0	0	0
	3	1	1	1	1	0	0	0	0	0	0
	4	0	1	1	1	0	0	0	0	0	0
	6	1	1	1	1	0	1	1	0	0	0
	14	1	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	1	1	1	0	0	0
	7	0	0	0	0	1	1	0	0	0	0
	11	0	0	1	0	1	1	1	0	0	0
	12	0	0	0	0	1	1	1	0	0	0
	5	0	0	0	0	0	0	0	1	0	0
	8	0	0	0	0	0	0	0	0	1	1
	9	0	0	0	0	0	0	0	1	1	1
	10	0	0	0	0	0	0	0	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	15	0	0	0	0	0	0	0	1	1	1

Binary problem set 9 SOM result
(Perf.Mea.=0.8959).

		MACHINES														
		5	6	7	11	15	1	10	14	3	13	2	4	8	9	12
	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	6	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	14	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	12	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	15	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
PARTS	4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	13	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Binary problem set 10 SOM result (Perf.Mea.=1).

		MACHINES														
		2	4	8	9	12	3	13	1	10	14	5	6	7	11	15
	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
	9	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	13	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	4	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1
	7	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
PARTS	8	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
	10	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
	12	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0
	15	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
	5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	6	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
	14	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1

Binary problem set 11 SOM result (Perf.Mea.=0.9308).

		MACHINES														
		2	4	8	9	12	1	10	14	3	13	5	6	7	11	15
	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	9	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
	13	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	12	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	15	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
PARTS	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
	6	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
	14	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Binary problem set 12 SOM result (Perf.Mea.=0.9).

		MACHINES														
		5	6	7	11	15	1	10	14	3	13	2	4	8	9	12
	5	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
	6	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
	14	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1
	10	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	12	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0
	15	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
PARTS	4	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
	3	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	9	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
	13	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Binary problem set 13 SOM result (Perf.Mea.=0.8285).

		MACHINES																
		2	8	9	13	14	17	3	4	5	11	16	1	6	7	10	12	15
PARTS	4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	10	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	11	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	13	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
	7	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	8	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
	9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Binary problem set 14 SOM result (Perf.Mea.=0.904).

		MACHINES																										
		4	10	17	18	22	28	5	9	12	25	3	16	24	26	2	6	8	13	20	1	7	14	15	21	11	19	23	27
PARTS	5	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	1	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	23	1	1	1	0	1	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	29	1	0	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	32	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	35	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	3	0	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
	10	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	17	0	1	0	0	0	0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
	18	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
	24	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	0	0	0	1	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	21	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	22	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	31	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	26	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1
	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
	8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1
	14	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0
	15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	0
	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
	4	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1
	27	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1
	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Binary problem set 15 SOM result (Perf.Mea.=0.8032).

APPENDIX A4

BINARY PROBLEM SET RESULT MATRICES WITH CNN

		MACHINES			
		1	3	2	4
PARTS	1	1	1	0	0
	3	1	1	1	0
	4	1	0	0	0
	2	0	0	1	1
	5	0	0	1	1

Binary problem set 1 CNN result
(Perf.Mea.=0.8836).

		MACHINES				
		1	3	2	4	5
PARTS	1	0	1	0	1	0
	2	1	1	0	0	0
	4	1	1	0	0	1
	7	1	1	0	0	0
	3	0	0	1	1	1
	5	0	0	1	0	1
	6	0	0	0	1	1

Binary problem set 2 CNN result
(Perf.Mea.=0.8282).

		MACHINES								
		1	5	2	6	9	3	4	7	8
PARTS	1	1	1	1	0	0	0	0	0	0
	4	0	1	0	0	0	0	1	0	1
	5	1	1	0	0	0	0	0	0	1
	2	1	0	1	1	1	0	1	0	0
	6	0	0	1	1	1	0	0	0	0
	9	0	0	1	1	1	0	0	0	0
	3	0	0	0	0	0	1	1	1	1
	7	0	0	0	0	0	1	0	1	1
	8	0	1	0	0	0	1	1	1	1

Binary problem set 3 CNN result
(Perf.Mea.=0.8388).

		MACHINES									
		6	7	8	4	5	1	2	3	9	10
PARTS	4	1	1	1	0	0	0	0	0	1	0
	5	0	1	1	0	0	0	0	0	0	0
	9	1	1	1	0	0	0	0	0	1	0
	10	0	1	1	0	0	0	0	0	0	0
	3	0	0	0	1	1	0	0	0	1	1
	7	0	0	0	1	0	0	0	0	1	0
	8	0	0	0	1	1	0	0	0	0	1
	1	0	0	0	0	0	1	1	1	1	1
	2	0	0	0	0	0	1	0	0	1	1
	6	0	0	0	0	0	1	0	1	1	1

Binary problem set 4 CNN result
(Perf.Mea.=0.8022).

		MACHINES									
		1	7	10	2	5	8	3	4	6	9
PARTS	2	1	1	1	0	0	0	0	0	0	0
	7	1	1	1	0	0	0	0	0	0	0
	10	1	1	0	0	0	0	0	0	0	0
	11	1	1	1	0	0	0	0	0	0	0
	12	1	1	1	0	0	0	0	0	0	0
	3	0	0	0	1	1	1	0	0	0	0
	5	0	0	0	1	1	1	0	0	0	0
	8	0	0	0	0	1	1	0	0	0	0
	13	0	0	0	1	1	1	0	0	0	0
	15	0	0	0	1	1	1	0	0	0	0
	1	0	0	0	0	0	0	1	1	1	0
	4	0	0	0	0	0	0	0	1	1	1
	6	0	0	0	0	0	0	1	0	1	1
	9	0	0	0	0	0	0	1	1	1	1
	14	0	0	0	0	0	0	1	1	1	1

Binary problem set 5 CNN result
(Perf.Mea.=0.94).

		MACHINES									
		4	5	6	1	2	3	7	8	9	10
	5	1	0	1	0	0	0	0	0	0	0
	6	0	1	1	0	0	0	1	0	0	0
	7	1	1	1	0	0	0	0	0	0	0
	8	1	1	1	0	0	0	0	0	0	0
	9	1	1	1	0	0	0	1	0	0	0
	10	1	1	1	0	0	0	0	0	0	0
	1	0	0	0	1	1	1	0	0	0	1
PARTS	2	0	0	0	1	1	0	1	0	0	0
	3	0	0	0	1	1	1	0	0	0	0
	4	0	0	0	1	1	1	0	0	0	1
	11	0	0	0	0	0	0	1	1	1	0
	12	0	0	0	0	1	0	1	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	14	0	0	0	0	1	0	1	1	1	1
	15	0	0	0	0	0	0	1	1	1	1

Binary problem set 6 CNN result
(Perf.Mea.=0.8705).

		MACHINES									
		1	6	7	2	5	10	3	4	8	9
	2	1	1	1	0	0	0	0	0	0	0
	6	1	1	1	0	0	0	1	1	1	1
	7	1	1	0	0	0	0	0	0	0	1
	11	1	1	1	0	0	0	0	0	1	0
	12	1	1	1	0	0	0	0	0	0	0
	5	0	0	0	1	0	0	0	0	0	0
	8	0	0	0	0	1	1	0	0	0	0
PARTS	9	0	0	0	1	1	1	0	0	0	0
	10	0	0	0	1	1	1	0	0	0	0
	13	0	0	0	1	1	1	0	0	0	0
	15	0	0	0	1	1	1	0	0	0	0
	1	0	0	0	0	0	0	1	1	1	0
	3	0	0	0	0	0	0	1	1	1	1
	4	0	0	0	0	0	0	0	1	1	1
	14	0	0	0	0	0	0	1	1	1	1

Binary problem set 7 CNN result
(Perf.Mea.=0.8764).

		MACHINES									
		3	4	8	9	1	6	7	2	5	10
PARTS	1	1	1	1	0	0	0	0	0	0	0
	3	1	1	1	1	0	0	0	0	0	0
	4	0	1	1	1	0	0	0	0	0	0
	6	1	1	1	1	0	1	1	0	0	0
	14	1	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	1	1	1	0	0	0
	7	0	0	0	1	1	1	0	0	0	0
	11	0	0	1	0	1	1	1	0	0	0
	12	0	0	0	0	1	1	1	0	0	0
	5	0	0	0	0	0	0	0	1	0	0
	8	0	0	0	0	0	0	0	0	1	1
	9	0	0	0	0	0	0	0	1	1	1
	10	0	0	0	0	0	0	0	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	15	0	0	0	0	0	0	0	1	1	1

Binary problem set 8 CNN result
(Perf.Mea.=0.886).

