AN APPROACH TO THE HYBRID ORGANIZATION OF THE SHOP LAYOUT

(İMALAT ATÖLYESİ YERLEŞİMİNDE
BİR KARMA DÜZENLEME YAKLAŞIMI)

by

Erdem ZENGİN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING

in the

INSTITUTE OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

June 2007

AN APPROACH TO THE HYBRID ORGANIZATION
OF THE SHOP LAYOUT

(İMALAT ATÖLYESİ YERLEŞİMİNDE
BİR KARMA DÜZENLEME YAKLAŞIMI)

by

Erdem ZENGİN, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Date of Submission : May 18, 2007

Date of Defense Examination : June 05, 2007

Supervisor : Asst. Prof. Dr. Orhan FEYZİOĞLU

Committee Members : Assoc. Prof. Dr. Gülçin BÜYÜKÖZKAN FEYZİOĞLU

Assoc. Prof. Dr. Y. İlker TOPÇU (İTÜ)

ACKNOWLEDGEMENTS

In first place, I wish to express my profound gratitude to Professor Orhan Feyzioğlu. More than a mentor and advisor, he was just like a father. He had the patience to support my weak attempts, encouraged me and never gave up providing me necessary guidance that allowed me to finish this venture.

I would like to thank Galatasaray University for providing comfortable learning and research environment.

I also wish to thank my father and my mother for their encouragement and emotional support.

Erdem ZENGİN
June, 2007

TABLE OF CONTENTS

Acknowledgements ii
Table of Contents iii
List of Figures v
List of Tables vi
Abstract vii
Résumé viii
Özet ix
1. Introduction 1
2. Literature Review: Hybrid Manufacturing Systems 3
   2.1. Traditional Types of Manufacturing Facility Layouts 3
   2.2. Performance of Manufacturing Cells and Functional Layout 4
   2.3. Configuring A Hybrid Workshop 9
3. Model Basics and Assumptions 14
   3.1. Fuzzy Sets Theory 14
      3.1.1. Fuzzy Sets and Fuzzy Numbers 14
      3.1.2. Fuzzy Constraints 15
      3.1.3. Constraints Aggregation with Fuzzy Logic 16
      3.1.4. Ordered Weighted Average (OWA) Operator 17
      3.1.5. Fuzzy Inference System 19
   3.2. Similarity of Parts 22
      3.2.1. Levenshtein Distance 24
      3.2.2. Selvam's Similarity Matrix 24
      3.2.3. Vakharia and Wemmerlöv's Similarity Coefficient 25
      3.2.4. A Dissimilarity Coefficient Integrated with Operation Commonality 25
      3.2.5. A Compliant Index based Similarity Coefficient 26
      3.2.6. An LCS based Similarity Coefficient 28

3.2.7. Merger Coefficient 28
3.3. Demand Pattern 29
3.4. Alternative Routing 30
4. Model Formulation 31
4.1. Decision Variables 31
4.2. Constraints 32
4.3. Objective Functions 34
5. Solution Method 40
5.1. Multiobjective Optimization 40
5.1.1. Domination 41
5.1.2. Relation Derived From Domination 44
5.1.3. Tradeoff Surface 47
5.2. Evolutionary Programming 49
5.2.1. Definitions 50
5.2.2. The Vector Evaluated Genetic Algorithm (VEGA) Method 52
5.2.3. The Multiple Objective Genetic Algorithm (MOGA) Method 53
5.2.4. The Nondominated Sorting Genetic Algorithm (NSGA) 54
5.2.5. The Niched Pareto Genetic Algorithm (NPGA) Method 57
5.3. Strength Pareto Evolutionary Algorithm (SPEA2) 58
5.4. Additional Settings Related to the Problem 61
5.4.1. Solution Encoding 61
5.4.2. Mutation 61
5.4.3. Constraints 62
6. Illustrative Example 63
7. Conclusion 67
References 69
Biographical Sketch 75

LIST OF FIGURES

Figure 2.1	: Traditional Types of Manufacturing Facility Layouts	4
Figure 3.1	: Example of Fuzzy Set “About b_i or less”	15
Figure 3.2	: Block Diagram for a Fuzzy Inference System	22
Figure 3.3	: Comparison Process for Compliant Indexes	27
Figure 4.1	: Maximum Similarity – Fuzzy Numbers	36
Figure 4.2	: Coefficient of Variation – Fuzzy Numbers	37
Figure 5.1	: Local Optimality in Pareto Sense	42
Figure 5.2	: Contact Theorem	43
Figure 5.3	: Preference Level and Domination Relation	44
Figure 5.4	: Representation of Trade-off Surface	48
Figure 5.5	: Common Shapes of Tradeoff Surface Considering Two Objective Functions	48
Figure 5.6	: Representation of the Ideal Point and the Nadir Point	49
Figure 6.1	: Representation of the Best Solution	66

LIST OF TABLES

Table 4.1	: Fuzzy If-Then Rules	37
Table 6.1	: Routings, Maximum Similarities, Coefficient of Variations, Demands, Costs of Machining in CM-cells, Costs of Machining in FL-cells for the Illustrative Example	64
Table 6.2	: Unit Processing Times on Machine Types	65
Table 6.3	: Number of Machines of Each Type for the Single Optimum	66

ABSTRACT

While the benefits of organizing machines in form of manufacturing cells are highly praised, certain industrials do not want a comprehensive transformation due to the possible changes that may occur, for example, in the parts' demands or routings. A hybrid manufacturing system that associates functional departments and machine cells may then appear much more attractive given that it would takes advantages of both. Nevertheless, the design of such systems is a much less researched issue compared to the effort spent to the efficient design of each mentioned organization separately. In this work, we suggest an approach that penalizes the allocation of parts that are less stable to manufacturing cells, and highly stable to functional departments. The abstract concept of part stability is associated with two part characteristics: routing similarity and demand variability. In order to determine the similarities between parts, sequence based similarity principles and alternative process routings are considered simultaneously. Meanwhile, demand variability is assumed to be related with the coefficient of variation. While such a discrimination of parts between manufacturing cells and functional departments is desired, the flow of materials in the planned cells should also be appropriate. Suitable machine investment is another issue to consider. Hence, the hybrid manufacturing system design problem is formulated as a constrained multiobjective optimization problem. Evolutionary algorithms present important features for the solution of this type of problems. Among them, a very efficient algorithm which incorporates pareto dominance and elitism, namely SPEA2, is adapted. Both the proposed optimization model and the solution method are illustrated with an artificially generated example.

RESUME

Même si les bénéfices d'organiser l'atelier sous forme des ilots de machines sont largement manifestés, les industriels ne veulent pas une transformation totale à cause des changements probables, comme par exemple la fluctuation de la demande ou changements de gammes. Alors, un atelier hybride où les sections homogènes et les ilots coexisteront pourrait être plus attractif en tant compte qu'il en tirera ses avantages. Malheureusement, la conception du compromis est un sujet moins abordé par rapport aux efforts engagés à la conception efficace des deux systèmes séparément. Dans ce travail, nous proposons une approche qui défavorise l'affectation des produits qui sont moins stable aux ilots, et ceux qui sont très stable aux sections homogènes. Le concept abstrait de stabilité est associé avec deux caractéristiques du produit: similarité de gammes de production et variation de la demande. Pour établir la similarité des produits, les principes concernant la similarité des séquences d'opérations et gammes alternatives sont considérés simultanément. Cependant, on suppose que la variation de la demande est liée au coefficient de variation. Même une telle séparation des produits entre les ilots et sections homogènes est souhaitable, le flux du matériel dans les ilots doit être aussi adéquat. L'investissement dans les nouvelles machines est un autre sujet à envisager. Ainsi, le problème de conception de l'atelier hybride est formulé comme un problème d'optimisation multiobjectif sous contraintes. Les algorithmes évolutionnistes présentent des aspects très intéressants pour la résolution de ce type de problèmes. Parmi eux, un algorithme très efficace qui intègre la dominance Pareto et l'élitisme, ou SPEA2, est adapté. Le modèle d'optimisation proposé ainsi que la méthode de résolution sont illustrés a l'aide d'un exemple artificiel.

ÖZET

Üretim araçlarının imalat hücreleri biçiminde düzenlenmesinin faydaları çokça vurgulanmıştır. Ancak pek çok üretici, ürün talebi veya rotası gibi konularda zamanla oluşabilecek değişikliklerden dolayı toptan bir düzenleme arzu etmemektedir. Bu durumda işlevsel bölümlerin ve imalat hücrelerinin bir arada bulunduğu karma imalat sistemleri daha çekici olabilmektedir. Bununla birlikte, karma imalat sistemlerinin tasarımı konusunda yapılan çalışmalar işlevsel bölümlerin ve imalat hücrelerinin ayrık sistemler olarak ele alınarak tasarımın bu sistemlerden sadece birine göre yapıldığı çalışmalarla karşılaştırıldığında oldukça sınırlı sayıda kalmaktadır. Bu çalışmada, değişken parçaların hücrelerde ve sabit parçaların işlevsel bölümlerde işlenmesinin cezalandırıldığı bir yaklaşım önerilmektedir. Soyut parça değişkenliği kavramı, parçanın iki karakteristik özelliği ile ilişkilidir: parça rotası benzerliği ve talep değişkenliği. Parçalar arasındaki benzerliklerin belirlenmesi için, operasyon sıralarının göz önünde bulundurulduğu benzerlik prensipleri ve alternatif operasyon rotaları bir arada dikkate alınmıştır. Bu arada talep değişkenliğinin de değişkenlik katsayısı ile ilişkili olduğu varsayılmıştır. Ürünlerin bu sayede imalat hücreleri ve işlevsel bölümler arasında paylaşılması istenen bir unsur olmakla beraber, oluşturulan imalat hücreleri içindeki malzeme akışının da düzgün olması gereklidir. İhtiyaç duyulan makineler için yapılan yatırımlar dikkate alınması gereken bir başka unsurdur. Bu nedenle karma imalat sistemleri tasarımı problemi, kısıtlı çok amaçlı eniyileme problemi olarak düzenlenmiştir. Evrimsel algoritmalar bu tipteki problemlerin çözümünde faydalı olan önemli özellikler barındırmaktadır. Bu çalışmada, pareto etkinlik ile seçicilik özelliklerini bir arada bulunduran SPEA2 algoritması etkin bir evrimsel algoritma olarak uyarlanmıştır. Önerilmekte olan eniyileme modeli ve çözüm yöntemi yapay biçimde oluşturulan bir örnekle açıklanmaktadır.

1. INTRODUCTION

To suppress some disadvantages of job shops –where the most notables are complicated part flows and increased waiting times, which in result cause large manufacturing cycle times and production costs– group technology (GT) is proposed. The benefits of cellular manufacturing (CM) systems have been widely reported in GT literature [1, 2, 3, 4, 5]. By organizing independent machine groups a reduction of work in process, a reduction of lead-times and an improvement of quality are expected. On the other hand, in some circumstances –for example where the variation in the product mix and demand pattern is high– cellularized organization can lead to a great disorder in the flow of parts, and severe bottlenecks in some cells while idle machines in others therefore yield poor performances. Many practices have proved that, in some cases, a hybrid system, where both manufacturing cells and functional departments reside within the same shop, can be more efficient [6, 7, 8, 9].

There are few studies concerning the performance of hybrid shops [10, 11, 12, 13], and less attention is paid on how to find intermediate solutions [6, 14, 15, 16, 17]. In this study, we propose a new approach to form hybrid manufacturing systems where CM cells and functional layout (FL) departments can reside at the same shop, and thus intend to achieve the benefits of both organizations.

The proposed approach is mainly inspired by the work due to Viguier and Pierreval [9], but differs from it in many different aspects. The main idea in this work is that parts that are "stable" enough are encouraged to be processed in CM cells while the remaining ones in functional departments. It is argued that the idea of stability can be associated with the parts that are expected to be produced on the long-term, and to the strategic character of certain parts in particular cases [9]. Consequently, an index which aims to measure the part stability is proposed and an objective function is designed to penalize undesired allocations. Meanwhile, not much is explicitly given on how to determine the index value of each part, and most probably, it relies on the shop planner experience. Here, we attempt to better formulize this stability idea by incorporating demand variability and part similarity.

Production volumes, operation times, machine capacities, alternative routings and operational sequences are mentioned as important factors in the design of cellular systems [18, 19]. All of these factors are also taken into account in this study.

This study is organized as follows. Chapter 2 reviews the contemporary literature on the design and performance evaluation of some traditional manufacturing systems, and also various approach to design hybrid manufacturing systems. This design problem is modeled as a constrained multiobjective integer programming problem and Chapter 3 mainly provides the ingredients to arrive such an optimization model. Chapter 4 is devoted to model formulation. As multiobjective optimization problems differ significantly form the mono objective problems, basic definitions are first introduced in Chapter 5. Then, evolutionary programming strategies available in the literature to solve this type of problems are investigated in details. Among them, a superior performance algorithm, namely SPEA2, is further analyzed and the ways how it is adapted to solve our problem are presented. To both illustrate the proposed model and the associate solution method, an artificial example is given in Chapter 6 such that the single feasible shop setup is efficient in all objectives and dominates all remaining solutions. It is shown that the method is able to find this single solution. Finally in the last chapter, conclusions are given and research directions are suggested.

2. LITERATURE REVIEW: HYBRID MANUFACTURING SYSTEMS

2.1. TRADITIONAL TYPES OF MANUFACTURING FACILITY LAYOUTS

Traditionally, three types of layouts are considered appropriate for a manufacturing facility. These are flow line (product), cellular (group) and functional (process) layouts, as shown in Figure 2.1. The difference between these layout patterns is mostly based on the material flow between departments or workstations.

In a flow line manufacturing system, all of the machines and support services required to make a single part (or a family of variants of product) are located in a single department. Flow line layouts have advantages over other types of layouts including high potential production volume, low throughput time, and low work in process inventories. However, product lines cannot be justified for companies that make frequent changes in the production mix since the cost to rearrange machine is significant [19].

In a functional layout, machines with identical manufacturing capabilities are grouped into a single department. The different process-specialized departments are located relative to each other.

A functional manufacturing organization has a high degree of adaptability to changes in the work mix [20]. This manufacturing system is useful for firms that produce a large mix of products at low volumes. Since each part in a process layout is assumed to have its own routing, the challenge is to determine the relative location of departments to reduce the total volume-distance of products. However, it results in complicated part flows and increased waiting times, hence large manufacturing cycle times and production costs [21].

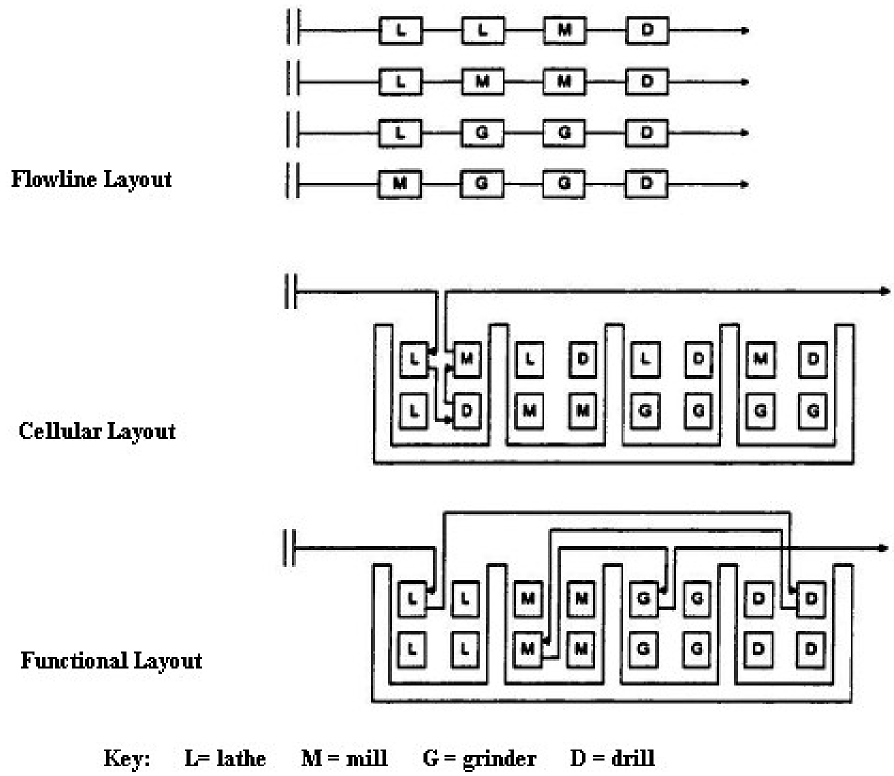

Figure 2.1: Traditional Types of Manufacturing Facility Layouts

To reduce inefficiencies and difficulties arising from the management of large functional job shops, similar products are grouped into part families and production cells with a group of dissimilar machines and processes which are dedicated to produce just this family of parts [9]. In essence, a functional layout has a process focus whereas a cellular layout has a part family focus.

2.2. PERFORMANCE OF MANUFACTURING CELLS AND FUNCTIONAL LAYOUT

In this section, an overview of the researches that focuses on the performances of manufacturing systems is given. There are several studies in which cellular and functional manufacturing system performances are compared. A number of researchers support the relative performance supremacy of cellular layout over functional layout, while others doubt this supremacy.

