2007

RULE EXTRACTION TO ESTABLISH CRITERIA FOR MINICELL DESIGN IN MASS CUSTOMIZATION MANUFACTURING

Smitha Thuramalla
University of Kentucky, tsmitha11@yahoo.com

ABSTRACT

RULE EXTRACTION TO ESTABLISH CRITERIA FOR MINICELL DESIGN IN MASS CUSTOMIZATION MANUFACTURING

Minicell-based manufacturing system is used in identifying best minicell designs. The existing method of minicell design generates best minicell designs by designing and scheduling minicells simultaneously. While in this research designing of minicells and scheduling of jobs in minicells is done separately. This research evaluates the effectiveness of 'hierarchical approach' and compares with 'simultaneous' method. Minicell designs with respect to average flow times and machine capacities and both are identified in a multi-stage flow shop environment. Rules for the extraction of good minicell designs in mass customization manufacturing systems are also established.

KEYWORDS: Minicell, Hierarchical, Scheduling, Mass customization, Average Flow time.

Smitha Thuramalla

7/21/2007

RULE EXTRACTION TO ESTABLISH CRITERIA FOR MINICELL DESIGN IN MASS CUSTOMIZATION MANUFACTURING

By

Smitha Thuramalla

Dr. Fazleena Badurdeen

Director of Thesis

Dr. L.S.Stephens

Director of Graduate Studies

7/21/2007

RULES FOR THE USE OF THESIS

Unpublished thesis submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name	Date

THESIS

Smitha Thuramalla

The Graduate School

University of Kentucky

2007

RULE EXTRACTION TO ESTABLISH CRITERIA FOR MINICELL DESIGN IN MASS CUSTOMIZATION MANUFACTURING

THESIS

A thesis submitted in partial fulfillment of the

requirements for the degree of Master of Science in Mechanical Engineering

in the College of Engineering at the University of Kentucky

By

Smitha Thuramalla
Lexington, Kentucky

Director: Dr. Fazleena Badurdeen

Assistant Professor of Mechanical Engineering

University of Kentucky Lexington, Kentucky

2007

Dedicated To My Parents

ACKNOWLEDGEMENTS

I would like to sincerely acknowledge the mentorship and support that my advisor, Dr. Fazleena Badurdeen has extended throughout the course of my M.S. It would not be an overstatement to say without her enthusiasm for research, and encouragement towards success, I would not have successfully completed my thesis. I also thank her for funding me through out my MS. I would like to express my sincere gratefulness to Dr. Rouch, for his kindness in agreeing to be on my committee. I would then like to extend my thankfulness to Dr.Holloway, for agreeing to be on my committee. I would like to thank my D.G.S. Dr. L.S.Stephens, for giving me an opportunity to pursue my MS at University of Kentucky. I would also like to thank all the faculty members and staff of ME and MFS dept. for their continued cooperation. I also have to acknowledge each and every member of my research team and all my friends, whose support has made my life in US very comfortable and also helped me achieve my tasks with ease.

Above all, I would like to thank my parents and my brother for supporting and motivating me to pursue my master's. My brother's encouragement has raised my spirits all through out my study. I am ever indebted for the motivation and encouragement rendered by my family, which made every hurdle surmountable and every task achievable. I dedicate each and every achievement to them and consider myself blessed to have such a family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FILES

1. Smitha-Thesis.pdf-1.56MB(FileSize)

1 INTRODUCTION

1.1 Overview

In the earlier times customer demand was satisfied by producing standardized products in large volumes using mass production techniques. This manufacturing strategy enabled manufacturers to achieve lower unit costs. During this period of time, the main focus of manufacturers was to reduce the cost of products. Due to the limitation in technological developments it was difficult to meet customer's specific needs.

Using the mass production strategy, a high volume of products can be produced but with little variety. But customized products demanded better quality, which were produced in low volumes by manufacturing firms. Hence with the improvement in technology, cellular manufacturing was developed, which could produce medium to large quantity of products incorporating certain degree of product variety. In order to extend this product variety to meet each individual customer's need, mass customization strategy was developed.

With the improvements and advances in information technology and manufacturing, tremendous opportunities are opened for manufacturers and customers. This improvement enables customers to demand for customized products which would meet their individual requirements. In this scenario, manufacturing firms have started to focus on customer needs in particular chosen markets to remain in competition. Hence customization has become an important element in today's manufacturing system.

1.2 Mass Customization: Definition and Challenges

Mass customization is the process of delivering customized products and services for individual customers at near mass production efficiency and price [77]. Manufacturers previously met customer requirements by product differentiation. But in the present era customers are not content with the different products offered in distinct markets. They demand customized products of better quality and also expect at low cost and fast delivery [4].

Mass customization was first anticipated in 1970 by Alvin Toffler in his book *Future Shock* and the term was introduced by Stan Davis in his book *Future Perfect* in 1987[24]. Mass customization gained popularity only when Pine [77] explained it as a manufacturing strategy. During this period, manufacturing methods could either produce standard products at low cost or customized products at high costs using job shop principles. Hence it was difficult to produce customized products during this period, due to the constraints in technology. But with the progress in time, due to the improvement in technology successful mass customization has become more feasible. The role of information technology, such as the Internet, provides direct interaction with customer and helps in increasing the responsiveness of the company [76]. Due to the continuous dialogue with customers, manufacturers can improve their ability to analyze customers' requirements. Thus companies now think about product and process designs in order to meet low cost, high quality, customized products.

The latest definition of mass customization given by Pine as 'low cost, high-volume, efficient production of individually customized offerings' (which, incidentally, may be goods, services, experiences, or transformations) aptly fits in the present market condition [78]. The customers now demand products which are of better quality but at the same time expect them to be delivered at low cost.

There are offerings beyond commodities which cannot be customized as mentioned in Pine and Gilmore's book *The Experience Economy* [79]. According to them, "goods and services are no longer enough; what customers want today are experiences – memorable events that engage each person in an inherently personal way; and transformations, effectual outcomes that change each individual to achieve his aspirations". It was observed by Pine and Gilmore that "little that has been done to mass customize either experiences or transformations, and a world of opportunity exists for the firms wishing to start mass customization.”

Customer requirements in the present era of manufacturing are very diverse. Hence design-to-order and build-to-order tools are used to capture heterogeneous market segments [39]. Gathering customer requirements, finalizing product design and eventual manufacturing are the main tasks, which need to be seamlessly integrated to achieve customization at mass production efficiencies.

One of the challenges of mass customization manufacturing lies in the variety dilemma exhibited by frequent design changes and recurrent process variations [61]. This shows the importance of variety coordination from design to production. The essential problem is to minimize process variations in production in order to fulfill design changes resulting from customization within a given product platform [51].

Most of the literature on mass customization focuses on customized product development through platform-based designs, developing interfaces to achieve customer integration and the problems in these areas [4]. Only few studies have focused on developing the systems for mass customization manufacturing. Various manufacturing strategies have been developed to meet the requirements of mass customization manufacturing. With changing customer demands and increasing competitive markets, manufacturing firms started using cellular manufacturing, Just- In-time and flexible manufacturing system strategies.

Using modular systems to achieve flexibility needed in mass customization manufacturing is one approaches suggested more recently [59, 109]. The application of group technology concepts to form integrated, multi-stage cells and design modular processes to accommodate dynamic demand situations is another suggestion available in literature [48]. However, not many studies have focused on developing systems for mass customization manufacturing. A new approach to design a manufacturing system for mass customization using minicells was proposed by Badurdeen in [6].

A traditional cell consists of products and necessary machines to make these products i.e. traditional cells contain product families and machine cells. A typical traditional product structure is shown in Figure 1-1.

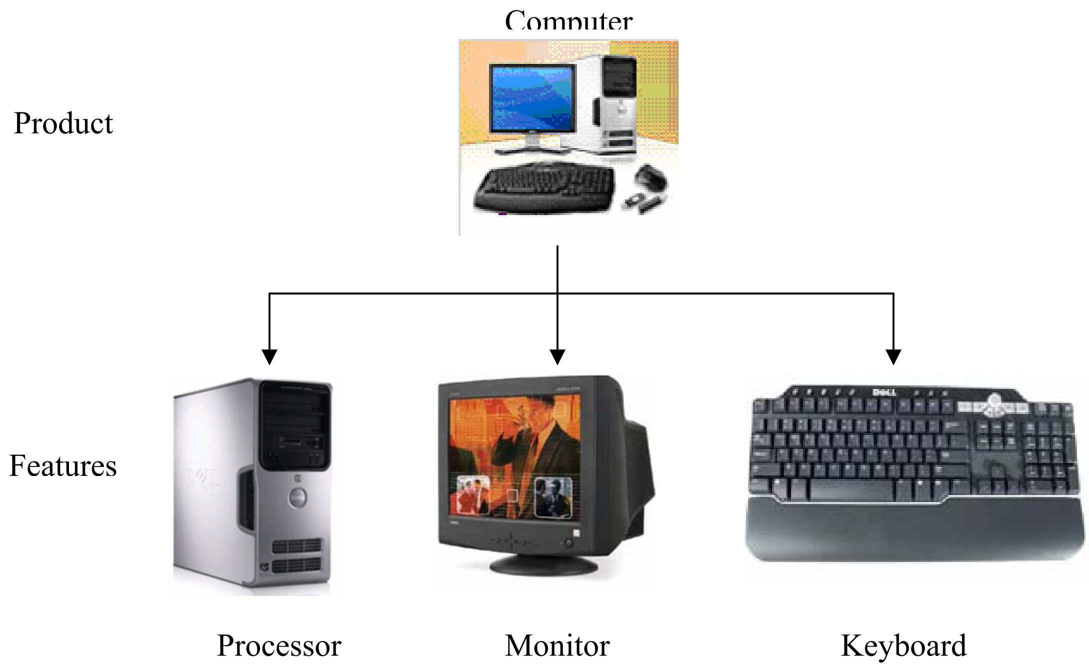


Figure 1-1: Traditional Product structure (Images Adapted from [114])

But in the present mass customization environment, large numbers of options are available for each feature of a product. The product variants are built using different options available for each feature [5]. Hence with this kind of product structure, the total number of product variants would be very high and containing all the processes using traditional cells would result in large cells [6]. Large cells are difficult to manage and could be less flexible ([112], [49]) particularly with dynamic demand, contrary to the requirements of mass customization [4]. Also processing of the product variants using large cells in traditional manufacturing would result in long lead times and large in process inventory [6].

An alternative approach to the use of traditional cells is to form smaller cells which use options available for features, rather than product variants. The demand for the options is likely to be less dynamic than that for the product variants since demand for an option is the sum of the demand of several product variants [4]. These smaller cells are designated as 'minicells' by Badurdeen [6] and are dedicated to producing options families as opposed to large traditional cells for product families [6].

In a minicell based approach, minicells are formed by considering options, which make up a feature, and their processing requirements. Minicells are the manufacturing cells used to process a set of option families as opposed to traditional cells used to produce product families [6]. A platform-based mass customization product structure is shown in Figure 1-2.

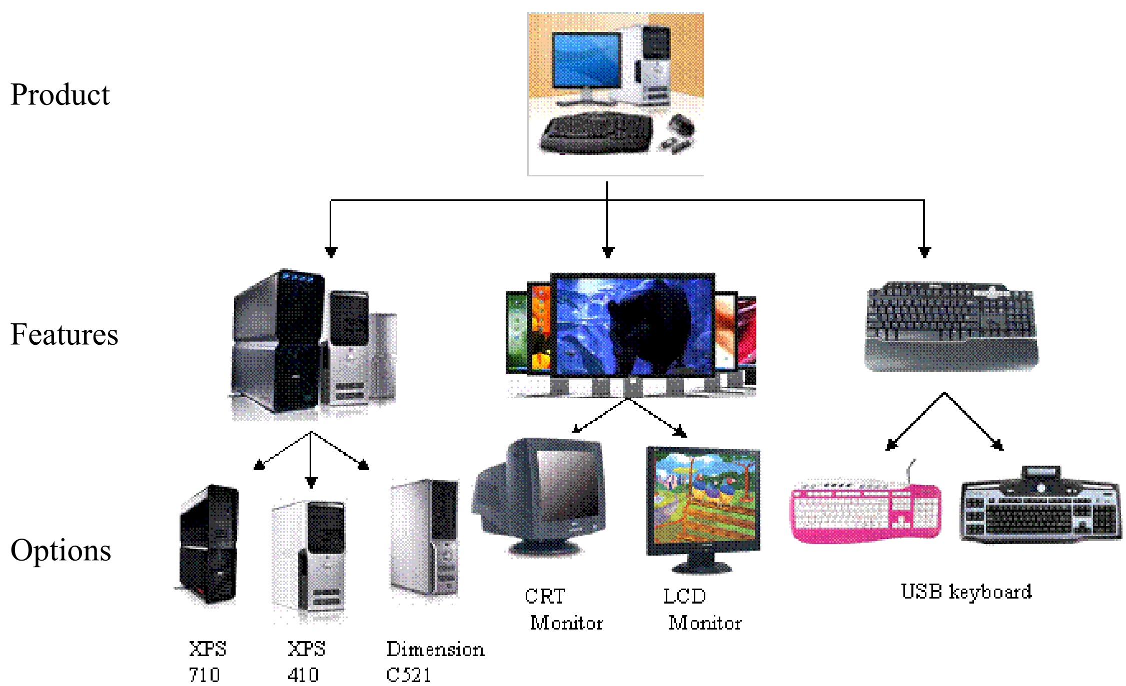
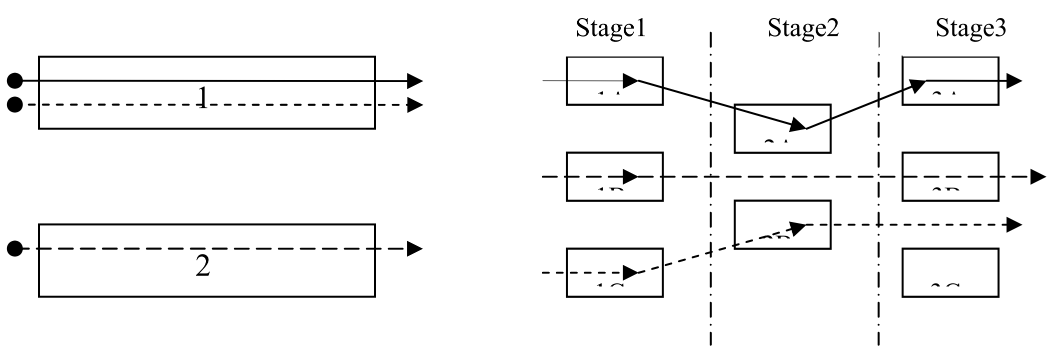


Figure 1-2: Mass Customization Product structure (Images Adapted from [114])

The minicell configuration is developed using the option-machine matrix as opposed to the product-machine matrix in traditional cellular manufacturing. The matrix is divided into multiple stages and option families. Minicells are then formed within each sub-matrix, by grouping machines required for processing a set of options. Separation into multiple stages helps increase the modularity of the system [6]. In order to complete processing the product variants would be directed to several minicells based on the options selected by the customers for each product variants.

In traditional manufacturing, all parts in a family pass through a cell from start to finish as shown in Figure 1-3(a). With the minicell configuration, the large cell is replaced by several small minicells as shown in Figure 1-3(b); Machines available to one large cell are made available to several option families in traditional manufacturing, thus increasing the flexibility of the system.

Options available for the features may need similar processes but with different tools and setups [6]. Depending on the requirement of an option family, machines and operators are grouped together to form a minicell. The multi-stage minicell structure requires products to be routed to several smaller minicells based on the options selected for the product.



(a): Traditional Cells

(b): Minicells

Figure 1-3(a): Traditional Cells and (b): Minicells (Adapted from [6])

1.3 Motivation for Research

Badurdeen [6] developed a robust minicell configuration by considering the design and scheduling of minicells simultaneously i.e. the effectiveness of minicell designs was evaluated by considering its performance with the job scheduled as well. The desired performance criterion was based on the weights assigned to different performance measures, makespan and machine count. They used two different approaches to design a minicell configuration in order to identify the best design while minimizing two performance measures. The first approach was to separate operations (machines) required to process options into multiple stages and the second approach was to separate the option- machine matrix into stages based on the features available [4].

The objective of previous research was to identify the best minicell design while minimizing makespan and machine capacity simultaneously. The minicells were first designed and product demand is then scheduled and the best minicell design is obtained based on the particular fitness value. Hence all designs would be evaluated based on the desired performance criteria and relative weights assigned. Thus, in this process, there is a possibility that the designs which generate the minimum machine requirements may not be selected because the emphasis has been on both measures. Also, based on the previous work, Badurdeen and others have not been able to establish any rules/steps that can be used to design minicells without pursuing the lengthy process of developing a metaheuristic model as they did. Hence an alternate approach would be to first design and then schedule jobs in minicells to determine minicell designs that meet the desired performance criteria.

This method may enable identifying rules, to first find good minicell designs and then screening them to find the best design without enumerating through all possible minicell configurations. Hence it is necessary to evaluate the effect of designing and scheduling in minicells separately, by focusing on a single objective at a time and analyze the results under these situations. This alternate approach of selecting the best minicell design by achieving each objective (designing and scheduling in minicells) independently is to identify if it generates better results than the previous method.

1.4 Research Objective

The objective of the thesis is to identify the best minicell design by using a hierarchical approach to design minicells first and then schedule jobs in minicells for mass customization manufacturing. This approach is an attempt to verify if better results would be generated with a ‘hierarchical’ method than with the ‘simultaneous’ method of designing and scheduling to select a design followed previously. The performance of hierarchical and simultaneous approaches will be compared in this research work. The present approach is termed as hierarchical, since several of best minicell designs will be first identified in the design stage based on minimum machine requirement criteria, then using this data, jobs are scheduled in minicells to obtain minimum average flow time. Based on the desired objective, the design and scheduling results will be studied and the best minicell designs will be chosen depending on the objective. Designing minicells using either method, hierarchical or simultaneous, involves extracting a large number of alternate designs to select the most appropriate minicell design subject to the desired performance. This can be time consuming, particularly when the problem size is large. Therefore, the other major objective of this research is to evaluate minicell designs to guide in the process of extracting rules of designing the best minicells for a given problem.

1.5 Thesis Organization

A literature survey about recent developments in mass customization, cell formation strategies, flow shop scheduling, particularly for minimizing average flow time in a flow shop manufacturing system are presented in chapter 2. The methodology used for designing and scheduling in minicells is presented in chapter 3. Details about the experimentation conducted and results obtained are discussed in chapter 4. Chapter 5 focuses on the conclusions and discusses about the future work.

2 LITERATURE REVIEW

A summary of literature related to topics covered in this research are presented in this chapter. Initially a review of mass customization literature is presented. This is followed by the review of literature on cell formation strategies used in manufacturing cell design and strategies to minimize flow time.

2.1 Mass Customization

The term 'Mass Customization' was introduced by Davis [24] in 1987 in his book *Future Perfect.* According to Davis, it is possible to "customize each product in a batch of products while still process the batch as a whole as in mass production" i.e. each is understood to be both part (customized) and whole (mass) simultaneously.

In the term 'Mass Customization', mass relates to "mass production efficiency" and the term customization corresponds to the needs of "individual customers" [70]. The mass customization concept was popularized by Pine [77] as a manufacturing strategy. According to Pine [77], mass production strategy is no longer sufficient to cater to the changing demands of the consumers. The advances in manufacturing and information technology have made the transition from mass production to mass customization strategy feasible [77]. Also, Kotha [30] valued it as "the emerging paradigm for competitive advantage".

The old production methods like craftsmanship, craft production and mass production has led to the development of mass customization. In craftsmanship method, production of goods and delivery of services was carried out utilizing few employees. Continuing industrialization reduced the importance of this method and led to craft production where part of the manual production was automated [77].

Manufacturing companies later focused on mass production, since it was assumed production costs could be significantly reduced by substituting human work for machines [3]. This production system resulted in the large volume manufacture of low cost, high quality, standard goods and services. But factors like increase in product diversity, changing customer preferences, short product life cycles, have led to the emergence of a new business strategy called mass customization. This strategy led to the development of low cost, high quality, customized goods and services produced on a large scale to a mass market [3].

The improvements in technologies for manufacturing processes and development of new concepts such as cellular manufacturing (CM), Just-in-time (JIT), Flexible manufacturing systems (FMS) helped in achieving some of the objectives of mass customization in manufacturing [6]. CM involves the application of Group Technology (GT) concepts to manufacturing. Tools and machines used to manufacture a similar product family [40] are identified and grouped into a separate manufacturing cell. However, literature shows that the cellular system is not efficient when there is a frequent change in product sequence or in the composition of part or product family [25] meaning that this system is not feasible in a mass customization environment which involves dynamic change in product demand.

Just-In-Time (JIT) strategy emphasizes on reducing wastage by reducing the amount of inventory and decreasing set-up times and achieves a lot size of one [16]. The strategy is primarily applied to industries where similar products or components are manufactured repeatedly [47]. But JIT cannot be efficiently used to produce customized products on a make-to-order basis, particularly if products are fabricated after receiving customer orders.

Flexible Manufacturing system (FMS) provides flexibility in the system adapting to the different changes in market. Production level can be controlled by the numerical controlled machines. Hence introducing a FMS that can 'make-to-order' along with proper supporting infrastructure such as advanced information technology systems will help in increasing the flexibility [47]. However, the high expenditure of advanced information technology systems such as the numerical controlled machines and internet, limits the use of this system in low volume manufacturing plants [16].

Customization is being followed by companies in its own way. In simple words, the level of customization varies from company to company. Different approaches are suggested by the researchers to classify mass customization from an organization's perspective. Lampel and Mintzberg [63] presented a systematic approach in classifying mass customization based on customer involvement in the value chain activities. They identified five manufacturing strategies: pure standardization, segmented standardization, customized standardization, tailored customization and pure customization. Their mapping of the strategies in relation to organizational value chain and the point at which customization takes place is shown in Figure 2-1.

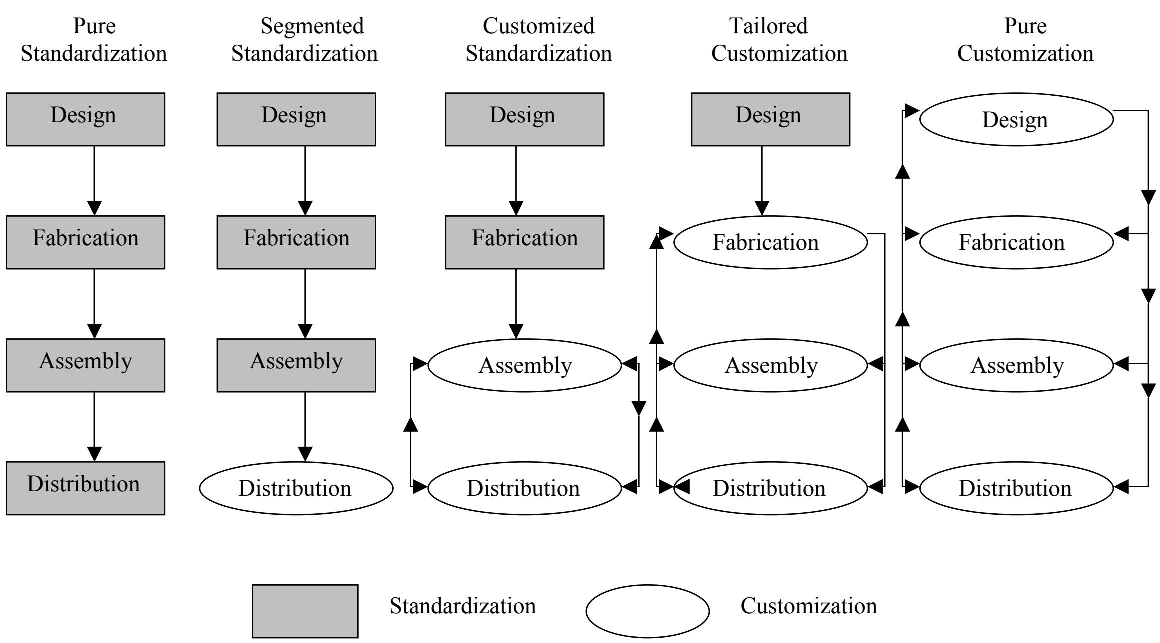


Figure 2-1: Lampel and Mintzberg's Continuum of Strategies [63]

Gilmore and Pine [39] identified four approaches to mass customization derived mostly from empirical observations. The four approaches are defined as collaborative, adaptive, cosmetic, and transparent customization. In collaborative customization customers are involved right from the design stage. In this type, the product is manufactured as per customer's specifications. An adaptive mass customizer offers a product which can be altered by the customer himself as per his individual requirements. Differentiation of a standard product mainly through packaging is done in cosmetic customization. Transparent customization provides "individual customers with unique goods or services without letting them know explicitly that those products and services have been customized for them" [39].

Firms adopting mass customization should also include customer's specifications in product design [28]. Duray, et al [28] suggested mass customization classification by identifying customer involvement in product design and the type of product modularity involved. It was also shown that mass customizers can be identified and classified based on customer involvement and modularity type [28].

The effects of modularity and customer involvement on production planning, channel management and materials management were explored by Duray [27]. This study identifies that the type of modularity plays a major role in the choice of channel integration, inventory and planning techniques. Customer involvement in the production cycle does not make significant impact in the choice of channels or inventory management but does contribute to the choice of planning systems [27].

Different companies follow mass customization to different extents. For example in Hewlett-Packard (HP) printers, power supply component was installed at distribution center rather than at production center. This introduces customization in the product at the assembly stage while helping the company to reduce holding large inventory [30].

Dell Computers uses standard mass produced modular components in their product (computers) to offer customized computers to their customers by allowing them to choose from a list of options for the different features [114]. National Bicycle Industrial Company (NBIC) of Japan offered customized bicycles to their consumers by custom building parts like frame and using standard parts for the rest of the bicycle. The bicycle is then assembled with the custom fabricated components along with the standard parts

[65]. Toyota is one more example which manufactures cars by offering few customized features in its product [115]. All these examples clearly show that customization is followed in both the assembly and distribution stages.

Several other companies like Nike and Adidas shoes also provide customization in their products by involving customization from fabrication stage. Even cell phones, clothes and houses can be custom built in the present changing market conditions. The main causes of this trend could be due to the increasing saturation of markets and the pressure of globalization on local economies [70].

Mass customization offers benefits not only to customers but also to the marketers. This method enables a consumer to get whatever he demands. This in a way translates into higher sales and better customer loyalty for the marketer. Greater efficiency is attained due to lower inventory levels throughout the distribution channel. Due to product's better fit with customer needs, product alterations or modifications are reduced significantly. Thus it gives scope for manufacturers to raise the price levels of products [11].

To successfully implement mass customization strategy, three major components of a system must be identified: elicitation, process flexibility and logistics [116]. Elicitation is the process by which marketers interact with customers and procure information about their specific needs. Internet is a major tool being used for communication between the manufacturers and consumers. But in some instances consumers may be reluctant to use the internet due to fear of error or being overwhelmed with the number of options. Firms can overcome this problem with the use of system choice boards and design technologies [116].

A methodology to classify the product design information, which can easily accommodate design variations based on product platform architecture, is proposed in [39]. Further, selection of most conformal design family and mapping of parameters was also discussed in [39]. Experienced, efficient sales personnel are required in companies not implementing Web-based elicitation systems in order to interpret costumer's response correctly. Hence “the challenge is if you are making units of one, your margin of error is zero” [94].

With traditional manufacturing, it is relatively simple to coordinate production with sales requirements and purchase parts to meet production requirements. But as the product variety increases it would be difficult to manage the information management system. Especially in the present scenario of mass customization, firms would have no knowledge of what parts would be needed, what products need to be manufactured and what goods needs to be transported until customer orders are received. Mass customization manufacturing begins with a customer order that forces companies to take different approaches to organize the workflow. One such approach in this unpredicted environment would be to use a well-integrated logistics information system.

Based on Zipkin’s ideas [116], Logistics is an important component of a mass customization system. Since a mass customized product would have special options in the same product, it would be a challenge for mass customized firms to create the product with special interest, with lower handling costs and to meet delivery expectations. Hence, a highly coordinated supply chain is necessary to efficiently deliver a customized component to right customer at right time.

The major capability needed for mass customization is the use of flexible production processes. These processes must be flexible enough to accommodate the varying needs of customers and produce them at mass production cost. One of the components of process flexibility is the use of modular design combined with postponement of product differentiation [11].

Successful mass customization must employ a production strategy that incorporates modularity into components and process [80]. Modularity is achieved in two stages. In the first stage, common parts of mass customized products are manufactured as standard modules. In the second stage, product distinctiveness can be achieved through combination or addition or modification of the modules. The second stage can be referred to as postponement of product differentiation [11].

A flexible manufacturing system enables efficient production of different varieties of products in small quantities using a single assembly line. For a known variety range, cost and time penalties and changeover times can be minimized. By analyzing the characteristics and components of mass customization manufacturing systems, it is observed that better perspective and understanding of manufacturing requirements is essential.

2.2 Manufacturing System requirements for Mass customization

A mass customized manufacturing system produces right parts at right time in the most effective manner using less number of tasks. The advances in the technology such as CM, JIT, FMS and the work of several researchers helped in achieving these objectives.

Qiao, et al. [84] proposed enabling technologies for mass customization manufacturing. They suggest an Extensible Markup Language (XML) based information integration platform to support dynamic reconfiguration of mass customization manufacturing. A simulation model of an aircraft major component assembly line is generated from the XML-based shop data specification which clearly demonstrates the powerful nature of mass customization manufacturing. The approach of simulation modeling helps manage a flexible customized manufacturing system in a flexibly modulated and customized fashion. The application of the XML can enable effective data exchange across various hierarchies in a system as XML uses a commonly accepted text based data structure [84]. A simulation system model for mass customization manufacturing was developed using valid, colored Petri Net by Qiao, et al. [87]. This model was able to represent solutions to dynamic rescheduling, shop reconfiguration, part rework processing and multi robot cooperation and coordination [86].

Process control is identified as one of the critical problems in mass customization manufacturing. With small batch size, high variety in products and irregular insertion of new orders from customers can make the process control more volatile. A process platform for coordinating product and process variety was proposed by Jianxin, et al. [52]. Process information is usually established manually in the industries. It is challenging to handle process control information efficiently when new scenario changes are incorporated. To address this, process control driven by an XML-based shop data file was developed by Qiao, et al. This simulation model yielded in optimized resource utilizations and improved system efficiencies [85].