		MACHINES									
		3	4	8	9	1	6	7	2	5	10
PARTS	1	1	1	1	0	0	0	0	0	0	0
	3	1	1	1	1	0	0	0	0	0	0
	4	0	1	1	1	0	0	0	0	0	0
	6	1	1	1	1	0	1	1	0	0	0
	14	1	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	1	1	1	0	0	0
	7	0	0	0	0	1	1	0	0	0	0
	11	0	0	1	0	1	1	1	0	0	0
	12	0	0	0	0	1	1	1	0	0	0
	5	0	0	0	0	0	0	0	1	0	0
	8	0	0	0	0	0	0	0	0	1	1
	9	0	0	0	0	0	0	0	1	1	1
	10	0	0	0	0	0	0	0	1	1	1
	13	0	0	0	0	0	0	0	1	1	1
	15	0	0	0	0	0	0	0	1	1	1

Binary problem set 9 CNN result
(Perf.Mea.=0.8959).

		MACHINES														
		5	6	7	11	15	1	10	14	3	13	2	4	8	9	12
	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	6	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	14	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	12	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	15	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
PARTS	4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	13	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Binary problem set 10 CNN result (Perf.Mea.=1).

		MACHINES														
		2	4	8	9	12	3	13	1	10	14	5	6	7	11	15
	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
	9	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	13	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	4	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1
	7	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
PARTS	8	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
	10	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
	12	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0
	15	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
	5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	6	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1
	14	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1

Binary problem set 11 CNN result (Perf.Mea.=0.9308).

		MACHINES														
		2	4	8	9	12	1	10	14	3	13	5	6	7	11	15
PARTS	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	3	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	9	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
	13	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	12	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	15	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
	6	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
	14	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Binary problem set 12 CNN result (Perf.Mea.=0.9).

		MACHINES														
		5	6	7	11	15	1	10	14	3	13	2	4	8	9	12
PARTS	5	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0
	6	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
	14	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1
	10	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	12	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0
	15	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	4	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0
	7	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
	3	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	9	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
	13	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Binary problem set 13 CNN result (Perf.Mea.=0.8285).

		MACHINES																
		2	8	9	13	14	17	3	4	5	11	16	1	6	7	10	12	15
PARTS	4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	5	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	10	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	11	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	13	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
	7	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	8	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
	9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Binary problem set 14 CNN result (Perf.Mea.=0.904).

		MACHINES																											
		9	12	25	2	5	6	8	13	20	4	10	17	18	22	28	1	7	14	15	21	11	19	23	27	3	16	24	26
PARTS	3	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	9	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
	10	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	17	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
	24	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	16	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	18	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	26	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	28	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	0	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0
	23	0	0	1	1	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0
	29	0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	1	0
	32	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	35	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	0
	14	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0	0	0
	15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0
	30	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0
	27	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	12	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
	13	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	1
	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1
	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	31	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Binary problem set 15 CNN result (Perf.Mea.=0.7748).

APPENDIX A5

THE NONBINARY PROBLEM SETS

PROCESSING TIME		MACHINES						
		1	2	3	4	5	6	7
PARTS	1	0	0.7	0.8	0.7	0	0.3	0.8
	2	0	1	0.7	0.7	0.1	0.2	1
	3	1	0.3	0	0	0.5	0.8	0.5
	4	0.8	0	0	0.3	0	0.8	0
	5	1	0	0.3	0	0.6	0.7	0
	6	0.1	0.8	0.8	0	0	0	0.9
	7	0.9	0	0	0.3	0.5	0.7	0

Nonbinary problem set 1.

WORK LOAD		MACHINES								
		1	2	3	4	5	6	7	8	9
PARTS	1	1	0	0	0	0	0	1	0	0
	2	0	1	0	0	1	0	0	0	0
	3	0	0.95	0.05	0	0	0	0	0	1
	4	1	0	0	1	0	0	1	0	0
	5	0	0	0	0.1	0.5	0.4	0	1	0
	6	0	0	1	0	0	1	0	0	1
	7	0.7	0	0.3	0	0	0	1	0	0
	8	0	1	0	0	1	0	0	1	0
	9	0	0.45	0.55	0	0	1	0	0	0

Nonbinary problem set 2.

VOLUME BASED		MACHINES														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PARTS	1	0	0	0	0	1200	0	0	800	800	0	0	800	400	0	0
	2	155	620	0	0	0	0	310	0	0	0	0	0	0	155	0
	3	0	0	0	0	1250	0	0	5000	3750	0	0	2500	2500	0	0
	4	0	350	0	0	0	700	350	0	0	0	0	0	0	0	0
	5	90	0	0	0	360	0	0	360	0	360	180	0	0	90	0
	6	0	0	120	120	0	0	0	0	0	120	0	0	0	0	120
	7	0	0	200	0	600	0	0	400	0	0	0	0	0	0	0
	8	1100	0	0	1100	0	0	0	0	0	0	0	0	0	1100	1100
	9	215	0	860	430	0	0	0	0	0	860	430	0	0	215	430
	10	140	560	0	280	0	560	280	0	0	0	0	0	0	140	280
	11	0	0	0	0	1560	0	0	1040	0	0	0	1040	520	0	0
	12	0	0	150	0	0	0	0	0	0	300	150	0	0	0	0
	13	45	180	0	0	0	0	270	0	0	0	0	0	0	45	0

Initial matrix of nonbinary problem set 3.

VOLUME BASED		MACHINES														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PARTS	1	0	0	0	0	1	0	0	0.67	0.67	0	0	0.67	0.33	0	0
	2	0.25	1	0	0	0	0	0.5	0	0	0	0	0	0	0.25	0
	3	0	0	0	0	0.25	0	0	1	0.75	0	0	0.5	0.5	0	0
	4	0	0.5	0	0	0	1	0.5	0	0	0	0	0	0	0	0
	5	0.25	0	0	0	1	0	0	1	0	1	0.5	0	0	0.25	0
	6	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1
	7	0	0	0.33	0	1	0	0	0.67	0	0	0	0	0	0	0
	8	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1
	9	0.25	0	1	0.5	0	0	0	0	0	1	0.5	0	0	0.25	0.5
	10	0.25	1	0	0.5	0	1	0.5	0	0	0	0	0	0	0.25	0.5
	11	0	0	0	0	1	0	0	0.67	0	0	0	0.67	0.33	0	0
	12	0	0	0.5	0	0	0	0	0	0	1	0.5	0	0	0	0
	13	0.17	0.67	0	0	0	0	1	0	0	0	0	0	0	0.17	0

Nonbinary problem set 3.

PROCESSING TIME		MACHINES													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
PARTS	1	0	0	0	1	0.36	0	0.72	0	0	0	0	0	0	0
	2	0	0	0	1	0.7	0	0.27	0	0	0	0	0	0	0
	3	0	0.7	0.22	0	0	0	0	0	0	0.46	1	0	0	0
	4	1	0	0.5	0	0	0	0	0	0	0	0.82	0	0	0
	5	0	0	0	0	0	0	0	1	0.8	0	0	0	0	0
	6	1	0	0	0	0	0	0	0	0	0	0	0	0.7	0
	7	0.69	0	0	0	0	0	1	0	0	0	0	0.24	0.03	0
	8	0	0	0	0	0	0	0	0	0	0	0	0.93	1	0
	9	0	0	0	0	0	0.83	0	0.57	0.28	0	0	0	0	1
	10	0	0	0	0	0	0.73	0	1	0	0	0	0	0	0
	11	0	0	0	0	0	1	0	0	0	0	0	0	0	0.24
	12	0	0	0	0	0	1	0	0.85	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0.97	0	0	0	0	0	1
	14	0	0	0	0	0	1	0	0.3	0	0	0	0	0	0
	15	0	0	0	0	0	0.8	0	0.45	0.72	0	0	0	0	1
	16	0	0	0	0	0	0.66	0	1	0	0	0	0	0	0
	17	0	0	0	0.26	1	0	0.5	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0	0	0	0	0	0.68	0
	19	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	20	0	0	0	0.74	1	0	0.5	0	0	0	0	0	0	0
	21	0	0	0.33	0	0	0	0	0	0	0	1	0	0	0
	22	0	0	0	0	0	0.88	0	1	0	0	0	0	0	0
	23	0	0	0	0.65	1	0	0	0	0	0	0	0	0.46	0
	24	0	0	0	0	0	0	0	0	0	1	0.76	0	0	0

Nonbinary problem set 4.