The pioneers of cellular manufacturing, Burbidge, Durie, Gombinski, Mc Auley, Opitz have promoted the viability of completely independent cells. On the other hand; experiences and practices have made this concept open to discussions [22]. As observed in some practices, a comprehensive transformation to cellular layout is not always advantageous. Conversion to cells can reduce some abilities of the organization. Thus; in some cases, these losses counteract the impact of cellular manufacturing system.

Burgress et al. [11] compare a factory structured as a traditional job shop with the same factory structured as a hybrid factory containing a cellular manufacturing unit. A simulation study is conducted, using combinations of the capacities, allocations of jobs between the cell and the rest of the plant, and levels of productivity improvements achieved in the cell. Performance is evaluated in terms of flow times and delays for the hybrid factory and in a normal factory. The conclusions show that for a cell using 37.5 % of the theoretical capacity (in terms of machines and operators) and loaded with 40 or 45% of the demand the plant experienced lower job flow times than in the traditional job shop. For job mixes of 30% and 35% of cell family parts, the flow times achieved were higher than the traditional Job shop setting. The authors note that the hybrid factory with a manufacturing cell performs better than the traditional job shop. The productivity gains allow the hybrid factory to achieve lower optimum flow times than the traditional JS when the cell is operated at a relatively high operating level than the non-cell work centers. They stress the importance of measuring the overall hybrid factory impact instead of measuring the effectiveness of a cell by itself.

Wemmerlöv and Hyer [23], in their research study, give the reasons that affect the performance of cellular organization. In conclusion he points out numerous factors which threaten independent structure of cells and force inter-cellular part flows among these cells. Some of these factors are machine breakdowns, the need to keep expensive one-off type machines loaded, parts visiting vendors for some operations, changes in part mix or production quantities for part families, non-integer machine requirements of bottleneck machines required by two or more part families, feasibility of using handling systems to make inter-cell moves between adjacent cells, alternative routings for parts when identical machines get duplicated in cells, etc. Hence, the need for a new hybrid manufacturing organization design concept and appropriate algorithms is enounced.

Shambu and Suresh [13] make another comparison. They present a computer simulation investigation of a single data problem with 12 machine types (63 machines in total) and 6 families of parts, covering a pure functional layout (but two scheduling rules are used: traditional job shop and part family oriented). Unlike most of the past researches in group technology, their work examines the entire shop floor, in which the Cellular manufacturing systems have cells and another workshop as a functional layout. They use 5 hybrid systems, from 1 cell dedicated to a part family, to 5 cells dedicated to 5 part families, with the 6th family (formed by a collection of parts with little similarity) processes in the remaining cell (last machines still on the functional layout). Their experimental design employs four experimental factors: (i) SYS, representing the extent of cellularization and group technology application; (ii) d, the degree of setup reduction in cells; (iii) RULE, the dispatching rule used; (iv) Q, lot size. The performance measures used are flow time, work-in-process inventory, machine utilization and flow ratio. Flow ratio is defined by Suresh [13] as the ratio of the flow time of a cellular layout to the time (or work in process) of a functional layout performing at peak performance, and flow ratio is used to report the results of these experiments. In terms of results, for small savings of setup times in the cells, there is little difference between the overall flow ratios among the hybrid systems. They show that part family oriented system significantly outperformed job shop.

The study presented by Molleman et al. [24] concerns a longitudinal case study covering 13 years of experience with a cellular manufacturing system. They consider three factors, which may drive a redesign of the cellular manufacturing system on the impact of redesign decisions on the entire cellular manufacturing system. These factors are the market, manufacturing technology and managerial choices. While many plants were converting the system gradually and implementing cells sequentially, the cells in the selected plant were created in a one-time factory conversation. During the period; changes in customer demands, changing management strategy to strengthen the market orientation of the firm and focusing on decreasing costs led managers to redesign of the cellular manufacturing system not less then nine times. An audit showed advantages as well as disadvantages of the remaining cellular manufacturing system design. Most of the advantages were connected to the controllability of the manufacturing process, such as a good delivery performance, short throughput times and low inventory levels. However, it was striking that there proved to be several disadvantages concerning the quality of working life, machine load and efficiency. Some of the operators no longer viewed the induced job enrichment and job enlargement as an improvement. It was shown that the poor performance of the cutting department was closely related to fluctuating machine loads and low efficiency. Further analysis indicated that most of the problems causing these deficiencies were related to the grouping of resources. Although, according to the existing grouping it is mostly possible to move part types to another cell for a particular process, in practice it is not that easy and requires much coordination effort. These experiences led to the decision to merge some of the cells to remain bigger cells of which the functionality was higher. Approximately by the 13 years' CM experience, as the result, number of cells reduced from 16 to 6. Although management decided to stop producing a large number of parts, the total number of part types grew slightly, from 9053 to 9516 (+5.1%). The total number of machines decreased substantially, from 206 to 111 (−46.1%), as did the number of workers (from 162 to 95; −41.4%). In many situations, the rearrangements were consisting of fusion of cells. As the result of their study, Molleman et al. [24] conclude that a CM system is vulnerable to changes in markets, while functional layouts are less sensitive to such changes. According to the authors, their findings support the statement that a CM system is especially unstable in dynamic markets in which the demand pattern is highly variable. The need of a market-oriented CM design method and a hybrid structure with independent cells and identical machine pools is emphasized.

Asssad et al. [25] compares functional and cellular layout performances across a test bed from the literature by using a single simulation model. They examine many analytical or simulation conversion studies in using flow time as the primary performance measure for comparing job shop and cellular manufacturing layouts. For the simulation model, they suppose that they pay attention to select studies from the literature that provides sufficiently specific information in terms of the parts, machines and operations. They study on examining the benefits of conversion of job shops to cellular layouts. They use the term pooling loss to refer to the reduced efficiency due to segregating into individual cells the resources commonly shared in the job shop. The key trade-off in the study is between pooling loss and set-up reduction. Based on simulation studies, they conclude that cellular manufacturing is not efficient in every case: According to the simulation results, conversation to cells improves flow time. Some additional factors as reduced batch sizes, transfer batches or move times provide a higher performance. As the result, set-up reduction can overcome the effects of pooling loss as long as the magnitude of the set-ups is not too small and no significant bottlenecks are developed in the cells upon conversations. Meanwhile, the tests indicate that conversion to cellular manufacturing systems may not be advantageous if the utilization level is high or if there is not sufficient potential to reduce set-ups. In their study, they show that total cellularization may not always remain effective because of the dynamic nature of environmental factors like demand rate and number of product types. They indicate that the extraction of a few cells from the job shop may provide most of the benefits of full conversion in some conditions. However, they do not identify these conditions in their proposed study.

A comprehensive transformation to cellular layout is not always desirable [6, 10, 26]. Often, small and medium companies are skeptical to adopt a total reorganization, since the variation in their product mix or demand pattern can render the cellular shop no more appropriate. Johnson and Wemmerlöv [26] make a survey research to identify factors that arrest continued implementation of cells in manufacturing plants. They search the factors that prevent firms to continue to cellularization. They conduct a mail survey. Among responding companies; 79 % of them are cell users. About two-thirds of these plants have cell penetration levels of less then 50 %. The reasons for stopping implementation of additional cells are questioned. As the result, many of the companies state four dominant factors. These factors are demand volume, demand stability, service processes and cost justification.

In conclusion, several studies indicate that a new cell design concept and appropriate algorithms, which can enable a company to limit machine grouping and machine duplication by having a semi-functional layout and allowing limited inter-cell flows among the cells, is needed.

2.3. CONFIGURING A HYBRID WORKSHOP

Although there are few studies concerning the performance of hybrid shops, very less attention is paid on how to find comprise solutions.

Balasubramanian and Panneerselvam [27] describe an integrated method that takes into account costs associated with material handling, machine idle times and overtime using a covering heuristic developed for warehouse location problems. Their solution results in the definition of the manufacturing cells, plus a "general pool" of machines (a small job shop) to cover operations not available in some cells. Their proposed approach also ensures the adoption of process layout, if the production volume is low enough, it is capable of defining separate manufacturing cells for each product if the production volume justifies this option, even if the products could be grouped in only two cells by other methods covered in the literature (e.g., rank order clustering). The work just described is only applicable for a stable demand, for a single definition of cells. The scheduling problem of the jobs is not even considered.

Delaney et al. [14] suggest a multistage approach to convert FL shop to a hybrid shop. The first stage consists in decomposing the set of machines in cells and identifying the most appropriate cells. The other stages address the machine layout problem within and between cells. The proposed cell formation procedure aims at minimizing the normalized intercellular traffic. By first putting each machine in one cell, the objective is evaluated, and two cells generating the highest traffic are merged. This procedure is reiterated until the predetermined cell size bound limit further fusion. It is necessary to make two evaluations before implementing the cells proposed at the end of the algorithm: calculating the benefit resulting from the configuration of each cell by evaluating the decrease in intercellular traffic, and estimating the robustness of the proposed organization based on the variation in products and/or the their volume on different time horizon. Then, the most promising cells are select according to this evaluation.

Gravel et al. [15, 16] propose forming cells in a functional shop in the particular case where the majority of products require most of the equipment types available in the shop. Moreover, the same machine type must be used at more than one stage of the manufacturing process. Given that the number of different products is high and the production volumes are low in their case, it becomes difficult to use conventional cell design techniques that try to create product families to be processed within a single cell. The essence of their five-step approach is to assign some common operation sequences of products in cells and the remaining ones to non-cellular part of the shop. The first step requires the identification of frequently occurring machine sequences. Then, candidate cells are formed by selecting longest sequences among the most frequently encountered ones. Operation sequences are divided as necessary by withdrawing operations having durations significantly above an average (bottleneck operations). Next, all sequences are assigned to previously and newly defined cells with specific capacity concerns determined by the authors. Finally, some least busy machines in cells are either shared with non-cellular part of the shop or with another cell. The approach can partially be automated but requires also high user intervention.

Harhalakis et al. [6] pay attention to the fact that cells defined in the past become inappropriate when there are modifications in product routings and/or variation in their demands. The authors define a hybrid shop as the coexistence of cells and individual machines. They assume that there is only one routing for each product, the capacity is sufficient for the production, machines of the same type are identical, and the number and size of the cells are known a priori. Their approach targets the minimization of material handling in the shop and includes four steps: (i) identification of candidate cells, (ii) evaluation and selection of the cells to be implemented, (iii) determination of the intra-cell layout, and (iv) determination of the shop layout. In the first step, cell formation aspects such as the distribution of functionally identical machines among cells and the preservation of setup families are addressed. The second step retains only the cells that yield substantial reductions in material handling. The traffic is calculated based on the long term demand of products. While respecting all physical constraints, the third step determines the layout of cells that minimize a cumulative measure of traffic and distance. Finally, the intercellular layout is determined by considering dimensions of the shop, dimensions of manufacturing resources, restrictions to the shop area and realistic flow paths.

An interesting cost model is described in Needy el al. [28]. Beginning with all parts being processed in a pure job shop environment, an iterative method is used, where parts deemed "similar" are placed into a cell. The total cost of each solution is evaluated (including machine investment, machine setup, and material move costs). The potential solutions range from a pure job shop to a pure cellular configuration, but the authors did not allow for the sharing of resources between cells or between cell and job shop areas like Hybrid Manufacturing Systems do, and concentrate their efforts in a single configuration problem, without including scheduling issues.

Based on their experience, Baker and Maropoulos [17] propose to reorganize only a small part of the shop floor into cells instead of a complete transformation. In other words, their approach creates a hybrid layout where functional departments and manufacturing cells exist concurrently. Without precisely pointing which one, the authors propose some criteria such as inter machine traffic or 80/20 rule to select parts and operations for forming cells (80% of the profit is generated by 20% of the products). With selected products, they create the classical product-machine incidence matrix and manipulate it according to a clustering algorithm so as to obtain a block diagonal structure. Each machine block consists of a set of machines that will form a new cell. However, not taking account to the whole shop, as in this approach, may lead to disastrous results. As an example, a product that is high demanded but not ranked in the top profitable 20% (hence excluded from the cell formation product set) may generate an enormous traffic between new cells.

Irani and Huang [29] introduce the concept of layout as a network of basic modules. Modular layouts are hybrid layouts for systems with complex material flows that cannot be described as functional, flow line, or cellular. In their research on modular layouts; they assume, at least in the short term, a known product mix and fairly stable demand. As the mix and demand change, some modules are eliminated and others added. With such modular layouts, manufacturers can scale their activities up or down quickly. Irani and

Huang [29] seek to answer the following fundamental questions. Can a layout other than the three traditional layouts - perhaps a combination of the three traditional layouts - better fit the material flows of multi-product manufacturers? Can a network of layout modules provide a meta-structure for designing multi-product manufacturing facilities in general? Does grouping and arranging resources into modules corresponding to specific traditional layouts minimize total flow distances or costs? They designed a modular layout for a Motorola facility. The company wants to assess the feasibility of changing the layout in one of their semiconductor fabrication plant from functional to cellular. The functional layout comprises seven bays (or process departments): diffusion, etching, film deposition, implant, photolithography, metrology, and backend. Motorola provided four product routings representative of the plant's product flows. The authors find that a cellular layout is not viable because it requires duplicating equipment and processes. However, a visual string-matching analysis of the routings reveals that different pairs of routings has substrings of operations that are identical or has many operations in common. Based on this observation, they designed a new layout that combines the three traditional layouts. In this layout, all pairs of consecutive operations in all the product routings are performed in the same layout module or in adjacent modules, where a layout module is a group of machines whose flow pattern is characteristic of a traditional layout.

Irani and Huang [30] propose a new method including design of the cascading flow line layout. This layout depends on the routing similarity between parts. The cells are designed such that simple parts get routed to small cells and complex parts get routed to larger, more complex cells. The complexity of a routing is determined by the number of unique work centers (or machines) that are required to make it, regardless of multiple non-consecutive occurrences that could result in flow backtracking. There is no inter-cell movement for any part i.e. every part gets processed in at most one cell. A novel string-to-graph aggregation and planar graph embedding method that allows machine duplication in the layout is introduced. Additionally, the designs of modular and cascading flow lines hybrid manufacturing methods are discussed.

Viguier and Pierreval [9] propose an approach to organize hybrid shops that aims to regroup the parts that will be considered as stable enough into manufacturing cells and the other parts into FL-cells. This idea of stability can be associated with parts that are expected to be produced on the long-term; but it can also be related to the strategic character of certain parts in particular cases. Thus, the hybrid structure tries to combine the well-known advantages of group technology where it is possible, and those of the flexibility of functional departments, when it is more suited. However, stability relation is associated with a stability index and an objective function is designed so as to penalize undesired allocations. Meanwhile, not much is explicitly given on how to determine the index value of each part. Probably, shop planner estimates the degree of stability by choosing between linguistically stated alternatives, which is in fact hard to quantify.

3. MODEL BASICS AND ASSUMPTIONS

3.1. FUZZY SETS THEORY

Fuzzy analysis is an important tool to represent vagueness and a kind of imprecision and uncertainty. The term fuzzy [31] is meant to represent expressions and judgments that have no clear (crisp) value or boundary. For example, when it's needed to express the performance of the personnel as outstanding, this linguistic expression is fuzzy since it cannot be precisely associated with a real number. However, it can always be associated outstanding to an expression like "performance close to 90", which is vague (fuzzy).

3.1.1. Fuzzy Sets and Fuzzy Numbers

A fuzzy set is first introduced by Zadeh [31] and is a generalization of crisp sets. The degree of membership of an individual in a fuzzy set expresses the degree of compatibility of the individual with the concept represented by the fuzzy set. Each fuzzy set, *A*, is defined in terms of a relevant universal set, *X*, by a function called a membership function which assigns to each element *x* of *X* a number *A*(*x*) in the closed interval [0,1]. This characterizes the degree of membership of *x* in *A*. Therefore, membership functions are of the form $\mu_A(x): X \to [0,1]$. Larger values of $\mu_A(x)$ imply higher degrees of set membership. The fuzzy set A is normal, if $\sup_x \mu_A(x) = 1$. The fuzzy set *A* is defined as convex if and only if $\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}$ for all $x_1, x_2 \in \Re^n$ and for all $\lambda \in [0,1]$. A normal and convex fuzzy set defined on $\Re^n$ whose membership function is piecewise continuous is called a fuzzy number. In other words, a fuzzy number represents the conception of a set of real numbers close to $\bar{x}$, where $\bar{x}$ is the number being fuzzified. As an example, a triangular fuzzy number (*l*, *m*, *u*) is expressed as:

$$\mu_A(x)=\begin{cases}0, & x<l \text{ or } x>u\\(x-l)/(m-l) & l\le x\le m\\(u-x)/(u-m) & m\le x\le u\end{cases} \tag{3.1}$$

3.1.2. Fuzzy Constraints

A fuzzy constraint can be given by the following fuzzy inequality $a_i^T x \lesssim b_i$ where a_i^T is the row i of A matrix and b_i is the row i of B matrix. The coefficients of the matrix A are crisp and $\lesssim$ implies that the left hand side of the constraint $a_i^T x$ should be "less or almost equal" to the right hand side b_i. An example of the membership function $\mu_i(a_i^T x)$ for this type of fuzzy inequalities is shown in Figure 3.1.