A high level of reconfigurability and flexibility in production is required in areas like production planning and control and effective information systems for a mass customized manufacturing system. Internet based concepts with the application of multi- agent systems provide necessary interoperability and organizational alignment in mass customization. Benefits of this integrated view arise especially for production planning and control and for the reconfiguration of the production environment. The MAS Internet based production concepts lead to an enabling of mass customization [12].

Flexibility in mass customization environment can be achieved through the use of modular processes [59, 109]. The role of modularity for effective mass customization and the way modularity helps the manufacturing system in meeting rapidly changing customer demands have been discussed in the literature previously [77, 10, 31]. The minicell based manufacturing system, proposed by Badurdeen [6] and further studied by Badurdeen and others [4] apply the concept of modularity to manufacturing system design for mass customization. In this proposed manufacturing system, the option machine matrix is divided into different stages and option families (minicells) are then formed within each stage. Products are routed through minicells, based on options used in each product variant. Separation into several stages serves to increase the modularity of the system. A modular manufacturing system results because of the multi-cell, multi-stage environment [6].

The effectiveness of a minicell based manufacturing system design was measured in [4, 5, 8]. The results of two minicells per stage in a 3 stage minicell design and two cell traditional cell manufacturing system designs are compared in order to study the minicell based and traditional manufacturing systems. When the demand for product variants was high, minicells outperformed the traditional cells. The minicell-based approach generated lower average flow time than the traditional cells. Also makespan was found to be comparable to that with traditional cell design but with a lower machine count [8].

Designing of cells in manufacturing is defined as the allotment of machines and products/parts to a cell. The minicells are dedicated to the production of options similar to a manufacturing cell being dedicated to the production of similar processing requirement parts. Hence the cell formation concepts of cellular manufacturing are most applicable and useful in designing minicells. The procedures used to identify and define part families and/or machine cells are referred to as cell formation procedures [102]. A brief review of manufacturing cell formation strategies is given in the following section.

2.3 Manufacturing Cell Formation

Several different approaches were suggested for traditional cell formation (CF) in the literature. In the cellular manufacturing field, cell formation is concerned with grouping of machines and labor into cells. The cells are dedicated to the production of part families which have similar processing requirements. As per Shafer and Rogers [101] a cell formation problem should, ideally, consider multiple objectives.

A number of researchers have developed taxonomies for CF techniques. The contributions of several researchers were integrated by Shafer [102] and presented in Figure 2-2. According to the taxonomy manual methods, classification and coding, statistical clustering techniques, algorithms for sorting machine part matrix, mathematical techniques and Artificial Intelligence techniques were the six basic methodologies developed for cell design [102]. CF strategies were classified as descriptive procedures, cluster analysis, graph partitioning, artificial intelligence and mathematical programming by Selim, et al. [92].

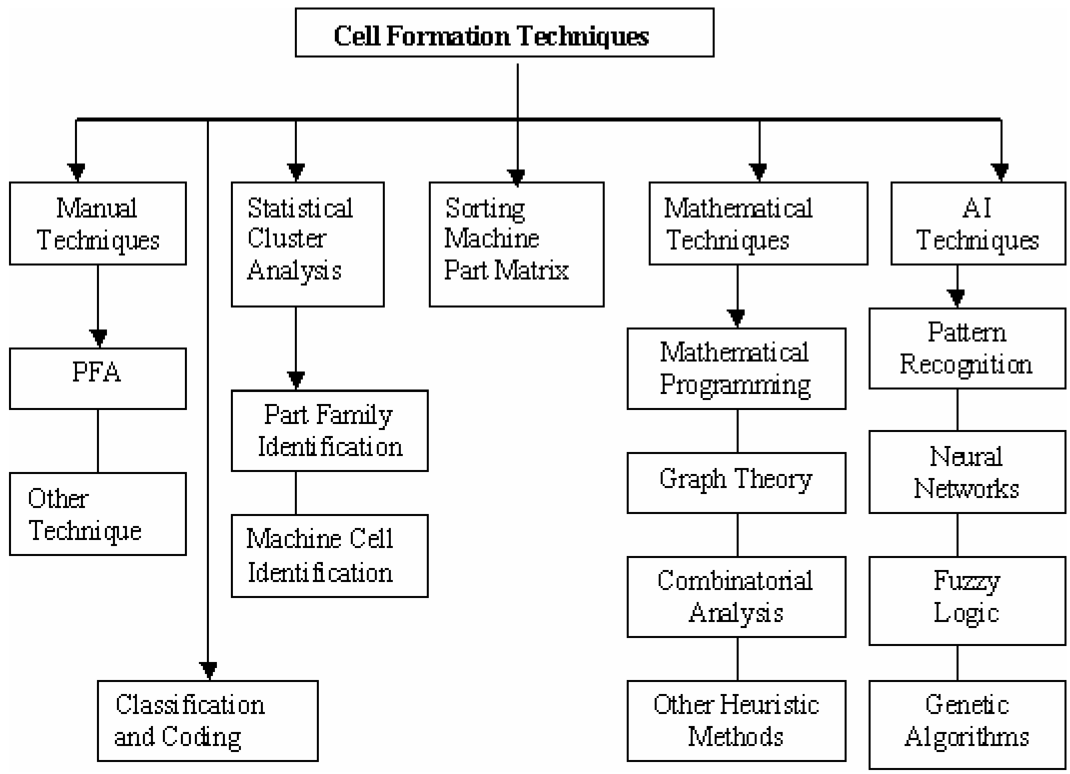


Figure 2-2: Classification Framework of Cell Formation Techniques [102]

Wemmerlov and Hyer [28] have classified CF procedures into four categories while Chu [22] classified into six categories. Singh [105] identified eight categories and Kaparthi and Suresh [108] added three additional categories of cell formation procedures.

Manual CF techniques require the analyst to identify part families and machine cells iteratively. Production Flow Analysis (PFA) developed by Burbidge [101] is a part of manual techniques. In this method part families and machine cells are based on part routing information. Though PFA is one of the most comprehensive CF procedure developed to date, manual techniques do not lend themselves to being implemented on a computer [57]. El-Essawy and Torrance [13] developed a CF procedure similar to PFA called component flow analysis. In some respects CFA differs from PFA since the latter partitions the problem whereas the former does not.

A number of classification and coding systems were developed in the initial days to facilitate the practice of group technology. This system assigns alphanumeric codes to each part. Based on the codes, parts are grouped into families [102]. A group of machines is assigned to each part family. But same set of machine tools may not be used for manufacturing parts of similar shape and size. Another disadvantage of classification and coding system is its use is time consuming [29].

Cluster analysis consists of diverse techniques to identify part families or machine cells. The main objective of this statistical tool is to group entities or their attributes into clusters such that individual elements within a cluster have a high degree of “natural association” among themselves and that there is very little "natural association" between clusters [98]. Selim et al [98] classified clustering procedures as: array-based clustering techniques, hierarchical clustering techniques and non-hierarchical clustering techniques. In solving the cell formation problem using array manipulations, the machine route of each part is noted and converted to machine-component incidence matrix. The rows and columns of this matrix are rearranged until a block-diagonal structure emerges. This structure groups parts into part families and machines into machine cells [98].

The literature yields several array-based clustering algorithms like Rank Order Clustering (ROC) by King [58] and King and Nakornchai [56], Modified Rank Order Clustering by Chandrasekharan and Rajagopalan [19], Direct Clustering Analysis by Chan and Milner [17]. But the array based methods are dependent on the initial configuration of the zero-one matrix and does not provide disjoint part families and machine cells for ill-structured matrices [66].

In hierarchical clustering, similarity or dissimilarity between machines or components are computed and clustered so as to maximize similarity [92]. A measure of proximity that quantifies either the similarity or distance (dissimilarity) between two parts or machines is referred to as similarity or distance coefficient [102].

With statistical cluster analysis, each part or machine is initially placed in its own separate cluster. Then the clusters are successively combined together based on selected clustering algorithm until all parts or machines are grouped into single cluster. If similarity coefficient is used, clusters are combined starting with clusters that are most similar and ending with combination of clusters that are least similar. On the other hand, with distance measure clusters are combined starting with two clusters that are least dissimilar and end with combination of clusters that are most dissimilar. Statistical cluster analysis is also referred to as hierarchical clustering [102].

Among the several approaches to the cell formation problem, those based on Similarity Coefficient Method (SCM) are more flexible in incorporating manufacturing data such as production volume, sequence of operations and processing times into process [97]. The early algorithms using clustering and similarity coefficients were developed by McAuley [67] and Carrie [15]. McAuley for the first time used Single Linkage Clustering (SLC) by applying similarity measure for machine pairs and formed machine cells. Carrie used similarity between the parts as a basis for grouping as opposed to similarity between machine types. Seifoddini and Wolfe used Average Linkage Clustering to form machine cells [96].

Single Linkage Clustering (SLC) and Average Linkage Clustering (ALC) are the most widely used clustering algorithms. With SLC, two clusters are combined based on the strongest single link between the clusters. The similarity between clusters is calculated considering the maximum value among the elements of cluster. In contrast, average value of all links between two clusters is the basis of combination of clusters in ALC. The average value of the elements of cluster is considered in ALC [102].

While Single linkage clustering is the simplest of all clustering algorithms and requires minimal computational requirements, it may generate machine cells in which a large number of machines are far apart in terms of similarity [97]. The problem of chaining due to the use of SLC was investigated and ALC was suggested for CF which eliminates the formation of enlarged cells [95]. ALC improves the solution by reducing the number of intercellular moves by assigning machines to cells, that their members have the largest number of common operations with them. Also the machine cells formed by ALC are better separated compared to the machine cells formed by using SLC [97]. But computational requirement of ALC is significantly higher than SLC.

Several other researchers have also proposed different similarity coefficient based cell formation methods. Choobineh [21] developed a similarity coefficient based CF method that considers sequence of operations. Selvan and Balasubramanian [99] developed a dissimilarity measure based on operations sequences. A similarity coefficient which considers within-cell machine sequence and machine loads was studied in [111].

Non-hierarchical clustering methods are iterative methods and they begin with either an initial partition of the data set or the choice of a few seed points. In both cases, number of clusters has to be decided in advance [98]. Part families and machine cells are formed alternatively until a good block-diagonal structure of the input machine-part incidence matrix is obtained [92].

A non-heuristic algorithm for group technology problems was developed and demonstrated by formulating the problem as a bipartite graph [18]. Chandrasekharan and Rajagopalan [20] developed an algorithm for the concurrent formation of part families and machine cells. The formation of part families and machine cells has been treated as a problem of block diagonalization of the zero-one matrix. An efficient nonhierarchical clustering algorithm was developed by Srinivasan and Narendran [106] that identified seeds for clustering by solving an assignment problem.

Based on production data, Gupta and Seifoddini [44] and Nair and Narendran [71] developed clustering algorithms. Similarity coefficient developed in [44] considered production data such as part type, production volume, routing sequence and unit operation time. Nair and Narendran in [71] considered part sequence while considered production sequence, volumes, processing times and machine capacity in [72].

Non-hierarchical algorithms do not impose any restriction on cell size or maximum number of cells and obtain a natural grouping from the input matrix. But the quality of solution obtained from these algorithms usually depends on the initial machine clusters. Also, these algorithms do not address the problem of alternate routings [92].

Mathematical techniques include a variety of analytical cell formation techniques including mathematical programming, graph theory etc. Mathematical programming techniques use linear programming or quadratic integer programming or goal programming to identify part families and their corresponding manufacturing cells [98]. These approaches can consider a variety of objectives and also include a number of problem limitations. Due to their computational limitations to solve large problems and requirement of sophisticated algorithms to solve mathematical models, they are not widely used [98].

Rajagopalan and Batra [88] were the first to use graph theory to solve the cell formation problem. They developed a machine graph with as many vertices as the number of machines. The limitation of this method was that machine cells and part families were not formed simultaneously. Minimum spanning tree for machines was constructed by Srinivasan in [107] from which seeds to cluster components were generated. A Hamiltonian path algorithm for the grouping problem was suggested by Askin et al [2]. With this approach, actual machine groups are not evident from the solution.

The techniques of neural networks, fuzzy logic and genetic algorithms are considered under artificial intelligence approaches. Kaparthi and Suresh [54] applied Adaptive Resonance Theory [ART] for the part-machine grouping problem. Later in [55] they showed that the performance of a basic ART model could be improved by reversing the zeros and ones. Kusiak [62] developed a pattern recognition based parts grouping.

However, these techniques cannot be applied directly to the minicell formation. Minicell formation needs evaluating the processing requirements of options not products as in traditional manufacturing. Since minicell configuration consists of multiple stages, traditional cell formation methods must be applied separately to each stage [6].

Also cell formation techniques that proceed by diagonalizing the product-machine [57] matrix cannot be used in minicell based configuration. Since in the minicell formation, column sequence in the option machine matrix must be preserved to maintain sequential flow across stages.

Among the different cell formation methods, similarity coefficient method identifies the manufacturing cells with part families and processing of the parts is performed in each cell. The similarity coefficient method first form part families and then assign machines to the families and hence increase the flexibility of the system. Since the concept of minicells requires processing of options in each minicell as the similarity coefficient method, this method is most apt for minicell design and hence is selected for the present research. Also similarity coefficient method outperformed other methods to identify machine cells by providing greater flexibility to assign duplicate machines to alternate cells with part family identification [100].

Cellular manufacturing systems, though achieve efficiency by exploiting part similarities are not as flexible as functional layouts [104].This fact enlightens the need to design manufacturing systems for mass customization environment. This also shows that mass customization production systems must incorporate the features of CM, but at the same time include some of the functional layout characteristics. This shall enable to increase flexibility required for product variety.

Once the product/part families are assigned to machine cells, it becomes critical to schedule these jobs in order to achieve desired performance objectives. Hence it becomes important to schedule jobs in the cells [48]. Scheduling helps reduce the cost of production and delivery time in the system. Hence to improve the performance of minicells it is important to determine an efficient scheduling strategy.

2.4 Scheduling

The scheduling or sequencing problem involves determination of the relative position of job ‘i’ to all other jobs as they proceed from one machine to another [73]. A flow shop is a manufacturing system that consists of series of ‘m’ machines in which given ‘n’ jobs are processed [62,9]. In permutation flow shop problem, jobs are processed in the same order on all machines [82, 83]. In contrast, non-permutation schedule gives the sequence which does not remain same for all machines in a flow shop. Most of the research is associated with finding good permutation schedules because it is less complex compared to finding a non-permutation schedule.

The scheduling problem in flow shops is to find a sequence of jobs for each machine according to certain performance measure(s). Most of the research during the last decades has concentrated upon the minimization of the makespan [35]. Makespan and flow time are two important performance measures considered in scheduling. Makespan is defined as the least time in which all tasks are completed while flow time is the length of the time that a job remains in the system [7]. Flow shop scheduling is classified as NP hard [36] and this complexity makes it difficult to have exact solution methods for more than two machines [37]. The complexity increases even further for non- permutation flow shop scheduling.

Based on the variability of data (such as demand), flow shop scheduling can be classified as Deterministic and Stochastic. A flow shop scheduling problem is termed as deterministic when data about all the jobs is available. While if any one of the parameter such as demand is variable, it is called a stochastic problem.

The job arrival rate plays a vital role in scheduling jobs on different machines. If the job arrival times are known in advance or if the jobs are available at the beginning of scheduling period, then the process can be considered as static. On the other hand if the job arrival is random it is called dynamic. The latter depicts the situation encountered in the real world. All the jobs have to be processed on all machines in a pure flow shop while in a general flow shop case a couple of jobs may skip processing on some machines [32]. The complexity of scheduling problem increases with the increase in the number of variable parameters in a system [84].

Optimization algorithms for the two and three machine flow shop problems with respect to different objectives have been developed by Johnson [83] and using branch-and-bound technique by Ignall and Schrage [53]. However they were found to be not very effective on large or even medium sized problems. As the vast majority of flow shop scheduling problems were proved to be NP-complete (Garey, et al. in [37]), research was directed towards the development of heuristic or near optimal methods. Many heuristics were developed to obtain good solutions within a limited amount of computation time.

Enumeration methods or analytical methods, heuristics and meta-heuristics are some of the solution procedures used for solving complex scheduling problems. While enumeration methods give the optimal solution and are the most efficient, the other two procedures give optimal or near optimal solutions. But there is restriction on the size of the problem that can be solved using the enumeration methods [93]. In order to solve this, heuristics were developed to obtain optimal or near optimal solutions for large size problems.

Most of the flow shop research to date has focused on the problem of minimizing makespan, because minimizing the total production run can be achieved through this [90]. But in recent years, flow time minimization is drawing more attention from researchers possibly due to, this criterion being directly related to measuring effectiveness of responding to customer orders. Also, with the minimization of flow time, stable or even utilization of resources, rapid turn-around of jobs and minimization of in-process inventory [90] and reduction of scheduling costs [1] can be achieved.

A heuristic is regarded as Composite if it employs another heuristic for one or more of its phases of solution [35]. These kinds of heuristics are developed to modify or enhance the previous successful heuristics. Allahverdi and Aldowaisan [93] developed one such composite heuristic (IH6) which combines WY and RZ heuristics. Also pair-wise exchange procedure was applied to NEH, WY, RA heuristics to obtain new heuristics. This yielded significant improvement in flow time with negligible CPU time [93].

While Liu et al. [64] studied the effectiveness of combining constructive heuristics and local search methods. Comparing constructive heuristics (WY, RZ) and composite methods, it was seen that applying local search improves the solution quality on the solutions generated by constructive heuristics. Composite methods are more effective than constructive heuristics alone, at the expense of more computation time. Hence a trade-off between solution quality and computational expense needs to be determined for the application of composite heuristic [64].

Several heuristic procedures developed in literature for flow time minimization in permutation flow shops are shown below in a tree diagram in Figure 2-3. The tree diagram for the scheduling heuristics is drawn in such a way as one proceeds up from bottom, the heuristic above performs significantly better than the lower heuristics in terms of quality and time i.e. in Figure 2-3 the heuristic by Framinan and Lesiten [33] and B5FT [32] outperforms Allahverdi and Aldowaisan [1].

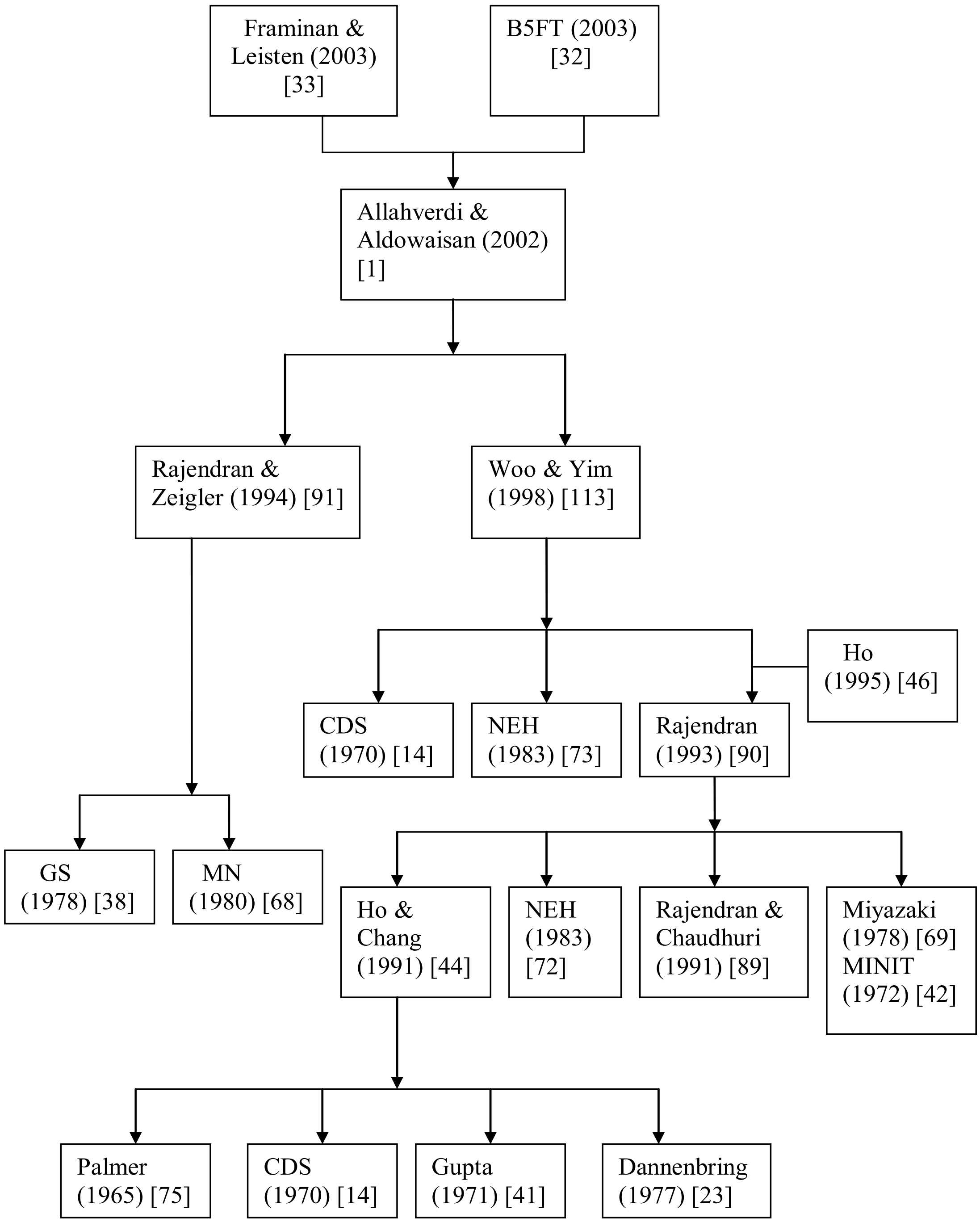


Figure 2-3 : Tree Diagram of Scheduling Heuristics [Compiled from literature]

The Rajendran and Ziegler (RZ) heuristic [91] is one of the earlier heuristics for minimizing of flow time. RZ heuristic consists of two phases, the first one involves generating a seed sequence and the second improves the solution. Woo and Yim [113] (WY) developed a heuristic for minimization of mean flow time in an 'm' machine flow shop. Unlike NEH [73] and RZ, WY does not require an initial starting sequence. The method consisted of two phases where in jobs are ranked according to ascending sum of processing times in first phase and then partial sequences are obtained by inserting non-scheduled jobs in all possible positions. The WY heuristic and the RZ heuristic were outperformed by the heuristic developed by Liu and Reeves [64] (LR heuristic). This heuristic is based on developing an index function to sort the jobs.

Framinan and Leisten [41] (FL) proposed a heuristic for total flow time minimization in permutation flow shops. The heuristic exploits the idea of optimizing partial schedules present in the NEH heuristic. Additionally, Framinan, et al. [82] evaluate all possible five-tupels among the 177 approaches and select the best of them (named Best 5 Flow time i.e. B5FT in the following). According to their results, the B5FT outperforms both WY and RZ in terms of quality of the solutions.

With respect to the rest of the heuristics, FL heuristic and B5FT are best in terms of producing good quality of the solutions as can be observed from Figure 2-3. The Average Relative Percentage Deviation (ARPD) and CPU time (in seconds) are compared for both the heuristics [35] for different number of jobs and considering the maximum number of machines used in [35]. As seen Figure 2-4 and Figure 2-5, if the numbers of jobs are less than 100, the computational requirements of FL and B5FT heuristics are very closely comparable. The difference in the CPU time between the two heuristics is significant only after 100 jobs i.e. if the jobs are more than 100, then FL heuristic consumes more time than B5FT. Due to the complexity involved in calculating the average flow time with B5FT heuristic, FL heuristic is selected in the present research for scheduling of jobs in minicells.

		B5FT		FL	
Machines	Jobs	ARPD	CPU	ARPD	CPU
20	10	2.359	0.01	2.749	0
20	20	2.471	0.05	2.882	0.05
20	30	2.424	0.14	2.633	0.28
20	40	2.637	0.34	2.893	0.93
20	50	2.65	0.63	2.843	2.3
20	60	2.458	1.09	2.697	4.67
20	70	2.605	1.67	2.506	8.56
20	80	2.674	2.46	2.545	14.39
20	100	1.768	4.73	1.463	34.99
20	200	1.815	36.89	1.046	543.87

Figure 2-4 : Comparison of FL and B5FT heuristics [35]

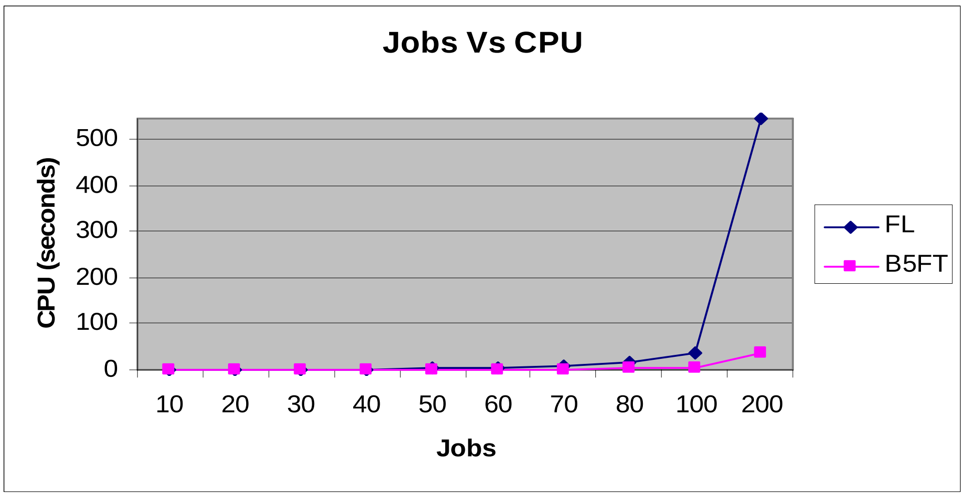


Figure 2-5 : Comparison of FL and B5FT heuristics using CPU requirements [35]

Some of the heuristics discussed above use meta-heuristic search techniques such as simulated annealing, tabu search or genetic algorithm in their solution methods. Meta-heuristics are classified as approximate algorithms since they obtain near optimal solutions whereas deterministic algorithms obtain optimal solutions. Scheduling of jobs in minicells in previous work [4, 6, 5, 8], was done by using Genetic Algorithm and the objective was minimizing the makespan. However, given the importance of flow time and the superior performance of minicells in minimizing average flow time [6, 5, and 8], minimization of average flow time is chosen as the objective for scheduling in research. This is achieved by applying FL heuristic.

3 METHODOLOGY

This chapter focuses on the methodology used for designing minicells and scheduling of jobs in minicells. Since the minicells are first designed and then jobs scheduled in them in this research, a detailed explanation of design and scheduling methods is presented. A brief description of minicell configuration is also given in this chapter.

3.1 Comparison with Previous Research

Badurdeen [6] developed an integrated approach to build a robust minicell configuration by considering the design and scheduling of minicells simultaneously. Badurdeen [6] used a genetic algorithm-based method to determine the best minicell subject to two objectives: minimizing machine requirements and /or makespan.

In the previous research, based on the population size a large number of designs were generated. Then scheduling of jobs is performed on all these designs. This is followed by makespan and machine count calculations for each of the generated designs. Based on the weights assigned to each objective, a weighted objective function value is computed for each design. Finally, the best minicell design is selected based on the weighted objective function value. For example, if 100 generations are considered for a problem with 50 chromosomes in each generation, then each generation would produce 50 designs and since there are 100 generations a total of 5000 (50*100) designs would be generated. Then jobs are scheduled on each of the designs, followed by makespan and machine capacity calculations. Hence, it can be observed that a minicell design is selected from a large solution space. But since the emphasis has been on minimizing both objectives, there could be a design generating minimum machine requirements which may not be selected in this process.

Hence an alternate approach to this process is to first design minicells and then schedule product variant demand in minicells to determine the minicell designs which meets the desired performance criteria. This approach is an attempt to identify best minicell designs without having to go through the lengthy process of developing a metaheuristic model as used in the previous research.

In the present research, minimum machine requirement minicell designs are identified in the designing stage. Then using these designs, jobs are scheduled in minicells to determine the minimum average flow time. Then the results from designing and scheduling stages are analyzed and best minicell design is selected based on the desired objective. The details of the procedure to develop minicell design are given in the following section.

3.2 Framework of Hierarchical Approach to Minicell Design and Rule Extraction

There are two objectives in this research: identifying the best minicell designs using the hierarchical method (to determine if this method can find better solutions) and to extract rules to identify the criteria to design the best minicell design, subject to given objectives. The detailed methodology followed to achieve these objectives is presented in Figure 3-1. Best minicell designs and their machine requirements are identified in design stage and average flow times are calculated in scheduling stage. These designs are then evaluated to identify similarities in their design to establish rules that can simplify the minicell design process.

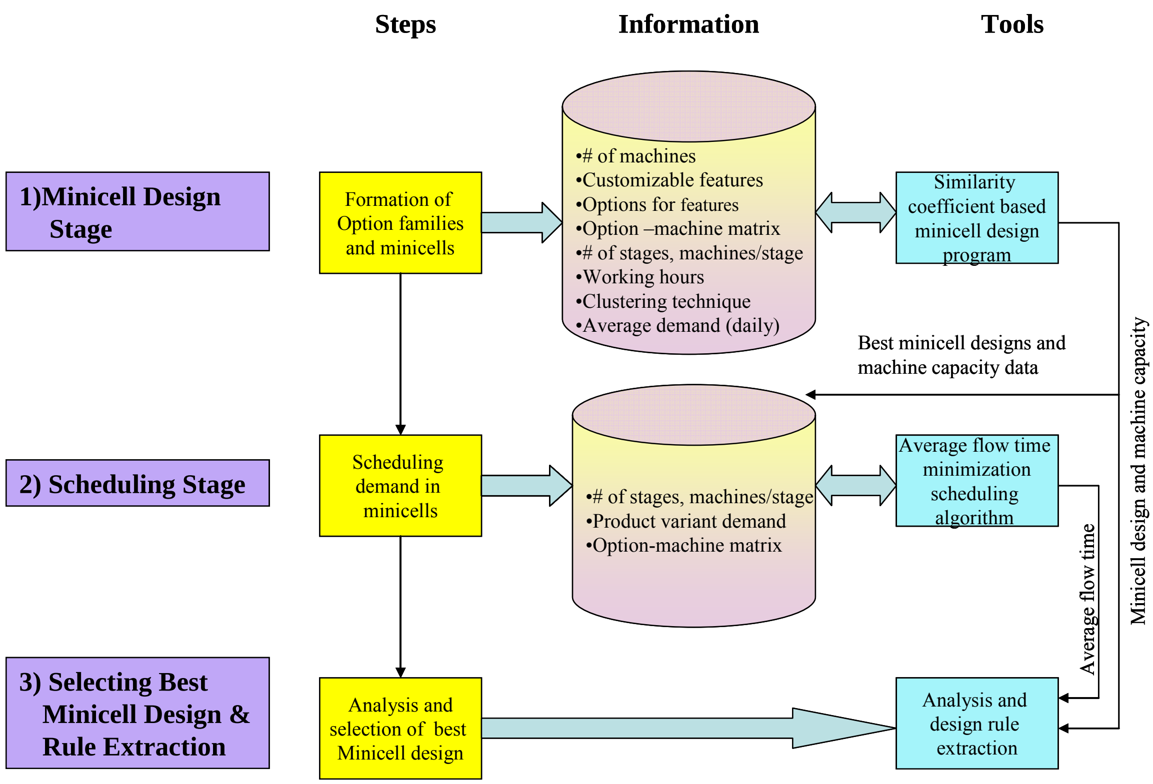


Figure 3-1 : Framework for hierarchical approach to Minicell Design and Rule Extraction

Minicell design and scheduling stages are elucidated clearly in this chapter and the third stage of selecting of best minicell designs and design rule extraction are given in next chapter. Before proceeding to minicell design, a clear understanding of minicell configuration and minicell formation method is necessary. These topics are elaborated in the following sections.