WORK LOAD		MACHINES																							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PARTS	1	0.62	0	1	0	0	0	0	0	0	0.38	0.75	0	0	0	0	1	0	0	0	0	0	0.62	0.75	0.88
	2	0	0	0	0	0.33	0	1	0	0	0	0	0	0.89	0	0.22	0.44	0	0	0	0	0	0	0	0
	3	0.62	0.38	0	0	0	0	0	0	0	0	0.75	0	0	0	0	0.88	0	0	0.88	0	0	0	1	0
	4	0.38	0	0.38	0	0	0	0	0	0	0	0	0.88	0	0	0	0	0	0.75	0	0	0	1	0.5	0.62
	5	0.75	0	0	0.38	0	0	0	0	0	0.75	1	0	0	0	0	0.38	0	0	0	0	0	0	0	0
	6	0.5	0.75	0	0	0	0	0	0	0	0.88	0	0	0	0	0	0.88	0	0	0	0	0	0	1	0
	7	0	0	0	0	0	0	0.43	1	0.57	0	0	0	0.43	0	0	0	0	0	0	0	0	0	0	0
	8	0.88	0	0.62	0	0	0	0	0	0	0	0	0.75	0	0	0	0.25	0.75	0	0	0	0	0	1	0
	9	0	0.5	0	0	0.88	0	1	0	0	0.38	0	0	0	0	0	0	0	0	0.88	0.5	0	0.88	0.5	0.5
	10	0	0.22	0	0	0	0.22	0.33	0.33	1	0	0	0	0	0	0	0	0.33	0.89	0	0	0	0	0.67	0
	11	0	0	0	0	0.38	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	0.38	0.62	0	0	0	0	0	0	0	0.62	0.88	0	0	0	0	1	0	0	0.38	0	0.75	0	0	0
	13	0.75	0	0	0.75	0	0	0	0	0	1	0	0	0	0.75	0	0.5	0	0	1	0	0.38	0	0	0
	14	0.22	0	0	0	0	0	0	0	0	0	0	0.56	0	0	0	1	0.56	0	0	0	0	0	0.33	0
	15	0.78	0	0	0	0	0	0	0	0	0	0	0.89	0	0	0	0	0	0	0.44	0.56	0	1	0.89	0.89
	16	0	0	0	0	0.75	0	0.38	0	0	0	0	0	0	0	0.88	0	0	0	1	0	1	0	0	0
	17	0	0	0	0	0	1	0.71	0	0	0	0	0	1	0	0	1	0	0	0.43	0.43	0	0	0	0
	18	0.5	0	0.5	0	0	0	0	0	0	1	0	0	0	0	0	0.67	0	0	0	0	0	0	0	0
	19	1	0	0.83	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	0	0	0.12	0	0	0	0	0	0	0	0	0.38	0	0	0	1	0.38	0	0	0	0	0	0.88	0

Nonbinary problem set 5.

PROCESSING TIME		MACHINES																													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
PARTS	1	0	0	0.3	0	0	0.6	0.6	0.2	0.2	0.5	0.7	0	0	0	0	0	0.4	0	0.6	0	0	0	0	0	0	0	0	0	0	0
	2	0.4	0	0.5	0	0	0.7	0.3	0.4	0.3	0.6	0.8	0	0	0	0	0	0.9	0	0.2	0	0	0	0	0	0	0	0	0	0	0
	3	0.6	0	0.7	0.3	0	0.2	0.4	0.9	0.6	0.2	0.2	0	0	0	0	0.4	0.3	0	0.5	0	0	0	0	0	0	0	0	0	0	0
	4	0	0.2	0	0	0.3	0	0	0	0	0	0	0	0.4	0.7	0.5	0.6	0	0.2	0	0.4	0	0.4	0	0	0	0.5	0.6	0	0	0
	5	0	0.2	0	0	0.3	0	0	0	0	0	0	0.4	0	0.5	0	0.7	0	0.8	0	0.9	0	0.6	0	0	0	0.8	0.2	0	0	0
	6	0	0.8	0	0	0.9	0	0	0	0	0	0	1	0	0.7	0	0.2	0	0.3	0	0.4	0	0.5	0	0	0	0.6	0.8	0	0	0
	7	0.8	0	0.9	0	0	0.3	0.5	0.5	0.7	0.3	0.5	0	0	0	0	0	0.6	0	0.9	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0.8	0	0.3	0	0.9	0	0.2	0	0.3	0	0	0	0.4	0.5	0	0	0
	9	0	0.4	0	0	0.5	0	0	0	0	0	0	0.6	0	0.9	0	0.5	0	0.6	0	0.7	0	0.8	0	0	0	0.9	0.5	0	0	0
	10	0.6	0	0.2	0	0	0.3	0.9	0.2	0.3	0.4	0.5	0	0	0	0	0	0.6	0	0.8	0	0	0	0	0	0	0	0	0	0	0
	11	0.3	0	0.3	0.2	0	0	0	0	0	0	0	0	0.3	0	0.4	0	0	0	0	0	0.5	0	0.9	0.2	0.5	0	0	0.6	0.7	0.8
	12	0	0	0	0.6	0	0	0	0	0	0	0	0	0.7	0	0.8	0	0	0	0	0	0.9	0	0.9	0.3	0.5	0	0	0.5	0.6	0.7
	13	0	0	0	0.7	0	0	0	0	0	0	0	0	0.5	0	0.6	0	0	0	0	0	0.8	0	0.5	0.3	0.4	0	0	0.5	0.7	0.8
	14	0	0	0	0.2	0	0	0	0	0	0	0	0	0.6	0	0.8	0	0	0	0	0	0.5	0	0.5	0.4	0.6	0	0	0.8	0.2	0.8
	15	0	0	0	0.5	0	0	0	0	0	0	0	0	0.7	0	0.9	0	0	0	0	0	0	0	0.3	0	0.7	0	0	0.9	0.3	0.4

Nonbinary problem set 6.

APPENDIX A6

NONBINARY PROBLEM SET RESULT MATRICES WITH FUZZY SOM

		MACHINES						
		1	5	6	2	3	4	7
PARTS	3	1	0.5	0.8	0.3	0	0	0.5
	4	0.8	0	0.8	0	0	0.3	0
	5	1	0.6	0.7	0	0.3	0	0
	7	0.9	0.5	0.7	0	0	0.3	0
	1	0	0	0.3	0.7	0.8	0.7	0.8
	2	0	0.1	0.2	1	0.7	0.7	1
	6	0.1	0	0	0.8	0.8	0	0.9

Nonbinary problem set 1 Fuzzy SOM result
(Perf.Mea.=0.837).

		MACHINES								
		2	5	8	3	6	9	1	4	7
PARTS	2	1	1	0	0	0	0	0	0	0
	5	0	0.5	1	0	0.4	0	0	0.1	0
	8	1	1	1	0	0	0	0	0	0
	3	0.95	0	0	0.05	0	1	0	0	0
	6	0	0	0	1	1	1	0	0	0
	9	0.45	0	0	0.55	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	1
	4	0	0	0	0	0	0	1	1	1
	7	0	0	0	0.3	0	0	0.7	0	1

Nonbinary problem set 2 Fuzzy SOM result with 3 cells
(Perf.Mea.=0.7948).

		MACHINES								
		2	5	8	9	1	4	7	3	6
PARTS	2	1	1	0	0	0	0	0	0	0
	5	0	0.5	1	0	0	0.1	0	0	0.4
	8	1	1	1	0	0	0	0	0	0
	3	0.95	0	0	1	0	0	0	0.05	0
	1	0	0	0	0	1	0	1	0	0
	4	0	0	0	0	1	1	1	0	0
	7	0	0	0	0	0.7	0	1	0.3	0
	6	0	0	0	1	0	0	0	1	1
	9	0.45	0	0	0	0	0	0	0.55	1

Nonbinary problem set 2 Fuzzy SOM result with 4 cells
(Perf.Mea.=0.8077).