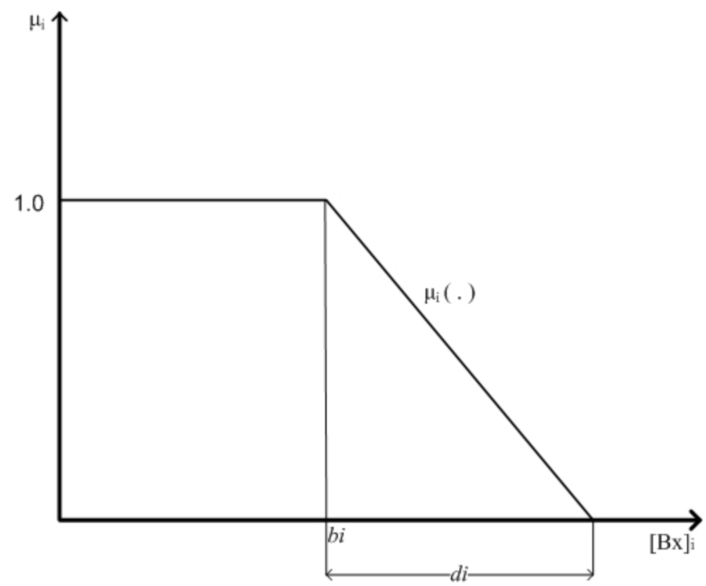

Figure 3.1: Example of Fuzzy Set "About b_i or less"

where d_i is the maximum tolerance for the constraints. The membership function represented by $\mu_i(a_i^T x)$, $i=1,2,...,m$, can be shown as:

$$\mu_i(a_i^T x) = \begin{cases} 1, & a_i^T x < b_i \\ 1 - \dfrac{a_i^T x - b_i}{d_i}, & b_i \le a_i^T x \le b_i + d_i \\ 0, & a_i^T x > b_i + d_i \end{cases} \tag{3.2}$$

3.1.3. Constraint Aggregation with Fuzzy Logic

Weighted aggregation is used quite extensively especially in fuzzy decision making, where the weights are used to represent the relative importance that the decision maker attaches to different decision criteria. Since it is possible to satisfy a constraint partially in fuzzy optimization, the weight factors indicate to what degree various constraints can be interchanged.

In many cases an averaging operator has been used for the weighted aggregation, such as the generalized means, fuzzy integrals or the ordered weighted average operators. Consequently, the weighted aggregation of fuzzy sets has been studied with averaging type of operators.

Weighted aggregation of fuzzy sets considered by Yager [32] is proposed to modify the membership functions with the associated weight factors before the fuzzy aggregation. The weighted aggregation is then the aggregation of the modified membership functions. A generalized form of this idea leads to the weighted aggregation function [33]

$$D(x,w) = T[I(G_1(x),w_1), I(G_2(x),w_2),..., I(G_m(x),w_m)] \tag{3.3}$$

where w is a vector of weight factors $w_i \in [0,1]$, $i = 1,2,...,m$ associated with the aggregated membership functions $G_i(x)$, T is a t-norm and I is a function of two variables that transforms the membership functions. Usually, the power-raising method is used for the transformation and the minimum operator for the t-norm, so the aggregation function becomes

$$D(x,w) = \overset{m}{\underset{i=1}{\wedge}} [G_i(x)]^{w_i} \tag{3.4}$$

Another weighted aggregation function that fits the general scheme (3.3) is given by [34]

$$D(x,w) = \overset{m}{\underset{i=1}{\wedge}} [G_i(x) \vee (1 - w_i)], \tag{3.5}$$

where $\vee$ denotes the maximum operator and the weight factors satisfy

$$\overset{m}{\underset{i=1}{\vee}} w_i = 1 \tag{3.6}$$

3.1.4. Ordered Weighted Average (OWA) Operator

A system needs the knowledge supplied by information sources to be reliable and extend on the whole domain of actuation. However, the information supplied by a single information source is often not reliable enough and/or too narrow in relation to the working domain. In this case, the information provided from several sources can be combined to improve data reliability and accuracy and to include some features that are impossible to perceive with individual sources.

When the objects to synthesize are numeric values (e.g., numbers in the [0, 1] interval) two classical aggregation functions are considered: the arithmetic mean and the weighted mean. Alternatively Yager [35] defined an alternative combination function to synthesize also numeric values—the OWA operator.

Definition 3.1: Let w be a weighting vector of dimension n $(w = [w_1 w_2 ... w_n])$ such that

$$\text{(i) } w_i \in [0,1] \tag{3.7}$$

$$\text{(ii) } \sum_{i=1}^{n} w_i = 1 \tag{3.8}$$

In this case, a mapping $f_{owa}:\mathbb{R}^n \to \mathbb{R}$ is an *Ordered Weighted Averaging (OWA)*operator [35] of dimension *n* if

$$f_{owa}(a_1,\ldots,a_n)=\sum_{i=1}^{n} w_i a_{\sigma(i)} \tag{3.9}$$

where $\{\sigma(1),\ldots,\sigma(n)\}$ is a permutation of $\{1,...,n\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i=2,...,n$. (*i.e.,* $a_{\sigma(i)}$ is the i^{th} largest element in the collection $a_1,...,a_n$).

The OWA operator weights the values, because each w_i is attached to the i^{th} value in decreasing order without considering from which information source the value comes from. OWA operator is commutative. This is, all information sources have an equal contribution to the final solution. With this kind of weight, the OWA operator calculates the output, for example, without considering extreme values ($w_1 = w_n = 0$); or considering only the values that most of the experts give where most is a fuzzy quantifier.

To decide the weighting vector; fuzzy majority is one of the concepts, which can give some meaning to weights. Fuzzy majority is a soft majority concept, which is manipulated via a fuzzy logic-based calculus of linguistically quantified propositions. Two types of linguistic quantifiers can be distinguished: absolute and proportional. Absolute quantifiers are used for representing the amounts that are absolute in nature such as about 2 or more than 5. These absolute linguistic quantifiers are closely related to the concept of the count or number of the elements and for any given $x \in \Re_+$, the membership degree $Q(x)$ of x indicates the degree of which the amount x is compatible with the quantifier represented by Q which is a fuzzy subset. Proportional quantifiers, such as most, at least half may be represented by fuzzy subsets of the unit interval [0,1]. Then for any $x \in [0,1]$, $Q(x)$ indicates the degree to which the proportion x is compatible with the meaning of the quantifier it represents.

The weights can be calculated by Yager's method [35]. The identification of the weights of the OWA-type operators is an important concern and there exist numerous methods to assist. For a non decreasing relative quantifier, *Q*, the weights are obtained as follows:

$$w_i = Q\left(\frac{i}{m}\right) - Q\left(\frac{i-1}{m}\right), \quad i = 1, 2, ..., m \tag{3.10}$$

where *Q* is defined as

$$Q(x) = \begin{cases} 0 & \text{if} \quad x<a \\ \dfrac{x-a}{b-a} & \text{if} \quad a \le x<b \\ 1 & \text{if} \quad x \ge b \end{cases} \tag{3.11}$$

with $a, b, x \in [0,1]$ and $Q(x)$ indicating the degree of which the proportion x is compatible with the meaning of the quantifier it represents. Some examples for the relative quantifiers are "most" (0.3, 0.8), "at least half" (0, 0.5) and "as many as possible" (0.5, 1).

3.1.5. Fuzzy Inference System

Fuzzy Logic and Linguistic Variables: Fuzzy sets provide quantitative methods to express linguistic variables using the precepts of mathematics [36]. As an example, a linguistic term and its quantitative interpretation is shown. Suppose that a specific term "Young" in the universe of natural language is wanted to express in terms of age, y, by a membership function that expresses the term young. A is a fuzzy set and might be one interpretation of the term young expressed as a function of age,

$$A = \text{"young"} = \int_0^{25} \frac{1}{y} + \int_{25}^{100} \frac{1}{y}\left(1+\left(\frac{y-25}{5}\right)^2\right)^{-1} \tag{3.12 }$$

or alternatively,

$$\mu_M(\text{young, y}) = \begin{cases} \left(1+\left(\dfrac{y-25}{5}\right)^2\right)^{-1} & y > 25 \text{ years} \\ 1 & y \leq 25 \text{ years} \end{cases} \quad (3.13)$$

IF-THEN rule based form is one of the most common ways to represent human knowledge by forming it into the type of language expressions,

$$\text{IF premise (antecedent), THEN conclusion (consequent)} \quad (3.14)$$

IF-THEN can be expressed as,

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B, \quad (3.15)$$

where *A* and *B* represent fuzzy propositions.

A rule-based system can involve more than one rule. The process of obtaining the overall consequent (conclusion) from the individual consequents contributed by each rule in the rule-base is known as aggregation rules [37].

Conjunctive system of rules is one of the aggregation rules. In the case of a system of rules that must be jointly satisfied, the rules are connected by "and" connectives. In this case of aggregated output (consequent), *y*, is found by the fuzzy intersection all individual rule consequents, y^i where i = 1, 2, …, r as

$$y = y^1 \text{ and } y^2 \text{ and ... } y^r \quad (3.16)$$

or

$$y = y^1 \cap y^2 \cap \ldots \cap y^r \quad (3.17)$$

defined by the membership function

$$\mu_y(y) = \min(\mu_{y^1}(y), \mu_{y^2}(y)) \text{ for } y \in Y \quad (3.18)$$

Composition: Let R be a relation that relates, or maps elements from universe X to universe Y, and let S be a relation that relates, or maps, elements from universe Y to universe Z. A relation T can be found that relates the same elements in the universe X that R contains to the same elements in Z that S contains.

Suppose R is a fuzzy relation on the cartesian space $X \times Y$, S is a fuzzy relation on $Y \times Z$, and T is a fuzzy relation on $X \times Z$; then fuzzy max-min composition is defined in terms of the set-theoretic notation and membership function-theoretic notation in the following manner:

$$T = R \circ S$$
$$\mu_T(x,z) = \vee_{y \in Y} (\mu_R(x,y) \wedge \mu_S(y,z)) \tag{3.19}$$

Defuzzification: For some situations outputs of a fuzzy process needs to be a single quantity as opposed to a fuzzy set. Defuzzification is the conversion of a fuzzy quantity to a precise quantity. Centroid method (also called center of area, center of gravity) is one of the most prevalent and physically appealing of all the defuzzification methods. It is given by the algebraic expression

$$Z^* = \frac{\int \mu_Ç(Z).Zdz}{\int \mu_Ç(Z)dz} \tag{3.20}$$

The Mamdani Fuzzy Rule Based System: The Mamdani controller is constituted of a group of rules and fuzzy sets in which the inference mechanism is the Mamdani one [38]. Designing a Mamdani rule base requires three steps: (i) determine appropriate fuzzy sets over the input domain and output range; (ii) determine a set of rules between the fuzzy inputs and the fuzzy outputs that model system behavior; (iii) create a framework that maps crisp inputs to crisp outputs, given (i) and (ii).
The operation of the Mamdani rule base can be broken down into four parts: (i) mapping each of the crisp inputs into a fuzzy variable (fuzzification); (ii) determining the output of

each rule given its fuzzy antecedents; (iii) determining the aggregate output(s) of all of the fuzzy rules; (iv) mapping the fuzzy output(s) to crisp output(s) (defuzzification). Figure 3.2 illustrates the operation steps of a fuzzy inference system.

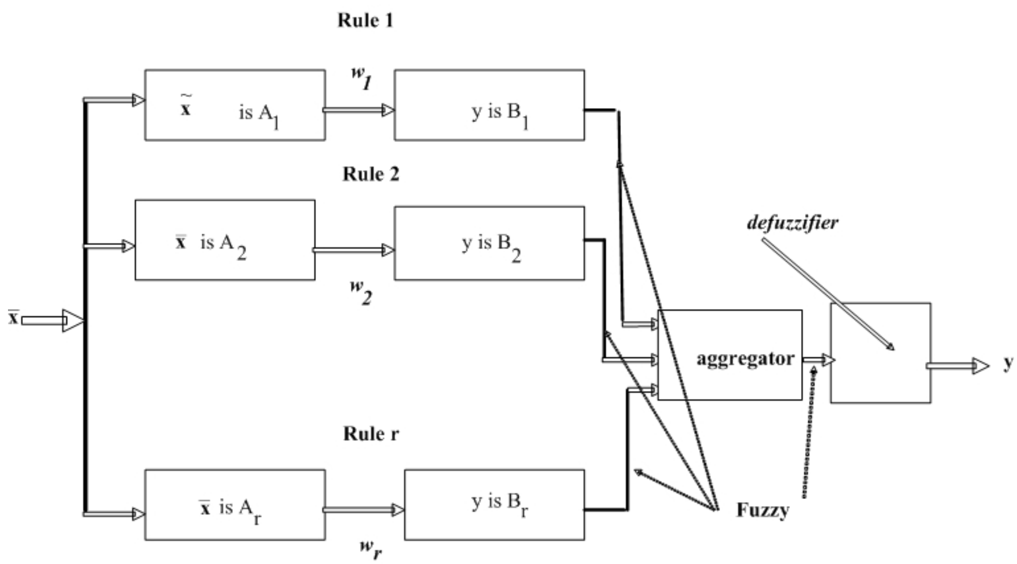

Figure 3.2: Block diagram for a fuzzy inference system

The implication of a fuzzy Mamdani system may be characterized by a rule of the kind:

IF [premise 1 is qualification 1] AND [...is...] THEN [consequent is qualification] (3.21)

The formulization of the implication of the kind and aggregated output is given by:

$$\mu_{B^k}(y) = \max[\min\{\max[\mu_{A_1^k}(x) \wedge \mu(x_1)], \max[\mu_{A_2^k}(x) \wedge \mu(x_2)]\}] \tag{3.22}$$

3.2. SIMILARITY OF PARTS

By the exploration of group technology; cell formation problem become one of the most important issues of cellular manufacturing systems [39].

The cell formation problem can be extremely complex, because of various different production factors, such as alternative process routings, operational sequences, production volumes, machine capacities, tooling times and others, need to be considered. Many approaches and methods have been developed to solve the cell formation problems. These approaches can be classified into three groups: (i) Mathematical Programming Models, (ii) (meta-) Heuristic Algorithms, (iii) Similarity Coefficient Methods.

Among these approaches, similarity coefficient method is the application of cluster analysis to cell formation procedures. Since the basic idea of group technology depends on the estimation of the similarities between part pairs, the cluster analysis is the most basic method for estimating similarities, it is concluded that similarity based method is one of the most basic cell formation method and more flexible in incorporating manufacturing data into the machine-cells formation process than the others [40].

Similarity coefficient measures the similarity between two parts' attributes. According to the similarity coefficient, the parts, which have a larger value of the similarity coefficient, can be grouped into the same part-family efficiently. Therefore, creation of an effective similarity or dissimilarity coefficient is very important for the success of the similarity based method.

The operation sequence is defined as an ordering of the machines on which the part is sequentially processed [39]. According to Choobineh [41] the achievement of the purpose and objectives of the group technology task in a production system is dependent on the choice of machines and impact of the materials flow. However, only considering the machine requirements can just reflect the choice of the machines. It is obvious that just considering the machine requirement cannot reflect the impact of materials flow. Therefore, in recent years, operation sequence, a very important factor in the manufacturing system, is emphasized by some researchers, because it can reflect not only the machine requirements of the parts but also the flow pattern of the parts. On the other hand, the parts do not have a direct relationship with the machines, but, actually, each part and machine is associated with a set of operations [42]. Therefore, considering the operation sequence in the cell formation problem is realistic to integrate the choice of the machine and the impact of the materials flow in group technology successfully.

3.2.1. Levenshtein Distance

Levenshtein distance has been the most commonly used measure for comparison of the difference (similarity) of two operation sequences. Levenshtein distance is defined as the smallest number of substitutions, deletions and insertions required to change source sequence into target sequence [43]. It uses listings as the acceptable analyses of sequence difference and a simple length function without weights or parameters. Considering all listings from source sequence to target sequence, let the length of each listing be the number of transformations it contains, then the distance is the minimum length of any listing.

3.2.2. Selvam's Similarity Matrix

Selvam and Balasubramanian [44] defined a similarity matrix based on the operation sequences and inter-machine movements to solve the cell formation problem. The values of this similarity matrix reflect the similarity between the parts and the main part that have been chosen before the cell formation procedure. The concept of the main part is defined as the part based on whose operation sequence the production line is set up, and this production line is defined as the main line. According to Selvam's definition of this similarity matrix, the value of the similarity matrix S can be represented by the following mathematical formula:

$$S = [S_{ij}] = D_i M_{ij} \tag{3.23}$$

where S_{ij} is the i^{th} row and j^{th} column value of the similarity matrix, in which row indicates parts and column indicates main lines; D_i is the production volume per period for part i; M_{ij} is the number of movements when part i is processed through the main line j. It makes this similarity coefficient reasonable on considering the demand and materials movements simultaneously. The value of this similarity coefficient depends on the selection of main part and main line.

3.2.3. Vakharia and Wemmerlöv's Similarity Coefficient

Vakharia and Wemmerlöv [39] developed a square, symmetric similarity matrix S for the set of the parts group that their operation sequences do not contain backtracking. This similarity matrix can reflect the proportion of machine types used by two part groups in the same sequence. In their method, according to this similarity measure, any two part-groups are identified for potential merging. This similarity measure is defined as:

$$S_{pq} = \frac{1}{2}\left(\frac{\sum_{i \in C_{pq}} A_{ip}}{\sum_{i=1}^{M} A_{ip}} + \frac{\sum_{i \in C_{pq}} A_{ip}}{\sum_{i=1}^{M} A_{iq}} \right) \quad (3.24)$$

In this formula, S_{pq} is the similarity between part groups p and q; i =1, 2, ..., M is machine type index; A_{ip} is equal to 1 if the parts in part-group p need to be processed on machine type i, otherwise, A_{ip} is equal to 0; C_{pq} is the set that contains the machine types which exist in both the composite operation sequences of part-group p and q in the same relative order. The range of this similarity coefficient is (0-1). Bounded and symmetry are two important characters of this coefficient, and make it attractive. Based on this similarity coefficient, a clustering procedure is applied to form the part-families.