3.3 Description of Minicell Configuration

In classical cellular manufacturing environment, a manufacturing cell is dedicated to produce product family and machines required to process them are grouped together. Then the products are manufactured within the cell. Hence, a product-machine matrix which consists of all the products and machines on which the products get processed is used to form product families and machine cells. An example of product-machine matrix is given in Table 3-1.

Table 3-1 : Product-Machine matrix used in traditional manufacturing system (in terms of processing times)

Products\Machines	M1	M2	M3	M4
Product1	3	4	5	0.2
Product2	1	2	3.2	5.6
Product3	2	3.1	5	2.9
Product4	2	3	1	2.3

In mass customization environment often a platform based product structure is used as a number of options are available for each feature of product. Therefore, different product variants can be formed based on the choice of options for each feature, though there may be only one or few products. Hence developing traditional cells to contain processes for all these product variants would result in large cells. Also large cells would be difficult to manage [48] and would limit flexibility in mass customization. Hence considering these factors, forming small manufacturing cells called 'minicells' which are dedicated to an option-family was suggested by Badurdeen [6].

For example, assume that a product consists of three customizable features. Feature 1 and 2 consists of two options and feature 3 has as one option as shown in Table 3-2. The option-machine matrix provides information about the processing time requirements of all options on each of the required machines. If an option does not require processing on any machine, then the processing time will be zero in the matrix. The dashed line in Table 3-2 shows that machines M1 and M2 are assigned to stage 1 and M3 and M4 to stage 2.

Table 3-2: Option-Machine matrix used in mass customization environment (in terms of processing times)

		M1	M2	M3	M4
Feature 1	Option 1	0.5	0.2	1.3	2.6
	Option 2	1	0	2.8	3
Feature 2	Option 3	2.3	1	0	4
	Option 4	1	2	3	0
Feature 3	Option 5	8.5	0	0	5.6

Two different approaches have been used to divide the option-machine matrix into multiple stages [6]. In the 1st approach, machines are first listed in processing sequence and are then separated into multiple stages. This is done by dividing the matrix vertically [6]. For example, in Table 3-2, option-machine matrix is divided into two stages, with M1 and M2 in stage 1 and M3, M4 in stage 2. Within each stage, the options are combined based on the processing requirement in that stage and option-families are formed. Machines required to process the option family are then grouped into a minicell.

Depending on the option allotment in minicells, a product variant may need to visit more than one minicell in each stage. Each minicell will contain machine types required to process a subset of operations for each option.

In the 2nd approach to minicell design, the option-machine matrix is separated into stages based on the available features. The matrix is divided horizontally so that each stage will contain all the machines to process options for one or more features [6] i.e. Features 1 and 2 could form stage 1 and feature 3 as stage2 as seen from Table 3-2. Hence all operations for an option would be performed in one minicell in each stage i.e. a product variant would visit only one minicell in each stage.

Badurdeen and others used GA based model to design minicells following both strategies and it was found that designing minicells using the 2nd approach resulted in significant machine duplication [5]. It was also noted that minicells designed using 1st approach produced results, comparable in terms of makespan and better average flow times, even though it was not a performance measure that was considered in previous work [4]. The 2nd approach also produced better makespan and average flow time values but with the use of more machines. But the 2nd approach converged to the overall best solution much faster than the 1st approach [20]. Hence in the present research, stages in the option machine matrix are formed using the 1st approach as seen in Table 3-2.

3.4 Similarity Coefficient Technique for Minicell Formation

A brief description about the minicell configuration and operation is given in the literature section. The basis to form minicells and the importance of option-machine matrix has also been presented. This section provides information about the method used to group options and generate option families in this research.

Different strategies suggested for cell formation in the literature were discussed in chapter 2. However, all these techniques cannot be applied to minicell formation. Minicell formation needs evaluating the processing requirements of options not products. Further, traditional cell formation methods must be applied separately to each stage since minicell configuration consists of multiple stages [6]. For example, with the similarity coefficient method, similarity of the options has to be considered in terms of their processing requirements in a particular stage. Also cell formation techniques that proceed by diagonalizing the product-machine [57] matrix like King's algorithm cannot be used in minicell-based configuration because the column sequence in the option-machine matrix must be preserved to maintain sequential flow across stages.

Among the several cell formation techniques, clustering techniques are most apt for the minicell formation. The part-machine matrix used in clustering technique is similar to the part of the option-machine matrix assigned to each stage in minicell design. Also the similarity coefficient methods are more flexible in incorporating manufacturing data into the minicell formation process [97].

In statistical cluster analysis method, a measure of association or proximity that quantifies the similarity between two parts or machines is developed. This measure of association is called similarity coefficient. The similarity coefficients are calculated for each pair of parts or machines and stored in a matrix. This matrix is used as input for Single Linkage Clustering and Average linkage clustering methods to identify part families or machine cells [102].

McAuley developed a clustering algorithm to form machine cells by considering parts as rows and machines as columns. On the other hand Carrie illustrated the clustering algorithm by considering similarity between to generate a part-part similarity coefficient matrix. According to Shafer and Meredith [100], statistical cluster analysis to identify part families outperformed similar procedures to identify machine cells due to the greater flexibility the analyst has in assigning duplicate machines to alternative cells with part family identification. Hence in the present research, Carrie's method of part-part cell formation technique is used.

In statistical cluster analysis, each part or machine is initially placed in a separate cluster. These clusters are successively combined starting with the clusters that are most similar and ending with the combination of two clusters that are least similar [102].

Carrie illustrated how numerical taxonomy can be applied to cellular manufacturing using the Jaccard Similarity Coefficient (JSC). Similarity between two parts with JSC is calculated as the ratio of number of machines the two parts require in common to the number of machines either or both parts together require i.e. JSC is defined as:

$$JSC_{jk} = \frac{N_{jk}}{\left(N_{jj} + N_{kk} + N_{jk}\right)}$$

Where JSC_{jk} = Jaccard similarity coefficient between parts j and k; N_{jk} = Number of machines that components j and k have in common in their production; N_{jj} = Number of machines component j requires in its production; N_{kk} = Number of machines component k requires in its production.

To illustrate the calculation of JSC, consider the following option-machine matrix. Each value represents the processing time for each option on each machine. The corresponding binary matrix is shown in Table 3-4.

Table 3-3: Option-machine matrix

Options\Machines	M1	M2
11	0.5	0.2
12	1	0
21	2.3	1
22	1	2
31	8.5	0

Table 3-4: Binary matrix

Options\Machines	M1	M2
11	1	1
12	1	0
21	1	1
22	1	1
31	1	0

$$JSC_{11,12} = \frac{N_{11,12}}{\left(N_{11} + N_{12} + N_{11,12}\right)} = \frac{1}{2+1-1} = \frac{1}{2} = 0.5$$

JSC between all other options is calculated in the similar way and the option-option JSC matrix can be generated as shown in Table 3-5.

Table 3-5: Similarity Coefficient Matrix

Options\Options	11	12	21	22	31
11	-	0.5	1	1	0.5
12		-	0.5	0.5	1
21			-	1	0.5
22				-	0.5
31					-

Statistical cluster analysis begins with each component in its own cluster and successively combines clusters until all components are combined into a cluster i.e. starting from Table 3-5, the procedure begins by placing all the five options into their own cluster. Clustering is done based on the similarity between options and specifying a threshold value for the similarity level, which helps in determining which options should be clustered together to form option families. The threshold value is the smallest similarity value acceptable for two clusters to be combined [102] i.e. any two options could be combined only if their JSC is greater than or equal to the threshold value. The similarity coefficient of '1' indicates the highest similarity between two options. Any value less than one indicate that the options combined would be less similar.

From Table 3-5, it can be observed that the options 11 and 21, options 11 and 22, options 12 and 31, options 21 and 22 all have the highest pair-wise similarity. From the above combinations, any one of the options can be selected for combining. Option 11 and 21 is selected arbitrarily from the above list. After this step, there would be four clusters: {11, 21}, {12}, {22} and {31}.

The next step is to update the option cluster matrix based on the new cluster. This depends on the clustering procedure used. Single linkage clustering (SLC) and Average linkage clustering (ALC) are two popular clustering algorithms. Two clusters are combined based on the strongest single link between the two clusters in SLC method. Alternatively, with ALC, combining two clusters is based on the average value of all links between two clusters.

If SLC is used then the similarity between the new cluster {11, 21} is based on the maximum similarity between either option 11 or option 21 and the options in other clusters. With ALC, similarity between the new cluster {11, 21} is based on the average similarity of both option 11 and option 21 to each of the options in the other clusters.

To illustrate, based on SLC similarity between clusters {11, 21} and {12} is calculated as the max ($JSC_{11,\ 12}$, JSC $_{21,\ 22}$) i.e. max (0.5, 0.5). Similarity between clusters based on ALC would be ($JSC_{11,\ 12}$ + $JSC_{21,\ 22}$) / 2 which is equal to (0.5+0.5)/2. The option cluster matrix in this example is updated using ALC method.

Table 3-6: Similarity Coefficient Matrix after combining options 11, 21

Options\Options	11,21	12	22	31
11,21	-	0.5	1	0.5
12		-	0.5	1
22			-	0.5
31				-

Values in Table 3-6 indicate that clusters {11, 21} and {22} should be combined next. After combining these clusters the JSC matrix is updated as follows.

Table 3-7: JSC Matrix

Options\Options	11,21,22	12	31
11,21,22	-	0.5	0.5
12		-	1
31			-

The clusters {12} and {31} have the next highest similarity in sequence in the updated JSC matrix. Then, clusters {12} and {31} are combined into a single cluster.

Table 3-8: Option Family for Stage 1

Options\Options	11,21,22	12,31
11,21,22	-	0.5
12,31		-

Finally clusters {11, 21, 22} and {12, 31} are combined into a single cluster i.e. clusters {11, 21, 22} and {12, 31} are combined at a similarity level of 0.5. If a threshold value of 0.75 is specified for the above example then two option families, {11, 21, 22} and {12, 31} would be created i.e. option families would be combined for all values which are equal to or above the threshold.

3.5 Minicell Design Stage

In this stage of minicell design, option families and machine requirements of each minicell at different threshold values are obtained. As the threshold values are changed the grouping of options also varies. Hence it becomes essential to explore forming option families and corresponding minicells for a range of threshold values. Therefore, a lower bound (LB) and higher bound (UB) for threshold values is utilized in this research and alternate minicell designs are determined by varying the threshold value from the LB in steps of 0.1 until UB.

The JSC method is used to generate clusters of options and the machines needed for them are then grouped to form minicells. The difference in the present research work in comparison with the previous method developed by Badurdeen [6] is that here the allotment of machines to different stages is decided by the user. By doing so, this method provides flexibility to the user to decide which operations need to be performed in which stage. However, in the previous research, though the number of stages required and maximum number of minicells per stage were predefined, the GA has been used to determine the optimal assignment of machines to stages. In the present work, number of stages into which machines are distributed is decided by the user but the number of minicells in each stage is determined by the clustering technique used. In the previous method, number of stages was restricted to three, while in the present work there is no restriction on the number of stages. With different stage division, different minicell designs are generated and give an opportunity to identify the best minicell design. Also this flexibility of minicell design method helps to perform in scenario based planning to select the best minicell configuration. However, in comparison with the previous work, the present approach limits the exploration of solution space as the machines allotment to different stages is decided by the user. Also in previous research used a metaheuristic like GA to explore the solution space.

The details of the minicell design developed to solve the multi stage flow shop scheduling problem are explained in this section. The inputs procured from the user in designing minicells are:

a) Number of customizable features
b) Number of options available for each feature
c) Total number of machines
d) Daily demand for each product variant
e) Processing times for each option on each machine
f) Number of working hours in a day
g) Statistical clustering technique to be used (SLC or ALC)
h) Threshold values – lower threshold value, upper threshold value and setup size for threshold
i) No of stages
j) Allotment of machines in each stage

The steps followed in the data entry and calculations of minicell design are shown in Figure 3-2. The screenshots for each of the numbered steps in Figure 3-2 are shown in Appendix. The second step in the following chart uses the input data a), c), f) and g) from the above list.

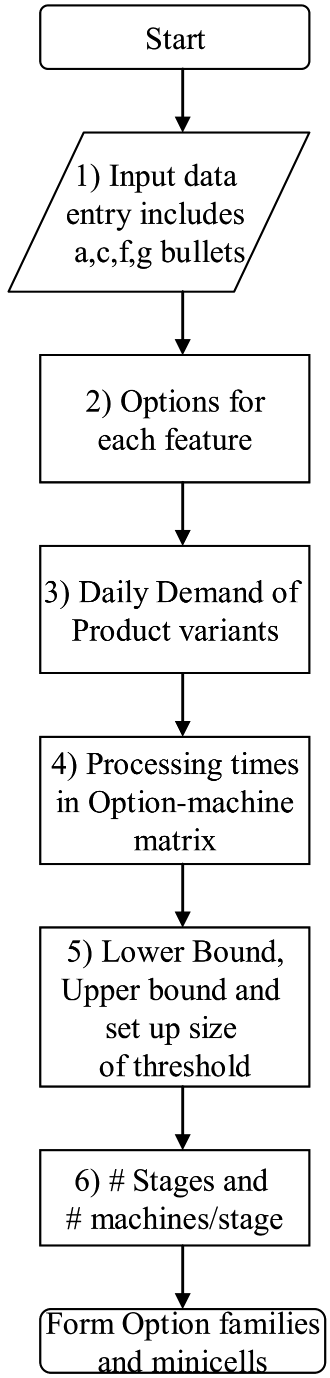


Figure 3-2: Flow chart representation of minicell design

Consider an example with three customizable features (F1, F2, and F3). The first feature consists of two options (11, 12) and second feature also contains two options (21, 22).

The third feature contains one option (31). The resulting product structure can be represented pictorially in Figure 3-3.

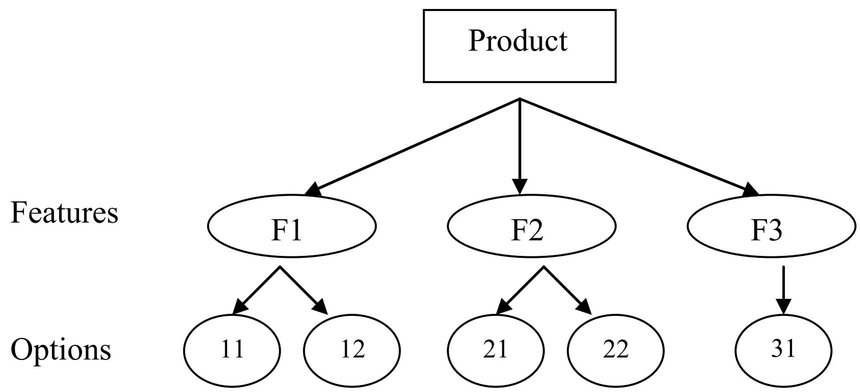


Figure 3-3: Product-Feature-Option diagram

In the above example, the total number of options of all features is equal to five (2+2+1 = 5). The number of options available would now form the rows of option-machine matrix and the total number of machines will be equal to the number of columns of the option-machine matrix. Therefore, assuming four machine types are needed, there would be four columns and five rows in the option-machine matrix. The processing requirement of each option on each of the machine is entered in minutes by the user as shown in Table 3-9.

Table 3-9: Option-machine matrix

Options/Machines	M1	M2	M3	M4
11	0.5	0.2	1.3	2.6
12	1	0	2.8	3
21	2.3	1	0	4
22	1	2	3	0
31	8.5	0	0	5.6

The option machine matrix in terms of processing times is transformed into a binary matrix. All the options which require processing on any machine would be assigned '1' and all other options would be assigned '0' i.e. '1' indicates that the particular option requires processing on that machine and number '0' indicates that the option doesn't need processing on that machine. The binary matrix for the example is shown in Table 3-10.

Table 3-10: Binary matrix for Table 3-9

Options/Machines	M1	M2	M3	M4
11	1	1	1	1
12	1	0	1	1
21	1	1	0	1
22	1	1	1	0
31	1	0	0	1

The option- machine matrix is divided into two stages by dividing the matrix vertically using 1st approach [Section 3.3]. The first two machines M1 and M2 are allotted to 1st stage and M3 and M4 are assigned to second stage. After assigning machines to different stages, each stage is considered as a separate manufacturing unit. Statistical clustering technique is applied to each stage and minicells in each stage are identified. The following example elucidates this procedure clearly. The resulting option-machine matrices for stage 1 and stage 2 are shown in Table 3-11 respectively.

Table 3-11: Option-machine matrix for (a) Stage 1 and (b) Stage 2

(a) Stage 1

Options/Machines	M1	M2
11	0.5	0.2
12	1	0
21	2.3	1
22	1	2
31	8.5	0

(b) Stage 2

Options/Machines	M3	M4
11	1.3	2.6
12	2.8	3
21	0	4
22	3	0
31	0	5.6

Binary matrices are formed for the two stages using the option machine matrices in Table 3-11. Jaccard similarity coefficients are then calculated for each option with every other option, following the method described in Table 3-4 through Table 3-8 for stage 1. The procedure is repeated for stage 2. Option families {11, 21, 22} and {12, 31} are formed in stage 1(seen from Table 3-8) and option families {11, 12}, (21, 31} and {22} are generated for stage 2 (shown in Table 3-13).

Table 3-12: Option-Option Jaccard Similarity Coefficient matrix for stage 2

Options/ Options	11	12	21	22	31
11	-	1	0.5	0.5	0.5
12		-	0.5	0.5	0.5
21			-	0	1
22				-	0
31					-

Table 3-13: Option family matrix for stage 2

Options/ Options	11,12	21,31	22
11,12	-	0.5	0.5
21,31		-	0
22			-

By specifying a threshold value for the similarity level, it is possible to determine which options can be clustered to form families. From Table 3-13 it is observed that three clusters are formed at a similarity level of 0.6 and three option families are created. But at a threshold of 0.5 the option families in Table 3-13 would reduce to one because the clusters with JSC values equal to or above threshold would be combined into one family.

After applying clustering technique, the number of option families and machine usage in each minicell is determined. The machines needed for each option family in each stage is summarized in Table 3-14.

Table 3-14: Option families, machine usage in minicells for (a) Stage 1, (b) Stage 2

(a) Stage 1

Option Family	Machines in minicells
11,21,22	M1, M2
12,31	M1

(b) Stage 2

Option Family	Machines in minicells
11,12	M3, M4
21,31	M4
22	M3

A product variant is formed by considering one option from each feature. In the present example since there are three features, the product variant would contain one option from each of the three features. Hence there would be three options in one product variant. The total number of product variants formed with the available options is obtained from the product of the number of options available for each feature i.e. 2*2*1 = 4. The configuration of each product variant is shown below.

Table 3-15: Product Variant Configuration

Product Variant ID	Product Variant
1	11,21,31
2	11,22,31
3	12,21,31
4	12,22,31

After identifying the total number of product variants, the average demand for each product variant is used to determine the average demand for each option considering the options needed for each variant. The product variant demand and option demand are shown in Table 3-16(a) Product variant Demand and (b) Option Demand respectively.

Table 3-16: (a) Product Variant Demand and (b) Option Demand

(a) Product Variant Demand

Product Variant	Demand
11,21,31	8
11,22,31	9
12,21,31	4
12,22,31	3

(b) Option Demand

Option	Demand
11	17
12	7
21	12
22	12
31	24

The number of machines of each type in each minicell in each stage (capacity) is calculated using the demand of each option and the processing time for that option in corresponding machine in a particular minicell. In the above example, as seen from the Table 3-8, at a threshold value of 0.6 two minicells with option families {11, 21, 22} and {12, 31} are formed in stage 1. Three minicells with option families {11, 12}, {21, 31} and {22} are formed in stage 2 .The machine capacity calculations are enumerated below. In minicell 1 of stage1, there are three options 11, 21, 22. Stage1 has two machines M1 and M2. The machine usage of option 11 on M1 is obtained by multiplying the demand for option 11 with the processing requirement of option 11 on M1 i.e.17*0.5 = 8.5. The sum of all the options on an individual machine in a minicell determines the usage of that machine. Machine capacity requirement for minicell 1 in stage 1, calculated as described above, is given in Table 3-17.

Table 3-17: Machine capacity requirement (minutes) for minicell 1 in stage 1

Options\Machines	M1	M2
11	8.5	3.4
21	27.6	12
22	12	24
Total	48.1	39.4

The number of working hours per day is also entered by the user. Thus, for an 8-hour shift the number of available minutes is 480 and one unit of M1 and M2 are required in minicell 1 in stage 1. The requirements of all machines in all minicells are calculated accordingly, as shown in Table 3-18.

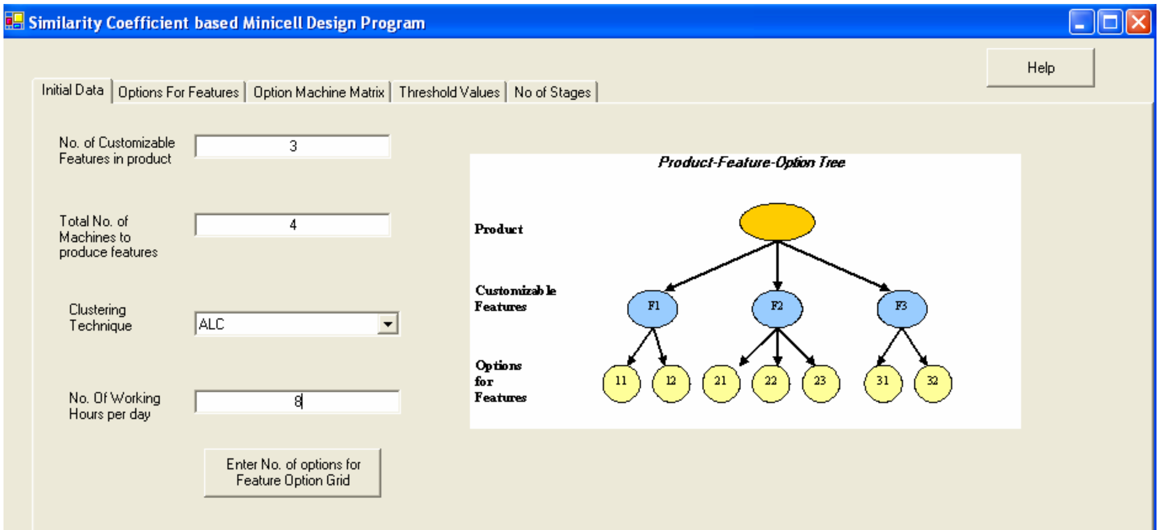
Table 3-18: Final minicell design output matrix

			Machine Capacity			
Stage	Minicell	Options	M1	M2	M3	M4
1	1	11,21,22	1	1	-	-
	2	12,31	1	0	-	-
2	1	11,22	-	-	1	1
	2	21,31	-	-	0	1
	3	22	-	-	1	0

From Table 3-18, the machine requirement of each minicell at each stage can be obtained. Thus, at a threshold value of 0.6, the total machine requirements are given by the usage of M1 (2), M2 (1), M3 (2) and M4 (2) units which is equal to seven. The machine requirements at any threshold value are calculated accordingly.

As observed from Figure 3-1, the minicell design stage identifies the best minicell designs and the required number of machines for those designs. As the objective of the research is to identify criteria that can be used to determine the best minicell designs, considering machine capacity and/or flow time or both, several designs that perform well with respect to machine requirements are chosen to be carried forward to the next stage. The second step involves scheduling the product demand in the selected designs to identify the minimum average flow time processing sequence.

The software program to perform the above mentioned calculations for minicell designs was developed by using Microsoft Visual Basic.NET programming language, 2003 version. Visual Basic.NET provides ease of creating an interface for the user and has the ability to present the work as an executable file. A few screenshots of the interface created for the Minicell Design are shown in Figure 3-4. An example using Visual Basic.Net is given in the Appendix.



(a): Screenshot of interface in minicell design stage; Input data Entry

	Output						
	Stage	Threshold	Minicell	Machine1	Machine2	Machine3	Machine4
	1	0.6	12,31	1	0	(null)	(null)
	2	0.6	11,12	(null)	(null)	1	1
	2	0.6	21,31	(null)	(null)	0	1
	2	0.6	22	(null)	(null)	1	0

(b): Screenshot of interface in minicell design stage; Final Output

Figure 3-4 : Screenshots of Interface in Minicell Design Stage

3.6 Scheduling Stage

In this stage, the average flow time to process the given product variant demand is taken as the performance measure for evaluating minicell designs. Several of the best minicell designs are selected with respect to machine capacity from the minicell designs generated at different threshold values from the minicell designing stage. The jobs in these selected designs are scheduled to identify average flow times. Therefore it is essential to determine an effective scheduling strategy which identifies the minimum average flow time processing sequence in the given minicell configuration.

Since minicells are similar to flow shops, different flow shop scheduling heuristics which generate minimum average flow time were reviewed. Among the several heuristics, the heuristic by Framinan and Leisten [33] produces good quality results to minimize average flow time in reasonable computational time. Hence the Framinan and Leisten (FL) heuristic is selected for scheduling jobs in minicells in the multi-stage flow shop environment. Each stage in a minicell configuration could have multiple minicells representing a separate flow shop. Therefore, the multi-stage configuration results in a nested multi-stage flow shop configuration. Hence the application of FL heuristic has to be modified for scheduling minicells.

FL heuristic [33] is specifically developed for a single stage permutation flow shop. Hence minor modifications are needed in the heuristic before applying it to scheduling in a minicell configuration, because minicells are also considered as non-permutation flow shops that can process jobs as they become available. The heuristic is individually applied to each minicell considering it as a separate flow shop scheduling problem to get the best possible sequence. In the most basic form FL heuristic has only been applied in literature to situations where there is no machine duplication. However, there could be more than one machine of each type in any minicell, i.e. machine duplication is allowed in order to provide the required capacity to process daily demand. Therefore, it is essential to modify the application of FL heuristic in the present research, to be applicable to multi-stage flow shop problem allowing machine duplication.

After identifying minicells from the design stage, the product demand is scheduled for processing in respective minicells. Since a multi-stage flow shop manufacturing system is considered, the modified FL heuristic is applied to all the minicells in all stages. It is assumed that all jobs are available at a ready time of zero for processing in the first minicell in the initial stage. For all subsequent minicells, the start times of jobs will depend on the completion times in the previous minicells.

3.6.1 Application of FL Heuristic for Scheduling in Minicells

A detailed description of the scheduling method along with step-wise procedure is explained using the same example discussed earlier.

<u>Procedure for First minicell in Stage 1</u>

Due to differences in ready times for jobs entering the first minicell in stage 1 and the remainder of minicells, different approaches are used when applying the FL heuristic in the two situations. The approach followed for the fist minicell in stage 1 is discussed here.

Step1:

Identify the product variants which needs to be fabricated in the particular minicell based on the options assigned to the minicell. Based on the information provided in Table 3-15 and Table 3-18, the routing of product variants to different minicells in each stage can be established as shown in table below.

Table 3-19 : Product Variant Routing

Stage	Minicell	Options	Product Variant
1	1	11,21,22	1, 2, 3, 4
	2	12,31	1, 2, 3, 4
2	1	11,12	1, 2, 3, 4
	2	21,31	1, 3
	3	22	2, 4

Step 2:

Calculate the total processing time of each product variant in each minicell on each machine. Obtain the sum of processing times of options present in the minicell. Then get the product of this sum and the product variant demand.

In the present example, two options (11, 21) of the first product variant (11, 21, 31 = Job1) are processed in the first minicell, the sum of processing times for options 11 and

21 is computed. As mentioned previously, the first two machines are allotted to the first stage. Therefore, the processing times needed for Job1 (demand = 8 units) on M1 and M2 in minicell 11 will be 8*(0.5+2.3) and 8*(0.2+1), respectively. The times needed to process all jobs in minicell 11, computed as described above are shown in Table 3-20.

Table 3-20 : Processing Times for jobs in Minicell 11 in Stage 1

	M1	M2
# Machines	2	1
11, 21, 31 = Job1	8*(0.5+2.3)=22	8*(0.2+1)=10
11, 22, 31 = Job2	9*(0.5+1)= 14	9*(0.2+2)=20
12, 21, 31 = Job3	4*(2.3)=9	4*(1)=4
12, 22, 31 = Job4	3*(1)=3	3*(2)=6

The values in Table 3-20 are rounded off to the nearest minute in this example. The processing time of each job on the available machines and the number of copies of each machine is given in Table 3-20.

Step 3:

Determine the initial job sequence by calculating the sum of processing times of each job on the all the available machines in each stage and arranging the jobs in the ascending order of their total processing times as illustrated in Table 3-21. The job sequence for minicell 11 is Job4-Job3-Job1-Job4.

Table 3-21 : Job order of Minicell 11

	M1	M2		
No of Machines	2	1	Sum	Order
Job1	22	10	32	3
Job2	14	20	34	4
Job3	9	4	13	2
Job4	3	6	9	1

Step 4:

A partial sequence S is developed by considering the first two jobs in the job sequence. If the number of jobs is equal to two then the jobs are interchanged and the sequence which produces lowest flow time is selected and updated as S. The first two jobs are then removed from the job sequence list.

The first two best jobs, Job4 and Job3 with the shortest processing times are selected and considered as a partial schedule. They are then scheduled on the machines considering all the units of the particular type of machines available. The assignment of jobs to machines and cumulative times are shown in Table 3-22.