		MACHINES														
		1	3	4	10	11	14	15	5	8	9	12	13	2	6	7
	6	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0
	8	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0
	9	0.25	1	0.5	1	0.5	0.25	0.5	0	0	0	0	0	0	0	0
	12	0	0.5	0	1	0.5	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	1	0.67	0.67	0.67	0.33	0	0	0
	3	0	0	0	0	0	0	0	0.25	1	0.75	0.5	0.5	0	0	0
PARTS	5	0.25	0	0	1	0.5	0.25	0	1	1	0	0	0	0	0	0
	7	0	0.33	0	0	0	0	0	1	0.67	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	1	0.67	0	0.67	0.33	0	0	0
	2	0.25	0	0	0	0	0.25	0	0	0	0	0	0	1	0	0.5
	4	0	0	0	0	0	0	0	0	0	0	0	0	0.5	1	0.5
	10	0.25	0	0.5	0	0	0.25	0.5	0	0	0	0	0	1	1	0.5
	13	0.17	0	0	0	0	0.17	0	0	0	0	0	0	0.67	0	1

Nonbinary problem set 3 Fuzzy SOM result with 3 cells (Perf.Mea.=0.7137).

		MACHINES														
		2	6	7	1	3	4	14	15	10	11	5	8	9	12	13
	2	1	0	0.5	0.25	0	0	0.25	0	0	0	0	0	0	0	0
	4	0.5	1	0.5	0	0	0	0	0	0	0	0	0	0	0	0
	10	1	1	0.5	0.25	0	0.5	0.25	0.5	0	0	0	0	0	0	0
	13	0.67	0	1	0.17	0	0	0.17	0	0	0	0	0	0	0	0
	6	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0
	8	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0
PARTS	9	0	0	0	0.25	1	0.5	0.25	0.5	1	0.5	0	0	0	0	0
	5	0	0	0	0.25	0	0	0.25	0	1	0.5	1	1	0	0	0
	12	0	0	0	0	0.5	0	0	0	1	0.5	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	1	0.67	0.67	0.67	0.33
	3	0	0	0	0	0	0	0	0	0	0	0.25	1	0.75	0.5	0.5
	7	0	0	0	0	0.33	0	0	0	0	0	1	0.67	0	0	0
	11	0	0	0	0	0	0	0	0	0	0	1	0.67	0	0.67	0.33

Nonbinary problem set 3 Fuzzy SOM result with 4 cells (Perf.Mea.= 0.7362).

		MACHINES														
		5	8	9	12	13	2	6	7	1	4	14	15	3	10	11
	1	1	0.67	0.67	0.67	0.33	0	0	0	0	0	0	0	0	0	0
	5	1	1	0	0	0	0	0	0	0.25	0	0.25	0	0	1	0.5
	7	1	0.67	0	0	0	0	0	0	0	0	0	0	0.33	0	0
	11	1	0.67	0	0.67	0.33	0	0	0	0	0	0	0	0	0	0
	3	0.25	1	0.75	0.5	0.5	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	1	0	0.5	0.25	0	0.25	0	0	0	0
PARTS	4	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
	10	0	0	0	0	0	1	1	0.5	0.25	0.5	0.25	0.5	0	0	0
	13	0	0	0	0	0	0.67	0	1	0.17	0	0.17	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
	6	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
	9	0	0	0	0	0	0	0	0	0.25	0.5	0.25	0.5	1	1	0.5
	12	0	0	0	0	0	0	0	0	0	0	0	0	0.5	1	0.5

Nonbinary problem set 3 Fuzzy SOM result with 5 cells (Perf.Mea.= 0.7575).

		MACHINES													
		6	8	14	2	3	9	10	11	1	12	13	4	5	7
PARTS	5	0	1	0	0	0	0.8	0	0	0	0	0	0	0	0
	9	0.83	0.57	1	0	0	0.28	0	0	0	0	0	0	0	0
	10	0.73	1	0	0	0	0	0	0	0	0	0	0	0	0
	11	1	0	0.24	0	0	0	0	0	0	0	0	0	0	0
	12	1	0.85	0	0	0	0	0	0	0	0	0	0	0	0
	13	0	0.97	1	0	0	0	0	0	0	0	0	0	0	0
	14	1	0.3	0	0	0	0	0	0	0	0	0	0	0	0
	15	0.8	0.45	1	0	0	0.72	0	0	0	0	0	0	0	0
	16	0.66	1	0	0	0	0	0	0	0	0	0	0	0	0
	22	0.88	1	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0.7	0.22	0	0.46	1	0	0	0	0	0	0
	21	0	0	0	0	0.33	0	0	1	0	0	0	0	0	0
	24	0	0	0	0	0	0	1	0.76	0	0	0	0	0	0
	4	0	0	0	0	0.5	0	0	0.82	1	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	1	0	0.7	0	0	0
	7	0	0	0	0	0	0	0	0	0.69	0.24	0.03	0	0	1
	8	0	0	0	0	0	0	0	0	0	0.93	1	0	0	0
	18	0	0	0	0	0	0	0	0	0	0	0.68	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	1	0.36	0.72
	2	0	0	0	0	0	0	0	0	0	0	0	1	0.7	0.27
	17	0	0	0	0	0	0	0	0	0	0	0	0.26	1	0.5
	19	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	20	0	0	0	0	0	0	0	0	0	0	0	0.74	1	0.5
	23	0	0	0	0	0	0	0	0	0	0	0.46	0.65	1	0

Nonbinary problem set 4 Fuzzy SOM result with 4 cells (Perf.Mea.= 0.7065).

	MACHINES													
PARTS	6	8	14	9	1	2	3	10	11	12	13	4	5	7
9	0.83	0.57	1	0.28	0	0	0	0	0	0	0	0	0	0
10	0.73	1	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0.24	0	0	0	0	0	0	0	0	0	0	0
12	1	0.85	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0.97	1	0	0	0	0	0	0	0	0	0	0	0
14	1	0.3	0	0	0	0	0	0	0	0	0	0	0	0
15	0.8	0.45	1	0.72	0	0	0	0	0	0	0	0	0	0
16	0.66	1	0	0	0	0	0	0	0	0	0	0	0	0
22	0.88	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0.8	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0.7	0.22	0.46	1	0	0	0	0	0
4	0	0	0	0	1	0	0.5	0	0.82	0	0	0	0	0
21	0	0	0	0	0	0	0.33	0	1	0	0	0	0	0
24	0	0	0	0	0	0	0	1	0.76	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0	0.7	0	0	0
7	0	0	0	0	0.69	0	0	0	0	0.24	0.03	0	0	1
8	0	0	0	0	0	0	0	0	0	0.93	1	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0.68	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0.36	0.72
2	0	0	0	0	0	0	0	0	0	0	0	1	0.7	0.27
17	0	0	0	0	0	0	0	0	0	0	0	0.26	1	0.5
19	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0.74	1	0.5
23	0	0	0	0	0	0	0	0	0	0	0.46	0.65	1	0

Nonbinary problem set 4 Fuzzy SOM result with 5 cells (Perf.Mea.= 0.7192).

PARTS \ MACHINES	6	8	9	14	12	13	2	3	10	11	1	7	4	5
10	0.73	1	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	0.24	0	0	0	0	0	0	0	0	0	0
12	1	0.85	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0.3	0	0	0	0	0	0	0	0	0	0	0	0
16	0.66	1	0	0	0	0	0	0	0	0	0	0	0	0
22	0.88	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	0.8	0	0	0	0	0	0	0	0	0	0	0
9	0.83	0.57	0.28	1	0	0	0	0	0	0	0	0	0	0
13	0	0.97	0	1	0	0	0	0	0	0	0	0	0	0
15	0.8	0.45	0.72	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0.93	1	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0.68	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0.7	0.22	0.46	1	0	0	0	0
4	0	0	0	0	0	0	0	0.5	0	0.82	1	0	0	0
21	0	0	0	0	0	0	0	0.33	0	1	0	0	0	0
24	0	0	0	0	0	0	0	0	1	0.76	0	0	0	0
6	0	0	0	0	0	0.7	0	0	0	0	1	0	0	0
7	0	0	0	0	0.24	0.03	0	0	0	0	0.69	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0.72	1	0.36
2	0	0	0	0	0	0	0	0	0	0	0	0.27	1	0.7
17	0	0	0	0	0	0	0	0	0	0	0	0.5	0.26	1
19	0	0	0	0	0	0	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0.5	0.74	1
23	0	0	0	0	0	0.46	0	0	0	0	0	0	0.65	1

Nonbinary problem set 4 Fuzzy SOM result with 6 cells (Perf.Mea.=0.7324).