3.2.4. A Dissimilarity Coefficient integrated with Operation Commonality

Tam [42] presented a dissimilarity coefficient method, based on the difference between operation sequences, for part grouping. In the method used, the dissimilarity coefficient denoted by S_c, of two operation sequences i and j is defined as follows:

$$S_c[i, j] = w_n d_n[i, j] + w_c(1 - c[i, j]) \quad (3.25)$$

$$w_n + w_c = 1 \text{ and } w_n, w_c \geq 0 \tag{3.26}$$

$d_n[i, j]$ is the normalized weighted Levenshtein distance between sequences i and j:

$$d_n[i, j] = \frac{d_w[i, j]}{\max\{d_w[x, y] \mid 1 \leq x, y, \leq \text{number of parts}\}} \tag{3.27}$$

$$d_w[i, j] = \min(w_z n_z + w_d n_d + w_i n_i) \tag{3.28}$$

where w_z, w_d and w_i are non-negative weights assigned to substitutions, deletions and insertions respectively; n_z, n_d and n_i are number of substitutions, deletions and insertions respectively.

$c[i,j]$ is a coefficient representing the commonality of operations between operation sequences i and j. It is defined as the number of common operations between operation sequences i and j divided by the total number of the distinct operations in these two sequences.

3.2.5. A Compliant Index based Similarity Coefficient

Ho et al. [45] proposed a heuristic pattern matching algorithm for multi-flow lines layout design. In the proposed algorithm, a similarity coefficient approach is used to compare the difference (similarity) between an operation sequence and a flow path. In order to calculate the sequence similarity coefficient, a compliant index based on trace analyses of differences between operation sequences is defined first.

The compliant index of the sequence of a part compared with a flow path is determined by the number of operations in the sequence that have either "in-sequence" or "bypassing" relationship with the sequence of the flow path. There are two kinds of compliant indexes: forward compliant index and backward compliant index. These two compliant indexes can be calculated by comparing the operation sequence of the part with the sequence of the flow path forwards and backwards. The process of calculating these two compliant indexes is illustrated in Figure 3.3

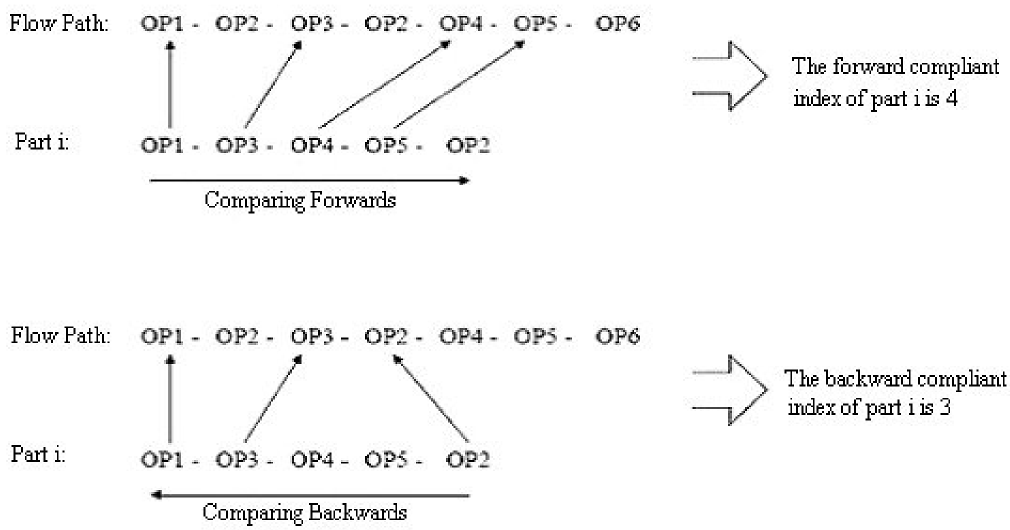

Figure 3.3: Comparison Process for Compliant Indexes

Once the compliant indexes of both directions have been calculated, the sequence similarity coefficient of this part can be calculated by dividing the sum of both compliant indexes by twice the number of operations in this part:

$$CO = \frac{CF + CB}{2N} \tag{3.29}$$

where

CO: Sequence similarity coefficient of part

CF: Compliant index of the part in forward direction

CB: Compliant index of the part in backward direction

N: Number of operations in the sequence of the part

Obviously, a part with higher sequence similarity coefficient means its sequence is more similar to the sequence of the flow path.

3.2.6. An LCS based Similarity Coefficient

Askin and Zhou [46] proposed a similarity coefficient based on the longest common subsequence (LCS) between parts for forming flow line manufacturing cells. Like Ho et al's [48] similarity measure, this coefficient is also based on trace analyses of operation sequence differences. The similarity coefficient S_{ij} between two operation sequences O_i and Oj is defined as:

$$S_{ij} = \max\left\{\frac{|LCS_{ij}|}{|O_i|}, \frac{|LCS_{ij}|}{|O_j|}\right\} \tag{3.30}$$

where LCS_{ij} is the longest common subsequence between O_i and O_j, and $|x|$ is the number of operations in sequence x.

3.2.7. Merger Coefficient

Irani and Huang [29] proposed a new similarity measure for string clustering called merger coefficient which evaluates the feasibility of merging or absorbing one operation sequence completely into another operation sequence.

In order to calculate the merger coefficient between two operation sequences, the merger distance and the interruption distance need to be defined first. In terms of the three types of transformations for deriving one operation sequence from another operation sequence -substitution, deletion, and insertion- the merger distance for the absorption of sequence x into sequence y, denoted by $md(x,y)$, is defined as the smallest number of substitutions and insertions required to derive x from y. Keeping $md(x,y)$ fixed, the smallest number of deletions required between two consecutive basic transformations, between two consecutive matching operations, and between two consecutive transformation and matching operation, is defined as the interruption distance for the absorption of x into y, denoted by $id(x,y)$.

Once the merger distances and interruption distances between any two operation sequences *x* and *y* have been identified, the merger coefficient between *x* and *y*, denoted by *mc*(*x,y*), can be calculated as follows:

$$
\begin{aligned}
& mc(x, y) \\
& = mc(y, x) \\
& = \begin{cases}
1 - \dfrac{md(y,x) + \dfrac{id(y,x)}{N_x}}{N_y + 1}, & \text{if } N_x > N_y \\[2ex]
1 - \dfrac{md(x,y) + \dfrac{id(x,y)}{N_y}}{N_x + 1}, & \text{if } N_x < N_y \\[2ex]
\max\left(1 - \dfrac{md(y,x) + \dfrac{id(y,x)}{N_x}}{N_y + 1}, 1 - \dfrac{md(x,y) + \dfrac{id(x,y)}{N_y}}{N_x + 1} \right) & \text{if } N_x = N_y
\end{cases}
\end{aligned}
\qquad (3.31)
$$

where N_x and N_y represent the number of operations in sequences *x* and *y*, respectively. Note that $0 < mc(x, y) \leq 1$. The higher the merger coefficient between two operation sequences, the more similar are they.

3.3. DEMAND PATTERN

Industry has for many years been dealing with the problem of making batch production more efficient and responsive to changes in demand and technology. Manufacturing cells dedicated to families of parts significantly reduce the problem of set-up time. However, conversation to cells can result in a reduced ability to respond to changes in demand and range of parts produced since machines, fixtures, tooling and people are dedicated to product families. Dedicating machines to specific parts typically results some inconsistencies since the shop cannot respond to even short-term fluctuations in demand. These kinds of inconsistencies with the capacities of cells lead to severe bottlenecks in some cells and idle machines in others [47, 48, 49, 50]. Thus, variability in the parts' demand is an important criterion for layout design. In such cases coefficient of variation of demand is used as a demand stability/volatility indicator [51].

Coefficient of variation of demand is the ratio of the standard deviation (σ_d) to the mean ($\bar{d}$) of the demand of a product. If coefficient of variation is high, this would suggest that σ_d is very significant compared to $\bar{d}$, and that the demand is highly variable/volatile.

$$CV = \sigma_d / \bar{d} \tag{3.32}$$

3.4. ALTERNATIVE ROUTING

In most manufacturing system design methods, parts are assumed to have a unique part process plan. However, it is well known that alternatives may exist in any level of a process plan. In some cases, there may be several alternatives for making a specific part, especially when the part is complex [52].

Consideration of alternative process plans generally complicates the grouping problem, but lower capital investment in machines, more independent manufacturing cells and higher machine utilization can be achieved [53, 54]. A number of methods are proposed for cell formation in the presence of alternative routing under the Group Technology approaches [55, 56, 57, 58].

Alternative routing arises from two sources: there may be multiple process plans for each part type and independently, there may be multiple machines of each type [59].

For the multiple machining cases the process is considered as operation sequence of machine types [55, 56, 57, 58]. In this case, the number of identical machines of each type is generally given and the problem is to allocate each operation to a machine of the required type.

4. MODEL FORMULATION

Based on the previous discussion, we now formulate the hybrid manufacturing design problem in this chapter. Lets define a CM-cell as a group of machines of dissimilar types, if possible, dedicated to the production of a set of parts with similar processing requirements. This group has to be defined such that a given set of parts can be completed or nearly completed within this group of machines. A FL-cell is a shop department which consists of machines having a common technological capability. Then, a hybrid manufacturing system is assumed to contain both FL and CM-cells concurrently.

4.1. DECISION VARIABLES

Our problem consists of designing a hybrid manufacturing system that is a good compromise between the functional layout and CM layout. We consider that the number of CM-cells is not given and must be determined by the method. To better reflect the real situation, we assume that each part's operation can be carried by identical machines of a given type. Again with practical concerns, we only upper bound the number of similar machines.

Let I $(i = 1, \ldots, I)$ be the number of distinct machine types and J $(j = 1, \ldots, J)$ be the number of parts. Each part is assumed to have K_j $(k = 1, \ldots, K_j)$ operations where each operation can be handled by only one machine type denoted as I_{jk}. As mentioned previously, there can be several machines of the same type within a cell or in different cells. However, the number of FL-cells is known since we can only have a single FL-cell for each machine type. For notational and computational convenience, we assume without loss of generality that all FL-cells reside within the cell $c = 0$ and the number of CM-cells, *C*, is set to a large number $(c = 0, 1, \ldots, C)$.

To design the system according to a hybrid organization, we have to identify which cells should be opened and allocate machines to those cells. The following variables are defined to represent above decisions:

x_{ic} = number of type i machines in cell c

$$x'_{ic} = \begin{cases} 1, & \text{if } x_{ic} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

$$z_c = \begin{cases} 1, & \text{if cell } c \text{ is opened} \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

$$y_{jkic} = \begin{cases} 1, & \text{if operation } k \text{ of part } j \text{ is completed} \\ & \text{with a machine of type } i \text{ in cell } c \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

4.2. CONSTRAINTS

Let d_j be the demand of part *j*, t_{jki} be the unit processing time of part *j* operation *k* on any machine of type *i*, and b_i be the capacity (in time units) of a machine of type *i*. Then, the first set of constraints (4.4) is related with the capacity of the machines. They guarantee that there are enough duplicated machines in each cell.

$$\sum_{j=1}^{J} \sum_{k=1}^{K_j} d_j t_{jki} y_{jkic} \leq b_i x_{ic} \qquad i = 1, \ldots, I; c = 0, \ldots, C. \tag{4.4}$$

The constraint set (4.5) assures that each operation is surely assigned. Note that the overall operation's workload can be split among identical machines as long as they reside within the same cell.

$$\sum_{c=0}^{C} \sum_{i \in I_{jk}} y_{jkic} = 1 \qquad j = 1, \ldots, J; k = 1, \ldots, K_j. \tag{4.5}$$

Equations (4.6) ensure that no operation is assigned to unopened cells.

$$\sum_{j=1}^{J} \sum_{k=1}^{K_j} \sum_{i \in I_{jk}} y_{jkic} \le M z_c \qquad c = 0, \ldots, C. \tag{4.6}$$

Here *M* is a big number. We also want a CM-cell to contain at least two different machine types, since otherwise machines of a single type would be assigned to an FL-cell. The set of constraints (4.7) and (4.8) imply this idea.

$$x_{ic} \le M x'_{ic} \qquad i = 1, \ldots, I; c = 1, \ldots, C \tag{4.7}$$

$$\sum_{i=1}^{I} x'_{ic} \ge 2 z_c \qquad c = 1, \ldots, C. \tag{4.8}$$

The last two constraint sets involve bounds related to the design: (4.9) limits CM-cell sizes (i.e. the number of machines that a CM-cell can contain) and (4.10) limits the number of similar machines. These bounds are, of course, not easy to quantify. A CM-cell, for example, can have one more machine than the limit set. The result may be less agreeable but also acceptable at some extent. Therefore, instead of hardly bounding cell sizes and machine numbers, we can allow to violate them but with a decreasing satisfaction level. There exists, of course, a maximal value that the design would be judged not satisfactory at all. This idea can be implemented with the help of the fuzzy set theory. Let α and β_i be the soft bounds on the CM-cell size and the number of type *i* machines respectively. If we define

$$\bar{x}_c = \sum_{i=1}^{I} x_{ic} \lesssim \alpha z_c \qquad c = 1, \ldots, C \tag{4.9}$$

and

$$\bar{\bar{x}}_i = \sum\nolimits_{c=0}^{C} x_{ic} \lesssim \beta_i \qquad i = 1,\ldots,I, \tag{4.10}$$

then the related membership functions can have the following forms:

$$\mu_{\bar{x}_c} = \begin{cases} 1, & \bar{x}_c < \alpha^1 \\ \dfrac{\alpha^2 - \bar{x}_c}{\alpha^2 - \alpha^1}, & \alpha^1 \le \bar{x}_c \le \alpha^2 \quad c = 1,\ldots,C \\ 0, & \bar{x}_c > \alpha^2 \end{cases} \tag{4.11}$$

$$\mu_{\bar{\bar{x}}_c} = \begin{cases} 1, & \bar{\bar{x}}_c < \beta_i^1 \\ \dfrac{\beta_i^2 - \bar{\bar{x}}_c}{\beta_i^2 - \beta_i^1}, & \beta_i^1 \le \bar{\bar{x}}_c \le \beta_i^2 \quad i = 1,\ldots,J \\ 0, & \bar{\bar{x}}_c > \beta_i^2 \end{cases} \tag{4.12}$$

In other words, the constraints (4.9) and (4.10) will be considered completely satisfied if their LHS are less than α^1 and β_i^1 $i = 1,\ldots,I$ (acceptable levels), and not at all satisfied if greater than α^2 and β_i^2 $i = 1,\ldots,I$ (unacceptable levels), respectively, and decreasing satisfied in between.

4.3. OBJECTIVE FUNCTIONS

Since we aim to process stable parts in CM-cells and non-stable parts in FL-cells, we have to first define how the stability can be identified. In this study, stability is associated with the maximum similarity (*MS*) of a part to all other remaining parts, and the coefficient of variation (*CV*) of the demand of that part. As CM-cells are designed to process similar type of products by definition, *MS* is an important discrimination factor. If *MS* is high, then this would imply that there exist other parts similar to the one investigated that we can put together in a cell. Otherwise, if *MS* is low, then the part would hardly fit in a well defined CM-cell. *CV* is another factor to measure what the demand of a part is likely to be. From previous sections we know that if *CV* is high, the demand is highly variable and processing this type of part(s) in a CM-cell is not feasible. Therefore, it is more desirable to route these products to FL-cells. As we want to penalize inadequate allocation of products to cells, we will incur a high cost for machining parts with low *MS* and high *CV*, in other words instable parts, in CM-cells, and again a high cost for machining parts with high *MS* and low *CV*, in other words stable parts, in FL-cells.

While *CV* can be figured out with a readily available demand data, maximum similarity needs to be defined. Similarity index (SI) measures the similarity between two parts' attributes. According to the index, the parts which have a larger value of the similarity index can be grouped into the same part-family efficiently. Therefore, creation of an effective SI is very important for the success of the similarity based method. Here, we preferred to work with the operation sequence based SI due to Ho et al. [45], since the sequence can reflect not only the machine requirements of the parts but also the flow pattern. The similarity index proposed in [45] is well suited for the cell formation problem [60], but also posses some deficiencies. For example, SI value of the sequences ABC and DEAFGBHIJCKL is 1 (maximum similarity value). While the first sequence is a subsequence of the second, it is clearly difficult to assess such high similarity. Meanwhile, sequences ABCD and EFBGCHI have a similarity value of 0.5, which we believe not very realistic too. To overcome this type of situations, we implemented the following procedure:

compute SI with the method proposed in [45];

if SI < 1 then

$$\text{SI} = \text{SI} \times (2 \times a)/(2 \times a + b + c)\ ; \tag{4.13}$$

otherwise

$$\text{SI} = 1 - ((m - n)/m)\,/(n + 1); \tag{4.14}$$

where *a* is the number of common machines in two sequence, *b* is the number of machines belonging only to one of the sequences and *c* is the number of machines belonging only to the other sequence, *m* is the size of the longer sequence, and *n* is the size of the sorter sequence. Consequently, we have a similarity value of 0.8125 for the first mentioned example, while 0.1818 for the second example. The maximum similarity value *MS* will be the maximum among all pair wise comparison of a given part, i.e.

$$MS_j = \max_{j' \neq j,\, j'=1,\ldots,J} \mathrm{SI}_{jj'} \quad j = 1, \ldots, J. \tag{4.15}$$

Meanwhile, it is not easy to directly quantify whether *MS* and/or *CV* values are high or low and the resulting cost of allocation too. On this line, fuzzy set theory is a widely used and useful tool. In this work, fuzzy implications are carried out with Mamdani's fuzzy implication and the max-min composition rule [38].