Table 3-22: Job-Assignment and Cumulative Flow time in Minicell 11 with Job4-Job3

	M1		M2	Cumulative Flow Time
	Unit 1	Unit 2	Unit 1	
Job4	3		3+6 =9	9
Job3		9	9+4 =13	13+9 =22

Therefore, the total flow time is equal to sum of completion times of Job4 and Job3, which is equal to the sum of 9 and 13, equal to 22. The same procedure is repeated for Jobs3 and 4 by interchanging their positions, i.e. Job3 would now be in the 1^{st} position and Job4 in the 2^{nd} position. The calculations are shown in Table 3-23.

Table 3-23 : Job Assignment and Cumulative Flow time in Minicell 11 with Job3-Job4

	M1		M2	Cumulative Flow Time
	Unit 1	Unit 2	Unit 1	
Job3	9		9+4 =13	13
Job4		3	13+6=19	13+19 =32

The total flow time obtained from the Job3-Job4 sequence is 32 (13+19). Since the lowest flow time is to be chosen, the Job4-Job3 sequence is selected. Let this partial job sequence be designated as S = {Job4-Job3}.

Step 5:

The next job in the job sequence is then appended to the partial sequence S and depending on the number of jobs in S, the new job is inserted in all possible positions and the partial schedule with lowest flow time is retained as best solution. This job is then removed from the original job sequence.

The next step is to add the third job in job list, Job1 to the partial sequence S in all the possible positions. As the total number of jobs is three, Job1 is added in all three positions, while retaining the sequence of other jobs in S i.e. sequence Job4-Job3 must be retained since it has given lower flow time for two jobs. Then, flow time must be calculated for the sequences {Job4-Job3-Job1}, {Job4-Job1-Job3} and {Job1-Job4-Job3}. The calculations for these sequences are shown below.

Table 3-24 : Job Assignment and Cumulative Time in Minicell 11 with 3 jobs

(a) Job1-Job4-Job3

	M1		M2	Cumulative Flow Time
	Unit 1	Unit 2	Unit 1	
Job1	22		22+10 =32	32
Job4		3	32+6=38	32+38 =70
Job3		3+9 =12	38+4=42	70+42=112

(b) Job4-Job1-Job3

	M1		M2	Cumulative Flow Time
	Unit 1	Unit 2	Unit 1	
Job4	3		3+6 =9	9
Job1		22	22+10=32	9+32 =41
Job3	9+3=12		32+4=36	41+36=77

(c) Job4-Job3-Job1

	M1		M2	Cumulative Flow Time
	Unit 1	Unit 2	Unit 1	
Job4	3		3+6 =9	9
Job3		9	9+4=13	9+13 =22
Job1	22+3= 25		25+10=35	22+35=57

The lowest flow time generated sequence, Job4-Job3-Job1 is selected and partial sequence S is updated as S = {Job4-Job3-Job1}.

Step 6:

If the number of jobs is greater than 2, then a general pair wise interchange is applied to the partial schedule S, i.e. all the possible combinations of the jobs are checked for flow time and the best partial solution which generates minimum flow time is retained. The pair wise interchange sequences are checked for the lower flow time values than the one generated by partial sequence S. If the new sequences do not produce flow time lower than S then partial sequence S is retained.

The jobs in the sequence Job4-Job3-Job1 are interchanged pair wise i.e. two jobs are interchanged at a time while keeping the other jobs in their positions. The pair-wise interchange on the sequence Job4-Job3-Job1 generates Job3-Job4-Job1, Job1-Job3-Job4, and Job4-Job1-Job3 sequences. The flow times are evaluated to be 67, 110 and 77 minutes respectively. Since the sequence Job4-Job3-Job1 generates lower flow time value, it is retained.

Step 7:

If the total number of jobs in the job order list becomes null then Stop else go to Step 5.

In this example, since there is one more job (Job2) left, Step 5 is repeated. The last job in the job order list (Job2) is then appended to Job4-Job3-Job1 in all possible positions, i.e. Job2-Job4-Job3-Job1, Job4-Job2-Job3-Job1, Job4-Job3-Job2-Job1 and Job4-Job3-Job1-Job2. The flow times of 172, 129,106 and 112 are obtained, respectively. The sequence with lowest flow time is selected and pair-wise interchange is performed. It is noted that the sequence Job4-Job3-Job2-Job1 generated minimum flow time of 106 minutes and is hence retained. The calculations of Job4-Job3-Job2-Job1 are shown in Table 3-25.

Table 3-25 : Completion times of jobs in Minicell 11 with Job4-Job3-Job2-Job1 sequence

	M1		M2	Cumulative
	Unit 1	Unit 2	Unit 1	Flow Time
Job4	3		3+6 =9	9
Job3		9	9+4=13	9+13 =22
Job2	3+14=17		17+20=37	22+37=59
Job1		9+22=31	37+10=47	59+47= 106

Procedure for Subsequent Minicells

The completion times of jobs from minicell 11 are passed on to the next minicell, Minicell 21 and become the start times of the jobs in minicell 21. Minicell 21 consists of options 12 and 31 as observed from Table 3-18 and requires only M1 of which there are two units available.

Table 3-26 : Job machine matrix for Minicell 21

	M0	M1
Job1	47	68
Job2	37	77
Job3	13	38
Job4	9	29

The processing times of jobs and initial job order are obtained by following *Steps 1, 2 and 3* as described previously. The completions of all jobs from the previous minicell are designated as processing times on a dummy machine M0 (as shown in Table 3-26). The partial sequence S with two jobs is obtained by considering the dummy machine M0 and using *Step 4* for calculations. Job4 will get processed on M1 only after completing processing on M0. Therefore, completion time of Job4 on M1 is the sum of 9 and 29. Similarly Job3 will get processed on M1 after completing its process on M0. Since a second copy of M1 is available, Job3 need not wait until Job4 is completed. The completion time of Job3 is calculated as 51(13+38). This can be clearly seen in Table 3-27.

Table 3-27: Job Assignment and Cumulative Flow time in minicell 21 with Job4-Job3

	M0	M1		Cumulative Flow Time
	Unit 1	Unit 1	Unit 2	
Job4	9	9+29=38		38
Job3	13		13+38=51	38+51=89

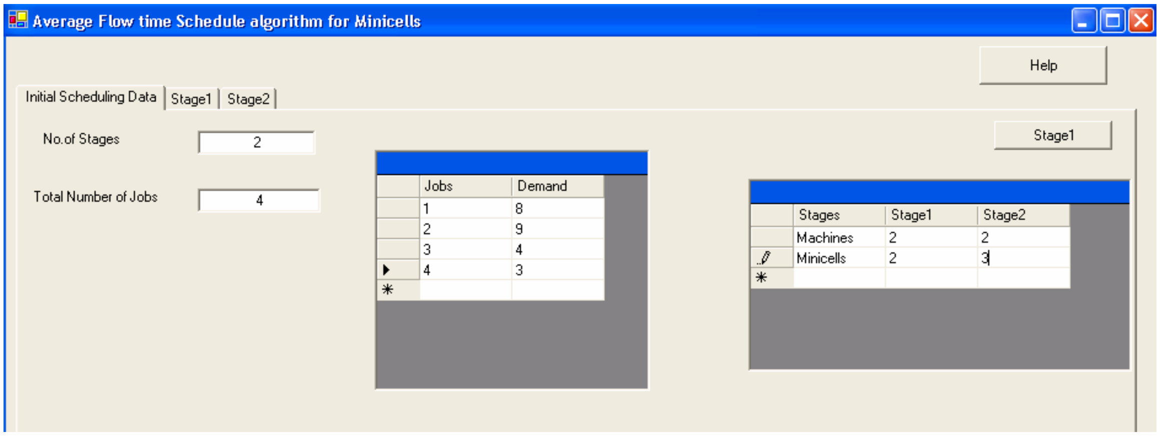
Steps 5 through *7* are repeated and the minimum flow time is identified for this minicell. The same procedure of passing the completion times from one minicell to another is adapted to the three minicells in second stage and minimum flow time sequence is obtained.

After applying the modified FL heuristic to the last minicell in the last stage, average flow time is calculated by taking the average of the completion times of all the jobs in the last stage. This procedure is applied to all minicells designs and the average flow time is evaluated. The best minicell designs are then identified using the minimum average flow time criteria.

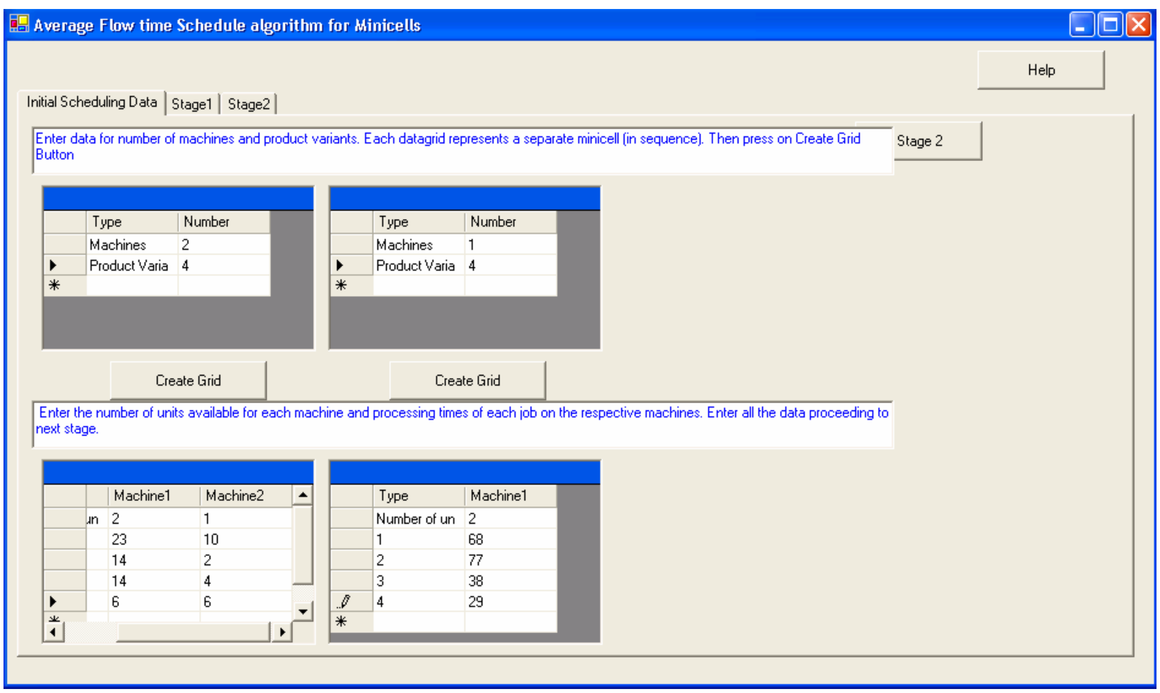
The best minicell design from the designing stage with the machine count and the best minicell design from the scheduling strategy with the minimum average flow time are considered and based on the desired objective the best minicell design is selected. I.e. if the objective is primarily focused on the minimum machine capacity then the minicell design which requires minimum number of machines and producing respective average flow time is selected. On the other hand, if the objective is to determine minimum average flow time, then minicell design with minimum average flow time is given importance. Hence the best minicell design is selected based on the assigned objective.

In order to study the impact on average flow time for minimum and maximum number of machines, the number of machines required at the low and high average flow times is studied by conducting a number of experiments by varying several parameters. This is discussed in detail in the following chapter.

The software program for the scheduling computations, too, was written using VisualBasic.Net. The screenshots of scheduling stage are shown in Figure 3-5.



(a): Screenshot of interface in Scheduling Stage; Input data Entry



(b): Screenshot of interface in Scheduling Stage; Stage 1 entry data

Figure 3-5 : Screenshots of Interface in Scheduling Stage

Selection of best minicell designs and extraction of design rules for the minicell designs is done by analyzing the results from designing and scheduling stages and explained in next chapter.

4 RESULTS AND DISCUSSION

In order to analyze minicell designs developed following methods discussed in the previous chapter, experimentations were conducted with problems of varying size. The test problems were studied to evaluate designing and scheduling to identify the best minicell designs using the performance measures of average flow time and/or machine capacity or both. The results procured from hierarchical method are then compared with the results obtained from the previous simultaneous method to evaluate the effectiveness of proposed method.

4.1 Experimentation Procedure

As outlined in Figure 4-1 the experimentation was conducted in three prime stages: Designing, Scheduling and Selecting best minicell designs to form rules for minicell design. The flowchart for experimentation with the hierarchical method is shown in Figure 4-1.

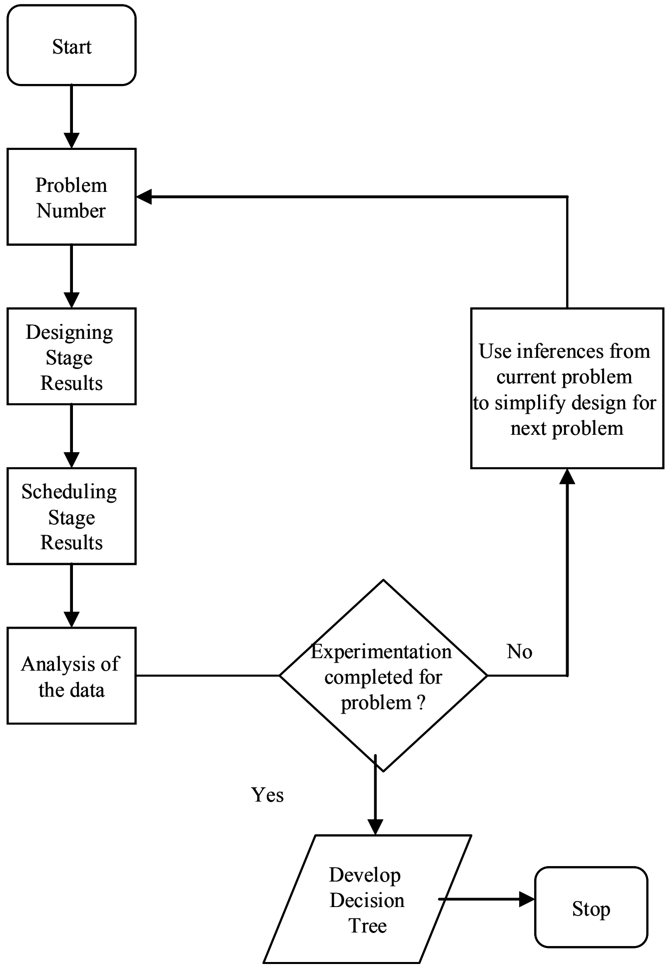


Figure 4-1 : Flow chart for Experimentation with Hierarchical Approach

Starting with a particular problem different minicell designs are identified in Minicell design stage. Several of the best minicell designs are chosen based on machine requirements and then the product demand for various product variants is scheduled in the respective minicells in the Scheduling stage. Minicell designs are evaluated for their effectiveness in meeting the objective(s), minimizing either machine capacity or average flow time or both, through detailed analysis of the results obtained from the first two stages. If the experimentation is to be continued then the inferences from previous problems are applied to the next set of problems thereby simplifying the process of best minicell design identification. After the experimentation is complete, a decision tree is developed using the data mining results for all the tested problems and to establish rules for minicell design.

Different problems which were used for the experimentation to evaluate best minicell design methodology for rule extraction of minicell designs are given in Table 4-1. The problem size was varied by using examples with different numbers of product variants and machine requirements. Data for all problems was generated randomly by considering different demands for the product variants and processing times of options on machines. For each of the tested problems, processing requirements of options was varied in order to test the results in different scenarios.

Table 4-1 : Problems used for Experimentation

Problem No.	No.of Product Variants(P)	No.of Machines (M)	Problem Size (P*M)
1	27	7	189
2	12	10	120
3	18	8	144
4	8	5	40

4.2 Analysis of Tested Problems

Detailed discussion of the experimentation conducted for each problem and the analysis of results is given below.

4.2.1 Problem No. 1 = 27 Product Variants

The example in the first problem has 27 product variants with three features (F) and each feature consists of three options. The product structure for 27 product variants problem is given in Figure 4-2. Seven machines are used to produce these variants.

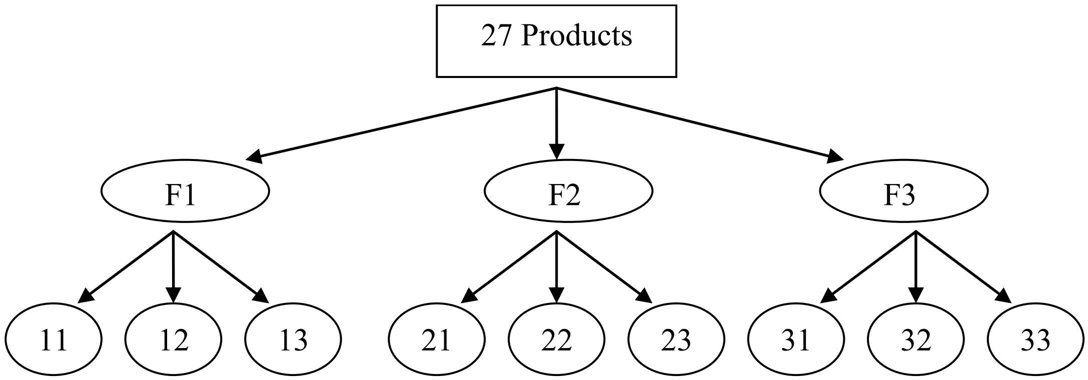


Figure 4-2 : 27 Products Structure

For initial experimentation, the machines were divided into 3 stages. The machine distributions into different stages (cut-offs) were identified using the genetic algorithm model developed by Badurdeen [6] for the simultaneous method. These machine requirements are anticipated to be determined based on the rules extracted through the analysis of experimental results subsequently. The inputs for the minicell design stage are discussed in section 3.5.The option-machine matrix (processing time in minutes) used for the 27 product variants problem is given below.

Table 4-2 : Option-machine matrix of 27 product variants problem

Options\ Machines	M1	M2	M3	M4	M5	M6	M7
11	1.02	0	0.57	0.84	0	1.9	0
12	0	0.2	0	0.75	0	0	1.2
13	0	0	0.19	0.02	0	0	0
21	0.18	0	0	0	1.18	0	0.84
22	0	0	0	0.09	0	0.09	0
23	1.65	0.06	0.86	1.27	0	0.02	0
31	1.73	0.05	1.22	1	1.58	0	1.06
32	1.65	0	0	0.33	0	0.43	0.14
33	0	0.67	0.89	0	1.08	0	0

Different minicell designs were found by varying the threshold values from 0.3 to 0.7 by incrementing the threshold value by 0.1 i.e. increasing similarity between options from a small level to a large level. As expected, the minicell designs formed changed as the threshold values were varied.

With the increase in threshold value, machine requirements were found to increase. At low threshold values, the similarity required between the options for clustering is less, hence more options are combined into one option family and minicell. Therefore, there would be less minicells and hence less machines. As the threshold value increases, similarity needed between options to be clustered together also increases, hence only options that are most similar get combined leaving out more options to stand out on their own. Therefore, this results in more minicells and consequently more machine usage. Generally, at high threshold values, minicells with single options are formed as the similarity required between options increase.

Two methods were used to cluster the options into families. From the results, it is noted that SLC generates less minicells and uses few machines in comparison with ALC. This is because in SLC, two option groups may join together merely because two of their members are similar while the remaining members may remain far apart in terms of similarity [65]. This may result in the formation of minicells containing options which do not have enough commonality with other options to justify their assignment to the minicell i.e. SLC generates minicells in which a large number of options are far apart in terms of similarity. Therefore, similarity of options within minicells generated using SLC is low. Hence more options are combined in SLC than in ALC which results in the formation of less minicells and consequently fewer machines are needed with SLC than with ALC.

On the other hand, with ALC the similarity coefficients between two minicells are a measure of similarity between all their options rather than the two most similar options as in SLC. In ALC the similarity coefficient is calculated considering the average of all the values unlike considering the maximum value in SLC. Since the average similarity coefficient between all the members of the two groups is considered in ALC, chaining problem is reduced and minicells are also better separated than those formed by SLC [65]. Since ALC generates minicells with higher overall similarity among its options, more minicells are formed resulting in more machine requirements.

When more machines are present in a minicell, the options can be processed in a shorter time, reducing the waiting time which in turn decreases average flow times. Between ALC and SLC, ALC method requires more machines and hence ALC designs at high threshold values appear to generate minimum average flow times.

In order to obtain minimum machine capacity, several combinations of machine distributions in 3 stages were tested. The results are presented in Table 4-3.

Table 4-3: Problem # 1 in 3 stages in Hierarchical and Simultaneous methods (MS=0 and MC=1)

# 3 Stages	MS=0, MC=1												
		Machines			Minicells			Average Flowtime			Makespan		
Stage Division	Threshold	Hierarchical		Simultaneous	Hierarchical		Simultaneous	Hierarchical		Simultaneous	Hierarchical		Simultaneous
		ALC	SLC		ALC	SLC		ALC	SLC		ALC	SLC	
5-1-1	0.3	12	7	8	5	3	4	128	131	181	418	424	348
	0.4	14	7		6	3		127	131		385	424	
	0.5	16	9		7	4		103	125		351	398	
	0.6	19	12		9	6		105	132		338	418	
	0.7	22	19		10	9		86	105		297	338	
1-1-5	0.3	10	7	8	4	3	5	134	134	138	391	384	393
	0.4	10	7		4	3		134	134		391	384	
	0.5	14	7		6	3		132	134		391	384	
	0.6	18	14		8	6		132	132		391	391	
	0.7	22	22		10	10		129	129		391	391	
3-2-2	0.3	11	7	10	7	3	5	103	134	205	344	392	342
	0.4	11	7		7	3		103	134		344	392	
	0.5	12	7		8	3		88	134		291	392	
	0.6	16	14		11	10		105	105		351	351	
	0.7	18	18		12	12		98	98		306	306	
2-1-4	0.3	10	7	9	5	3	6	133	135	271	391	392	372
	0.4	11	7		6	3		132	135		391	392	
	0.5	12	7		7	3		132	135		391	392	
	0.6	15	15		9	7		100	106		303	303	
	0.7	19	19		11	11		98	98		303	303	
1-5-1	0.3	11	7	7	5	3	5	132	135	137	391	391	393
	0.4	13	8		6	4		131	133		391	391	
	0.5	13	8		7	4		132	133		404	391	
	0.6	17	10		8	5		131	123		391	383	
	0.7	17	17		8	8		131	131		391	391	
1-3-3	0.3	11	7	9	6	3	5	131	133	164	391	392	339
	0.4	11	7		6	3		131	133		391	392	
	0.5	12	7		7	3		131	133		391	392	
	0.6	16	12		10	8		126	132		384	391	
	0.7	18	18		11	11		130	130		391	391	
1-2-4	0.3	10	7	10	5	3	6	134	138	138	391	384	304
	0.4	11	7		6	3		132	138		391	384	
	0.5	12	7		7	3		131	138		391	384	
	0.6	15	11		9	7		131	134		391	392	
	0.7	19	19		11	11		130	130		391	391	
3-1-3	0.3	11	7	9	7	3	5	107	135	207	351	392	341
	0.4	11	7		7	3		107	135		351	392	
	0.5	12	7		8	3		106	135		351	392	
	0.6	16	14		11	10		105	106		351	351	
	0.7	18	18		12	12		96	96		304	304	
3-3-1				9						205			345

The machine distribution combinations were procured from the GA program developed by Badurdeen [6]. From the data, it is observed that machines are distributed into three stages in three different formats:

1) More machines in the beginning stage or
2) More machines in the last stage or
3) Even distribution of machines in all three stages

Minicell designs are identified by the machine assignment to different stages. For example, the design 5-1-1 means, the total number of machines are distributed into 3 stages with 5 machines in first stage and one machine each in second and third stages. This design has more machines in the initial stage. On the contrary, 1-1-5 design has more machines in the end and 3-2-2 design indicates even distribution of 7 machines in 3 stages. Two objectives were used to evaluate the results of the minicell designs-machine requirements and average flow time. Experimental results were analyzed and a Pareto curve drawn to evaluate performance of designs as shown in Figure 4-3. The Pareto curve represents the best minicell design, given the relative importance of each objective. Thus the designs which lie on the bottom right side of the graph need more number of machines and generate minimum average flow times. Minimum machine capacities with high average flow times are given by the designs on the top left side portion of the graph. The results from both the hierarchical and simultaneous (previous GA) method are shown in Figure 4-3. In the figure, the text next to each data point denote the machine assignment to stages, threshold values used and clustering method (A or S) for hierarchical designs or 'Sim' indicate simultaneous method.

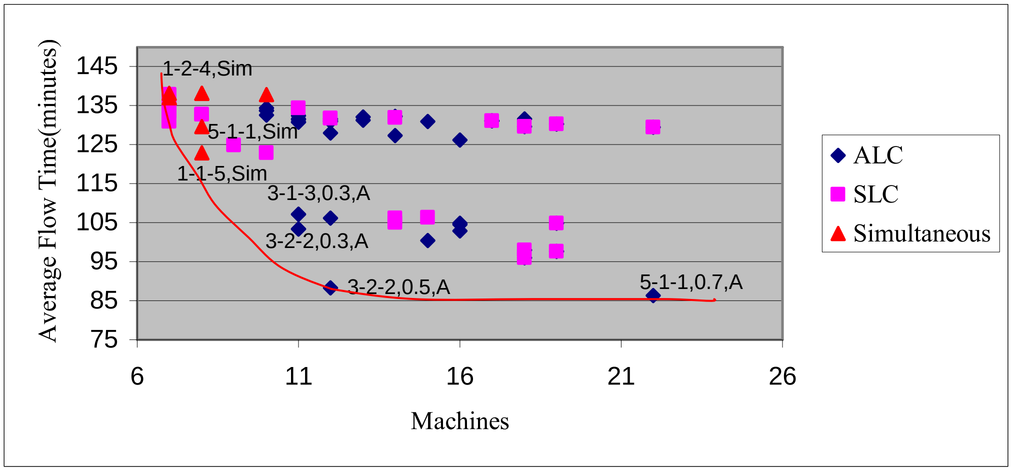


Figure 4-3: Variation of Machines Vs Average Flow Time - 3 stages

The data in Table 4-3 is studied, to identify best minicell designs. The designs generating low flow times were first identified with the corresponding number of machines. It is observed that low flow times were generated at 0.7 threshold value by both ALC and SLC methods. The next best minicell designs generating low flow flows were chosen. The pattern of machine distribution in all these minicell designs was observed and hence noted that having more machines in the initial stages or even distribution of machines yields good minicell designs.

This problem is also tested using GA from previously developed simultaneous method by assigning a weight of zero to makespan and one to machines count, since these two were the performance measures considered in previous research. The average flow times and number of machines is plotted for ALC, SLC and the simultaneous methods in Figure 4-3.

As, it can be observed the 5-1-1 design needs more machines and hence generates low average flow time while the 1-2-4 design needs less machines but produces high average flow time. From Figure 4-3, it is seen that the designs which lie away from Pareto front are 3-2-2 and 3-1-3 generated at low to medium threshold values. These designs need medium number of machines and produce mediocre average flow times. From the results it is also observed that having more machines in the initial stages would result in lower average flow times. It is also seen that most designs lying in the Pareto front region are produced by the hierarchical method and the designs generated by the simultaneous method lie in the left upper side of the graph, away from the Pareto front.

Comparing the average flow times produced by the hierarchical and simultaneous methods, it is seen that lower average flow times are generated with the hierarchical method. The designs from both clustering methods, as seen from the graph stand testimonial to this.

In order to verify and validate these observations, obtained using the 27 product variants problem, further experiments were conducted on a problem with 12 product variants that require 10 machines. The details of this problem are given below.

4.2.2 Problem No. 2 = 12 Product Variants

To test the observations from 27 product variant problem, the second problem was again tested with 3 stages. The product has 3 features and requires 10 machines which are divided into 3 stages. The product structure and option-machine matrix (in minutes) for this problem is given in Figure 4-4.

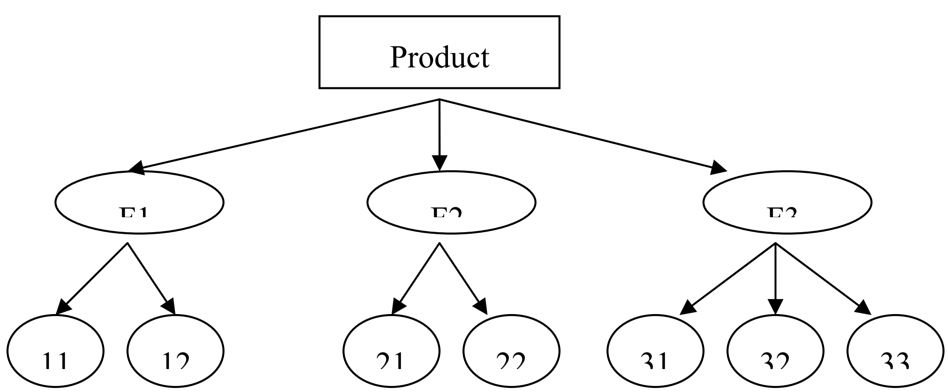


Figure 4-4 : 12 Products structure

Table 4-4: Option-Machine matrix of 12 product Variants problem

Options\ Machines	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
11	1.02	0	0.57	0.84	0	1.9	0	1	0	1
12	0	0.2	0	0.75	0	0	1.2	0	0	2
21	0	0	0.19	0.02	0	0	0	0	2	3
22	0.8	0	0.3	0	0	0	0.2	2.3	0	0
31	2.08	0	0.3	0	0.5	0	0	0	3	2.5
32	0.3	0	0.5	0.2	0	0	0	0	2	6
33	0	0	0.19	0.02	0	0	0	5	1	1

In order to verify the inferences from 27 product variant problem, this problem was also tested for threshold values ranging from 0.3 to 0.7 increasing in steps of 0.1. The machine distribution between the 3 stages (cut-offs) was obtained from the GA program. The results obtained through the experimentation with 3 stage designs are summarized in Table 4-5 below.