PARTS \ MACHINES	2	4	10	11	14	16	19	21	5	6	7	8	9	13	15	20	1	3	12	17	18	22	23	24
3	0.38	0	0	0.75	0	0.88	0.88	0	0	0	0	0	0	0	0	0	0.62	0	0	0	0	0	1	0
5	0	0.38	0.75	1	0	0.38	0	0	0	0	0	0	0	0	0	0	0.75	0	0	0	0	0	0	0
6	0.75	0	0.88	0	0	0.88	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	1	0
12	0.62	0	0.62	0.88	0	1	0.38	0.75	0	0	0	0	0	0	0	0	0.38	0	0	0	0	0	0	0
13	0	0.75	1	0	0.75	0.5	1	0.38	0	0	0	0	0	0	0	0	0.75	0	0	0	0	0	0	0
16	0	0	0	0	0	0	1	1	0.75	0	0.38	0	0	0	0.88	0	0	0	0	0	0	0	0	0
18	0	0	1	0	0	0.67	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0	0	0
2	0	0	0	0	0	0.44	0	0	0.33	0	1	0	0	0.89	0.22	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0.43	1	0.57	0.43	0	0	0	0	0	0	0	0	0	0
9	0.5	0	0.38	0	0	0	0.88	0	0.88	0	1	0	0	0	0	0.5	0	0	0	0	0	0.88	0.5	0.5
11	0	0	0	0	0	0	0	0	0.38	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	1	0.43	0	0	1	0.71	0	0	1	0	0.43	0	0	0	0	0	0	0	0
1	0	0	0.38	0.75	0	1	0	0	0	0	0	0	0	0	0	0	0.62	1	0	0	0	0.62	0.75	0.88
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0.38	0.88	0	0.75	1	0.5	0.62
8	0	0	0	0	0	0.25	0	0	0	0	0	0	0	0	0	0	0.88	0.62	0.75	0.75	0	0	1	0
10	0.22	0	0	0	0	0	0	0	0	0.22	0.33	0.33	1	0	0	0	0	0	0	0.33	0.89	0	0.67	0
14	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0.22	0	0.56	0.56	0	0	0.33	0
15	0	0	0	0	0	0	0.44	0	0	0	0	0	0	0	0	0.56	0.78	0	0.89	0	0	1	0.89	0.89
19	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	1	0.83	0	0	0	0	0	0
20	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.12	0.38	0.38	0	0	0.88	0

Nonbinary problem set 5 Fuzzy SOM result with 3 cells (Perf.Mea.=0.5332).

PARTS \ MACHINES	1	10	11	16	3	12	22	23	24	2	4	8	9	14	17	18	20	5	6	7	13	15	19	21
1	0.62	0.38	0.75	1	1	0	0.62	0.75	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0.62	0	0.75	0.88	0	0	0	1	0	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0.88	0
5	0.75	0.75	1	0.38	0	0	0	0	0	0	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0.5	0.88	0	0.88	0	0	0	1	0	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0.38	0.62	0.88	1	0	0	0	0	0	0.62	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0.75
13	0.75	1	0	0.5	0	0	0	0	0	0	0.75	0	0	0.75	0	0	0	0	0	0	0	0	1	0.38
18	0.5	1	0	0.67	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0.38	0	0	0	0.38	0.88	1	0.5	0.62	0	0	0	0	0	0	0.75	0	0	0	0	0	0	0	0
8	0.88	0	0	0.25	0.62	0.75	0	1	0	0	0	0	0	0	0.75	0	0	0	0	0	0	0	0	0
14	0.22	0	0	1	0	0.56	0	0.33	0	0	0	0	0	0	0.56	0	0	0	0	0	0	0	0	0
15	0.78	0	0	0	0	0.89	1	0.89	0.89	0	0	0	0	0	0	0	0.56	0	0	0	0	0	0.44	0
19	1	0	0	0	0.83	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0
20	0	0	0	1	0.12	0.38	0	0.88	0	0	0	0	0	0	0.38	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0.57	0	0	0	0	0	0	0.43	0.43	0	0	0
10	0	0	0	0	0	0	0	0.67	0	0.22	0	0.33	1	0	0.33	0.89	0	0	0.22	0.33	0	0	0	0
2	0	0	0	0.44	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33	0	1	0.89	0.22	0	0
9	0	0.38	0	0	0	0	0.88	0.5	0.5	0.5	0	0	0	0	0	0	0.5	0.88	0	1	0	0	0.88	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0	1	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0.38	0	0.88	1	1
17	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0.43	0	1	0.71	1	0	0.43	0

Nonbinary problem set 5 Fuzzy SOM result with 4 cells (Perf.Mea.=0.5537).

PARTS \ MACHINES	2	4	5	14	15	19	20	21	8	9	12	18	22	24	1	3	17	23	10	11	16	6	7	13
9	0.5	0	0.88	0	0	0.88	0.5	0	0	0	0	0	0.88	0.5	0	0	0	0.5	0.38	0	0	0	1	0
16	0	0	0.75	0	0.88	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0
4	0	0	0	0	0	0	0	0	0	0	0.88	0.75	1	0.62	0.38	0.38	0	0.5	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	0.57	0	0	0	0	0	0	0	0	0	0	0	0	0.43	0.43
10	0.22	0	0	0	0	0	0	0	0.33	1	0	0.89	0	0	0	0	0.33	0.67	0	0	0	0.22	0.33	0
15	0	0	0	0	0	0.44	0.56	0	0	0	0.89	0	1	0.89	0.78	0	0	0.89	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0.75	0	0	0	0.88	0.62	0.75	1	0	0	0.25	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0.56	0	0	0	0.22	0	0.56	0.33	0	0	1	0	0	0
19	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	1	0.83	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0.38	0	0	0	0	0.12	0.38	0.88	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0.62	0.88	0.62	1	0	0.75	0.38	0.75	1	0	0	0
3	0.38	0	0	0	0	0.88	0	0	0	0	0	0	0	0	0.62	0	0	1	0	0.75	0.88	0	0	0
5	0	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0	0	0.75	1	0.38	0	0	0
6	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	1	0.88	0	0.88	0	0	0
12	0.62	0	0	0	0	0.38	0	0.75	0	0	0	0	0	0	0.38	0	0	0	0.62	0.88	1	0	0	0
13	0	0.75	0	0.75	0	1	0	0.38	0	0	0	0	0	0	0.75	0	0	0	1	0	0.5	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	1	0	0.67	0	0	0
2	0	0	0.33	0	0.22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.44	0	1	0.89
11	0	0	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
17	0	0	0	0	0	0.43	0.43	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.71	1

Nonbinary problem set 5 Fuzzy SOM result with 5 cells (Perf.Mea.=0.5507).

PARTS \ MACHINES	4	14	15	19	21	6	8	9	18	5	7	13	20	12	22	24	3	17	23	1	2	10	11	16
13	0.75	0.75	0	1	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	1	0	0.5
16	0	0	0.88	1	1	0	0	0	0	0.75	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0.57	0	0	0.43	0.43	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0.22	0.33	1	0.89	0	0.33	0	0	0	0	0	0	0.33	0.67	0	0.22	0	0	0
2	0	0	0.22	0	0	0	0	0	0	0.33	1	0.89	0	0	0	0	0	0	0	0	0	0	0	0.44
9	0	0	0	0.88	0	0	0	0	0	0.88	1	0	0.5	0	0.88	0.5	0	0	0.5	0	0.5	0.38	0	0
11	0	0	0	0	0	0	0	0	0	0.38	1	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0.43	0	1	0	0	0	0	0.71	1	0.43	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0.75	0	0	0	0	0.88	1	0.62	0.38	0	0.5	0.38	0	0	0	0
15	0	0	0	0.44	0	0	0	0	0	0	0	0	0.56	0.89	1	0.89	0	0	0.89	0.78	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.62	0.88	1	0	0.75	0.62	0	0.38	0.75	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0	0.62	0.75	1	0.88	0	0	0	0.25
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.56	0	0	0	0.56	0.33	0.22	0	0	0	1
19	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0.83	0	0	1	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0	0	0.12	0.38	0.88	0	0	0	0	1
3	0	0	0	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.62	0.38	0	0.75	0.88
5	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0.75	1	0.38
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.5	0.75	0.88	0	0.88
12	0	0	0	0.38	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0.62	0.62	0.88	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0.5	0	1	0	0.67

Nonbinary problem set 5 Fuzzy SOM result with 6 cells (Perf.Mea.=0.6089).