All categories including *MS*, *CV*, the cost of assigning a product *j* to a CM-cell (c_j^{CMC}) and the cost of assigning a product *j* to a FL-cell (c_j^{CFC}) will be composed of five linguistic variables: Very Low (VL), Low (L), Moderate (M), High (H) and Very High (VH). The fuzzy numbers which model these situations for *MS* and *CV* is given in Figure 4.1 and Figure 4.2. *MS* and *CV* are the rule antecedents and, c_j^{CFC} and c_j^{CMC} are rule consequence. The corresponding rules are given in Table 4.1.

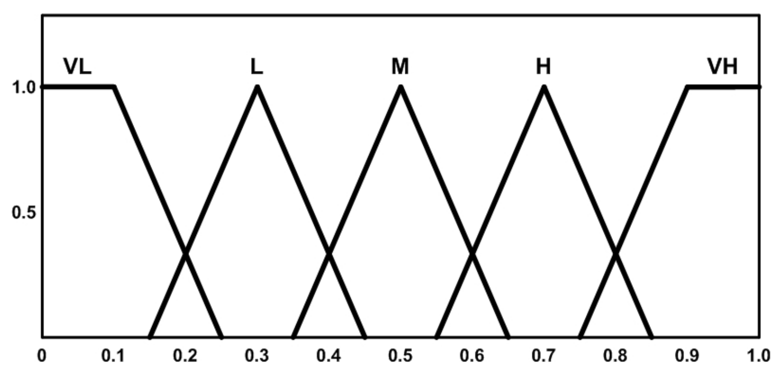

Figure 4.1: Maximum Similarity – Fuzzy numbers

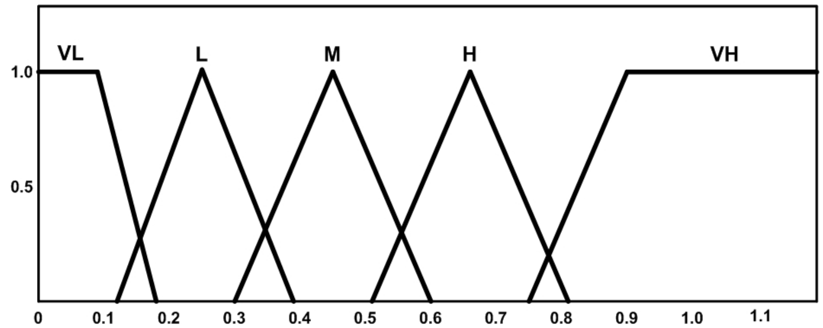

Figure 4.2: Coefficient of Variation – Fuzzy Numbers

We are now ready to give our first objective function related to the operation allocation costs of cells:

$$\min\ f_1 = \frac{\sum_{j=1}^{n} c_j^{CFC} d_j \sum_{k=1}^{K_j} \sum_{i\in I_{jk}} y_{ikj0} + \sum_{j=1}^{J} c_j^{CMC} d_j \sum_{k=1}^{K_j} \sum_{i\in I_{jk}} \sum_{c=1}^{C} y_{ikjc}}{\sum_{j=1}^{J} \sum_{k=1}^{K_j} \sum_{i\in I_{jk}} \sum_{c=0}^{C} d_j y_{ikjc}} \tag{4.16}$$

Table 4.1: Fuzzy If-Then Rules

c_j^{CMC}		Coefficient of Variation (*CV*) VL	L	M	H	VH
Maximum Similarity (*MS*)	VH	VL	VL	L	M	M
	H	VL	L	L	M	H
	M	L	M	M	M	H
	L	L	M	M	H	VH
	VL	M	M	H	VH	VH

c_j^{CFC}		Coefficient of Variation (*CV*) VL	L	M	H	VH
Maximum Similarity (*MS*)	VH	VH	VH	H	M	M
	H	VH	H	H	M	L
	M	H	M	M	M	L
	H	H	M	M	L	VL
	VL	M	M	L	VL	VL

We also want well defined CM-cells [5]. At this stage, several criteria are suggested to solve the cell formation problem [61]. A suitable criterion in our case can be the amount of pieces that is transferred immediately between two machines of a cell, or the intracellular transfer. If all the operations of the parts assigned to a cell are handled only with the machines residing in that cell, then we would have an independent cell as desired. Let's define

$$\gamma_{jkc} = \begin{cases} 1, & \sum_{i \in I_{jk}} y_{jkic} > 0 \\ 0, & \text{otherwise} \end{cases} \quad j = 1, \ldots, J; k = 1, \ldots, K_j; c = 1, \ldots, C. \tag{4.17}$$

In other words, γ_{jkc} is set to 1 if operation *k* of product *j* is allocated to cell *c*. Let also

$$N_{jc} = \begin{cases} 1, & \sum_{k=1}^{K_j} \gamma_{jkc} > 0 \\ 0, & \text{otherwise} \end{cases} \quad j = 1, \ldots, J; c = 1, \ldots, C \tag{4.18}$$

$$N_c = \sum_{j=1}^{J} N_{jc} \qquad c = 1, \ldots, C \tag{4.19}$$

Note that N_c denotes the total number of parts that goes in or out of cell *c*. Then r_c, or the intracellular transfer ratio of cell *c*, can be calculated as

$$r_c = \sum_{j=1}^{J} \left[\sum_{k=1}^{K_j} \gamma_{jkc} \gamma_{j(k+1)c} / (|K_j| - 1) \right] \big/ N_c \qquad c = 1, \ldots, C. \tag{4.20}$$

We consider the minimum of all r_c, which is associated to the CM-cell in which the assigned parts have to travel the most in and out to be completed. Then, the second objective is to maximize the minimum r_c, in other words

$$\max f_2 = \min_{c=1,\ldots,C} \{r_c\}. \tag{4.21}$$

Since the appropriateness of a cell configuration is measured with the minimum r_c, solutions with different number of cells can be compared. This is different from the objectives that deal with the minimization of intercellular move, since such objectives favor designs with fewer numbers of cells implicitly. Our last objective is to minimize the total machine investment and operating costs related to the design, denoted f_3 here. If c_i^{FIX} is the fixed purchasing cost of a machine of type i, then f_3 can be formulated as follows:

$$\min f_3 = \sum_{i=1}^{I} \sum_{c=0}^{C} c_i^{FIX} x_{ic}. \qquad (4.22)$$

5. SOLUTION METHOD

5.1. MULTIOBJECTIVE OPTIMIZATION

In general, the definition of an optimization problem consists of minimizing an objective function considering all the parameters of the problem. A lot of "classical" optimization methods exist to solve such problems. These methods can be used providing that certain mathematical conditions are satisfied: thus, linear programming efficiently solves problems where the objective function and constraints are linear with respect to the decision variables. Unfortunately, the situations encountered in practice often include one or more difficulties which make these methods inapplicable: for example, the problem may have contradictory objective functions. Two objectives are contradictory when a decrease in one objective leads to an increase in the other objective. To clarify this situation; multiobjective optimization is proposed.

The main difficulty in mono objective optimization comes from the fact that modeling a problem with just one equation can be a very difficult task. The goal of modeling the problem using just one equation can introduce a bias during the modeling phase.

Multiobjective optimization allows a degree of freedom which is lacking in mono-objective optimization. This flexibility is not without consequences for the method used to find an optimum for the problem when it is finally modeled. The search gives not a unique solution but a set of solutions. These solutions are called Pareto solutions, and the set of solutions that are found at the end of the search is called the tradeoff surface.

After having found some solutions of the multiobjective optimization problem, some difficulties occur: selection of a solution from the set of solutions. The solution selected by the user reflects tradeoffs performed by the user with respect to the various objective functions. The decision maker is "human"; he makes choices, and one of the goals of multiobjective optimization is to model the choices of the decision maker, or rather, his preferences.

5.1.1. Domination

When the multiobjective optimization problem is solved, a multitude of solutions is found. Only a small of these solutions are of interest. For a solution to be interesting; there must be a domination relation between the considered solution and the other solutions, in the following sense:

Definition 5.1: A vector x_1 dominates a vector x_2 if:

x_1 is at least good as x_2 for all the objectives, and

x_1 is strictly better than x_2 for at least one objective.

Solutions which dominate the others but do not dominate themselves are called optimal solutions in the Pareto sense. Local and global optimality in the Pareto sense can be defined as follows:

Definition 5.2: A vector $x \in \mathbb{R}^n$ is *locally optimal in the Pareto sense* if there exists a real $\delta > 0$ such that there is no vector $\vec{x}'$ which dominates the vector x with $x' \in \mathbb{R}^n \cap (x, \delta)$, where (x, δ) represents a bowl of center x and of radius δ.

A vector x is locally optimal in the Pareto sense if it is optimal in the Pareto sense with a restriction on the set $\mathbb{R}^n$. This definition is illustrated in Figure 5.1

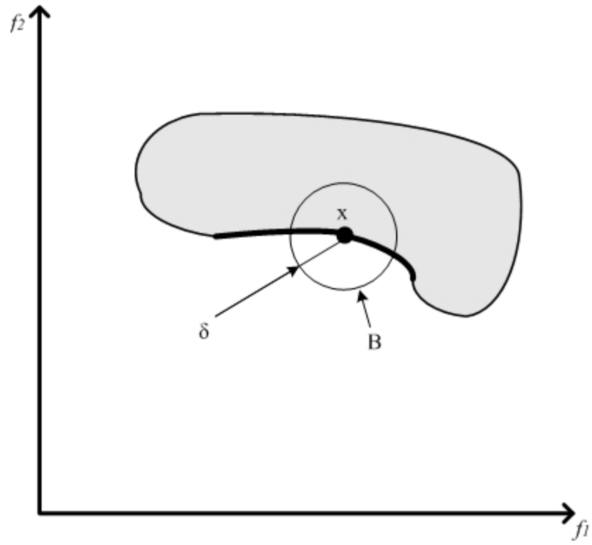

Figure 5.1: Local Optimality in Pareto Sense

Definition 5.3: A vector *x* is *globally optimal in the Pareto sense* (or optimal in the Pareto sense) if there does not exist any vector *x*' such that *x*' dominates the vector *x*.

The main difference between the definition of the global optimality and the definition of the local optimality lies in the fact that there is not any restriction on the set $\mathbb{R}^n$ anymore. A "graphical" version of the preceding definition uses the contact theorem. The way to use the contact theorem is illustrated in Figure 5.2

Definition 5.4: A negative cone is defined in $\mathbb{R}^k$ in the following way:

$$C^- = \left\{ x \mid f(x) \in \mathbb{R}^k \text{ and } f(x) \leq 0 \right\} \tag{5.1}$$

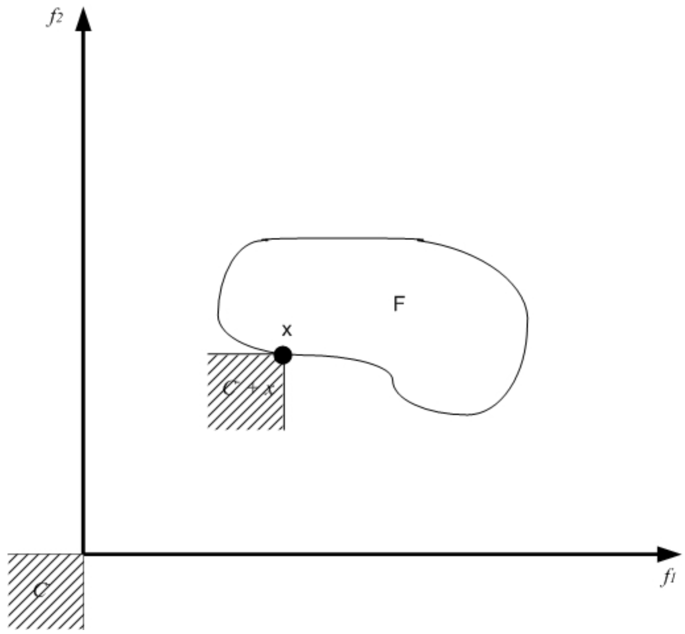

Figure 5.2: Contact Theorem

Definition 5.5: A vector *x* is optimal in the Pareto sense for a multiobjective optimization problem if

$$(C^{-} + x) \cap F = \{x\} \tag{5.2}$$

where *F* corresponds to the feasible subspace.

When the definition of domination is applied, four areas can be defined. A preference level can also be associated with each area. These areas are represented in Figure 5.3. This figure uses a splitting defined by use of the negative cone introduced above and spreads it out over the whole space.

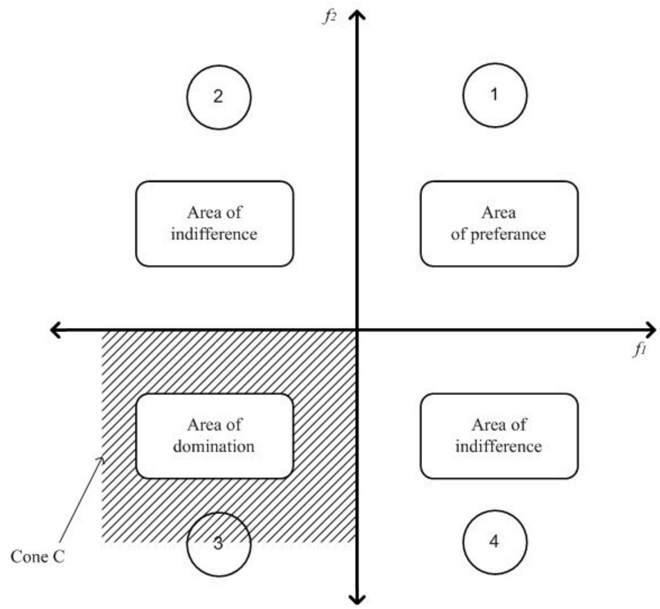

Figure 5.3: Preference Level and Domination Relation

For example, if this figure is centered on solution A and this solution is compared with solution B, there are three possibilities:

If solution B belongs to area 1, then solution A is preferred to solution B;

If solution B belongs to area 3, then solution A is dominated by solution B;

If solution B belongs to area 2 or 4, then we can not say solution A or B is preferable to each other.

5.1.2. Relation Derived From Domination

The domination relation does not offer too many degrees of freedom in its definition. For example, it is not possible to include in the definition of the domination relation a preference for one objective function against another one. To overcome this lack of flexibility, relations derived from the domination relation have been developed. Any solutions we can find with these relations are always optimal in the Pareto sense. The major change that we find with these relations is that the set of solutions that we obtain with derived relations is a subset of the set of solutions obtained using the domination relation.

Let S^k represents the set of feasible solutions of an optimization problem with k objective functions.

Lexicographic Optimality: This definition of optimality allows one to include preferences with respect to objective functions [62].

Definition 5.6: A solution $x^* \in S^k$ is *optimal in the lexicographic sense* if

$$x^* \leq_{lex} x, \ \forall x \in S^k - \{x^*\}. \tag{5.3}$$

If $x, y \in S^k$, we say that $x \leq_{lex} y$ if there exists an index value q^* such that $x_q = y_q$ for $q = 1, ..., (q^* - 1)$ and $x_{q^*} < y_{q^*}$. The relations between x_q and y_q for $q \geq q^*$ are not taken into account because of the stopping at index q^* (the first index for which $x_q < y_q$).

The definition involves the requirement that the decision maker has sorted the objective functions by increasing preference. The comparison between two solutions is done using this sorting of objective functions.

This relation can be illustrated by an example. Take two points *A* and *B*:

$$A = (1, 2, 3, 4, 5, 6), \quad B = (1, 2, 3, 9, 4, 9)$$

For these two points, we have $A \leq_{lex} B$ because, until the third position, we have $A_i = B_i, \ i = 1, 2, 3$ and, for the fourth position, we have $4 < 9$. We can conclude that solution *A* dominates solution *B* lexicographically.

Extremal Optimality: As with the lexicographic optimality relation, this relation allows one to add preferences between objective functions. This preference is modeled using weights. The more important an objective function, the heavier the weight [62].

Definition 5.7: A solution $x^* \in S^k$ is *extreme-optimal* if, given a weight vector $\lambda \in \mathbb{R}^k$ such that $\sum_{i=1}^{k} \lambda_i = 1$, x^* is an optimal solution of the mono criterion minimization problem with the following objective function:

$$\sum_{i=1}^{k} \lambda_i x_i \tag{5.4}$$

so,

$$\sum_{i=1}^{k} \lambda_i x_i^* \leq \sum_{i=1}^{k} \lambda_i x_i, \; \forall x \in S^k - \{x^*\} \tag{5.5}$$

For illustrating the notion of extremal optimality, preceding example is considered. Here, objective 6 is considered as a reference objective. It is also assumed that objectives 1, 3 and 5 are 20% more important than the reference objective and that objectives 2 and 4 are equivalent to the reference objective. The weight of each objective can be computed:

$$\begin{aligned} &w_1 = w_3 = w_5 = 1.2w_6 \\ &w_2 = w_4 = w_6 \\ &\sum_{i=1}^{6} w_i = 1 \end{aligned}$$

by solving these equations

$$\begin{aligned} w_1 = w_3 = w_5 = \frac{0.2}{1.1} = 0.18 \\ w_2 = w_4 = w_6 = \frac{1}{6.6} = 0.15 \end{aligned}$$

$\sum_{i=1}^{6} w_i A_i = 3.45$ and $\sum_{i=1}^{6} w_i B_i = 5.39$ are handled. So, *A* extreme-dominates point *B*.