Table 4-5 : Problem # 2 in 3 stages in Hierarchical and Simultaneous methods (MS=0 and MC=1)

# 3 Stages	MS=0, MC=1												
		Machines			Minicells			Average Flowtime			Makespan		
		Hierarchical			Hierarchical			Hierarchical			Hierarchical		
Stage Division	Threshold	ALC	SLC	Simultaneous	ALC	SLC	Simultaneous	ALC	SLC	Simultaneous	ALC	SLC	Simultaneous
3-3-4	0.3	12	11		5	5		232	256		538	547	
	0.4	13	13		7	8		232	232		538	538	
	0.5	13	13		7	8		232	232		538	538	
	0.6	17	15		10	9		213	228		462	525	
	0.7	19	19		11	11		197	197		424	424	
5-3-2	0.3	13	11		5	3		203	218		343	416	
	0.4	13	12		5	4		203	217		343	420	
	0.5	15	12		6	4		191	217		353	420	
	0.6	17	15		9	8		211	219		439	432	
	0.7	21	21		11	11		209	209		430	430	
4-2-4	0.3	13	11		6	4		231	255		538	546	
	0.4	14	14		7	7		231	231		538	538	
	0.5	14	14		7	7		231	231		538	538	
	0.6	18	14		9	7		203	231		414	538	
	0.7	22	22		11	11		195	195		406	406	
1-8-1	0.3	14	13	12	5	4	5	202	211	303	392	416	458
	0.4	17	13		6	4		251	211		541	416	
	0.5	17	14		6	5		251	207		541	379	
	0.6	20	17		7	6		195	202		374	364	
	0.7	20	20		7	7		195	195		374	374	
6-1-3	0.3	13	11	11	5	3	4	251	215	342	561	416	491
	0.4	13	12		5	4		251	220		561	429	
	0.5	15	12		7	4		202	220		358	429	
	0.6	19	13		9	6		206	204		427	368	
	0.7	23	23		11	11		186	186		400	400	

Table 4-5 (Continued)

# 3 Stages	MS=0, MC=1												
		Machines			Minicells			Average Flowtime			Makespan		
		Hierarchical			Hierarchical			Hierarchical			Hierarchical		
Stage Division	Threshold	ALC	SLC	Simultaneous	ALC	SLC	Simultaneous	ALC	SLC	Simultaneous	ALC	SLC	Simultaneous
4-5-1	0.3	14	11	13	6	3	5	200	216	346	355	416	543
	0.4	14	14		6	6		200	200		355	200	
	0.5	15	14		7	6		191	200		348	200	
	0.6	19	17		10	9		179	192		304	348	
	0.7	21	21		11	11		180	180		306	306	
1-4-5	0.3	13	11	12	5	3	5	229	215	283	526	416	477
	0.4	16	14		7	6		218	229		451	526	
	0.5	17	14		8	6		220	229		467	526	
	0.6	18	18		9	9		215	215		451	451	
	0.7	20	20		10	10		200	200		440	440	
6-2-2	0.3	12	11	12	5	3	4	196	220	317	409	422	480
	0.4	12	12		5	4		196	220		409	416	
	0.5	14	12		6	4		190	220		348	416	
	0.6	18	14		9	7		195	220		405	449	
	0.7	20	20		10	10		191	192		401	401	
5-2-3	0.3	13	11	12	6	4	5	213	217	352	406	419	489
	0.4	13	12		6	5		213	217		406	419	
	0.5	15	12		8	5		221	217		509	419	
	0.6	17	13		9	7		207	243		423	527	
	0.7	23	23		12	10		191	191		401	401	
7-2-1	0.3	14	13	12	5	4	5	237	215	300	496	416	517
	0.4	16	13		6	4		223	215		482	416	
	0.5	16	13		6	4		223	215		482	416	
	0.6	21	19		9	8		209	210		456	457	
	0.7	21	21		9	9		209	209		456	456	
3-6-1	0.3	14	11	12	6	4	5	205	216	444	362	418	704
	0.4	17	15		8	7		195	199		369	359	
	0.5	17	15		8	7		195	199		369	359	
	0.6	18	18		9	9		195	195		369	369	
	0.7	20	20		10	10		191	191		348	348	

From the results, it is observed that increase in threshold value increases the machine capacity. As noted in 27 product variants problem, low average flow times were obtained with ALC and less minicells and low machine capacities were produced using SLC. It was also seen that the average flow time values generally decreases with the increase in machine capacities i.e. average flow times decrease with increase in threshold values. Mostly, low average flow times are generated at 0.6 or 0.7 threshold values. The variation of machines required and average flow time values is shown in Figure 4-5.

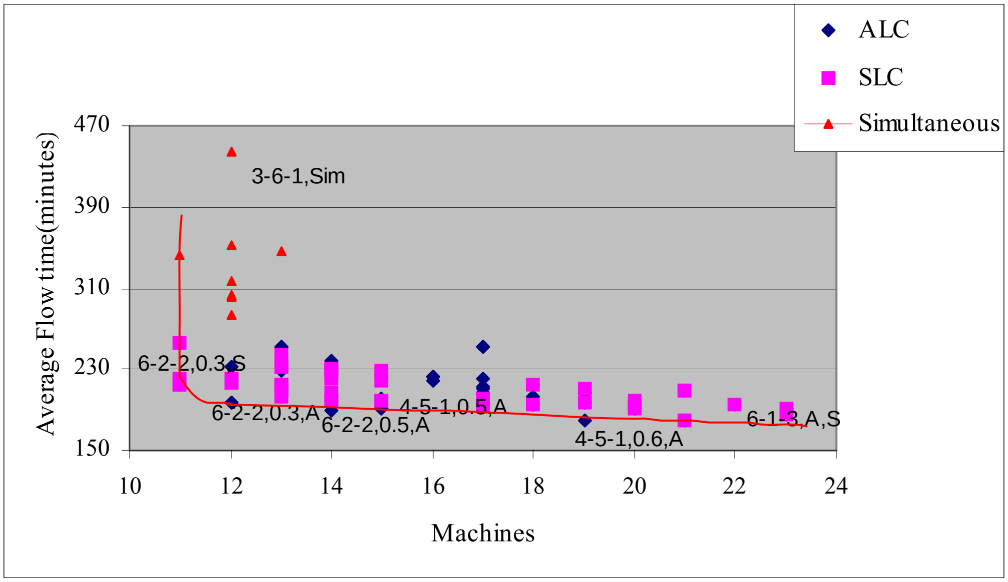


Figure 4-5: Variation of Machines Vs Average Flow Time - 3 stages

Best minicell designs generating low flow times using reasonable machines were identified on the analysis of the results in Table 4-5. Again low flow times were produced at high threshold by both ALC and SLC methods. From Figure 4-5, it is observed that minicell designs optimizing both objectives are given by ALC and designs with minimum average flow time are given by both ALC and SLC at high threshold values. At high threshold values, the similarity required is higher are combined into an option family only if they are exactly same; else minicells will contain single options. Therefore, minicell designs close to Pareto Front were selected.

The 'simultaneous' method produced designs with minimum machine requirements, as these design lie close to lower end of machine capacities. From the above, it can be seen that designs with more machines in the initial stage such as 6-2-2, 4-5-1 lie in the Pareto front region. Having maximum number of machines in the initial (7-2-1) or middle (1-8-1) stages does not produce better minicell design to minimize any of the objectives.

To observe the performance of the hierarchical method and minicell design patterns in other stages, the 12 product variants problem was tested in 2 stages i.e. 10 machines are divided into 2 stages. The results are tabulated in Table 4-6 and the variation of machines Vs average flow times- 2 stages minicell designs are shown below.

Table 4-6: Problem # 2 in 2 stages in Hierarchical method

# 2 stages		Machines		Minicells		Average Flowtime		Makespan	
		Hierarchical		Hierarchical		Hierarchical		Hierarchical	
Stage Division	Threshold	ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
8-2	0.3	13	13	3	3	213	213	393	393
	0.4	16	13	4	3	189	213	365	393
	0.5	18	13	5	3	185	213	376	393
	0.6	21	18	7	6	205	208	412	413
	0.7	26	26	9	9	203	203	412	412
5-5	0.3	13	10	4	2	230	256	526	548
	0.4	16	14	6	5	218	230	451	526
	0.5	18	14	7	5	210	230	434	526
	0.6	18	16	7	6	210	218	434	451
	0.7	24	24	10	10	174	174	395	395
3-7	0.3	13	13	5	4	238	192	489	378
	0.4	13	13	5	4	238	192	489	378
	0.5	16	13	6	4	214	192	458	378
	0.6	20	17	8	7	197	211	360	447
	0.7	20	20	8	8	197	197	360	360
1-9	0.3	14	11	4	2	230	211	543	416
	0.4	14	14	4	4	230	230	543	543
	0.5	18	14	5	4	216	230	462	543
	0.6	22	14	6	4	201	230	385	543
	0.7	22	22	6	6	201	201	385	385

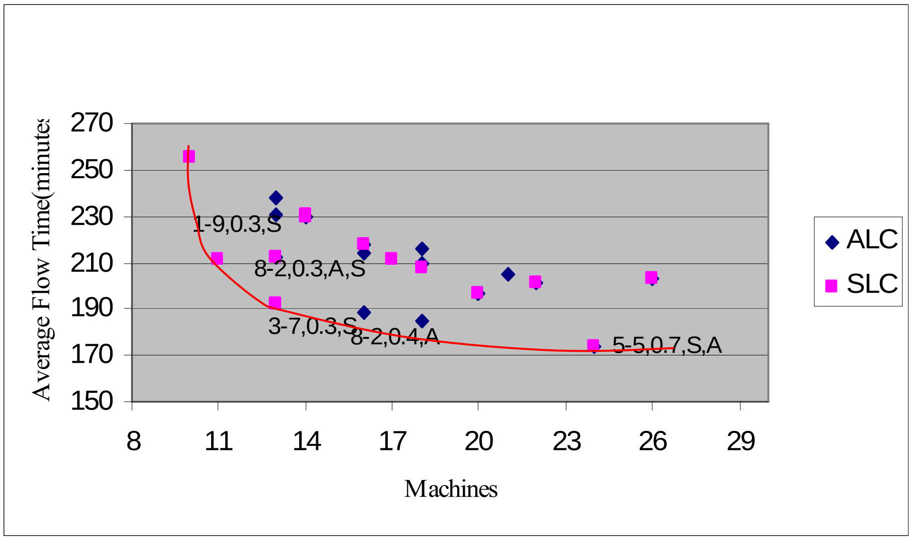


Figure 4-6: Variation of Machines Vs Average Flow Time - 2 stages

As observed previously, machine capacities increase with increase in threshold values and fewer minicells are generated at low threshold values. Also, lower average flow times are generated by both ALC and SLC at high threshold values with same machine usage. From the results and Figure 4-6 it is observed that more machines in the initial (8-2) or final stages (3-7) yields designs lying on the Pareto front. While having large number of machines in the final stage (1-9) generates designs with minimum machine requirements. It is seen that the minimum average flow time produced with 2 stage division is less than the value obtained in 3 stage division. The minimum value in 2 stages is due to the decrease in waiting time. Each part in order to get processed on each machine has to wait for all other parts to complete their processing. This would increase waiting times of the parts if there are more stages and hence increases average flow time. Therefore, in order to obtain minimum average flow time, the available machines must be divided into two stages. To evaluate the performance of a minicell design with a single stage further experimentation was conducted. The results are tabulated in Table 4-7 and presented graphically in Figure 4-7.

Table 4-7: Problem # 2 in 1 stage in Hierarchical method

# 1 Stage		Machines		Minicells		Average		Makespan	
		Hierarchical		Hierarchical		Hierarchical		Hierarchical	
Stage Division	Threshold	ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
10	0.3	13	11	2	1	217	200	460	402
	0.4	15	13	3	2	231	217	554	460
	0.5	20	15	4	3	199	231	469	554
	0.6	25	20	5	4	177	199	377	469
	0.7	25	25	5	5	177	177	377	377

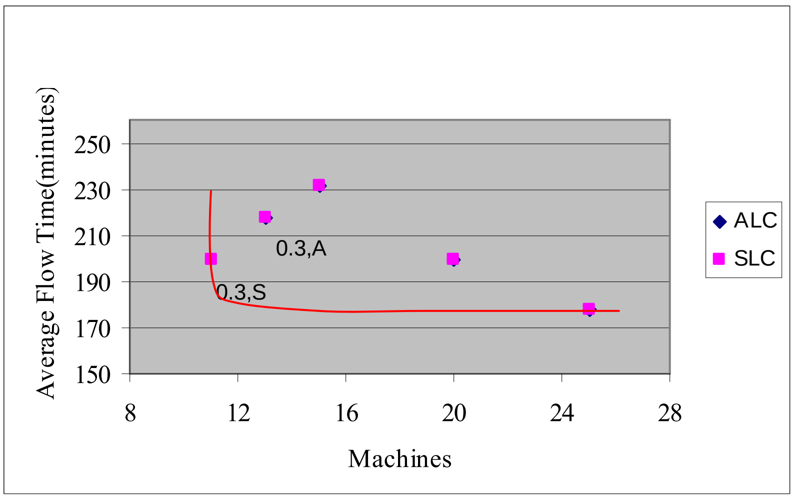


Figure 4-7 : Variation of Machines Vs Average Flow Time - 1 stage

In one stage minicell design, it was observed that increase in threshold increases machine capacity and decreases average flow time and the results with ALC and SLC are same at high threshold. But the minimum average flow time obtained with the 1 stage design was 177.353 minutes which is slightly higher than that obtained in two stage design. If all machines are assigned to a single stage, waiting times of options would be increased leading to increase in flow time. Hence it is clearly seen that in order to achieve lower average flow times it is necessary to divide the machines into at least 2 stages. Also from Figure 4-7, it is seen that there is no value in the Pareto front region and the minimum average flow time is generated using 25 machines while 174.057 minutes average flow time is generated in 2 stages using 24 machines.

In order to explore various solutions of this problem a branch and bound-like technique was applied for experimentation i.e. the 12 product variants problem is tested by varying the number of stages between the maximum and minimum number of stages into which 10 machines can be divided. 10 stages to a single stage minicell design. As seen earlier, one stage division does not generate a minicell design that satisfies both minimum average flow time or machine requirement criteria. The experimentation with 12 products problem is further continued by dividing 10 machines into 4, 5, 6 and 8 stages. The results are tabulated in Table 4-8 and the variation of machines vs. average flow times for the different stages is shown below.

Table 4-8 : Problem # 2 in a) 4 stages, b) 5 stages, c) 6 stages and d) 8 stages in Hierarchical method

a) 4 Stages

# 4 Stages		Machines		Minicells		Average Flow		Makespan	
		Hierarchical		Hierarchical		Hierarchical		Hierarchical	
Stage Division	Threshold	ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
1-1-2-6	0.3	13	11	6	4	230	222	528	432
	0.4	16	13	8	6	229	229	446	526
	0.5	16	13	8	6	229	229	446	526
	0.6	19	17	10	9	203	226	437	459
	0.7	21	21	11	11	190	190	342	342
7-1-1-1	0.3	13	13	5	5	216	216	416	416
	0.4	15	13	6	5	223	216	397	416
	0.5	15	13	6	5	223	216	397	416
	0.6	19	17	8	7	206	204	399	401
	0.7	19	19	8	8	206	206	399	399
4-1-1-4	0.3	13	11	6	4	239	217	536	430
	0.4	14	14	7	7	237	237	536	536
	0.5	14	14	7	7	237	237	536	536
	0.6	18	14	9	7	207	237	424	536
	0.7	22	22	11	11	194	194	406	406
3-2-3-2	0.3	12	11	7	6	198	221	374	416
	0.4	12	11	7	6	198	221	374	416
	0.5	12	11	7	6	198	221	374	416
	0.6	15	15	11	11	217	217	432	432
	0.7	15	15	11	11	217	217	432	432

b) 5 Stages

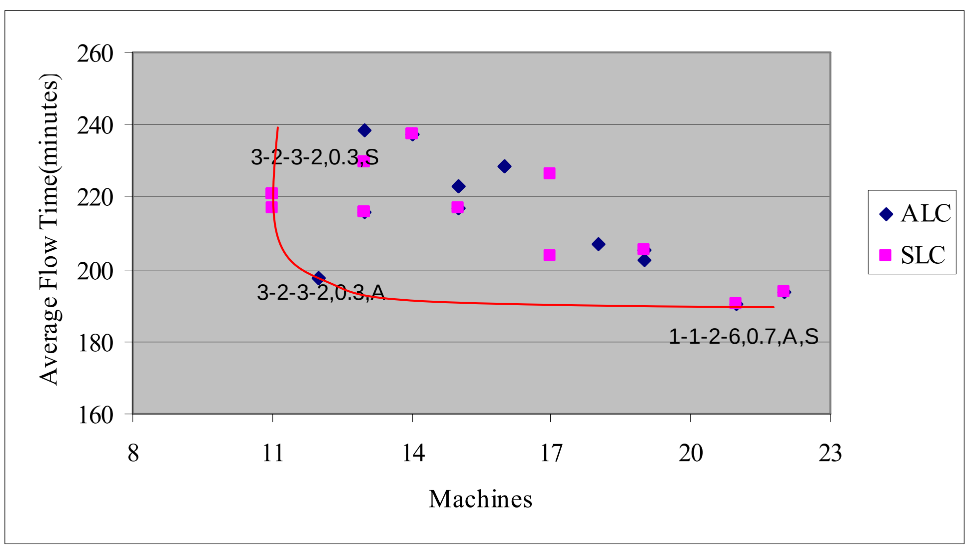
# 5 Stages		Machines		Minicells		Average Flow		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
5-2-1-1-1	0.3	12	11	7	6	224	225	416	416
	0.4	12	12	7	7	224	224	416	416
	0.5	14	12	8	7	204	224	394	416
	0.6	14	12	8	7	204	224	394	416
	0.7	18	18	10	10	205	205	399	399
1-2-4-2-1	0.3	13	11	9	7	257	219	528	416
	0.4	13	11	9	7	257	219	528	416
	0.5	13	11	9	7	257	219	528	416
	0.6	16	16	12	12	220	220	496	496
	0.7	16	16	12	12	220	220	496	496
1-1-1-1-6	0.3	12	11	6	5	229	222	526	432
	0.4	15	13	8	7	219	229	455	526
	0.5	15	13	8	7	219	229	455	526
	0.6	17	15	9	8	202	219	437	455
	0.7	19	19	10	10	188	189	342	342

c) 6 Stages

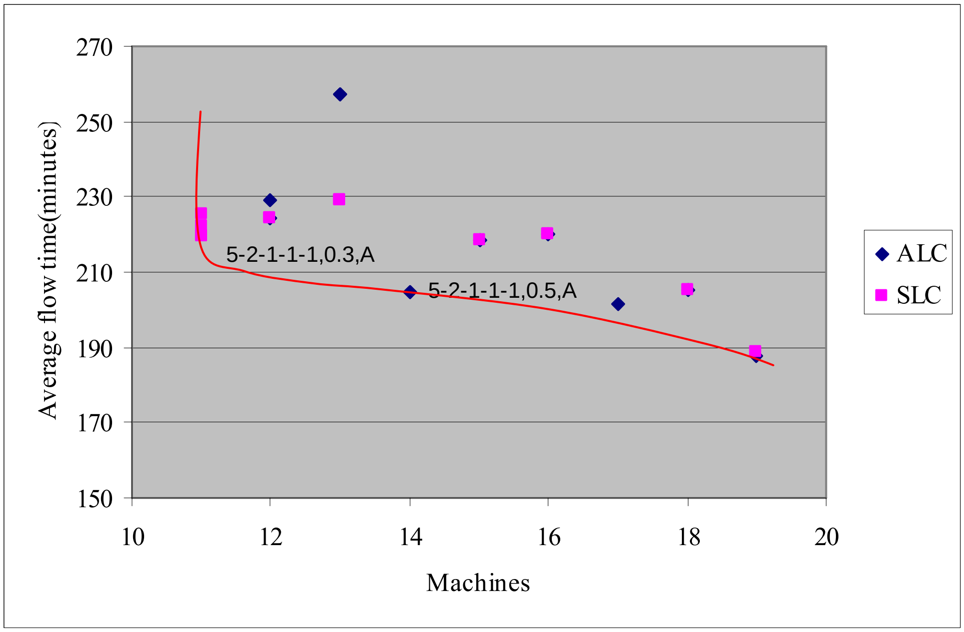
# 6 Stages		Machines		Minicells		Average		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
4-2-1-1-1-1	0.3	12	12	8	8	223	223	416	416
	0.4	12	12	8	8	223	223	416	416
	0.5	12	12	8	8	223	223	416	416
	0.6	14	12	9	8	204	223	404	416
	0.7	16	16	10	10	204	204	404	404
1-2-2-3-1-1	0.3	12	11	9	8	195	220	343	416
	0.4	12	11	9	8	195	220	343	416
	0.5	12	11	9	8	195	220	343	416
	0.6	14	14	11	11	199	199	348	348
	0.7	14	14	11	11	199	199	348	348
1-1-1-1-1-5	0.3	12	11	7	6	229	222	526	432
	0.4	15	13	9	8	218	229	451	526
	0.5	15	13	9	8	218	229	451	526
	0.6	15	15	9	9	218	218	451	451
	0.7	17	17	10	10	201	201	437	437

d) 8 Stages

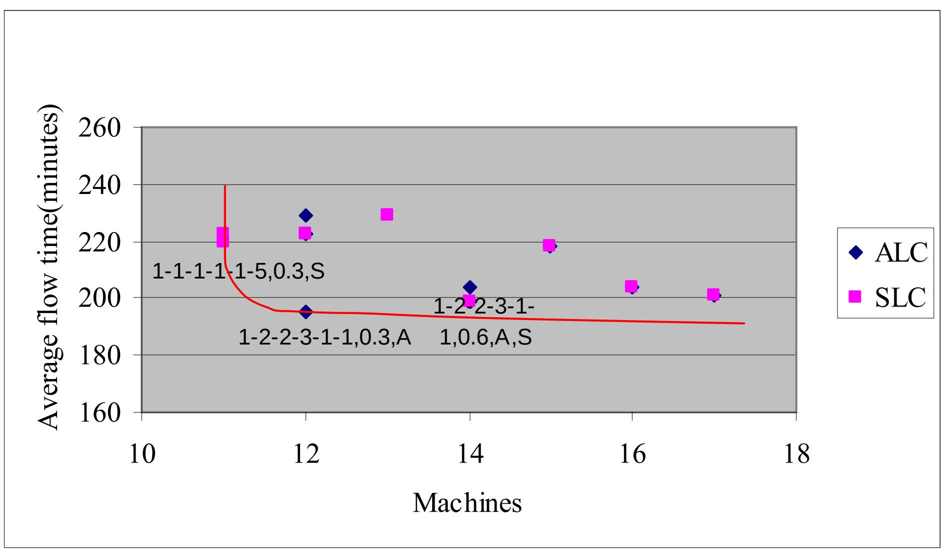
# 8 Stages		Machines		Minicells		Average Flow		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
2-2-1-1-1-1-1-1	0.3	12	11	10	9	216	225	416	416
	0.4	12	11	10	9	216	225	416	416
	0.5	12	11	10	9	216	225	416	416
	0.6	13	13	11	11	223	223	416	416
	0.7	13	13	11	11	223	223	416	416
1-1-1-2-2-1-1-1	0.3	11	11	10	10	251	251	540	540
	0.4	11	11	10	10	251	251	540	540
	0.5	11	11	10	10	251	251	540	540
	0.6	11	11	10	10	251	251	540	540
	0.7	11	11	10	10	251	251	540	540
1-1-1-1-1-1-1-3	0.3	12	11	10	8	211	219	406	421
	0.4	12	11	10	8	211	219	406	421
	0.5	12	11	10	8	211	219	406	421
	0.6	14	12	11	10	216	211	458	406
	0.7	16	16	12	12	205	205	436	436



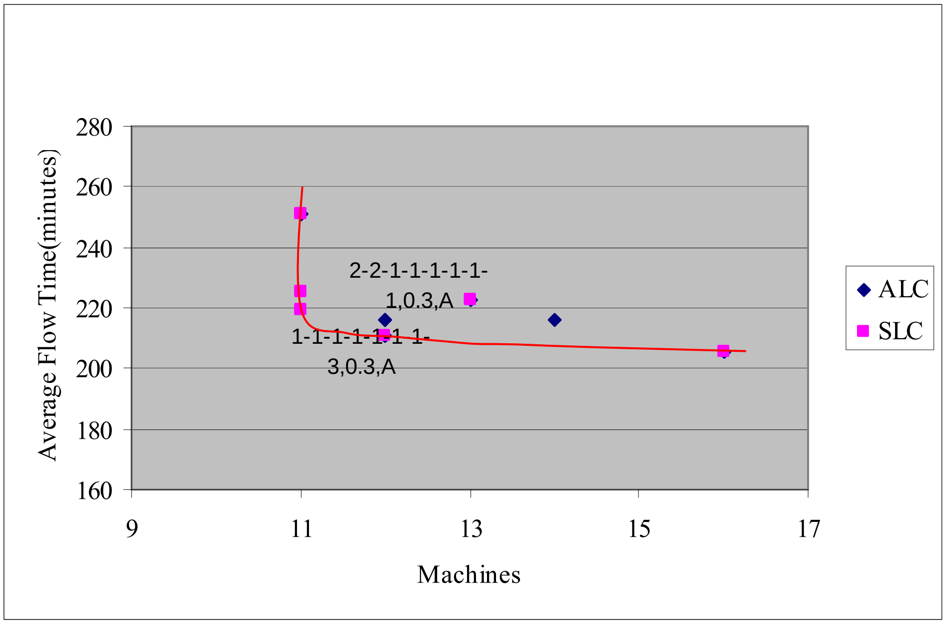
Variation of Machines Vs Average Flow Time – 4 stages



Variation of Machines Vs Average Flow Time – 5 stages



Variation of Machines Vs Average Flow Time – 6 stages



Variation of Machines Vs Average Flow Time – 8 stages

Figure 4-8: Variation of Machines Vs Average Flow Time in a) 4 stages, b) 5 stages c) 6 stages d) 8 stages

In all the above graphs, the same results are recorded as for the 2 stage and 3 stage minicell designs for 12 product variants. In 4 stage design, it is seen that minicell designs with more machines in the initial stage and designs with even distribution of machines were close to Pareto front. Designs with more machines in the initial stages were to the close Pareto front in 5 stage designs and designs with more machines in the final stage gave better results for minimization of average flow time objective.

In 6 stage designs, it is seen that designs with more machines in the middle gave better designs lying closer to the Pareto front while designs with more machines in the end with 8 stage designs gave better results. It was also observed that designs with fewer stages gave low average flow time values in comparison with average flow times obtained for designs with more stages.

The 12 product variants problem with 3 stages was also tested with GA by varying the Makespan (MS) and Machine Count (MC) parameters. Three different weights were assigned to these two parameters in order to observe their performance relative to hierarchical method. The Makespan with weight of zero and Machine count with weight of one have been tested. The performance of GA by assigning weights of 0.5 to Makespan and Machine count and weight of one to Makespan and weight of zero to Machine Count are further tested. These results are shown in Table 4-9.

Table 4-9 : Problem # 2 in 3 stages in Simultaneous methods with a) MS=0.5 and MC=0.5 and b) MS=1 and MC=0

a) MS=0.5 and MC=0.5

#3 Stages	MS-0.5,MC-0.5		
Stage Division	Machines	Average Flow Time	Makespan
5-4-1	12	394	577
2-5-3	12	248	398
5-1-4	13	261	407
4-4-2	12	329	475
7-1-2	11	290	486
6-1-3	12	299	441
2-7-1	12	354	567
2-1-7	12	305	529
1-5-4	12	278	453
5-4-1	11	396	566
4-2-4	12	317	433
1-8-1	11	274	470
7-2-1	13	235	386

b) MS=1 and MC=0

# 3 Stages	MS-1,MC-0		
Stage Division	Machines	Average Flow Time	Makespan
6-3-1	17	225	345
1-7-2	16	229	340
7-1-2	15	232	357
2-6-2	17	251	354
3-3-4	16	250	353
1-4-5	16	239	358
1-5-4	17	236	350
2-3-5	16	229	348
7-2-1	17	196	342

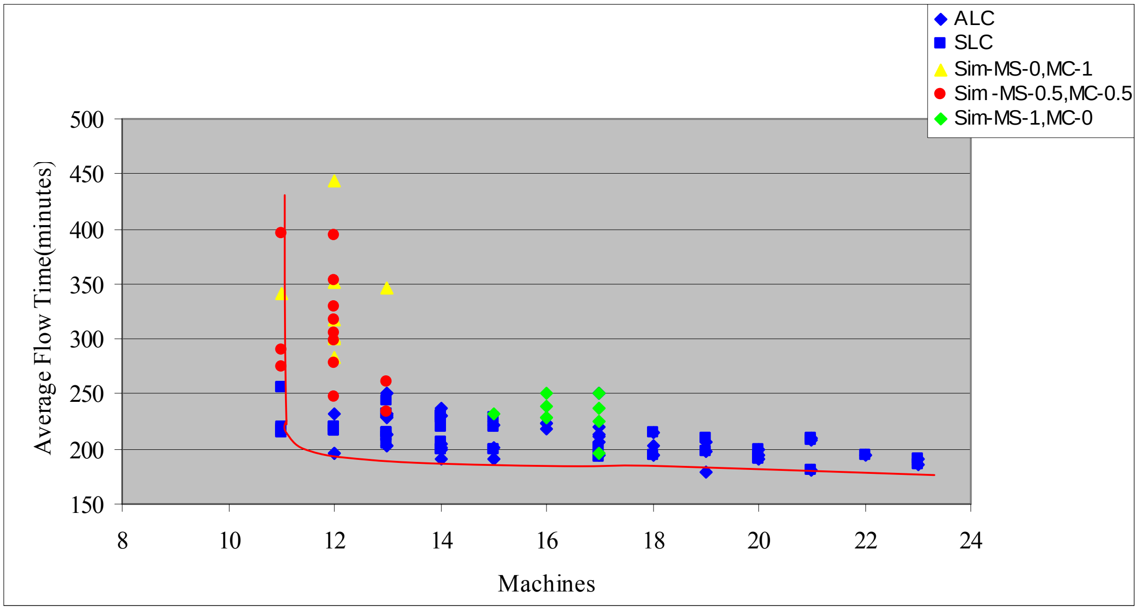


Figure 4-9 : Comparison of hierarchical and simultaneous methods in 12 product variants problem

From Figure 4-9, it is observed that the results from MS = 0 and MC = 1 and MS=0.5 and MC = 0.5 lie in the portion of minimum machine requirements and high average flow times. The results from MS = 1 and MC = 0 lie in the lower bound of average flow time with medium machine requirements region. However, the Pareto Front drawn considering the results from both methods(hierarchical and simultaneous) show that results from the hierarchical method are better and lie on the Pareto Front in most cases.

4.2.3 Problem No. 3 = 18 Product Variants

To authenticate the observations from the first two problems an 18 product variants problem 3 features and option assignment as shown in Figure 4-10 was tested. The option-machine matrix is shown below.