		MACHINES																													
		4	13	15	21	23	24	25	28	29	30	2	5	12	14	16	18	20	22	26	27	1	3	6	7	8	9	10	11	17	19
PARTS	11	0.2	0.3	0.4	0.5	0.9	0.2	0.5	0.6	0.7	0.8	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0	0	0	0	0	0	0	0
	12	0.6	0.7	0.8	0.9	0.9	0.3	0.5	0.5	0.6	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	0.7	0.5	0.6	0.8	0.5	0.3	0.4	0.5	0.7	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	0.2	0.6	0.8	0.5	0.5	0.4	0.6	0.8	0.2	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	0.5	0.7	0.9	0	0.3	0	0.7	0.9	0.3	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0.4	0.5	0	0	0	0	0	0	0	0.2	0.3	0	0.7	0.6	0.2	0.4	0.4	0.5	0.6	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0.2	0.3	0.4	0.5	0.7	0.8	0.9	0.6	0.8	0.2	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0.8	0.9	1	0.7	0.2	0.3	0.4	0.5	0.6	0.8	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.8	0.3	0.9	0.2	0.3	0.4	0.5	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0.4	0.5	0.6	0.9	0.5	0.6	0.7	0.8	0.9	0.5	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.6	0.6	0.2	0.2	0.5	0.7	0.4	0.6
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.5	0.7	0.3	0.4	0.3	0.6	0.8	0.9	0.2
	3	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0.6	0.7	0.2	0.4	0.9	0.6	0.2	0.2	0.3	0.5
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0.9	0.3	0.5	0.5	0.7	0.3	0.5	0.6	0.9
	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.2	0.3	0.9	0.2	0.3	0.4	0.5	0.6	0.8

Nonbinary problem set 6 Fuzzy SOM result (Perf.Mea.=0.8613).

APPENDIX A7

NONBINARY PROBLEM SET RESULT MATRICES WITH FUZZY CNN

		MACHINES						
		1	5	6	2	3	4	7
PARTS	3	1	0.5	0.8	0.3	0	0	0.5
	4	0.8	0	0.8	0	0	0.3	0
	5	1	0.6	0.7	0	0.3	0	0
	7	0.9	0.5	0.7	0	0	0.3	0
	1	0	0	0.3	0.7	0.8	0.7	0.8
	2	0	0.1	0.2	1	0.7	0.7	1
	6	0.1	0	0	0.8	0.8	0	0.9

Nonbinary problem set 1 Fuzzy CNN result (Perf.Mea.=0.837).

		MACHINES								
		2	5	8	3	6	9	1	4	7
PARTS	2	1	1	0	0	0	0	0	0	0
	5	0	0.5	1	0	0.4	0	0	0.1	0
	8	1	1	1	0	0	0	0	0	0
	3	0.95	0	0	0.05	0	1	0	0	0
	6	0	0	0	1	1	1	0	0	0
	9	0.45	0	0	0.55	1	0	0	0	0
	1	0	0	0	0	0	0	1	0	1
	4	0	0	0	0	0	0	1	1	1
	7	0	0	0	0.3	0	0	0.7	0	1

Nonbinary problem set 2 Fuzzy CNN result with 3 cells (Perf.Mea.=0.7948).

		MACHINES								
		1	7	3	6	9	4	8	2	5
PARTS	1	1	1	0	0	0	0	0	0	0
	4	1	1	0	0	0	1	0	0	0
	7	0.7	1	0.3	0	0	0	0	0	0
	6	0	0	1	1	1	0	0	0	0
	9	0	0	0.55	1	0	0	0	0.45	0
	5	0	0	0	0.4	0	0.1	1	0	0.5
	2	0	0	0	0	0	0	0	1	1
	3	0	0	0.05	0	1	0	0	0.95	0
	8	0	0	0	0	0	0	1	1	1

Nonbinary problem set 2 Fuzzy CNN result with 4 cells (Perf.Mea.=0.8082).

		MACHINES														
		1	3	4	10	11	14	15	5	8	9	12	13	2	6	7
PARTS	6	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0
	8	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0
	9	0.25	1	0.5	1	0.5	0.25	0.5	0	0	0	0	0	0	0	0
	12	0	0.5	0	1	0.5	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	1	0.67	0.67	0.67	0.33	0	0	0
	3	0	0	0	0	0	0	0	0.25	1	0.75	0.5	0.5	0	0	0
	5	0.25	0	0	1	0.5	0.25	0	1	1	0	0	0	0	0	0
	7	0	0.33	0	0	0	0	0	1	0.67	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	1	0.67	0	0.67	0.33	0	0	0
	2	0.25	0	0	0	0	0.25	0	0	0	0	0	0	1	0	0.5
	4	0	0	0	0	0	0	0	0	0	0	0	0	0.5	1	0.5
	10	0.25	0	0.5	0	0	0.25	0.5	0	0	0	0	0	1	1	0.5
	13	0.17	0	0	0	0	0.17	0	0	0	0	0	0	0.67	0	1

Nonbinary problem set 3 Fuzzy CNN result with 3 cells (Perf.Mea.=0.7137).

		MACHINES														
		3	10	11	5	8	9	12	13	1	4	14	15	2	6	7
PARTS	5	0	1	0.5	1	1	0	0	0	0.25	0	0.25	0	0	0	0
	6	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0
	9	1	1	0.5	0	0	0	0	0	0.25	0.5	0.25	0.5	0	0	0
	12	0.5	1	0.5	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	1	0.67	0.67	0.67	0.33	0	0	0	0	0	0	0
	3	0	0	0	0.25	1	0.75	0.5	0.5	0	0	0	0	0	0	0
	7	0.33	0	0	1	0.67	0	0	0	0	0	0	0	0	0	0
	11	0	0	0	1	0.67	0	0.67	0.33	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
	2	0	0	0	0	0	0	0	0	0.25	0	0.25	0	1	0	0.5
	4	0	0	0	0	0	0	0	0	0	0	0	0	0.5	1	0.5
	10	0	0	0	0	0	0	0	0	0.25	0.5	0.25	0.5	1	1	0.5
	13	0	0	0	0	0	0	0	0	0.17	0	0.17	0	0.67	0	1

Nonbinary problem set 3 Fuzzy CNN result with 4 cells (Perf.Mea.=0.7416).

		MACHINES														
		3	10	11	5	1	4	14	15	2	6	7	8	9	12	13
PARTS	6	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0
	9	1	1	0.5	0	0.25	0.5	0.25	0.5	0	0	0	0	0	0	0
	12	0.5	1	0.5	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	1	0.5	1	0.25	0	0.25	0	0	0	0	1	0	0	0
	7	0.33	0	0	1	0	0	0	0	0	0	0	0.67	0	0	0
	11	0	0	0	1	0	0	0	0	0	0	0	0.67	0	0.67	0.33
	8	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
	2	0	0	0	0	0.25	0	0.25	0	1	0	0.5	0	0	0	0
	4	0	0	0	0	0	0	0	0	0.5	1	0.5	0	0	0	0
	10	0	0	0	0	0.25	0.5	0.25	0.5	1	1	0.5	0	0	0	0
	13	0	0	0	0	0.17	0	0.17	0	0.67	0	1	0	0	0	0
	1	0	0	0	1	0	0	0	0	0	0	0	0.67	0.67	0.67	0.33
	3	0	0	0	0.25	0	0	0	0	0	0	0	1	0.75	0.5	0.5

Nonbinary problem set 3 Fuzzy CNN result with 5 cells (Perf.Mea.=0.7446).