Maximal Optimality: This relation, unlike the preceding relations, does not allow one to add preferences between objective functions [62].

<u>Definition 5.8:</u> A solution $x^* \in S^k$ is *max-optimal* if the value of the worst objective is as small as possible:

$$\max_{q \in \{1,\ldots,k\}} x_q^* \leq \max_{\substack{q \in \{1,\ldots,k\} \\ \vec{x} \in S^k - \{\vec{x}^*\}}} x_q \tag{5.6}$$

For illustrating the relation, preceding example is considered. Here, solution *A* max-dominates solution *B* because $\max A = 6 < \max B = 9$

5.1.3. Tradeoff Surface

The small number of solutions that we have selected using the sorting rule based on the definition of domination produces what we call the tradeoff surface (or Pareto front). A problem with two objective functions can be imagined as an example (minimize f_1 and minimize f_2 under the constraints $g(x) \leq 0$ and $h(x) = 0$). The set of values of the pair $(f_1(x),\ f_2(x))$ is denoted by *S* when *x* respects the constraints $g(x)$ and $h(x)$. The tradeoff surface is denoted by *P*.

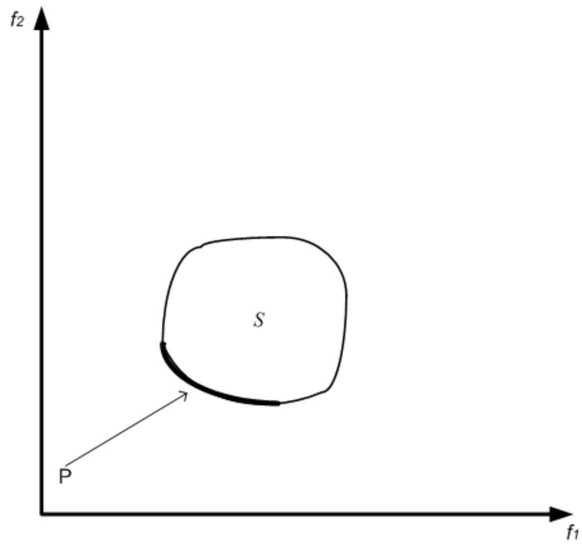

Figure 5.4: Representation of Trade-off Surface

S and *P* are represented in Figure 5.4. A noticeable property is that certain shapes for the tradeoff surface can be obtained, depending on the type of problem. These common shapes of tradeoff surface are shown in Figure 5.5. These shapes of tradeoff surfaces are typical of a multiobjective problem with a convex solution set we often face.

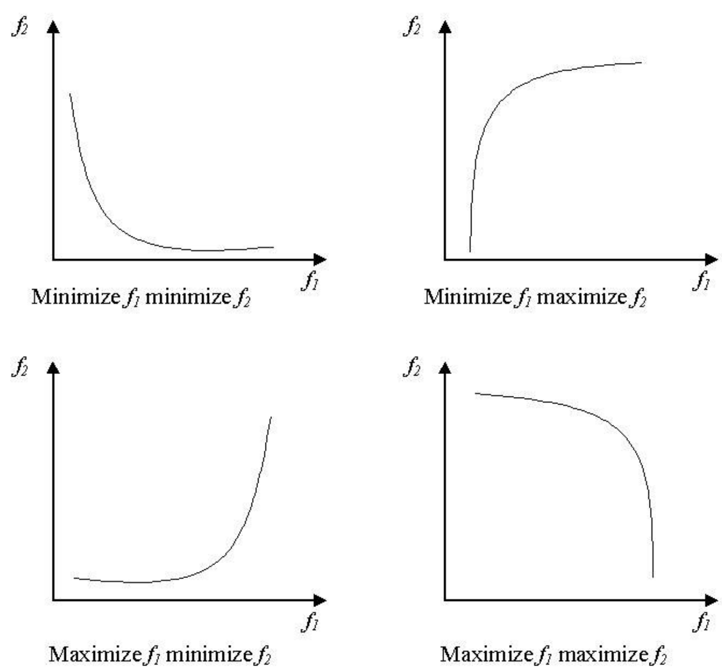

Figure 5.5: Common shapes of tradeoff surface considering two objective functions

Definition 5.9: Two characteristic points associated with a tradeoff surface are ideal point and nadir point. The coordinates of the *ideal point* are obtained by minimizing each objective function separately. The coordinates of *Nadir point* correspond to the worst values obtained for each objective function when the solution set is restricted to the tradeoff surface.

The ideal point is used in a lot of optimization methods as a reference point. The nadir point is used to restrict the search space; it is often used in interactive optimization methods. These two definitions are illustrated in Figure 5.6

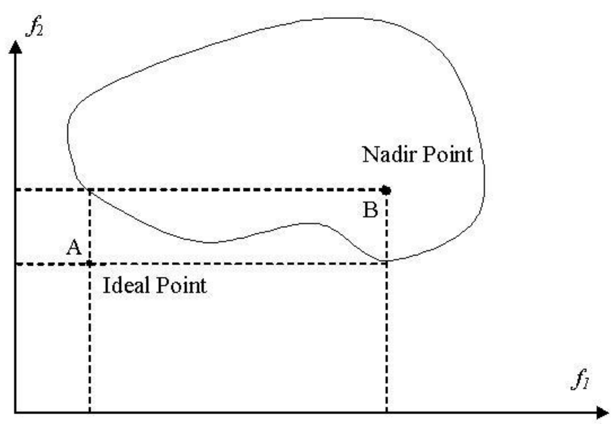

Figure 5.6: Representation of the ideal point and the nadir point

5.2. EVOLUTIONARY PROGRAMMING

The formulation proposed in chapter 3 yields a multiobjective combinatorial problem. Evolutionary algorithms (EAs) present interesting features to solve multiobjective optimization problems, in particular because they deal simultaneously with a set of possible solutions [63]. Additionally, EAs are less susceptible to the shape or continuity of the Pareto front [64] and have been successfully used in the design and organization of manufacturing systems [63].

5.2.1. Definitions

Genetic algorithms are inspired by classical genetics [65]. A "population" of points spread through the search space is considered. Before explaining the behavior of a genetic algorithm in detail, some words of the genetic algorithms shall be presented. These words are often used to describe a genetic algorithm.

• Genotype or chromosome: This is another way to say "individual".

• Individual: Corresponds to the coding, as a gene, of a potential solution of an optimization problem.

• Gene: A chromosome is composed of genes. With a binary coding, a gene has a value of either 0 or 1.

• Phenotype: Each genotype represents a potential solution of an optimization problem. The value of this potential solution is called a phenotype.

Each individual is "coded" as a gene. For example, most often, a correspondence between a binary chain and an individual is used. To each individual, an efficiency is assigned (this value is also known as the "adaptation"). This efficiency corresponds to the performance of an individual in solving a given problem. For example, if the problem of the maximization of a function is considered, the efficiency of the individual will increase with its ability to maximize this function.

After having determined the efficiency of the individuals, a reproduction is performed. The individuals are copied in proportion to their efficiency. For example, if individual 1 is reproduced two times, individual 2 will be reproduced four times and individual 3 will be reproduced one time. This behavior is similar to that in the real life. The better an individual is adapted to its environment (it protects itself better against predators, and it eats better), the more is its reproductive capacity.

After having determined the efficiency function and performing the breeding step, “crossovers” between individuals are performed. The more efficient an individual is in solving a problem, the more it shares parts of its chromosome with a large number of other individuals. To select an individual for this operation, a wheel, on which each individual fills an area proportional to its efficiency, is used (this process is called roulette wheel selection).

Merging the starting individual with the individual, which it is associated with, is performed. This operation is called a crossover. To do so, a position in the binary code of the individuals is selected randomly. At this position, the code is broken and the right hand side parts between the two individuals are swapped. After repeating this operation for all the individuals, two new elements in the results for each pair of individuals are obtained. So, the population increases its size (it doubles its size). At this point, to keep a constant number of individuals in the population; either the whole population is kept or less efficient individuals are removed.

Lastly, a mutation operation in a gene of an individual is applied. To do so, some individuals are selected randomly (in general, the probability of selection for a mutation is small). Lastly, at this position, a 0 is changed into 1 or vice versa.

The process is started and repeated until a stopping criterion is reached (for example, some maximum number of iterations).

Algorithm 5.1: A Genetic Algorithm

Population initialization

Objective function computation

Computation of the efficiency

For $i = 1$ to MaxIter

  Random selection,

  Selection proportional to the efficiency, Crossover,

Mutation,

Objective function computation,

Computation of the efficiency,

End For

An evolutionary algorithm is similar to a genetic algorithm with the exception that it does not contain crossover operator. Genetic algorithms are well adapted to dealing with multiobjective optimization problems. The huge number of papers which have been published about this subject testify to this. Moreover, this domain is very dynamic and has not stopped growing. The steps of a genetic algorithm, when it is used to solve a multiobjective optimization problem, are as follows:

- Population initialization
- Computation of the efficiency of the individuals in the population
- Vector/efficiency transformation
- Crossover
- Mutation
- Selection

In vector/efficiency transformation step a vector (which contains the objective function values of each individual) is transformed into efficiency. Various strategies, which will be described in the following subsections, can be adopted. All of the genetic algorithms which deal with multiobjective optimization problems follow this sequence.

5.2.2. The Vector Evaluated Genetic Algorithm (VEGA) Method

Using this method; a multiobjective optimization problem can be solved without having to merge all the objective functions into one [66]. The VEGA algorithm uses a population of N individuals. These N individuals are divided into k groups (k corresponds to the number of objective functions of the problem) of N/k individuals (where N is a multiple of k). At each group, an objective function is computed. By this objective function, the efficiency of an individual in the group is determined. Then, individuals are mixed and a crossover step is performed with respect to the efficiency of each individual. The method is composed of 5 steps. These steps are:

Iteration *i*. Initialize a population of size *N*.
Create *k* groups (subpopulations).
Compute the efficiencies. Mix the individuals.
Apply the classical genetic algorithm (crossover-mutation-selection).
Go on to the next iteration (*i+1*).

The danger with this method, which can be obtained, is that, at the end of the optimization, a population composed of mean individuals with respect to all the objective functions. Such a population does not allow one to obtain a good approximation of the tradeoff surface. Instead, the population will be concentrated around a mean "point". Tricks have been found to overcome this effect (use of species inside a group, etc.).

Moreover, it has been shown that this method is equivalent to the weighted sum of objective functions method. Therefore, it does not allow us to find all of the solutions hidden in concavities.

5.2.3. The Multiple Objective Genetic Algorithm (MOGA) Method

This method is presented in [67]. It uses a domination relation to compute the efficiency of an individual. This method is based on the relation of domination in the Pareto sense. Here, the "rank" of an individual (an order number which allows one to rank an individual with respect to the others) is given by the number of individuals which dominate the considered individual. For example, consider an individual at generation which is dominated by $p_i^{(t)}$ individuals, the rank of this considered individual is given by

$$rank(x_i, t) = 1 + p_i^{(t)}. \tag{5.7}$$

Algorithm 5.2: The MOGA Algorithm

Population initialization
Computation of the values of the objective functions
Assignment of a rank by use of domination
Assignment of efficiency by use of the rank
For i = 1 to G (G represents number of individuals)
    Random selection in proportion to the efficiency
    Crossover
    Mutation
    Computation of the objective functions
    Assignment of a rank by use of domination
    Assignment of efficiency by use of the rank
End for

In some cases the MOGA method does not provide a good diversity of solutions for the approximation of the tradeoff surface.

5.2.4. The Nondominated Sorting Genetic Algorithm (NSGA)

This method is based upon MOGA method described in the preceding section. The main difference occurs during the computation of the efficiency of an individual [68]. It is based on a classification using many levels of individuals. In a first step, before proceeding to the selection, a rank is assigned to each individual of the population. All the nondominated individuals with the same rank are ranked into this category. A dummy efficiency is assigned to this category. This efficiency is inversely proportional to the Pareto rank of the considered category.

The individuals of the considered category are uniformly spread inside it. So, a good approximation of the trade-off surface or a good diversity of solutions is desired.

To maintain this diversity inside the population, a new efficiency value is assigned to these ranked individuals. To do so, following formula is used:

$$m_i = \sum_{j=1}^{K} Sh(d(i,j)) \tag{5.8}$$

$$Sh(d(i,j)) = \begin{cases} 1 - \left(\dfrac{d(i,j)}{\sigma_{share}} \right)^2 & \text{if } d(i,j) < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

Here, K corresponds to the number of individuals in the considered category, and $d(i,j)$ corresponds to a distance between individual i and individual j. σ_{share} is the influence distance. As in the above expression, all the individuals which are sufficiently close (for which the distance $d(i,j)$ is lower than σ_{share}) are taken into account in the computation of m_i. The others are ignored.

The efficiency value of individual i inside the considered category is

$$f_i = \frac{F}{m_i} \tag{5.10}$$

where F is the value of the efficiency assigned to the category to which the individual belongs.

After performing computation for individual *I*, the individuals of this group is ignored. The process continues with the remaining individuals, where a new classification, a new categorization and a new sharing operation are made. This process is repeated until all the individuals have been treated. Because the individuals which have a Pareto rank equal to 1 have the best efficiency, they are reproduced more than the others. This property allows us to obtain a quicker convergence toward the tradeoff surface.

Algorithm 5.3: The NSGA Algorithm

Population initialization

Computation of the values of the objective functions
Assignment of a rank by use of the domination for each tradeoff surface
Computation of the enumeration of the neighboring points
Assignment of a shared efficiency
For i = 1 to G
    Random selection in proportion to the efficiency
    Crossover
    Mutation
    Computation of the values of the objective functions
    Assignment of a rank by use of the domination for each tradeoff surface
    Computation of the enumeration of the neighboring points
    Assignment of the enumeration of the neighboring points
    Assignment of a shared efficiency
End For

The efficiency of this method lies in the fact that all objective functions are reduced to a dummy efficiency value obtained using the ranking with respect to the Pareto rank. This method has the drawback that it is sensitive to the value of σ_{share}.

For this method, the number of comparisons performed on a population for one generation is the same as with the MOGA method. Nevertheless, an excess cost due to the sharing appears. This excess cost is proportional to $N \times (N-1)$.

5.2.5. The Niched Pareto Genetic Algorithm (NPGA) Method

NPGA is based upon the NSGA method described in the preceding section. The main difference occurs during the selection process [69].

In a classical genetic algorithm, the method of selection between two individuals uses a selection wheel. In the present method, individuals change in the selected way. Instead of comparing two individuals, a group of I individuals are used. If the individuals are either dominated or nondominated, a sharing of the efficiency value is used. The selection function chooses an individual from the population *S*. The values of t_{dom} and σ_{share} must be given by the user. As this method does not use the Pareto selection on the whole population, but just on a part at each run, it is very fast. Nevertheless, it requires from the user the parameters t_{dom} and σ_{share}, on which the performance of the algorithm depends. In [70] the techniques, which "automatically" tune these parameters, can be found.

Algorithm 5.4: The NPGA Algorithm

Population initialization

Computation of the values of the objective functions

For $i = 1$ to G

    Tournament selection between two individuals (use of t_{dom})

    Only candidate 1 is dominated: Select candidate 2

    Only candidate 2 is dominated: Select candidate 1

        Perform an efficiency sharing

        Select the candidate with the smaller number of neighbors (use of σ_{share})

    Crossover

    Mutation

    Computation of the values of the objective functions

End For

5.3. STRENGTH PARETO EVOLUTIONARY ALGORITHM (SPEA2)

In this study, we use SPEA2 algorithm to solve our multiobjective optimization problem given in chapter 4. Apart from the previously presented algorithms in this section, SPEA2 uses elitism.

Definition 5.10: Let P^t, a population obtained from a given genetic algorithm after t iterations (generations), be given. Let $P^t(x)$, the probability that an individual $x \in P^t$ is selected for the crossover and/or mutation step during generation t, be given. Then the generic algorithm is said to be "elitist" if and only if, for a given preference relation $\prec$ for a given decision problem, the following condition is satisfied:

$$\forall t \in \mathbb{N},\ \exists x \in \bar{P}^t \text{ such that } P^{t+1}(x) > 0$$

with

$$\bar{P}^t = \{x \in P^t \mid \nexists x' \in P^t,\ x' \prec x\} \tag{5.11}$$

$$P^t = \cup_{r \leq t} P^r \tag{5.12}$$

The operation $\cup_{r \leq t} P^r$ corresponds to the merging of all the sets P^r with $r \leq t$. So $\bar{P}^t$ corresponds to the non-dominated individuals of P^t.