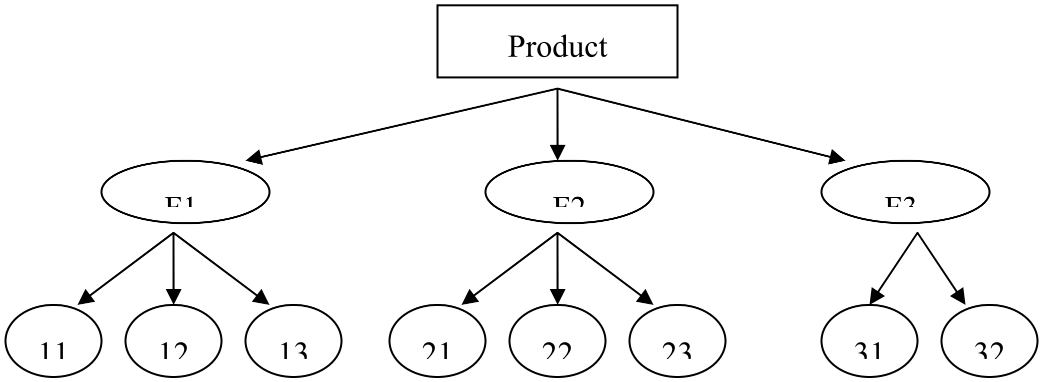


Figure 4-10 : 18 Products structure

Table 4-10 : Option-Machine matrix of 18 products problem

Options\ Machines	M1	M2	M3	M4	M5	M6	M7	M8
11	0.55	0	0.09	0.84	0	2.5	0	1
12	0	0.2	0	0.75	0	0	0.89	0
13	0	0.88	0.56	0.02	0	0	0	0
21	0.28	0	0	0	1.18	0	0.84	2
22	0	0	0.85	0.19	0	0.05	0	6
23	1.86	0	0	0	1.24	1.08	0.89	2.5
31	0	0.98	0	0.98	1.26	0	1.28	1
32	0.89	0.85	0	0	2.23	1.23	0	0

Using the inferences from 27 and 12 product variants problem, minicell designs are obtained for low threshold value 0.3 and high threshold values 0.6 and 0.7. This is because it was seen from the previous two problems, that high threshold values need more machines and result in low average flow times. Results similar to those observed with the previous problems were noted here too. The results for all the experimentation conducted for the 18 product variants problem are shown in Appendix I. This fact remains true in 18 products problem also. Again, ALC and SLC generated same number of minicells and machine requirements at high threshold value of 0.7. The results and plot of variation of machines and average flow times for 3 stages is illustrated below. The problem is also tested for MS =0.5 and MC = 0.5 and MS =1 and MC =0.

Table 4-11 : Problem # 3 in 3 stages in Hierarchical and Simultaneous methods (MS=0 and MC=1)

# 3 Stages	MS=0, MC=1												
	Machines				Minicells			Average Flowtime			Makespan		
Stage Division	Threshold	Hierarchical		Simultaneous	Hierarchical		Simultaneous	Hierarchical		Simultaneous	Hierarchical		Simultaneous
		ALC	SLC		ALC	SLC		ALC	SLC		ALC	SLC	
5-1-2	0.3	17	15	16	5	3	4	391	355	706	961	849	896
	0.6	21	17	15	8	6	5	351	383	487	973	961	1043
	0.7	26	26		11	11		229	229		592	592	
3-1-4	0.3	18	14	15	6	3	3	334	353	565	929	930	887
	0.6	21	21		11	11		380	380		961	961	
	0.7	21	21		11	11		380	380		961	961	
3-3-2	0.3	19	15		7	3		343	351		666	1075	
	0.6	24	24		14	14		342	342		705	705	
	0.7	24	24		14	14		342	342		705	705	

Table 4-12: Problem # 3 in 3 stages in Simultaneous methods (MS=0.5 and MC=0.5)

MS-0.5,MC-0.5			
Stages	Machines	Average Flow Time	Makespan
4-1-3	16	563	866
5-1-2	16	706	896
2-2-4	16	629	857
4-3-1	16	688	878
2-4-2	15	483	1023
3-4-1	16	520	951

Table 4-13 : Problem # 3 in 3 stages in Simultaneous methods (MS=1 and MC=0)

MS-1,MC-0			
Stages	Machines	Average Flow Time	Makespan
1-5-2	15	638	1479
1-2-5	15	563	1477
1-6-1	15	575	1372
4-3-1	16	948	1503
3-1-4	15	533	892
6-1-1	15	506	992
1-3-4	15	762	1175
5-1-2	15	487	1043

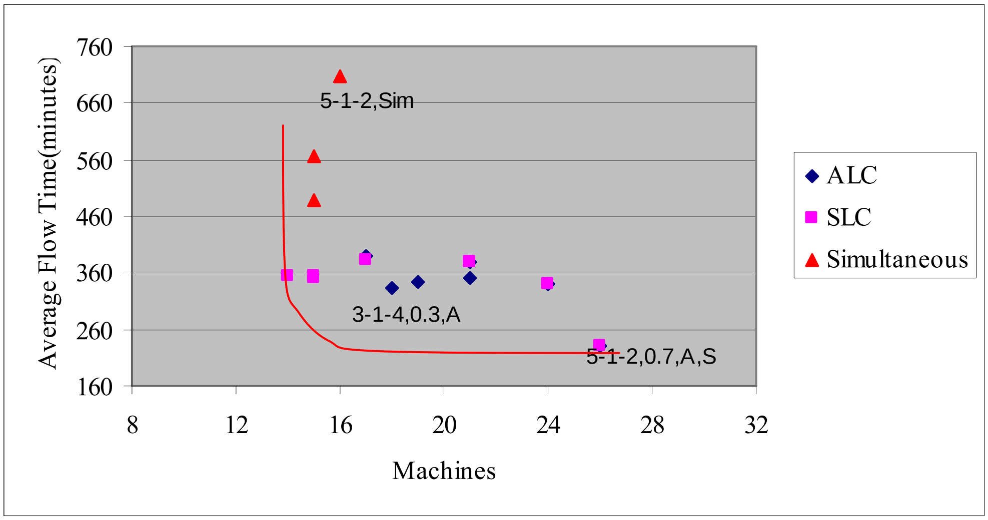


Figure 4-11: Variation of Machines Vs Average Flow Time - 3 stages

The minicell designs with minimum machine requirements and minimum average flow times and designs minimizing both parameters (Pareto Front) can be observed from Figure 4-11. It is seen that designs with sequential decrement in the distribution of machines with more machines in the initial stages like 3-3-2 yielded better results with regards to average flow time and machine capacity. Also minicell designs with more machines in the initial stage like 5-1-2 produced low average flow times. As seen from the above graph, minicell designs in the Pareto front are produced by the hierarchical method. In this problem, too the hierarchical method gave low average flow time values in comparison with the simultaneous method.

The 18 product variants problem with 8 machines is tested with 2 and 6 stage minicell designs in order to study the impact of machine division. Since there are 8 machines in total, the maximum number of stages in which the machines can be divided is 8. The results obtained with 2 stage and 6 stage minicell designs are shown Table 4-14. The variation of machines Vs average flow time for 18 product variants problem with 2 stage and 6 stage minicell designs is shown in Figure 4-12.

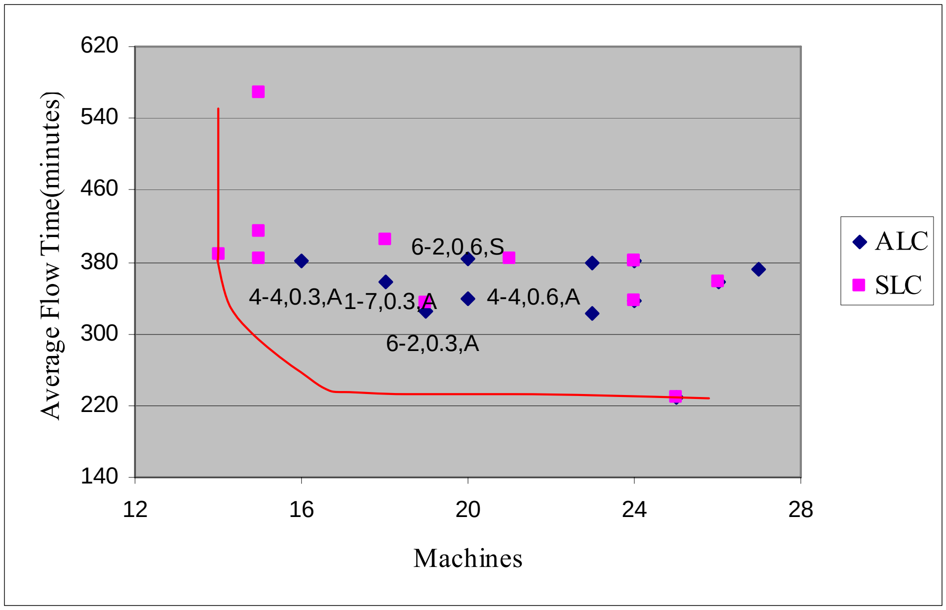
Table 4-14: Problem # 3 in a) 2 stages, b) 6 stages in Hierarchical method

a) 2 stages

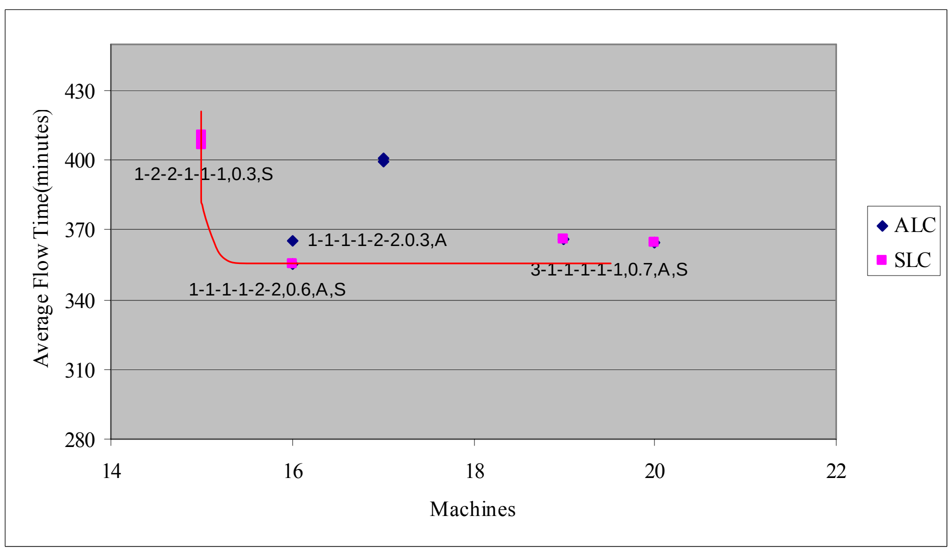
		Machines		Minicells		Average Flowtime		Makespan	
# 2 stages		Machines		Minicells		Average Flowtime		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
6-2	0.3	19	15	5	2	325	384	758	985
	0.6	23	19	8	6	323	334	747	920
	0.7	25	25	9	9	228	228	594	594
4-4	0.3	16	14	4	2	381	388	945	941
	0.6	20	18	8	7	340	405	1059	1079
	0.7	24	24	10	10	337	337	826	826
3-5	0.3	20	15	6	2	382	569	1710	905
	0.6	24	24	11	11	380	380	968	968
	0.7	27	24	12	11	372	380	968	968
1-7	0.3	18	15	3	2	358	415	1024	1029
	0.6	23	21	6	5	380	384	1001	1024
	0.7	26	26	7	7	358	358	1095	1095

b) 6 Stages

# 6 Stages		Machines		Minicells		Average		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
1-1-1-1-2-2	0.3	16	15	8	8	365	407	966	880
	0.6	16	16	10	10	356	356	751	751
	0.7	16	16	10	10	356	356	751	751
1-2-2-1-1-1	0.3	17	15	8	6	399	407	1083	1097
	0.6	19	19	10	10	366	366	1033	1033
	0.7	19	19	10	10	366	366	1033	1033
3-1-1-1-1-1	0.3	17	15	8	6	401	411	1073	1128
	0.6	20	20	11	11	365	365	1007	1007
	0.7	20	20	11	11	365	365	1007	1007



a) Variation of Machines Vs Average Flow Time - 2 stages



b) Variation of Machines Vs Average Flow Time - 6 stages

Figure 4-12 : Variation of Machines Vs Average Flow Time a) 2 stages and b) 6 stages

As seen earlier, results of 2 stage and 6 stage minicell designs show that an increase in threshold values increases machine capacity. ALC needs more machines and generates low average flow times. As observed earlier, minimum average flow time produced with two stage designs for 18 products is less than the minimum average flow time generated with three stage designs. From Figure 4-12 (a), it is noted that more machines in initial (6-2) or final stages (1-7) generates minimum average flow time designs using optimum number of machines in 2 stages. For 6 stage designs, more machines in the final stage (1-1-1-1-2-2) produced designs in Pareto front. Minimum average flow time generated with six stages is observed to be more than the minimum average flow time obtained in 2 and 3 stages. This is because increase in number of stages, requires the parts to be routed to different stages, which increase their waiting time and hence increases the average flow time.

In order to validate all the observations from the three problems, experimentation is continued with a small problem of size 40 i.e. using 8 products and 5 machines. Details of this problem are given in 4.2.4.

4.2.4 Problem No. 4= 8 product variants

To justify various observations from the 27, 18 and 12 product variants problems at different stages minicell designs are examined for 8 products problem. In this problem also initially 5 machines are divided into 3 stages and cut-offs are identified using GA. Later 8 products problem is tested in 2 and 4 stages. Product structure and option-machine matrix of 8 products problem is given below. The results from the experimentation are given in the Appendix I.

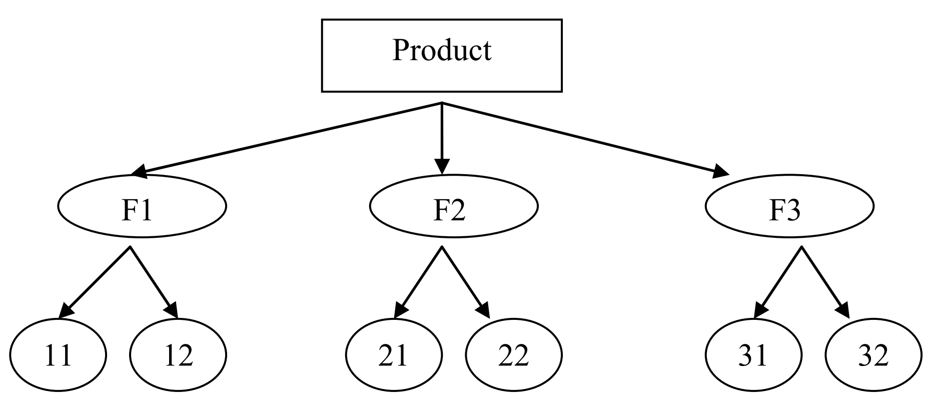


Figure 4-13 : 8 Products structure

Table 4-15: Option-machine matrix of 8 products problem

Options\Machines	M1	M2	M3	M4	M5
11	1.02	0	0.57	0.9	0
12	0	0.2	0	0.75	0
21	0	0	0.19	0	1.2
22	0.18	0	0	1.42	0
31	0	0	0	0.09	0
32	0.28	0	0.86	2.8	1

As can be observed machine capacity increase with increase in threshold values, ALC generating minimum average flow times and SLC producing less number of minicells with low machine usage were holding good in 8 product variants problem, too. Also, at high threshold value of 0.7, ALC and SLC generated same number of minicells and machines and average flow times. The results and variation of machines vs. average flow time graphs for 3, 2 and 4 stages are given below.

Table 4-16: Problem # 4 in a) 3 stages, b) 2 stages and c) 4 stages in Hierarchical method

a) 3 stages

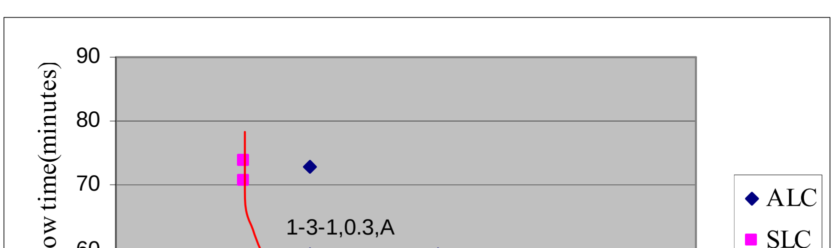
# 3 Stages	MS=0, MC=1												
		Machines			Minicells			Average Flowtime			Makespan		
Stage Division	Threshold	Hierarchical		Simultaneous	Hierarchical		Simultaneous	Hierarchical		Simultaneous	Hierarchical		Simultaneous
		ALC	SLC		ALC	SLC		ALC	SLC		ALC	SLC	
2-2-1	0.3	6	5		5	4		73	74		170	170	
	0.6	7	7		6	6		58	58		126	126	
	0.7	7	7		6	6		58	58		126	126	
1-1-3	0.3	6	5	5	4	3	3	56	74	96	106	170	164
	0.6	8	6		5	4		59	56		126	106	
	0.7	10	10		6	6		56	56		127	127	
1-3-1	0.3	6	5	5	4	3	3	59	71	96	127	143	164
	0.6	8	8		6	6		58	58		128	128	
	0.7	8	8		6	6		58	58		128	128	
3-1-1				5			3			98			150

b) 2 stages

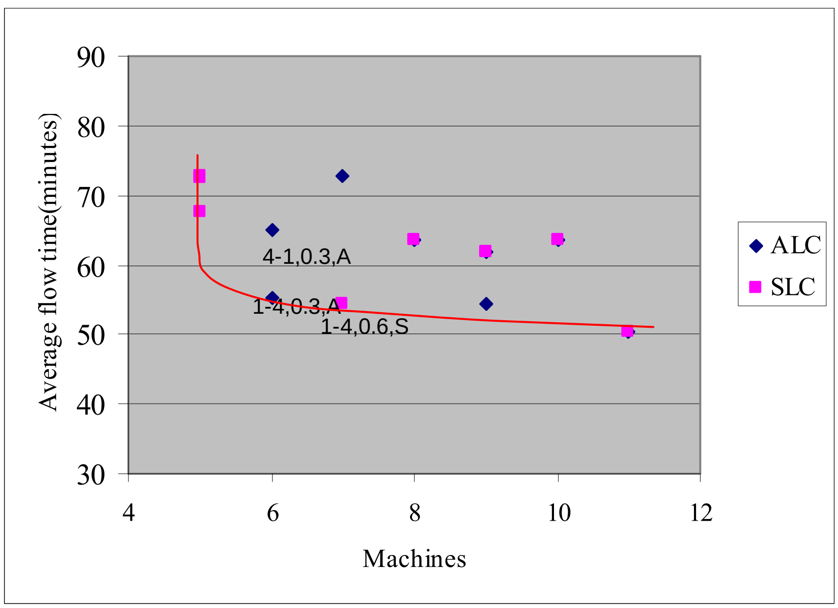
# 2 stages		Machines		Minicells		Average Flowtime		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
4-1	0.3	6	5	3	2	65	68	158	166
	0.6	8	8	5	5	64	64	128	128
	0.7	10	10	6	6	64	64	128	128
1-4	0.3	6	5	3	2	55	73	127	168
	0.6	9	7	5	4	54	54	106	106
	0.7	11	11	6	6	50	50	93	93
3-2	0.3	7	5	5	3	73	72	168	159
	0.6	9	9	7	7	62	62	125	125
	0.7	9	9	7	7	62	62	125	125

c) 4 stages

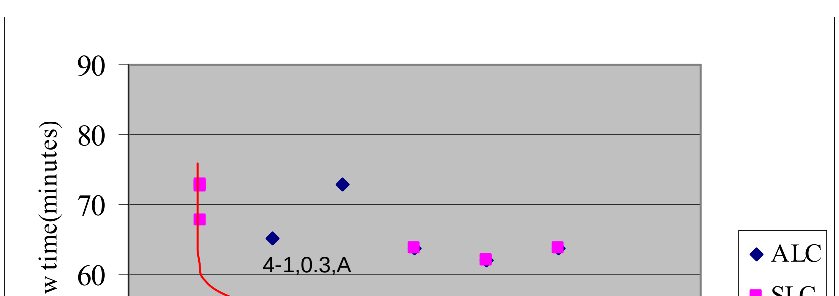
# 4 stages		Machines		Minicells		Average Flowtime		Makespan	
Stage Division	Threshold	Hierarchical		Hierarchical		Hierarchical		Hierarchical	
		ALC	SLC	ALC	SLC	ALC	SLC	ALC	SLC
1-1-1-2	0.3	6	5	5	4	73	72	170	160
	0.6	7	7	6	6	64	64	135	135
	0.7	7	7	6	6	64	64	135	135
2-1-1-1	0.3	5	5	5	5	74	74	170	170
	0.6	5	5	5	5	74	74	170	170
	0.7	5	5	5	5	74	74	170	170
1-2-1-1	0.3	5	5	5	5	73	73	169	169
	0.6	5	5	5	5	73	73	169	169
	0.7	5	5	5	5	73	73	169	169



a) Variation of Machines Vs Average Flow Time – 3 stages



b) Variation of Machines Vs Average Flow Time - 2 stages



c) Variation of Machines Vs Average Flow Time- 4 stages

Figure 4-14: Variation of Machines Vs Average Flow Time - a) 3 stages, b) 2 stages and c) 4 stages

From three stage designs, it is seen that hierarchical method produces low average flow times. In two and three stages, minicell designs with more machines in the initial or final stages lie in the Pareto front. In four stage designs, it is seen that minicell designs with more machines in the final stage produce better results with respect to average flow time and machine capacity. Again, minimum average flow time is generated in two stages followed by three and four stages. For average flow time minimization objective alone, designs with more machines in the final stage gave better results in all three stages.

The overall observations from all the above problems are used in developing rules for extracting minicell designs for a given problem. This is explained in the following section.

4.3 Final Observations

In the present research, minicell designs are developed for three criteria i.e. the objective of minicell design could be either to minimize average flow time or minimize machine capacity or minimization of both average flow time and machine requirements. In order to study the performance of minicell designs several experiments are conducted varying the problem size from 189 to 40 by changing the number of product variants and machines. A summary of the tested problem results is discussed in this section.

Using the inferences from 27, 18, 12 and 8 product variants problems, rules are extracted to facilitate designing the best minicell configuration for each objective.

4.3.1 Average Flow Time Minimization

From the tested problems, minicell designs which generate minimum average flow times are identified. A summary of the results of all the tested problems which generate minimum average flow time with the corresponding machine requirements are given in . These designs are identified by considering the points on the Pareto Front which lie on the bottom of Y-axis.

Table 4-17 : Summary of Minicell Designs for Average Flow time minimization

a) 27 products

Stages	Division	Average flow time	Machines
3	5-1-1	86.3	22
3	3-2-2	88.219	12

b) 18 products

Stages	Division	Average flow time	Machines
2	6-2	228.208	25
2	4-4	336.517	24
3	5-1-2	228.83	26
3	3-3-2	341.57	24
6	1-1-1-1-2-2	355.686	16
6	3-1-1-1-1-1	364.642	20

c) 12 products

Stages	Division	Average flow time	Machines
1	10	177.353	25
2	5-5	174.057	24
2	3-7	196.678	20
3	4-5-1	178.708	19
3	4-5-1	180.372	21
3	6-1-3	186.03	23
4	1-1-2-6	190.255	21
4	4-1-1-4	193.85	22
5	1-1-1-1-6	188.785	19
6	1-2-2-3-1-1	195.344	12
8	1-1-1-1-1-1-1-3	205.419	16

d) 8 products

Stages	Division	Average flow time	Machines
2	1-4	50.295	11
2	1-4	54.32	7
3	1-1-3	55.933	10
3	1-1-3	56.336	6
4	1-1-1-2	63.905	7

In the 27 product variants problem, minicell designs having more machines in the initial stage or even distribution of machines generate minimum average flow time. While in 18 product variants problem, 1, 2 and 3 stages, which are considered as designs with low number of stages for this problem using 8 machines, minicell designs with more machines in the beginning or middle stages gave better results. For higher number of stages, i.e. 6 stages in this case, more machines in the initial or final stages produced good minicell designs.

For 12 product variants problem, 2, 3 and 4 are considered as designs with low number of stages with 10 machines with these stages, minicell designs with more machines in the beginning or final stages gave good results. For 6 and 8 stage designs, more machines in the middle and final stages produced low average flow times. For a small problem like 8 product variants, for designs with low number of stages, having more machines in the end produced minimum average flow times. Most of these designs are developed by ALC, as ALC produces more minicells and more machines and consequently reduces the average flow time. Hence the rules developed for extracting minicell designs with minimum average flow time objective are:

- ❖ If number of products is less than 10 and high or low number of stages is required then minicell designs with more machines in the final stage yield good results.
- ❖ If number of products is more than 10 and fewer stages are desired then minicell designs with more machines in the initial or middle stages produce low average flow times.
- ❖ If number of products is more than 10 and more stages are desired then minicell designs with more machines in the middle or final stages produce low average flow times.

4.3.2 Machine Capacity Minimization

Minicell designs which need minimum number of machines and corresponding average flow times are selected for minimization of machine capacity objective. These designs lie on the Pareto front towards the left side of the X-axis. A summary of these designs is given below:

Table 4-18: Summary of Minicell designs for Machine Capacity minimization

a) 27 products

Stages	Division	Machines	Average flow time
3	5-1-1	7	130.91
3	1-1-5	7	133.727
3	1-3-3	7	133.437

b) 18 products

Stages	Division	Machines	Average flow time
2	4-4	14	387.619
2	6-2	15	384.18
3	3-1-4	14	353.248
3	3-3-2	15	351.35
6	1-2-2-1-1-1	15	406.895
6	1-1-1-1-2-2	15	407.243

c) 12 products

Stages	Division	Machines	Average flow time
1	10	11	199.615
2	5-5	4	255.635
2	1-9	11	211.403
3	3-6-1	11	215.985
3	1-4-5	11	215.113
3	4-2-4	11	255.135
4	4-1-1-4	11	217.156
4	3-2-3-2	11	220.631
5	1-2-4-2-1	11	219.489
6	1-2-2-3-1-1	11	219.631
8	-1-1-1-1-1-1-	11	219.102

d) 8 products

Stages	Division	Machines	Average flow time
2	4-1	5	67.63
2	1-4	5	72.733
3	1-3-1	5	70.695
3	2-2-1	5	73.856
4	1-1-1-2	5	72.231
5	1-2-1-1	5	73.263

From the above tables, it is seen that most of the designs have same machine requirements but generate different average flow times. Most of the above designs are obtained by the SLC method. As discussed in section 4.2, SLC combines most of the options into one minicell and hence requires less number of machines. From the results tables, it is seen that most of the minicell designs with minimum machine capacity have more stages with more machines distributed in the middle or final stages. Hence the rule for obtaining the minimum machine requirement minicell designs is:

- ❖ Irrespective of the number of product variants, having more machines in the middle or final stages produces designs with low machine requirements.

4.3.3 Average Flow time and Machine Capacity Minimization

The designs which need minimum number of machines and produce minimum average flow times are the designs which lie towards the middle of the Pareto Front. These designs lie in the middle of the graph between the extremes of average flow time and number of machines. Summary of all the designs that meet this criteria are shown below.

Table 4-19 : Summary of Minicell Designs for Average Flow time and Machine Capacity minimization

a) 27 products

Stages	Division	Average flow time	Machines
3	3-2-2	103.404	11
3	3-2-2	88.219	12
3	3-1-3	107.229	11
3	3-1-3	106.241	12

b) 18 products

Stages	Division	Average flow time	Machines
2	6-2	325.466	19
2	4-4	339.503	20
2	1-7	358.247	18
3	3-3-2	343.342	19
3	3-1-4	333.95	18
6	1-1-1-1-2-2	355.686	16
6	1-1-1-1-2-2	365.287	16

c) 12 products

Stages	Division	Average flow time	Machines
1	10	199.615	11
2	3-7	191.87	13
2	8-2	212.535	13
2	8-2	188.575	16
3	6-2-2	196.365	12
3	5-3-2	203.121	13
3	6-2-2	190.207	14
4	3-2-3-2	197.542	12
5	5-2-1-1-1	204.429	14
6	1-2-2-3-1-1	195.344	12
8	1-1-1-1-1-1-1-3	210.663	12

d) 8 products

Stages	Division	Average flow time	Machines
2	4-1	65.03	6
2	1-4	55.287	6
2	1-4	54.32	7
3	1-1-3	56.336	6
3	1-3-1	59.037	6
3	2-2-1	57.642	7
4	1-1-1-2	63.905	7

In the 27 product variants problem, minicell designs with even distribution (3-2-2) and more machines in the initial or final stages produces good results. Whereas in the 18 product variants case, with fewer stages, more machines in initial or final stages produce better minicell designs. For designs with more stages, more machines in the final stages produced minimum values for average flow time and machine capacity.

For the 12 product variants problem, with 2, 3, 4 stages minicell designs with more machines in the initial or final stages produced better results with more stages case in 12 products, designs with more machines in the middle or end stages gave good results. Minicell designs with fewer stages need more machines in the initial or final stages to produce better results for 8 products problem. While with more number of stages, minicell designs with more machines in the middle or final stages perform better compared to other designs. The rules formed to extract minicell design with minimum average flow times and minimum machine capacities criteria are:

- ❖ Irrespective of number of product variants if fewer stages are desired, minicell designs with more machines in initial or final stages should be selected.
- ❖ Irrespective of number of product variants if more stages are desired, minicell designs with more machines in middle or final stages should be selected.

In order to get a clear understanding of all the above rules for different sizes of problems, a decision tree is drawn. The decision tree can be seen in the next chapter. Apart from all the rules developed for each objective some basic observations which could be applied to any desired objective are:

- ➢ ALC uses more minicells and more machines and hence produces low average flow times.
- ➢ SLC produces less minicells and uses less machines and hence satisfies low machine requirement criteria.
- ➢ Increase in threshold value increases the number of minicells and machine usage
- ➢ At high threshold values, ALC and SLC generate same number of minicells and same average flow time and machine capacities.
- ➢ Increase in threshold generally decreases average flow time.

The results obtained from hierarchical and simultaneous approaches are compared in the following section.

4.3.4 Comparison of Hierarchical and Simultaneous approaches

The results procured from hierarchical method for each of the problems is compared with the results obtained from GA in the simultaneous method with machine distribution in 3 stages. The GA program is tested for different combinations of minicell designs for various problem sizes. The problem size was varied by varying the number of machines and product variants.

Minicell designs generated in hierarchical approach in all the product variants with 3 stages were compared with the simultaneous method. For comparison, minicell designs with the same machine assignment to stages that were generated using the simultaneous method were chosen. Because a large number of designs were generated using the hierarchical method with different threshold values, the design that most resembles the simultaneous minicell design, in terms of number of minicells and total machine requirement were selected.