		MACHINES													
		4	5	7	2	3	10	11	6	8	14	1	9	12	13
PARTS	1	1	0.36	0.72	0	0	0	0	0	0	0	0	0	0	0
	2	1	0.7	0.27	0	0	0	0	0	0	0	0	0	0	0
	17	0.26	1	0.5	0	0	0	0	0	0	0	0	0	0	0
	19	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	0.74	1	0.5	0	0	0	0	0	0	0	0	0	0	0
	23	0.65	1	0	0	0	0	0	0	0	0	0	0	0	0.46
	3	0	0	0	0.7	0.22	0.46	1	0	0	0	0	0	0	0
	4	0	0	0	0	0.5	0	0.82	0	0	0	1	0	0	0
	21	0	0	0	0	0.33	0	1	0	0	0	0	0	0	0
	24	0	0	0	0	0	1	0.76	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	1	0	0	0.8	0	0
	9	0	0	0	0	0	0	0	0.83	0.57	1	0	0.28	0	0
	10	0	0	0	0	0	0	0	0.73	1	0	0	0	0	0
	11	0	0	0	0	0	0	0	1	0	0.24	0	0	0	0
	12	0	0	0	0	0	0	0	1	0.85	0	0	0	0	0
	13	0	0	0	0	0	0	0	0	0.97	1	0	0	0	0
	14	0	0	0	0	0	0	0	1	0.3	0	0	0	0	0
	15	0	0	0	0	0	0	0	0.8	0.45	1	0	0.72	0	0
	16	0	0	0	0	0	0	0	0.66	1	0	0	0	0	0
	22	0	0	0	0	0	0	0	0.88	1	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	1	0	0	0.7
	7	0	0	1	0	0	0	0	0	0	0	0.69	0	0.24	0.03
	8	0	0	0	0	0	0	0	0	0	0	0	0	0.93	1
	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0.68

Nonbinary problem set 4 Fuzzy CNN result with 4 cells (Perf.Mea.=0.7048).

		MACHINES													
		1	12	13	6	8	4	5	7	2	3	10	11	9	14
PARTS	6	1	0	0.7	0	0	0	0	0	0	0	0	0	0	0
	7	0.69	0.24	0.03	0	0	0	0	1	0	0	0	0	0	0
	8	0	0.93	1	0	0	0	0	0	0	0	0	0	0	0
	18	0	0	0.68	0	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0.73	1	0	0	0	0	0	0	0	0	0
	11	0	0	0	1	0	0	0	0	0	0	0	0	0	0.24
	12	0	0	0	1	0.85	0	0	0	0	0	0	0	0	0
	14	0	0	0	1	0.3	0	0	0	0	0	0	0	0	0
	16	0	0	0	0.66	1	0	0	0	0	0	0	0	0	0
	22	0	0	0	0.88	1	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	1	0.36	0.72	0	0	0	0	0	0
	2	0	0	0	0	0	1	0.7	0.27	0	0	0	0	0	0
	17	0	0	0	0	0	0.26	1	0.5	0	0	0	0	0	0
	19	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	20	0	0	0	0	0	0.74	1	0.5	0	0	0	0	0	0
	23	0	0	0.46	0	0	0.65	1	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0.7	0.22	0.46	1	0	0
	4	1	0	0	0	0	0	0	0	0	0.5	0	0.82	0	0
	21	0	0	0	0	0	0	0	0	0	0.33	0	1	0	0
	24	0	0	0	0	0	0	0	0	0	0	1	0.76	0	0
	5	0	0	0	0	1	0	0	0	0	0	0	0	0.8	0
	9	0	0	0	0.83	0.57	0	0	0	0	0	0	0	0.28	1
	13	0	0	0	0	0.97	0	0	0	0	0	0	0	0	1
	15	0	0	0	0.8	0.45	0	0	0	0	0	0	0	0.72	1

Nonbinary problem set 4 Fuzzy CNN result with 5 cells (Perf.Mea.=0.7222).

		MACHINES													
		1	7	9	14	6	8	12	13	2	3	10	11	4	5
PARTS	4	1	0	0	0	0	0	0	0	0	0.5	0	0.82	0	0
	6	1	0	0	0	0	0	0	0.7	0	0	0	0	0	0
	7	0.69	1	0	0	0	0	0.24	0.03	0	0	0	0	0	0
	9	0	0	0.28	1	0.83	0.57	0	0	0	0	0	0	0	0
	13	0	0	0	1	0	0.97	0	0	0	0	0	0	0	0
	15	0	0	0.72	1	0.8	0.45	0	0	0	0	0	0	0	0
	5	0	0	0.8	0	0	1	0	0	0	0	0	0	0	0
	10	0	0	0	0	0.73	1	0	0	0	0	0	0	0	0
	11	0	0	0	0.24	1	0	0	0	0	0	0	0	0	0
	12	0	0	0	0	1	0.85	0	0	0	0	0	0	0	0
	14	0	0	0	0	1	0.3	0	0	0	0	0	0	0	0
	16	0	0	0	0	0.66	1	0	0	0	0	0	0	0	0
	22	0	0	0	0	0.88	1	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0.93	1	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0.68	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0.7	0.22	0.46	1	0	0
	21	0	0	0	0	0	0	0	0	0	0.33	0	1	0	0
	24	0	0	0	0	0	0	0	0	0	0	1	0.76	0	0
	1	0	0.72	0	0	0	0	0	0	0	0	0	0	1	0.36
	2	0	0.27	0	0	0	0	0	0	0	0	0	0	1	0.7
	17	0	0.5	0	0	0	0	0	0	0	0	0	0	0.26	1
	19	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	20	0	0.5	0	0	0	0	0	0	0	0	0	0	0.74	1
	23	0	0	0	0	0	0	0	0.46	0	0	0	0	0.65	1

Nonbinary problem set 4 Fuzzy CNN result with 6 cells (Perf.Mea.=0.7407).

		MACHINES																					
		1	3	12	17	18	22	23	24	2	4	10	11	14	16	19	21	5	6	7	8	9	13	15	20
PARTS	1	0.62	1	0	0	0	0.62	0.75	0.88	0	0	0.38	0.75	0	1	0	0	0	0	0	0	0	0	0	0
	4	0.38	0.38	0.88	0	0.75	1	0.5	0.62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0.88	0.62	0.75	0.75	0	0	1	0	0	0	0	0	0	0.25	0	0	0	0	0	0	0	0	0	0
	14	0.22	0	0.56	0.56	0	0	0.33	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	15	0.78	0	0.89	0	0	1	0.89	0.89	0	0	0	0	0	0	0.44	0	0	0	0	0	0	0	0	0.56
	19	1	0.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0
	20	0	0.12	0.38	0.38	0	0	0.88	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	3	0.62	0	0	0	0	0	1	0	0.38	0	0	0.75	0	0.88	0.88	0	0	0	0	0	0	0	0	0
	5	0.75	0	0	0	0	0	0	0	0	0.38	0.75	1	0	0.38	0	0	0	0	0	0	0	0	0	0
	6	0.5	0	0	0	0	0	1	0	0.75	0	0.88	0	0	0.88	0	0	0	0	0	0	0	0	0	0
	12	0.38	0	0	0	0	0	0	0	0.62	0	0.62	0.88	0	1	0.38	0.75	0	0	0	0	0	0	0	0
	13	0.75	0	0	0	0	0	0	0	0	0.75	1	0	0.75	0.5	1	0.38	0	0	0	0	0	0	0	0
	18	0.5	0.5	0	0	0	0	0	0	0	0	1	0	0	0.67	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0.44	0	0	0.33	0	1	0	0	0.89	0.22	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.43	1	0.57	0.43	0	0
	9	0	0	0	0	0	0.88	0.5	0.5	0.5	0	0.38	0	0	0	0.88	0	0.88	0	1	0	0	0	0	0.5
	10	0	0	0	0.33	0.89	0	0.67	0	0.22	0	0	0	0	0	0	0	0	0.22	0.33	0.33	1	0	0	0
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0	1	0	0	0	0	0
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.75	0	0.38	0	0	0	0.88	0
	17	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.43	0	0	1	0.71	0	0	1	0	0.43

Nonbinary problem set 5 Fuzzy CNN result with 3 cells (Perf.Mea.=0.5349).