The main idea of elitism is not to forget to include the best solution obtained, one generation after another. In mono-objective optimization, it is common to save the best solution obtained during the optimization process. In multiobjective optimization, a similar process consists in putting into an archive all the non-dominated individuals obtained during the optimization. This archive represents one of the best approximations of the trade-off surface. Moreover, this archive is used just to save non-dominated individuals; it does not change the behavior of the genetic algorithm.

Similar to most of the multiobjective evolutionary methods mentioned, SPEA2 uses a Pareto approach, which is based on non-dominated solutions. In SPEA2, the Pareto set is externally stored and updated after each generation, so that it contains all the non-dominated solutions detected from the beginning. The external set of a given iteration is obtained by selecting the non-dominated solutions among set formed by the current population and the external set of the previous iteration. If the size of the new archive is less than a predefined size, it is then filled with the dominated solutions having best fitness values. If this number is high than allowed, an archive truncation procedure which does not loose boundary points is invoked [71].

The fitness of solutions (external + current population) is obtained by first calculating the strength of each individual, i.e. the number of solutions among the external and current population it dominates. Next, an individual's raw fitness is set to the sum of the strengths of the solutions that dominate it. Note that zero raw fitness value for a solution implies that it is non-dominated. Although this raw fitness assignment provides a sort of niching mechanism based on the concept of Pareto dominance, it may fail when most individuals do not dominate each other. Therefore, additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique used in SPEA2 is an adaptation of the k^{th} nearest neighbor method, where the density at any point is a decreasing function of the distance to the k^{th} nearest data point. Then, the density estimate is the inverse of the distance to the k^{th} nearest neighbor [71]. This value is assured to be less than one. Finally, the fitness of a solution is set to the sum of its raw fitness and density estimate. This further implies that solutions having fitness close to zero are more preferable.

The population of the next generation is obtained by a selection based on binary tournaments with replacement; two solutions from the external set are randomly selected, the best one is copied, original solutions are returned to the external set and the copy is added to the mating pool. This process is repeated until new population size reaches its limit. At the end of the algorithm, we only keep from the last population the individuals that represent a feasible solution. That means that for all the survivors the value of the objective given in equation is strictly positive.

Algorithm 5.5: The SPEA II Algorithm

Input: N (population size), $\bar{N}$ (archive size),

T (maximum number of generations)

Output: A (non-dominated solutions set)

Begin

  $t = 0$;

  Generate initial population P^t;

  Create the empty archive $\bar{P}^t$;

  While ($t \leq T$) do

    Calculate fitness of individuals in P^t and $\bar{P}^t$;

    Copy all non-dominated individuals in P^t and $\bar{P}^t$ to $\bar{P}^{t+1}$;

    If size $\bar{P}^{t+1}$ is greater than $\bar{N}$ then

     reduce $\bar{P}^{t+1}$ by means of truncation operator;

    If size $\bar{P}^{t+1}$ is less than $\bar{N}$ then

     fill $\bar{P}^{t+1}$ with dominated individuals in P^t and $\bar{P}^t$;

    Perform binary tournament with replacement on $\bar{P}^{t+1}$ in order

    to fill P^{t+1} up to size N;

    Apply mutation operators to P^{t+1};

    $t = t + 1$;

  End;

  Set A to the set of non-dominated solutions in $\bar{P}^{t-1}$.

End.

5.4. ADDITIONAL SETTINGS RELATED TO THE PROBLEM

To solve our problem, we use an evolutionary programming approach as defined by [72], which has already provided good results on combinatorial optimization problems such as Traveling Salesman Problem [73]. The main feature of this evolutionary approach is that there is no crossover operator, and all individuals are parents, without any previous evaluation.

5.4.1. Solution Encoding

Given that any solution having more machines than necessary compared to the LHS of the equation (4.4) can not be optimum, x_{ic} variables are directly related how y_{jkic} variables are determined. Moreover, x'_{ic} is calculated from x_{ic} by Equation (4.7) and there is no interest to open a cell while there is no operation destined to that cell ($z_c = 0$) given Equation (4.6). Therefore, it suffices to code y_{jkic} variables for a solution of our formulation. In the vector coding of the y_{jkic} variables, the vector elements are related to part $j = 1, \ldots, J$ and operation $k = 1, \ldots, K_j$, and hence there are $\sum_{j=1}^{j} \left| K_j \right|$ elements. The solution vector hold respectively the indices of the cell (c) to which the operation is assigned. Since only one type of machine is allowed to undertake the operation, it is determined easily. With this coding, we can guarantee the feasibility regarding to equations (4.4) – (4.7).

5.4.2. Mutation

From a current solution population, we systematically obtain the offspring by duplicating and mutating each individual, and selecting the solutions that will belong to the next generation. To modify the cell where a part's operation is allocated, we only consider three mutations. With probability p_1, we randomly select two operations and create a new GT-cell with these setting. With probability $(1 - p_1)$, we randomly select two operations assigned to two different cells (FL-cell or CM-cell) and with probability p_2, we allocate one of the operations to the cell of the other operation. With probability $(1 - p_2)$, we switch the cells where the operations are allocated. These mutations ensure the connectivity of our search space. Moreover, any mutation violating constraints given in (4.8) is not allowed by construction.

5.4.3. Constraints

The manner a solution is coded and the mutation operator is used, we can always ensure feasibility with respect to constraint sets (4.4) – (4.8). To take constraint sets (4.9) – (4.10) into account, several possibilities exist [74, 75, 76]. In our multi-objective approach, the aggregated satisfaction degree of these two constraint sets is considered as a supplementary objective which is maximized to attain feasible solutions [77]. We assume that the problem addressed has solutions that satisfy these constraints with a positive satisfaction degree. The aggregated satisfaction degree is obtained by means of the quantifier guided OWA operator with the quantifier "the most" [35, 78]. Then, our fourth objective is given with the equation (5.12)

$$\begin{aligned} \max f_4 &= (w_1, w_2, \ldots, w_{C+I})(\mu_{\bar{X}_1}, \ldots, \mu_{\bar{X}_C}, \mu_{\bar{\bar{X}}_1}, \ldots, \mu_{\bar{\bar{X}}_I})^T \\ &= w_1 \times \mu_{[1]} + w_2 \times \mu_{[2]} + \ldots + w_{C+I} \times \mu_{[C+I]} \end{aligned} \qquad (5.13)$$

where $\mu_{[1]}$ is the largest element among $(\mu_{\bar{X}_1}, \ldots, \mu_{\bar{X}_C}, \mu_{\bar{\bar{X}}_1}, \ldots, \mu_{\bar{\bar{X}}_I})$, and $\mu_{\bar{X}_c}$ $c = 1, \ldots, C$ and $\mu_{\bar{\bar{X}}_i}$ $i = 1, \ldots, I$ are defined as in chapter 4.

6. ILLUSTRATIVE EXAMPLE

In order to validate the proposed approach, we created an artificial example for which the best solution (best on all objectives) is known by construction, and the algorithm is expected to find this solution. We consider an ideal workshop where the machines can be grouped into totally independent CM-cells and totally independent FL-cells. The investigated manufacturing system is to be designed for 30 parts and 10 machine types. Important information related to the parts is given in Table 6.1. Part unit processing times for machine types are given in Table 6.2. The cost of machines (c_i^{FIX}) is selected as (0.35, 0.70, 0.49, 0.44, 0.41, 0.37, 0.69, 0.60, 0.61, 0.86), the capacity of each machine type is equal to 2400 time units, and the acceptable and unacceptable levels (β_i^1 and β_i^2. respectively) for the number of identical machines of each type are selected as (2-5, 3-7, 3-7, 5-9, 4-8, 3-7, 5-9, 5-9, 5-9, 5-9). Similarly, the acceptable and unacceptable cell size limits (α^1 and α^2 respectively) are selected as 15 and 20 respectively.

The EA described in chapter 5 has been implemented in Matlab environment on a PC and run with a population of 200 individuals and 20 individuals as the maximum size for the external solution set. We compute the final Pareto set after ten runs of the algorithm, where each run consists of 700 iterations. It is observed that the algorithm was able to find the single optimum solution before 500 iterations. Figure 6.1 illustrates the associated optimum solution. The objectives' values for the optimum are (23.02, 0.1424, 1.0, 1.0). The composition of the cells is given in Table 6.3 . According to this solution, parts 1, 2, 7, 8, 9, 10, 12, 16, 17, 18 and 21 reside in CM-cell 1; parts 3, 4, 11, 15, 19, 20, 22 and 24 reside in CM-cell 2; parts 5, 6, 13, 14 and 23 reside in CM-cell 3 and finally all other parts 25-30 are machined in FL-cells.

Table 6.1: Routings, Maximum Similarities, Coefficient of Variations, Demands, Costs of Machining in CM-cells, Costs of Machining in FL-cells for the illustrative example

Parts	ROUTING	MS	CV	d_j	c_j^{CMC}	c_j^{CFC}
1	6 - 5 - 8 - 7 - 9 - 4 - 10	0.9844	0.0750	390	0.0937	0.9100
2	5 - 8 - 7 - 9 - 4 - 10	0.9796	0.2354	450	0.0973	0.9060
3	2 - 9 - 6 - 7 - 1 - 3	0.9796	0.0513	315	0.0937	0.9100
4	8 - 10 - 2 - 4 - 6 - 7 - 1 - 3	0.9877	0.1643	260	0.1120	0.8890
5	4 - 10 - 9 - 5 - 3 - 4 - 9 - 7	0.9844	0.0929	430	0.0944	0.9090
6	4 - 10 - 9 - 3 - 4 - 9 - 7	0.9844	0.0837	395	0.0937	0.9100
7	6 - 8 - 5 - 7 - 9 - 4 - 10	0.9844	0.0557	290	0.0937	0.9100
8	2 - 8 - 5 - 8 - 7 - 9 - 4 - 10	1.0000	0.0915	350	0.0941	0.9090
9	2 - 4 - 5 - 8 - 7 - 9 - 10	0.9722	0.0724	385	0.0937	0.9100
10	2 - 4 - 8 - 5 - 8 - 7 - 9 - 4 - 10	0.9877	0.1678	475	0.1110	0.8900
11	8 - 2 - 9 - 4 - 6 - 7 - 1 - 3	0.9877	0.2052	430	0.1030	0.8990
12	6 - 8 - 5 - 8 - 7 - 9 - 4 - 10	0.9877	0.2065	405	0.1030	0.9000
13	10 - 9 - 3 - 4 - 9 - 7	0.9796	0.0658	100	0.0937	0.9100
14	9 - 5 - 3 - 4 - 9 - 7	0.9796	0.2034	350	0.1040	0.8990
15	7 - 8 - 2 - 9 - 4 - 7 - 1 - 3	0.8167	0.1131	480	0.1000	0.9030
16	8 - 5 - 8 - 7 - 9 - 4 - 10	0.9844	0.0573	165	0.0937	0.9100
17	2 - 8 - 5 - 8 - 7 - 9 - 4 - 10	1.0000	0.0782	115	0.0937	0.9100
18	4 - 8 - 5 - 8 - 7 - 9 - 4 - 10	0.9877	0.1390	245	0.1080	0.8940
19	8 - 10 - 2 - 9 - 4 - 6 - 7 - 1 - 3	1.0000	0.0890	220	0.0937	0.9100
20	7 - 10 - 9 - 4 - 6 - 7 - 1 - 3	0.7347	0.2351	340	0.3000	0.7000
21	6 - 4 - 8 - 5 - 8 - 7 - 9 - 4 - 10	0.9877	0.1144	325	0.1010	0.9020
22	8 - 10 - 2 - 9 - 4 - 6 - 7 - 1 - 3	1.0000	0.0957	430	0.0952	0.9080
23	10 - 9 - 5 - 3 - 4 - 9 - 7	0.9844	0.2294	450	0.0984	0.9050
24	2 - 9 - 4 - 6 - 7 - 1 - 3	0.9844	0.1824	300	0.1080	0.8940
25	8 - 3 - 10 - 8 - 6 - 1 - 8	0.2571	0.6296	405	0.7000	0.3000
26	7 - 8 - 1 - 8 - 6 - 5 - 6	0.4000	0.7180	125	0.6000	0.4000
27	3 - 7 - 3 - 7 - 8 - 6 - 9	0.2571	0.6306	480	0.7000	0.3000
28	3 - 8 - 2 - 1 - 7 - 5 - 2	0.3117	0.8742	120	0.9050	0.0947
29	1 - 7 - 4 - 1 - 4 - 8 - 6	0.4000	0.8706	360	0.7900	0.2100
30	3 - 5 - 9 - 3 - 1 - 2 - 5	0.3117	0.6456	420	0.7000	0.3000

Table 6.2: Unit processing times on machine types

Parts	Processing times (mn)							
1	2.00	0.52	1.67	1.68	1.52	1.07	1.30		
2	1.83	1.69	0.46	1.17	1.09	1.31			
3	1.23	1.55	0.70	0.64	0.79	0.53			
4	1.70	1.89	0.32	1.09	1.48	0.73	0.78	0.91	
5	1.40	1.98	0.71	1.78	1.10	1.54	1.12	0.74	
6	0.56	1.51	0.67	1.43	0.51	0.41	1.62		
7	1.85	0.73	1.97	1.35	1.45	0.83	1.95		
8	1.65	1.39	1.38	1.60	1.20	0.59	1.49	1.85	
9	1.88	0.63	0.62	1.51	1.35	1.40	1.21		
10	2.00	1.25	1.20	1.77	0.57	1.83	1.32	1.34	1.32
11	1.14	1.09	1.04	1.28	1.30	0.87	0.92	0.48	
12	1.15	1.89	0.78	0.78	1.46	1.37	0.65	0.31	
13	1.33	0.72	0.95	1.03	0.57	1.36			
14	0.44	1.71	1.58	0.60	1.04	1.65			
15	0.27	1.36	1.23	1.27	0.81	0.59	0.88	1.04	
16	0.49	0.68	1.13	1.40	1.52	0.33	0.88		
17	1.03	1.02	1.72	1.86	1.73	1.46	1.20	0.78	
18	1.01	1.33	1.59	0.64	1.02	1.81	0.97	1.57	
19	0.75	1.67	1.00	0.40	1.60	1.23	0.68	0.75	0.95
20	1.07	1.78	0.70	1.26	1.12	0.83	0.94	0.98	
21	1.77	0.68	1.59	1.71	1.11	1.34	1.33	1.32	1.75
22	1.43	1.93	0.93	0.61	0.45	0.72	0.42	0.87	1.03
23	1.47	0.40	1.93	1.39	0.54	0.71	1.50		
24	0.45	0.61	0.45	0.76	0.73	0.53	0.92		
25	1.12	1.10	1.87	1.04	2.00	1.62	0.94		
26	0.95	1.18	1.41	0.66	0.97	0.70	1.38		
27	1.04	1.26	0.99	1.98	0.92	1.71	1.43		
28	1.48	1.50	1.68	0.90	1.39	0.58	1.44		
29	1.29	0.96	1.78	0.45	1.90	0.75	1.30		
30	1.25	0.94	1.87	0.49	0.82	1.86	1.51		

Table 6.3: Number of machines of each type for the single optimum

	Machine Types									
	1	2	3	4	5	6	7	8	9	10
CM-Cell 1	0	1	0	2	2	1	2	3	2	2
CM-Cell 2	1	1	1	1	0	1	1	1	1	1
CM-Cell 3	0	0	1	1	1	0	1	0	1	1
FL-Cells	1	1	1	1	1	1	1	1	1	1
Total	2	3	3	5	4	3	5	5	5	5

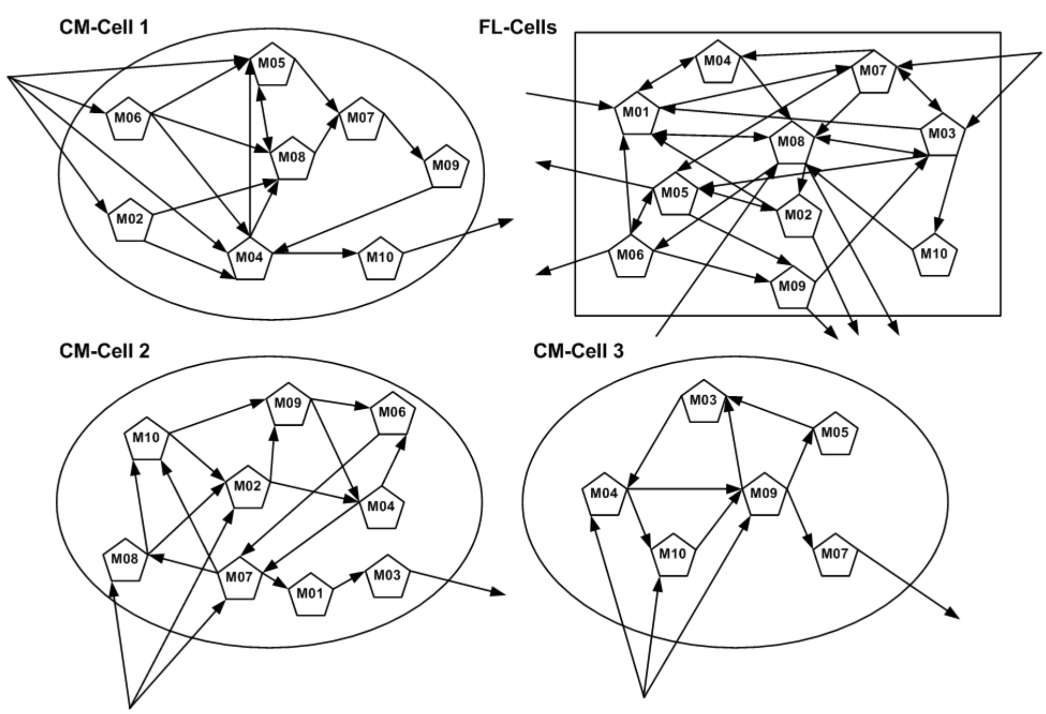

Figure 6.1: Representation of the best solution

7. CONCLUSION

This dissertation has presented a distinct approach to form hybrid manufacturing systems. The notion of part stability is introduced and it is associated with the part similarity and the coefficient of variation of its demand. The approach is open to incorporate other criteria as well to quantify the stability. Then, it is aimed to favor cellular organization of machines that process stable parts, and to benefit from the flexibility of the functional layout that enables to process less stable parts. We can underline that if all the parts are stable enough, our formulation will represent the classical cell formation problem, which can be again solved by our approach.