From the experimentation, it can be observed that, the first problem was tested with a large size and gradually the problem size was reduced. Initially it was assumed that with decrease in the problem size, GA based simultaneous approach would generate better results. Since the solution space of a small problem would be smaller than the large problem and hence GA would generate better results for small problems. But even with decrease in the problem size, for all the product variants tested (27, 18, 12 and 8 product variants), the minimum average flow times are generated for the best minicell designs found with the hierarchical approach. This is because the previous simultaneous method was developed using makespan minimization heuristic and its objective was to find best minicell designs using makespan and/or machine capacity or both measures. While in the present research, minicell designs are determined using average flow time and /or machine capacity requirements and uses minimization of flow time heuristic.

Comparing the makespan values between hierarchical and simultaneous methods, in 27 product variants problem, simultaneous method produced low makespan values than the hierarchical method. While in 18, 12 and 8 product variants problems, minimum makespan values are generated with hierarchical method, even though hierarchical approach used the flow time minimization heuristic to find the best minicell designs.

After comparing minicell design results with both of the approaches, it seems that hierarchical approach gives better results in terms of average flow times. This is due to the approach of designing minicells first and then scheduling jobs in minicells. This approach does not evaluate all the solutions in a solution space but tries to find a better solution within the tested region. Hence it could be said that designing minicells first and then scheduling demand of jobs in the minicells gives better results than doing designing and scheduling in minicells simultaneously.

5 CONCLUSIONS AND FUTURE WORK

A minicell-based manufacturing system was identified to have potential to provide flexibility and meet delivery expectations of mass customization. The performance of minicell configurations was investigated in the previous research by considering designing and scheduling in minicells simultaneously. In the present research, an attempt was made to evaluate the effect of designing minicells first and then scheduling demand of product variants in minicells subsequently on the performance of minicells. Using the results obtained for minicell design with the hierarchical method, rules were extracted to guide the design of good minicell configurations without following the lenghtly process of using metaheurictics (earlier work by Badurdeen and others). The experimentaion and results of this hierarchical approach has been presented in the previous chapter. Conclusions of the present research and directions for future work are given in this section.

5.1 Summary of Results

Experiments were conducted on several problems by varying parameters like number of stages, number of machines assigned to a stage, number of product variants, clustering technique and threshold values. The results obtained from all these problems with the variance in parameters are discussed in chapter 4. From the results, it is observed that the hierarchical method produced lower average flow time values in comparison with the simultaneous method for minicell designs with same/ similar number of minicells and machine requirements.

In the present research, best minicell designs are selected based on two performance measures, average flow time and machine capacity. In the first stage, minicell designs are selected based on the machine requirements criteria. The selected designs are then evaluated to calculate average flow times to produce given demand in scheduling stage. Finally, the results from design and scheduling stages are analyzed and best minicells are identified. In order to extract rules for developing good minicell designs, the results were all evaluated to identify similarities between different minicell designs that gave superior performance with different problem sizes and with different criteria. This was done considering minimization of average flow time objective, minimization of machine capacity and minimization of both parameters are established possible designs. The data from different problems was analyzed and the behavior observed was used to develop a framework to guide minicell design, as shown in Figure 5-1.

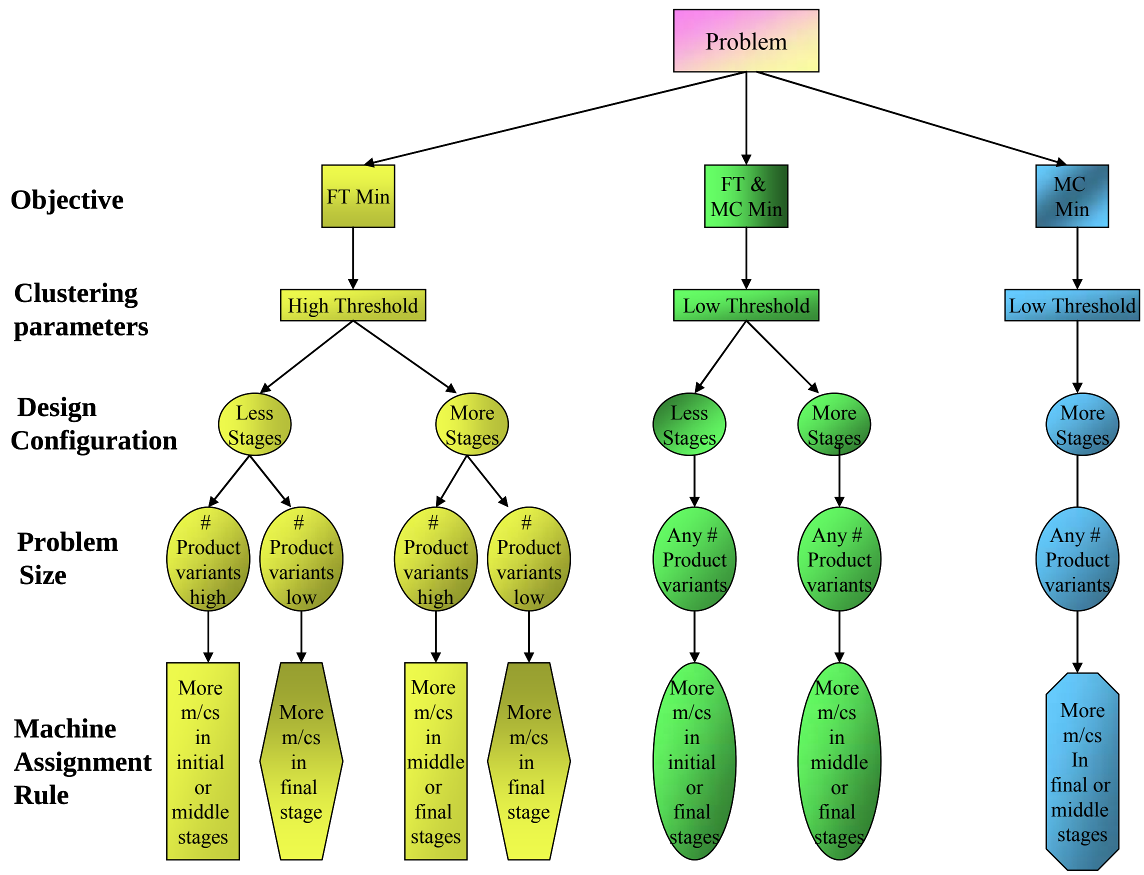


Figure 5-1 : Rules Extracted for Minicell Configuration Design

In the above figure, there could be three objectives to minicell design. The yellow colored sections indicate the path to achieve flow time minimization. Along green colored sections the steps to be followed in identifying minicell designs that optimize both flow time and machine capacity are given. Finally the blue colored section indicates steps that would lead to identify minicell designs that generate minimum machine capacity. The rectangles in the 'Machine Assignment Rule' row indicate the rules for minicell designs selection in large number of product variants, while hexagons give the rules for small number of product variants. Minicell designs in flow time and machine capacity minimization are chosen by following the rules in the green colored ovals. The rules to be followed for the selection of minicell designs in machine capacity minimization are given by the octagons.

From Figure 5-1, it is seen that for flow time minimization objective, difference in the number of product variants, varies the rules for minicell designs formation. Whereas irrespective of the number of product variants, the rules for minicell designs remains same for minimization of flow time and machine capacity and minimization of machine requirements objectives.

In comparison with the simultaneous method, the hierarchical approach produces lower average flow times for all problems. Also in 18, 12 and 8 product variants problems, the hierarchical method generated minimum makespan values albeit using flow time minimization as the objective, while in the 27 product variants problem, simultaneous method generated low makespan values when compared with hierarchical method. Hence it could be recommended that the simultaneous method be used to find minimum makespan values when the number of product variants in the problem product size is large. As seen from the above results, for flow time minimization, irrespective of the number of product variants in the problem, it is recommended to use hierarchical method.

5.2 Future Work

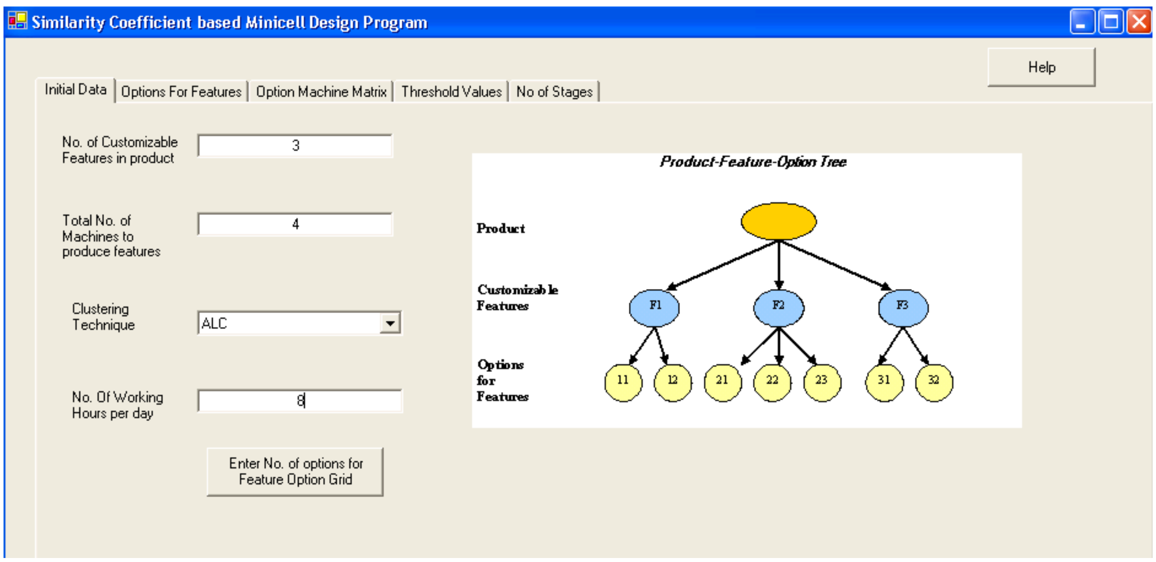
The software program developed for this research generates minicell designs in the design stage and schedules product variant demand in minicells subsequently. Both these tasks are done separately. Manual calculations are necessary to part-process data from the design stage to enter the processing times data for product variant demand in scheduling stage. Hence the software program can be further improved by combining both the designing and scheduling stages. This would eliminate the need for manual data processing and expedite the design and scheduling of minicells much more.

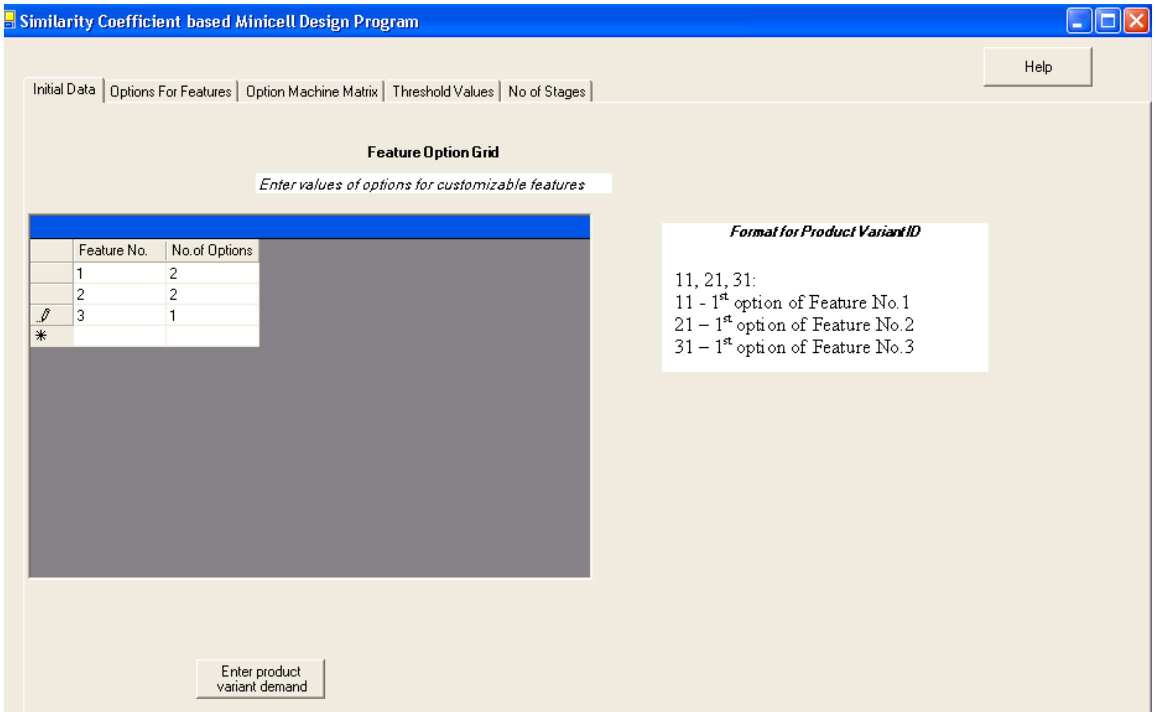
In this research, problems are tested by varying the number of machines and product variants. There are others parameters like number of customizable features, number of options available for each feature which can be varied and evaluate the performance of minicells, Also number of machines for each product variants problem can be varied and test the effect on minicell designs performance. In the present research, rules for minicell design for different product sizes have been established. These rules can be further checked with large sized problems and see if they are still applicable.

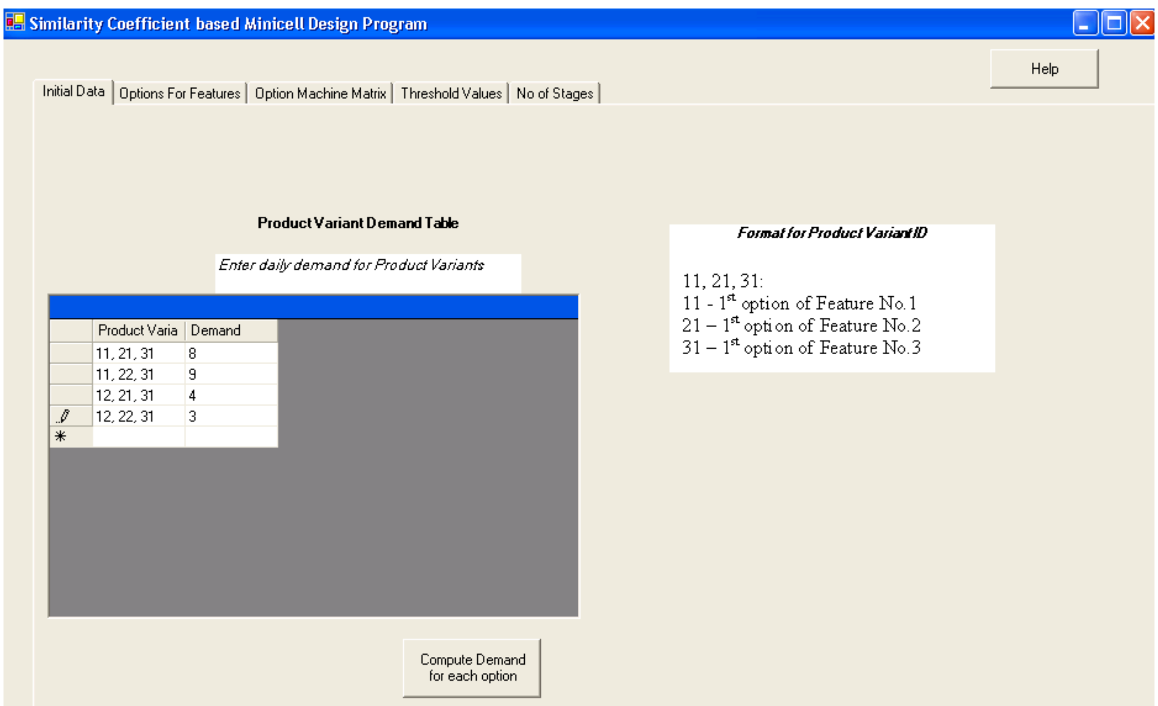
APPENDIX

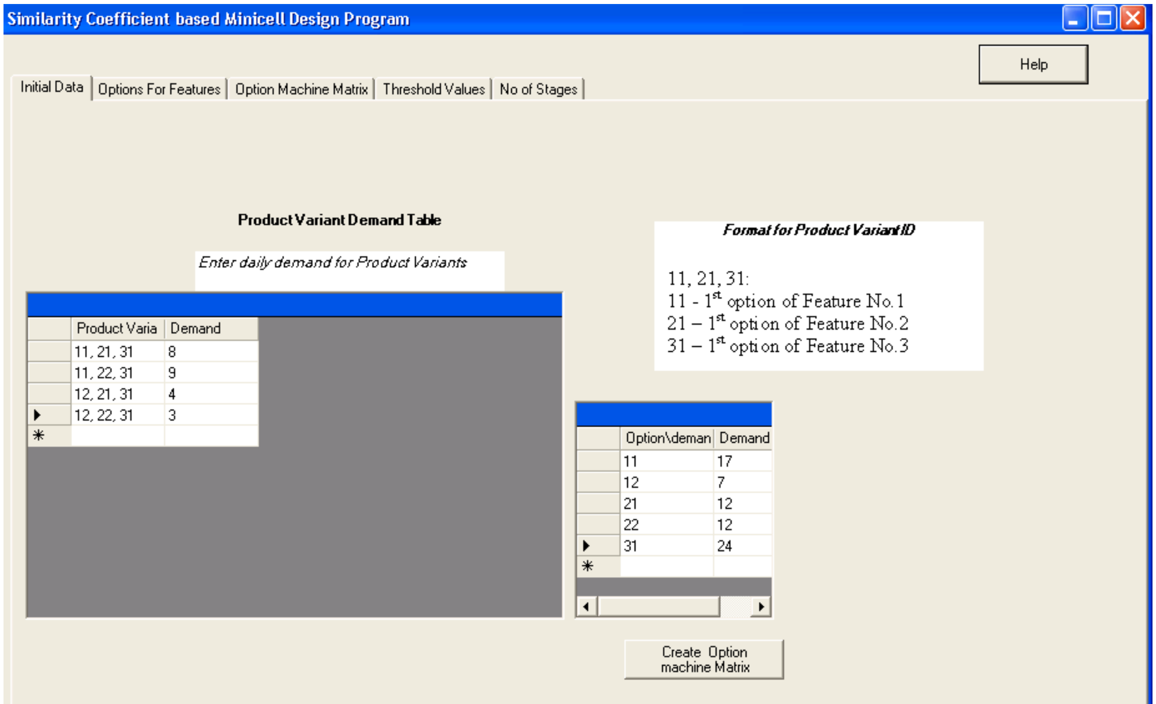
The screenshots of the example discussed in chapter 3 in designing and scheduling stages are given below.

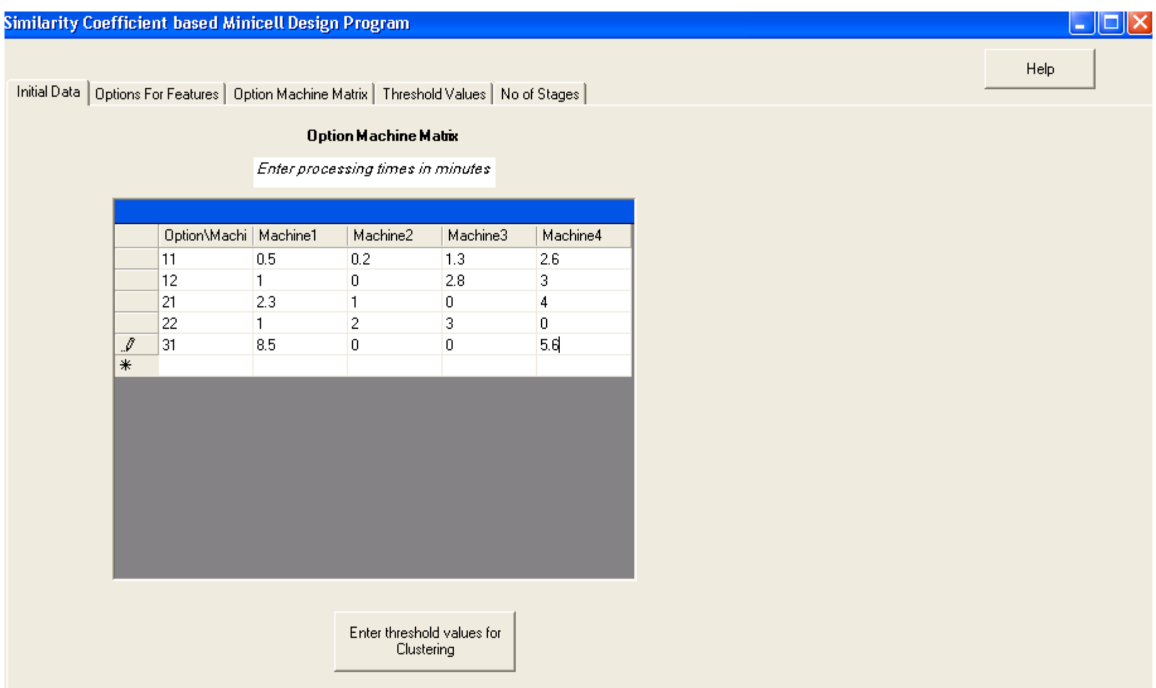
Minicell Design Stage

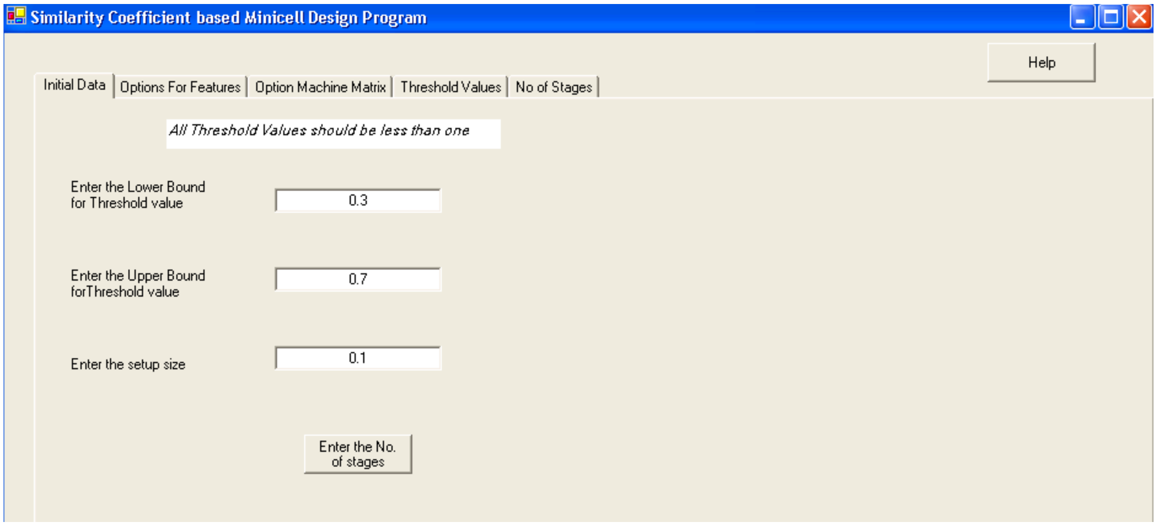


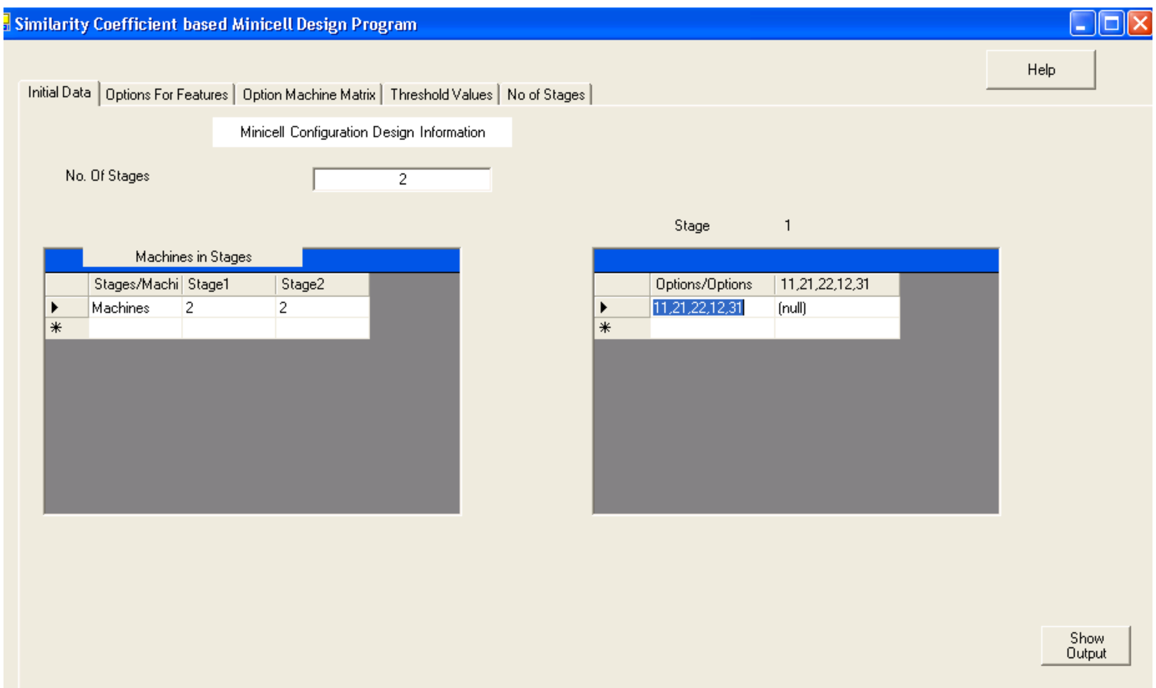

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Feature Option Grid
Enter values of options for customizable features
Feature No.
No.of Options
1
2
2
2
3
1
Format for Product Variant ID
11, 21, 31:
11 - 1st option of Feature No.1
21 – 1st option of Feature No.2
31 – 1st option of Feature No.3
Enter product variant demand

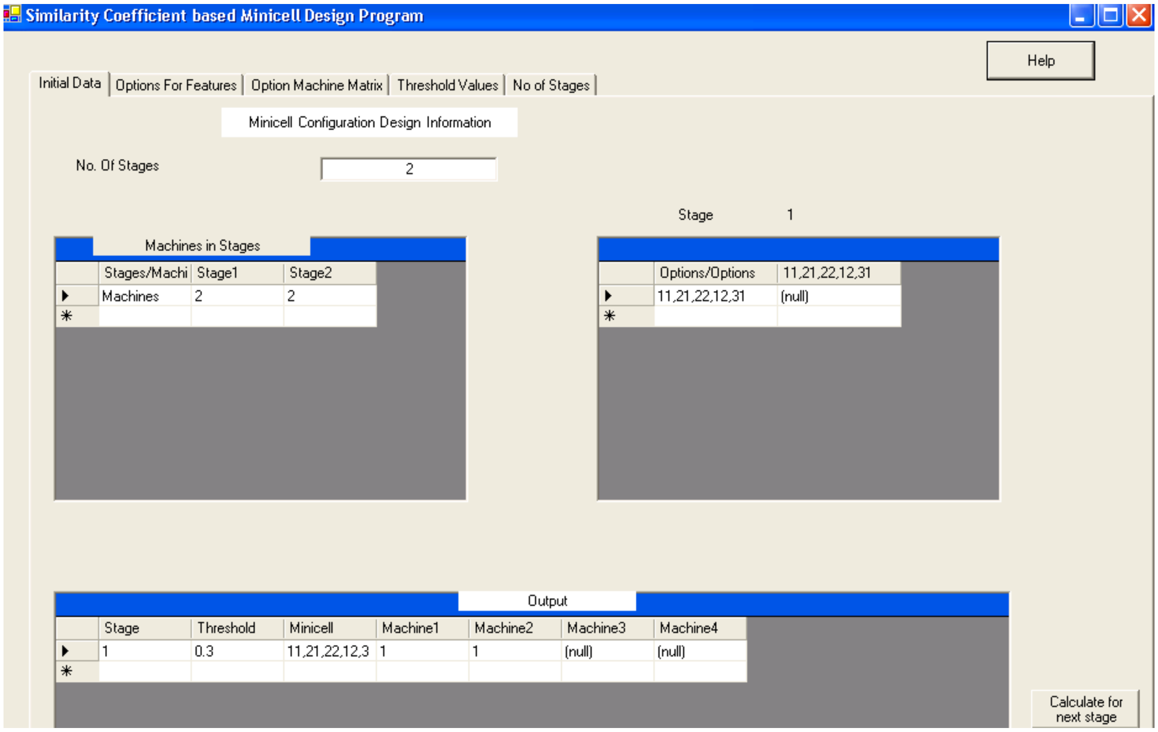

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Product Variant Demand Table
Enter daily demand for Product Variants
Product Varia
Demand
11, 21, 31
8
11, 22, 31
9
12, 21, 31
4
12, 22, 31
3
Format for Product Variant ID
11, 21, 31:
11 - 1st option of Feature No.1
21 – 1st option of Feature No.2
31 – 1st option of Feature No.3
Compute Demand for each option