	MACHINES																							
PARTS	10	11	16	1	3	12	17	22	23	24	6	7	8	9	13	18	2	4	5	14	15	19	20	21
1	0.38	0.75	1	0.62	1	0	0	0.62	0.75	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0.75	0.88	0.62	0	0	0	0	1	0	0	0	0	0	0	0	0.38	0	0	0	0	0.88	0	0
5	0.75	1	0.38	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0.38	0	0	0	0	0	0
6	0.88	0	0.88	0.5	0	0	0	0	1	0	0	0	0	0	0	0	0.75	0	0	0	0	0	0	0
12	0.62	0.88	1	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0.62	0	0	0	0	0.38	0	0.75
13	1	0	0.5	0.75	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0.75	0	1	0	0.38
18	1	0	0.67	0.5	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0.38	0.38	0.88	0	1	0.5	0.62	0	0	0	0	0	0.75	0	0	0	0	0	0	0	0
8	0	0	0.25	0.88	0.62	0.75	0.75	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0.22	0	0.56	0.56	0	0.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0.78	0	0.89	0	1	0.89	0.89	0	0	0	0	0	0	0	0	0	0	0	0.44	0.56	0
19	0	0	0	1	0.83	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0
20	0	0	1	0	0.12	0.38	0.38	0	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0.44	0	0	0	0	0	0	0	0	1	0	0	0.89	0	0	0	0.33	0	0.22	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0.43	1	0.57	0.43	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0.33	0	0.67	0	0.22	0.33	0.33	1	0	0.89	0.22	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0.38	0	0	0	0	0
17	0	0	1	0	0	0	0	0	0	0	1	0.71	0	0	1	0	0	0	0	0	0	0.43	0.43	0
9	0.38	0	0	0	0	0	0	0.88	0.5	0.5	0	1	0	0	0	0	0.5	0	0.88	0	0	0.88	0.5	0
16	0	0	0	0	0	0	0	0	0	0	0	0.38	0	0	0	0	0	0	0.75	0	0.88	1	0	1

Nonbinary problem set 5 Fuzzy CNN result with 4 cells (Perf.Mea.=0.5514).

	MACHINES																							
PARTS	5	7	13	15	12	20	22	23	24	1	3	16	17	2	4	10	11	19	21	6	8	9	14	18
2	0.33	1	0.89	0.22	0	0	0	0	0	0	0	0.44	0	0	0	0	0	0	0	0	0	0	0	0
11	0.38	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0.75	0.38	0	0.88	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
17	0	0.71	1	0	0	0.43	0	0	0	0	0	1	0	0	0	0	0	0.43	0	1	0	0	0	0
4	0	0	0	0	0.88	0	1	0.5	0.62	0.38	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0.75
9	0.88	1	0	0	0	0.5	0.88	0.5	0.5	0	0	0	0	0.5	0	0.38	0	0.88	0	0	0	0	0	0
15	0	0	0	0	0.89	0.56	1	0.89	0.89	0.78	0	0	0	0	0	0	0	0.44	0	0	0	0	0	0
1	0	0	0	0	0	0	0.62	0.75	0.88	0.62	1	1	0	0	0	0.38	0.75	0	0	0	0	0	0	0
8	0	0	0	0	0.75	0	0	1	0	0.88	0.62	0.25	0.75	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0.56	0	0	0.33	0	0.22	0	1	0.56	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0.5	0.5	0.67	0	0	0	1	0	0	0	0	0	0	0	0
19	0.5	0	0	0	0	0	0	0	0	1	0.83	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0.38	0	0	0.88	0	0	0.12	1	0.38	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	0	0.62	0	0.88	0	0.38	0	0	0.75	0.88	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0.75	0	0.38	0	0	0.38	0.75	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	0.5	0	0.88	0	0.75	0	0.88	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0.38	0	1	0	0.62	0	0.62	0.88	0.38	0.75	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0.75	0	0.5	0	0	0.75	1	0	1	0.38	0	0	0	0.75	0
7	0	0.43	0.43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.57	0	0
10	0	0.33	0	0	0	0	0	0.67	0	0	0	0	0.33	0.22	0	0	0	0	0	0.22	0.33	1	0	0.89

Nonbinary problem set 5 Fuzzy CNN result with 5 cells (Perf.Mea.=0.5667).

	MACHINES																							
PARTS	3	12	22	23	24	16	6	7	8	13	20	5	15	19	21	2	9	17	18	1	4	10	11	14
1	1	0	0.62	0.75	0.88	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.62	0	0.38	0.75	0
4	0.38	0.88	1	0.5	0.62	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0.38	0	0	0	0
8	0.62	0.75	0	1	0	0.25	0	0	0	0	0	0	0	0	0	0	0	0.75	0	0.88	0	0	0	0
15	0	0.89	1	0.89	0.89	0	0	0	0	0	0.56	0	0	0.44	0	0	0	0	0	0.78	0	0	0	0
3	0	0	0	1	0	0.88	0	0	0	0	0	0	0	0.88	0	0.38	0	0	0	0.62	0	0	0.75	0
6	0	0	0	1	0	0.88	0	0	0	0	0	0	0	0	0	0.75	0	0	0	0.5	0	0.88	0	0
12	0	0	0	0	0	1	0	0	0	0	0	0	0	0.38	0.75	0.62	0	0	0	0.38	0	0.62	0.88	0
14	0	0.56	0	0.33	0	1	0	0	0	0	0	0	0	0	0	0	0	0.56	0	0.22	0	0	0	0
20	0.12	0.38	0	0.88	0	1	0	0	0	0	0	0	0	0	0	0	0	0.38	0	0	0	0	0	0
2	0	0	0	0	0	0.44	0	1	0	0.89	0	0.33	0.22	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0.43	1	0.43	0	0	0	0	0	0	0.57	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	1	0	0	0	0.38	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	1	1	0.71	0	1	0.43	0	0	0.43	0	0	0	0	0	0	0	0	0	0
9	0	0	0.88	0.5	0.5	0	0	1	0	0	0.5	0.88	0	0.88	0	0.5	0	0	0	0	0	0.38	0	0
16	0	0	0	0	0	0	0	0.38	0	0	0	0.75	0.88	1	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0.67	0	0	0.22	0.33	0.33	0	0	0	0	0	0	0.22	1	0.33	0.89	0	0	0	0	0
5	0	0	0	0	0	0.38	0	0	0	0	0	0	0	0	0	0	0	0	0	0.75	0.38	0.75	1	0
13	0	0	0	0	0	0.5	0	0	0	0	0	0	0	1	0.38	0	0	0	0	0.75	0.75	1	0	0.75
18	0.5	0	0	0	0	0.67	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0	1	0	0
19	0.83	0	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0	1	0	0	0	0

Nonbinary problem set 5 Fuzzy CNN result with 6 cells (Perf.Mea.=0.5684).

		MACHINES																													
		4	13	15	21	23	24	25	28	29	30	2	5	12	14	16	18	20	22	26	27	1	3	6	7	8	9	10	11	17	19
	11	0.2	0.3	0.4	0.5	0.9	0.2	0.5	0.6	0.7	0.8	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0	0	0	0	0	0	0	0
	12	0.6	0.7	0.8	0.9	0.9	0.3	0.5	0.5	0.6	0.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	13	0.7	0.5	0.6	0.8	0.5	0.3	0.4	0.5	0.7	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	14	0.2	0.6	0.8	0.5	0.5	0.4	0.6	0.8	0.2	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	0.5	0.7	0.9	0	0.3	0	0.7	0.9	0.3	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0.4	0.5	0	0	0	0	0	0	0	0.2	0.3	0	0.7	0.6	0.2	0.4	0.4	0.5	0.6	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0.2	0.3	0.4	0.5	0.7	0.8	0.9	0.6	0.8	0.2	0	0	0	0	0	0	0	0	0	0
PARTS	6	0	0	0	0	0	0	0	0	0	0	0.8	0.9	1	0.7	0.2	0.3	0.4	0.5	0.6	0.8	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.8	0.3	0.9	0.2	0.3	0.4	0.5	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0	0.4	0.5	0.6	0.9	0.5	0.6	0.7	0.8	0.9	0.5	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.3	0.6	0.6	0.2	0.2	0.5	0.7	0.4	0.6
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0.5	0.7	0.3	0.4	0.3	0.6	0.8	0.9	0.2
	3	0.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0.6	0.7	0.2	0.4	0.9	0.6	0.2	0.2	0.3	0.5
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0.9	0.3	0.5	0.5	0.7	0.3	0.5	0.6	0.9
	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.2	0.3	0.9	0.2	0.3	0.4	0.5	0.6	0.8

Nonbinary problem set 6 Fuzzy CNN result (Perf.Mea.=0.8613).