The use of the fuzzy set theory has allowed the imprecision of the data available in the industry to be taken into account in a more satisfactory way, in particular when one has to evaluate the stability of products.

In this study, three objectives are investigated. One objective aims to separate parts into two groups as stables and non-stables by a stability and instability cost of part's operations. The second objective is proposed to keep within manufacturing cell flow appropriate. The third and final objective is related with the total machine investment and operating costs related to the design.

The hybrid manufacturing design problem is formulated as a constrained multiobjective integer programming problem and an evolutionary algorithm is used to solve it. An interesting aspect of using an evolutionary approach is that it uses a population of non-dominated solutions, which offers interesting possibilities to handle several criteria in the optimization search. Among various evolutionary strategies, strength pareto evolutionary algorithm (SPEA2) is selected due to its efficiency to provide quality solutions. This efficiency is related with elitism mechanism, in other words the ability of SPEA2 to keep best non-dominated solutions from iterations to iterations and the generation of the new populations based on these elites.

We are now interested in improving the performance of the optimization search in terms of the convergence speed, and trying different ways of handling criteria used. Running more experiments on other data sets (including concrete industrial cases) is also planned. Another perspective is to work on a user friendly graphical interface that can collect data easily and present solutions to the designer who can have minor knowledge on the details of the evolutionary programming. This will further increase the adoption of the theoretical findings.

REFERENCES

[1] Chu, C.H., "Cluster Analysis in Manufacturing Cellular Formation", *OMEGA*: *The International Journal of Management Science*, 17, 289 – 295, (1989).

[2] Heragu, S.S., "Group Technology and Cellular Manufacturing", *IEEE Transactions on Systems, Man, and Cybernetics*, 24, 203 – 215, (1994).

[3] Offodile, O.F., Mehrez, A., Grznar, J., "Cellular Manufacturing: A Taxonomic Review Framework", *Journal of Manufacturing Systems*, 13, 196 - 220, (1994).

[4] Selim, H.M., Askin, R.G., Vakharia, A.J., "Cell Formation in Group Technology: Review, Evaluation and Directions for Future Research", *Computers & Industrial Engineering*, 34, 3 – 20, (1998).

[5] Sing, N., "Design of Cellular Manufacturing Systems: An Invited Review", *European Journal of Operational Research*, 69, 284 - 291, (1993).

[6] Harhalakis, G., Lu, T., Minis, I., Nagi, R., "A Practical Method for Design of Hybrid-Type Production Facilities", *International Journal of Production Research*, 34, 897 – 918, (1996).

[7] Marsh, R.F., Shafer, S.M., Meredith, J.R., "A Comparison of Cellular Manufacturing Research Presumptions with Practice", *International Journal of Production Research*, 37, 3119 – 3138, (1999).

[8] Olorunniwo, F.O., "A Framework for Measuring Success of Cellular Manufacturing Implementation", *International Journal of Production Research*, 35, 3043 – 3061, (1997).

[9] Viguier, F., Pierreval, H., "An Approach to the Design of a Hybrid Organization of Workshops into Functional Layout and Group Technology Cells", *International Journal of Computer Integrated Manufacturing*, 17, 108 – 116, (2004).

[10] Ang, C.L., Willey, P.C.T., "A Comparative Study of the Performance of Pure and Hybrid Group Technology Manufacturing Systems Using Computer Simulation Techniques", *International Journal of Production Research*, 22, 193 - 233, (1984).

[11] Burgess, A.G., Morgan, I., Vollmann, T.E., "Cellular Manufacturing: Its Impact on the Total Factory", *International Journal of Production Research*, 31, 2059 - 2077, (1993).

[12] Christy D.P., Nandkeolyar, U., "A Simulation Investigation of the Design of Group Technology Cells", *Proceedings of the 1986 Annual Meeting of the Decision Sciences Institute*, 1201-1203, (1986).

[13] Shambu, G., Suresh, N.C., "Performance of Hybrid Cellular Manufacturing Systems: A Computer Simulation Investigation", *European Journal of Operational Research*, 120, 436 – 458, (2000).

[14] DeLaney, D., Herrmann, J.W., Ioannou, G., Lu, T., Minis, I., Mirza, A.H., Palmer, R., "Design and Implementation of a Hybrid Manufacturing Facility", *Computers and Industrial Engineering*, 29, 315 – 319, (1995).

[15] Gravel, M., Houde, S., Price, W., "Un Cas de Conception d'un Atelier Cellulaire Hybride Géré en Kanban", *Journal Européen des Systemes Automatisés*, 30, 801 – 822, (1996).

[16] Gravel, M., Price, W., Gagne, C., "An Interactive Tool for Designing Manufacturing Cells for an Assembly Job-Shop", *International Journal of Production Research*, 38, 309-322, (2000).

[17] Baker, R.P., Maropoulos, P.G., "Cell Design and Continuous Improvement", *International Journal of Computer Integrated Manufacturing*, 13, 522 – 532, (2000).

[18] Choobineh, F., Nare, A., "The Impact of Ignored Attributes on a CMS Design", *International Journal of Production Research*, 37, 3231 - 3245, (1999).

[19] Askin, R.G., Standridge, C.R., *Modeling and Analysis of Manufacturing Systems*, John Wiley & Sons, New York, (1993).

[20] Francis, R.L., McGinnis Jr., L.R., White, J.A., *Facility Layout and Location: An Analytical Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, (1992).

[21] Ham, I., Hitoni, K., Yoshida, T., *Group Technology: Applications to Production Management*, Kluwer-Nijhoff Publishing, Boston, (1985).

[22] Irani, S.A., Cavalier, T.M., Cohen, P.H., "Virtual Manufacturing Cells: Exploiting Layout Design and Intercell Flows for the Machine Sharing Problem", *International Journal of Production Research*, 31 (4), 791 – 811, (1993).

[23] Wemmerlöv, U., Hyer, N.L., "Research Issues in Cellular Manufacturing", *International Journal of Production Research*, 25, 413 – 431, (1987).

[24] Molleman E., Slomp, J., Rolefes, S., "The Evolution of a Cellular Manufacturing System - A Longitudinal Case Study", *International Journal of Production Economics*, 75 (3), 305 - 322, (2002).

[25] Assad, A.A., Kramer, S.B., Kaku, B.K., "Comparing Functional and Cellular Layouts: A Simulation Study based on Standardization", *International Journal of Production Research*, 41 (8), 1639 – 1663, (2003).

[26] Johnson, D.J., Wemmerlöv, U., "Why Does Cell Implementation Stop? Factors Influencing Cell Penetration in Manufacturing Plants", *Production & Operations Management*, 13 (3), 272 - 289, (2004).

[27] Balasubramanian, K.N., Panneerselvam, R., "Covering Technique-based Algorithm for Machine Grouping to Form Manufacturing Cells", *International Journal of Production Research*, 31 (6), 1479 - 1504, (1993).

[28] Needy, K.L., Billo, R.E., Colosimo Warner, R., "A Cost Model for the Evaluation of Alternative Cellular Manufacturing Configurations", *Computers & Industrial Engineering*, 34 (1), 119 - 135, (1998).

[29] Irani, S.A., Huang, H., "Custom Design of Facility Layouts for Multiproduct Facilities using Layout Modules", *IEEE Transactions On Robotics And Automation*, 16 (3), 259 – 267, (2000).

[30] Irani, S.A., Huang, H., "Cascading Flowlines and Layout Modules: Practical Strategies for Machine Duplication in Facility Layouts", *International Journal of Flexible Manufacturing Systems*, 17 (2), 119-149, (2005).

[31] Zadeh, L.A., "Fuzzy Sets", *Information and Control*, 8, 338 – 353, (1965).

[32] Yager, R.R., "Fuzzy Decision Making Including Unequal Objectives", *Fuzzy Sets and Systems*, 1, 87 – 95, (1978).

[33] Yager, R.R., "General Multiple Objective Decision Functions and Linguistically Quantified Statements", *International Journal of Man-Machine Studies*, 21, 389 – 400, (1984).

[34] Dubois, D., Fargier, H., Prade, H., "Propagation and Satisfaction of Flexible Constraints", *Journal of Intelligent Manufacturing*, 6, 215 – 234, (1994).

[35] Yager, R.R., "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making", *IEEE Transactions on Systems, Man and Cybernetics*, 18 (1), 183 – 190, (1988).

[36] Zadeh, L.A., "The Concept of a Linguistic Variable and its Application to Approximate Reasoning - I", *Information Sciences*, 8 (3), 199 - 249, (1975).

[37] Jamshidi, M., Vadie, N., Ross, T.J., *Fuzzy Logic and Control: Software and Hardware Applications*, Prentice Hall, Englewood Cliffs, New Jersey, (1993).

[38] Mamdani, E.H., "Applications of Fuzzy Algorithms for Control of a Simple Dynamic Plant", *Proc. IEEE*, 121, 1585 - 1588, (1974).

[39] Vakharia, A.J., Wemmerlöv, U., "Designing a Cellular Manufacturing System: A Materials Flow Approach based on Operation Sequences, *IIE Transactions*, 22, 84 – 97, (1990).

[40] Seifoddini, H., "Single Linkage versus Average Linkage Clustering in Machine Cells Formation Applications", *Computers & Industrial Engineering*, 16, 419 – 426, (1989).

[41] Choobineh, F., "A Framework for the Design of Cellular Manufacturing Systems", *International Journal of Production Research*, 26, 1161 - 1172, (1988).

[42] Tam, K.Y., "An Operation Sequence based Similarity Coefficient for Part Families Formations", *Journal of Manufacturing Systems*, 9 (1), 55 – 68, (1990).

[43] Levenshtein, V.I., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", *Soviet Physics Doklady*, 10, 707 – 710, (1966).

[44] Selvam, R.P., Balasubramanian, K.N., "Algorithmic Grouping of Operation Sequences", *Engineering Costs & Production Economics*, 9, 125 - 134, (1985).

[45] Ho, Y., Lee, C., Moodie, C.L., "Two Sequence Pattern, Matching-Based, Flow Analysis Methods for Multi Flow Lines Layout Design", *International Journal of Production Research*, 31, 1557 – 1578, (1993).

[46] Askin, R.G., Zhou, M., "Formation of Independent Flowline Cells based on Operation Requirements and Machine Capabilities", *IIE Transactions*, 30, 319 - 329, (1998).

[47] Rathmill, K., Leonard, R., "The Fundamental Limitations of Cellular Manufacture when Contrasted with Efficient Functional Layout", *Proceedings 4th International Conference on Production Research*, Taylor & Francis, London, (1977).

[48] Flynn, B.B., "Repetitive Lots: The Use of a Sequence Dependent Setup Time Scheduling Procedure in Group Technology and Traditional Shops", *Journal of Operations Management*, 7(1-2), 203 – 215, (1987).

[49] Garza, O., Smunt, T.L., "Countering the Negative Impact of Intercell Flow in Cellular Manufacturing", *Journal of Operations Management*, 10 (1), 92 – 118, (1991).

[50] Hyer, N.J., Wemmerlöv, U., *Reorganizing the Factory: Competing through Cellular Manufacturing*, Productivity Press, Portland, Oregon, (2002).

[51] Fransoo, J.C., Wouters, M.J.F., "Measuring the Bullwhip Effect in the Supply Chain", *Supply Chain Management*, 5 (2), 78 - 89, (2000).

[52] Qiao, L., Yang, Z., Wang, H.-P., "A Computer-Aided Process Planning Methodology", *Computers in Industry*, 25, 83 - 94, (1994).

[53] Zhao, C., Wu, Z.A., "Genetic Algorithm for Manufacturing Cell Formation with Multiple Routes and Multiple Objectives", *International Journal of Production Research*, 38, 385 – 395, (2000).

[54] Askin, R., Selim, H., Vakharia, A., "A Methodology for Designing Flexible Cellular Manufacturing Systems", *International Journal of Production Research*, 29, 599 – 610, (1997).

[55] Baykasoglu, A., Gindy., N.N.Z., "MOCACEF 1.0: Multiple Objective Capability based Approach to form Part-Machine Groups for Cellular Manufacturing Applications", *International Journal of Production Research*, 38, 1133 - 1161, (2000).

[56] Diallo, M., Pierreval, H., Quilliot, A., "Manufacturing Cells Design with Flexible Routing Capability in Presence of Unreliable Machines", *International Journal of Production Research*, 31, 175 - 182, (1993).

[57] Yin, Y., Yasuda, K., "Manufacturing Cells' Design in Consideration of Various Factors", *International Journal of Production Research*, 40, 885 - 906, (2002).

[58] Hwang, H. and Ree, P., "Routes Selection for the Cell Formation Problem with Alternative Part Process Plans", *Computers & Industrial Engineering*, 30 (3), 423 - 431, (1996).

[59] Adil, G.K., Rajamani, D. and Strong, D., "Cell Formation Considering Alternate Routings", *International Journal of Production Research*, 34 (5), 1361 - 1380, (1996).

[60] Yin, Y., Yasuda, K., "Similarity Coefficient Methods Applied to the Cell Formation Problem: A Taxonomy and Review", *International Journal of Production Economics*, 101, 329 – 352, (2006).

[61] Mansouri, S.A., Moattar Husseini, S.M., Newman, S.T., "A Review of the Modern Approaches to Multi-Criteria Cell Design", *International Journal of Production Research*, 38, 1201 – 1218, (2000).

[62] Colette, Y., Siarry, P., *Multiobjective Optimization: Principles and Case Studies*, Springer, Berlin, (2003).

[63] Pierreval, H., Caux, C., Paris, J.-L., Viguier, F., "Evolutionary Approaches to the Design and Organization of Manufacturing Systems", *Computers & Industrial Engineering*, 11, 5 – 19, (2003).

[64] Fonseca, C.M., Fleming, P.J., "An Overview of Evolutionary Algorithms in Multiobjective Optimization", *Evolutionary Computation*, 3, 1 - 16, (1995).

[65] Ehrgott, M., *Multiple Criteria Optimization: Classification and Methodology*, Shaker Verlag, Aaachen, (1997).

[66] Goldberg, D.E., "Algorithmes Génétiques", Addison-Wesley, (1994).

[67] Deb, K., "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems", *Evolutionary Computation*, 7 (3), 205 – 230, (1999).

[68] Srinivas, N., Deb, K., *Multi -objective Optimization using Nondominated Sorting in Genetic Algorithms*, Technical Report, Department of Mechanical Engeneering, Indian Institute of Technology, Kanput, India, (1993).

[69] Horn J. and Nafpliotis N., *Multi-objective Optimization using the Niched Pareto Genetic Algorithm*, IlliGAL Report No. 93005, University of Illinois, Urbana Champaign, (1993).

[70] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, (2001).

[71] Zitzler, E., Laumanns, M., Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. Evolutionary Methods for Design", *Proceedings of the EUROGEN2001 Conference*, International Center for Numerical Methos in Engineering (CIMNE), Barcelona, Spain, 95 – 100, (2002).

[72] Fogel, D.B., "A Comparison of Evolutionary Programming and Genetic Algorithms on Selected Constrained Optimization Problems", *Simulation*, 64, 399 – 406, (1995).

[73] Fogel, D.B., "Applying Evolutionary Programming to Selected Traveling Salesman Problems", *Cybernetics and Systems: an International Journal*, 24, 27 - 36, (1993).

[74] Powell, D., and Skolnick, M., "Using Genetic Algorithms in Engineering Design Optimization with Non-Linear Constraints", *5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, (1993).

[75] Homaifar A, Lai S.H.Y., Qi, X., "Constrained Optimization via Genetic Algorithms", *Simulation*, 62 (4), 242 – 254, (1994).

[76] Coello Coello, C.A., "Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art", *Computer Methods in Applied Mechanics and Engineering*, 191, 1245 – 1287, (2002).

[77] Michalewicz, Z., Dasgupta, D., Le Riche, R.G., Schoenauer, M., "Evolutionary Algorithms for Constrained Engineering Problems", *Computers & Industrial Engineering Journal*, 30, 851 – 870, (1996).

[78] Kaymak, U., Sousa, J.M., "Weighted Constraint Aggregation in Fuzzy Optimization", *Constraints*, 8, 61 – 78, (2003).

BIOGRAPHICAL SKETCH

Erdem Zengin was born in October 11, 1981 in Ürgüp, Nevşehir. He graduated from Adana Anatolian Teacher Training High School in 1999 and from the Department of Industrial Engineering, Çukurova University in 2003. Since then, he pursues his graduate studies in the Institute of Science and Engineering, Galatasaray University. At this period of his life he worked as an estimating engineer for energy production industry and then an application consultant for a software company.