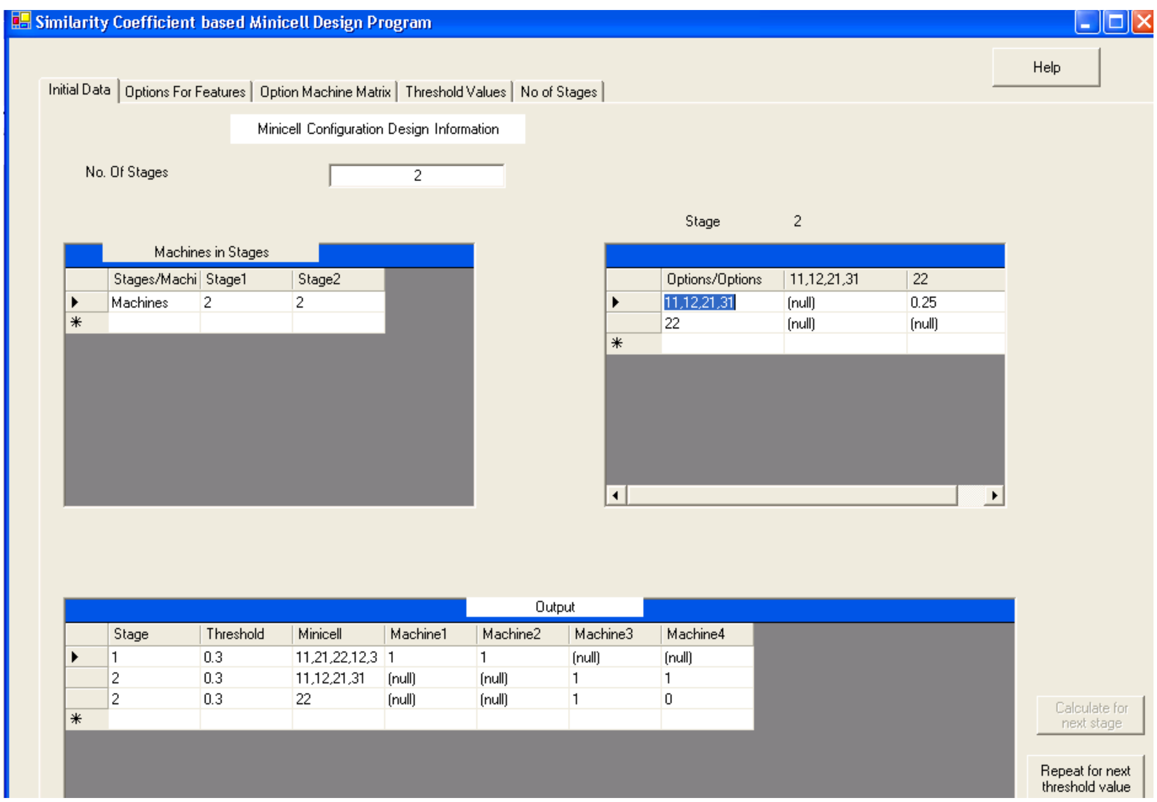

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Product Variant Demand Table
Enter daily demand for Product Variants
Product Varia
Demand
11, 21, 31
8
11, 22, 31
9
12, 21, 31
4
12, 22, 31
3
Format for Product Variant ID
11, 21, 31:
11 - 1st option of Feature No.1
21 – 1st option of Feature No.2
31 – 1st option of Feature No.3
Option\deman
Demand
11
17
12
7
21
12
22
12
31
24
Create Option machine Matrix

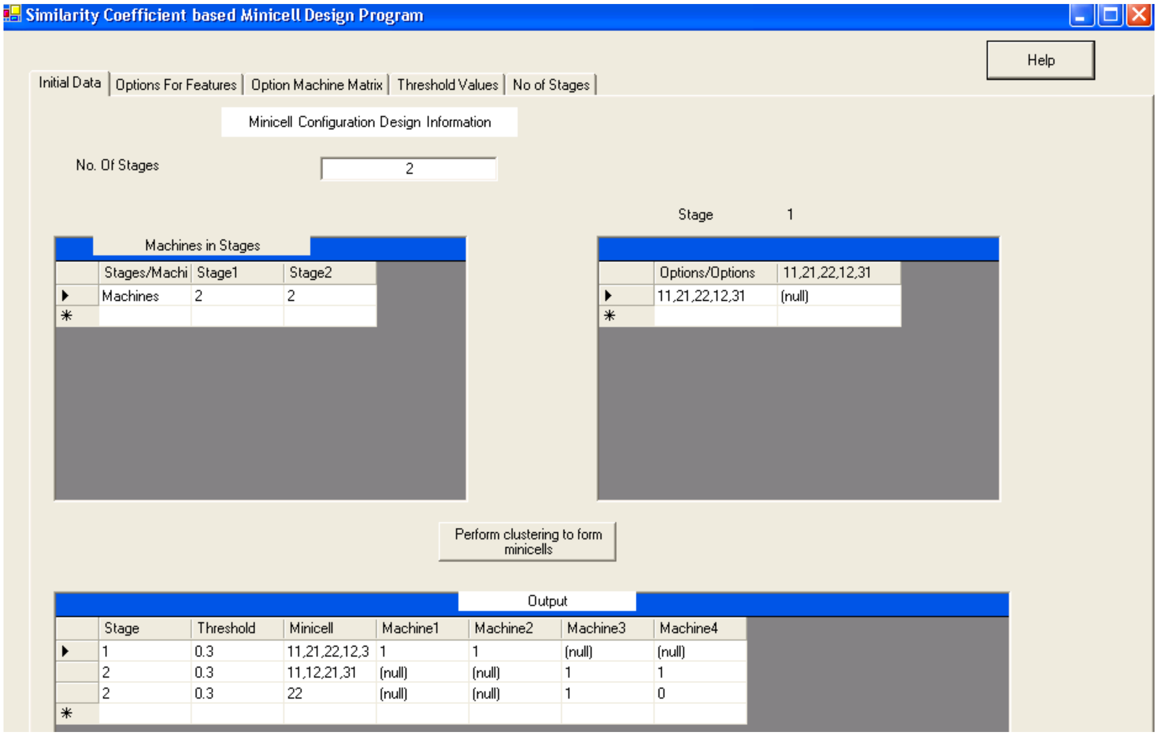

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Option Machine Matrix
Enter processing times in minutes
Option\Machi
Machine1
Machine2
Machine3
Machine4
11
0.5
0.2
1.3
2.6
12
1
0
2.8
3
21
2.3
1
0
4
22
1
2
3
0
31
8.5
0
0
5.6
Enter threshold values for Clustering

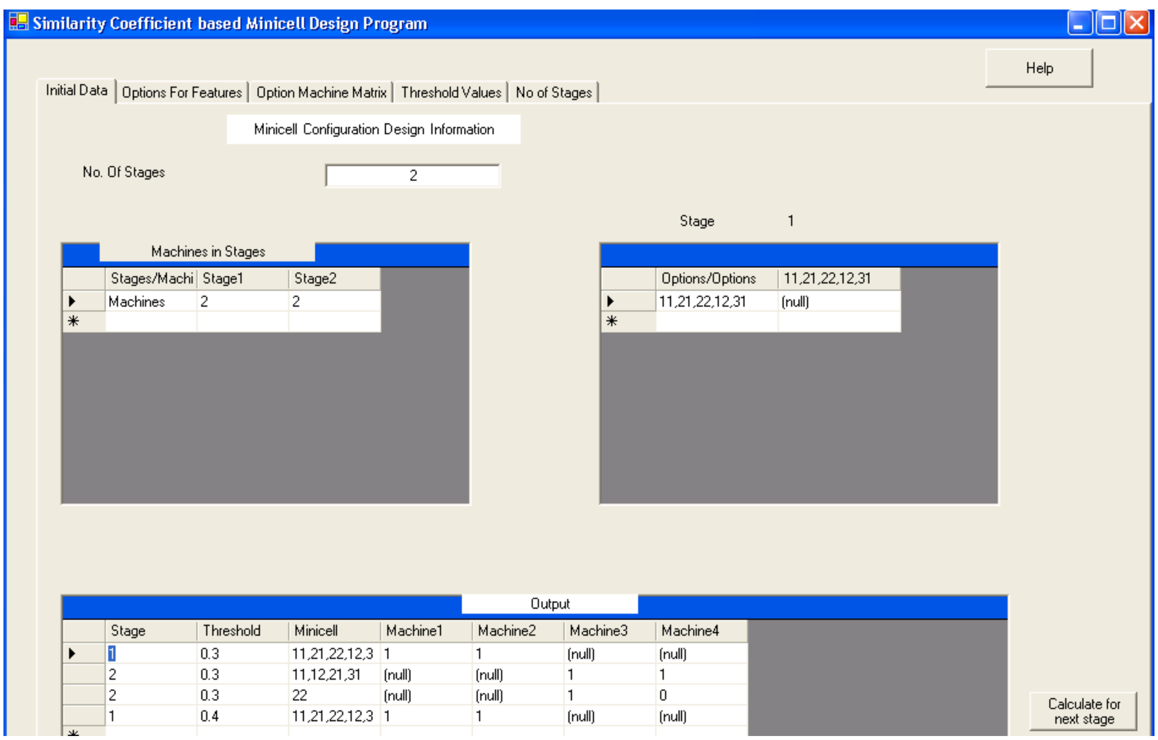

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
All Threshold Values should be less than one
Enter the Lower Bound for Threshold value
0.3
Enter the Upper Bound forThreshold value
0.7
Enter the setup size
0.1
Enter the No. of stages

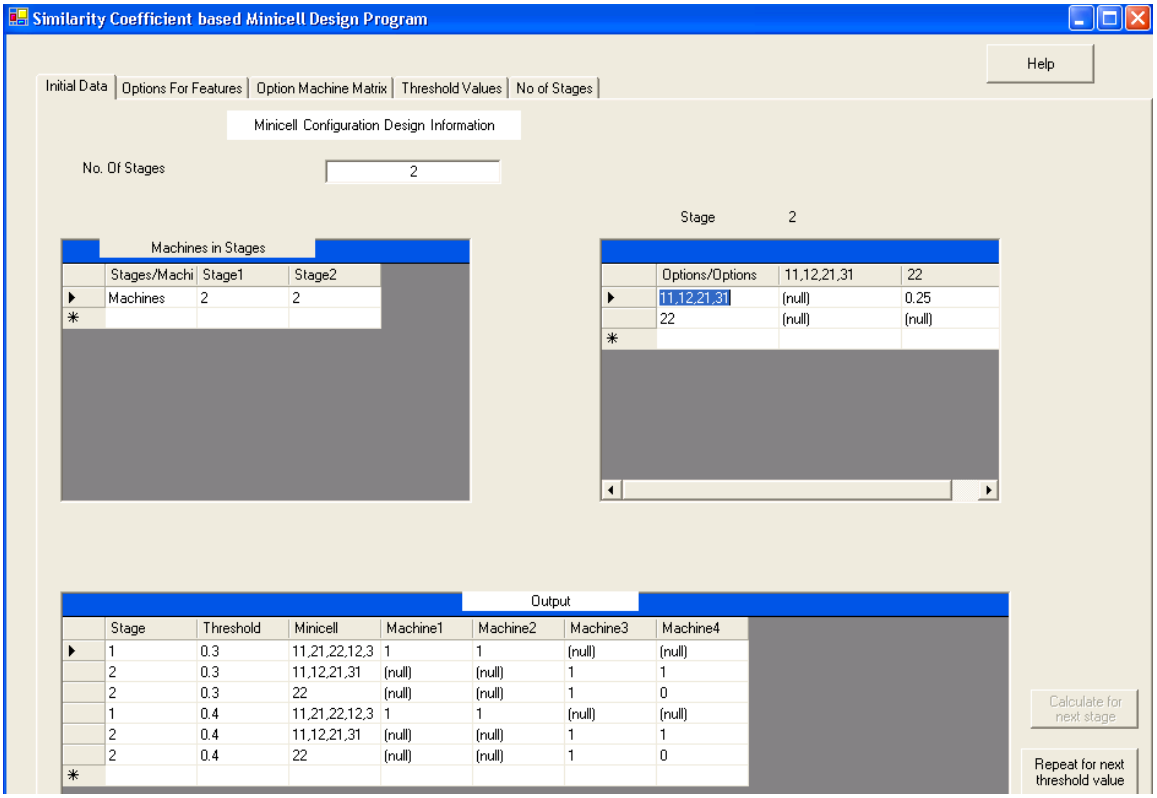

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage
1
Machines in Stages
Stages/Machi
Stage1
Stage2
Machines
2
2
Options/Options
11,21,22,12,31
11,21,22,12,31
(null)
Show Output

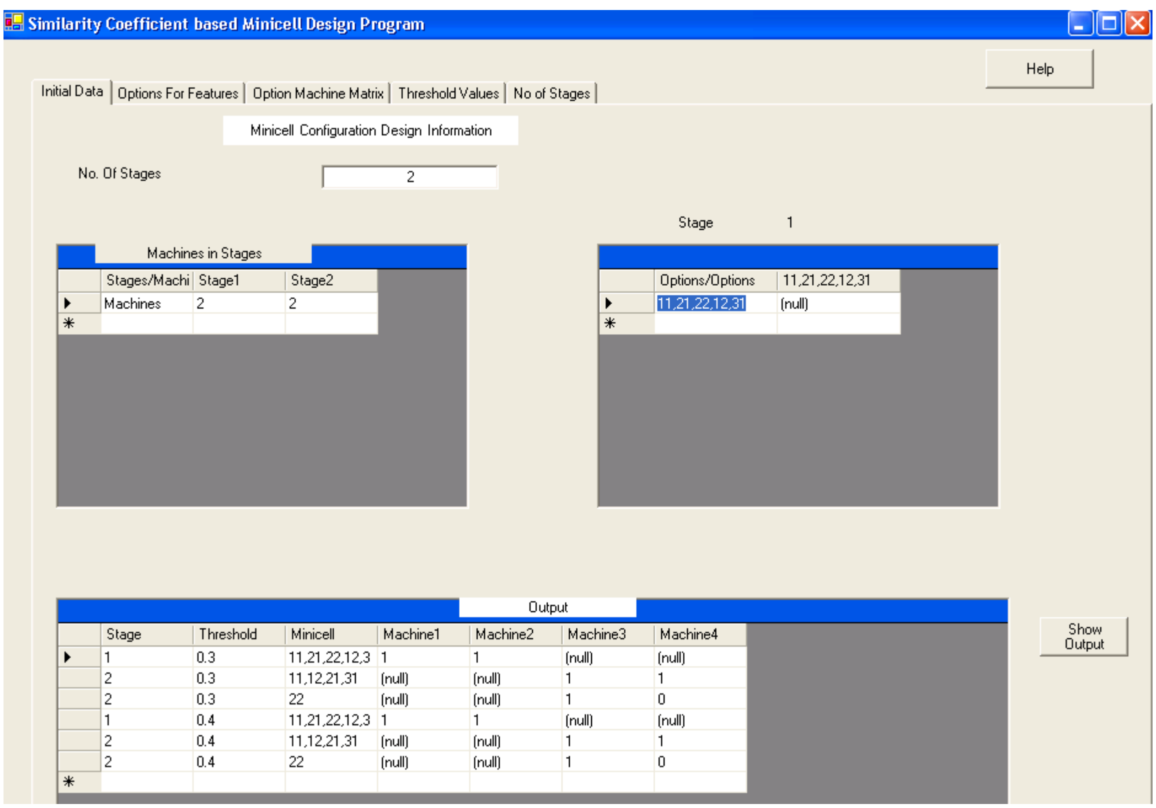

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 1
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,21,22,12,31
11,21,22,12,31 (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
Calculate for next stage

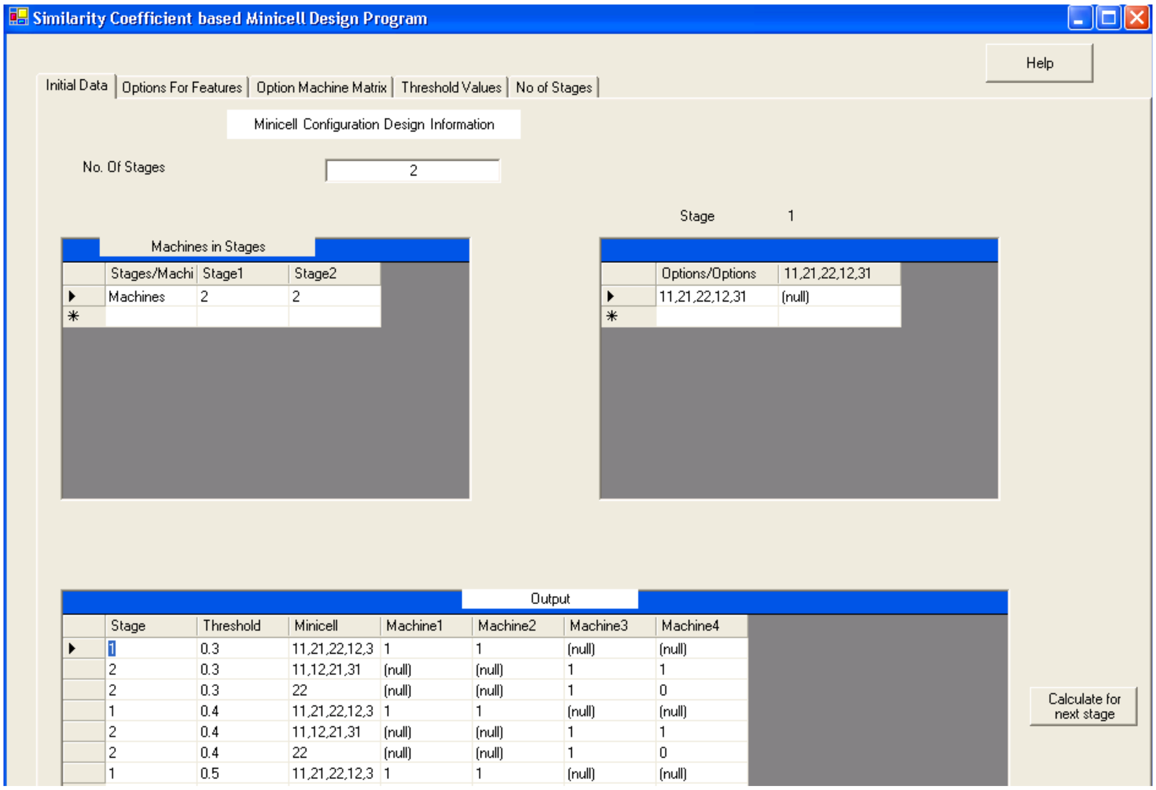

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 2
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,12,21,31 22
11,12,21,31 (null) 0.25
22 (null) (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0
Calculate for next stage
Repeat for next threshold value

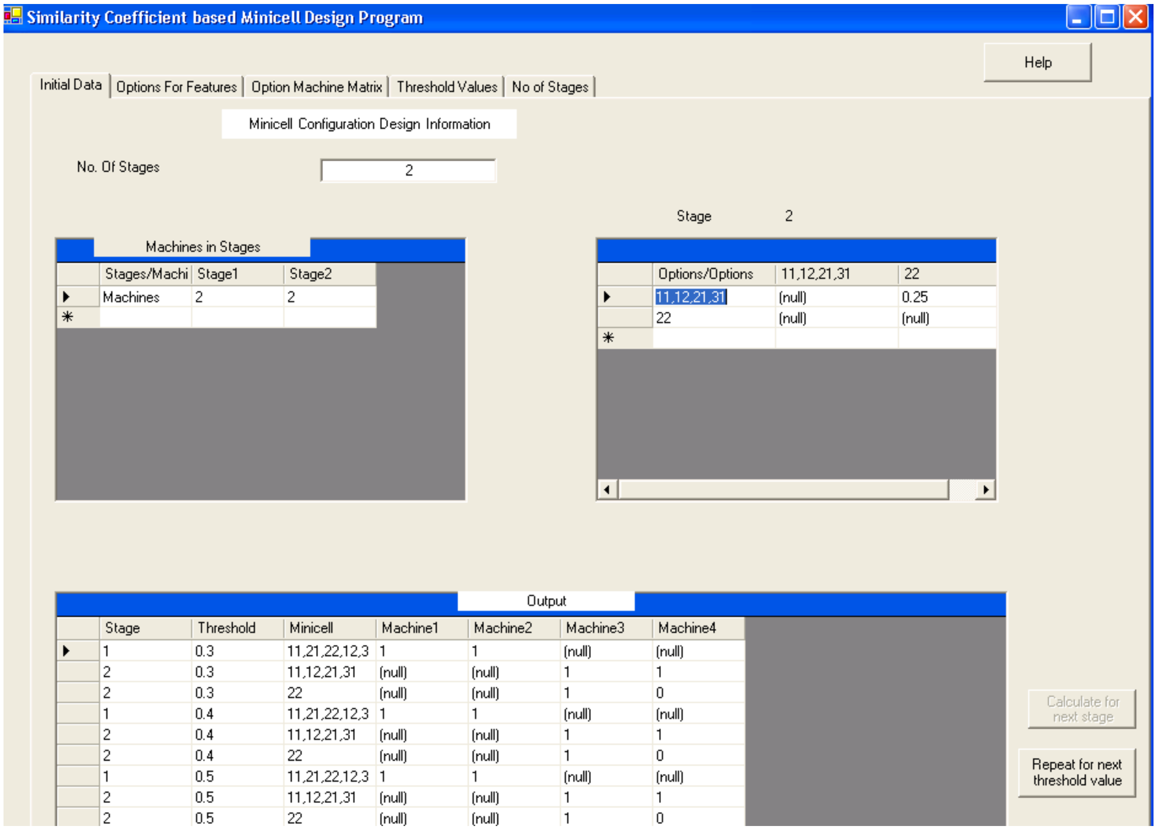

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 1
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,21,22,12,31
11,21,22,12,31 (null)
Perform clustering to form minicells
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0

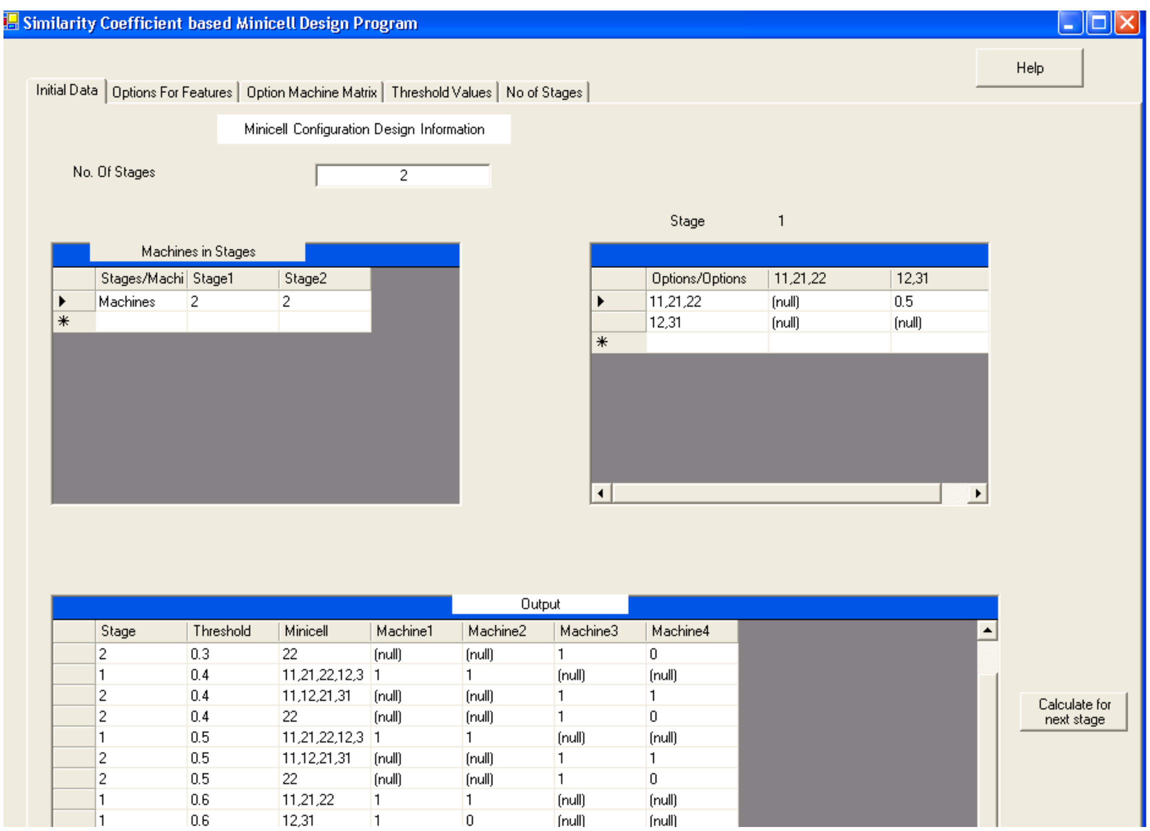

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 1
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,21,22,12,31
11,21,22,12,31 (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0
1 0.4 11,21,22,12,3 1 1 (null) (null)
Calculate for next stage

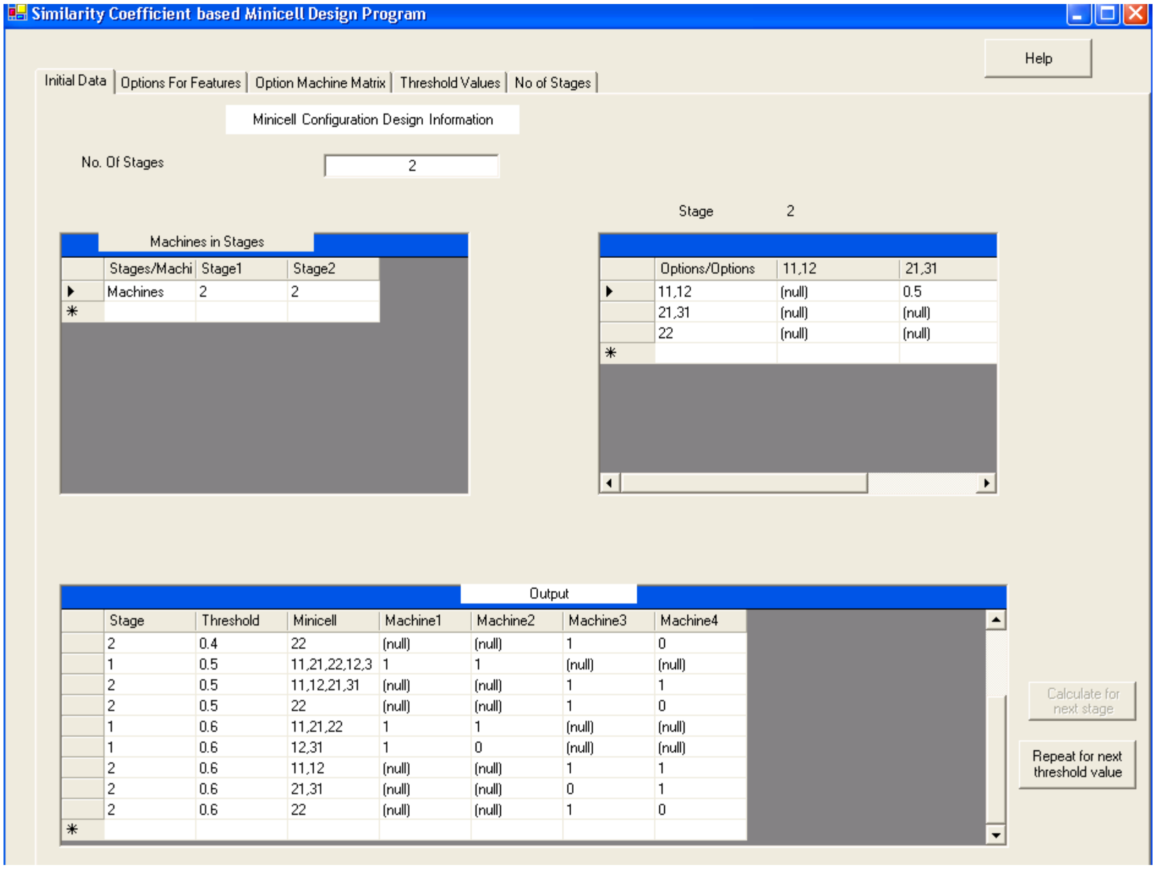

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 2
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,12,21,31 22
11,12,21,31 (null) 0.25
22 (null) (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0
1 0.4 11,21,22,12,3 1 1 (null) (null)
2 0.4 11,12,21,31 (null) (null) 1 1
2 0.4 22 (null) (null) 1 0
Calculate for next stage
Repeat for next threshold value

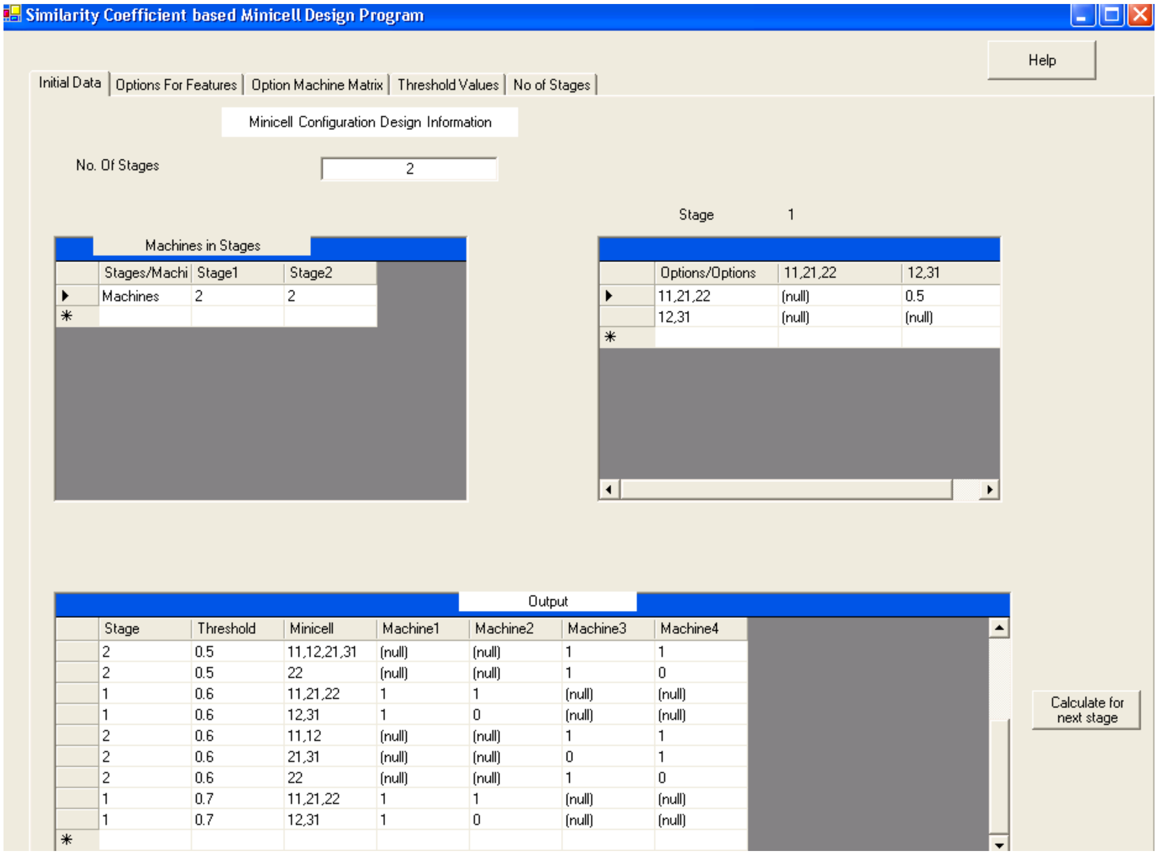

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 1
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,21,22,12,31
11,21,22,12,31 (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0
1 0.4 11,21,22,12,3 1 1 (null) (null)
2 0.4 11,12,21,31 (null) (null) 1 1
2 0.4 22 (null) (null) 1 0
Show Output

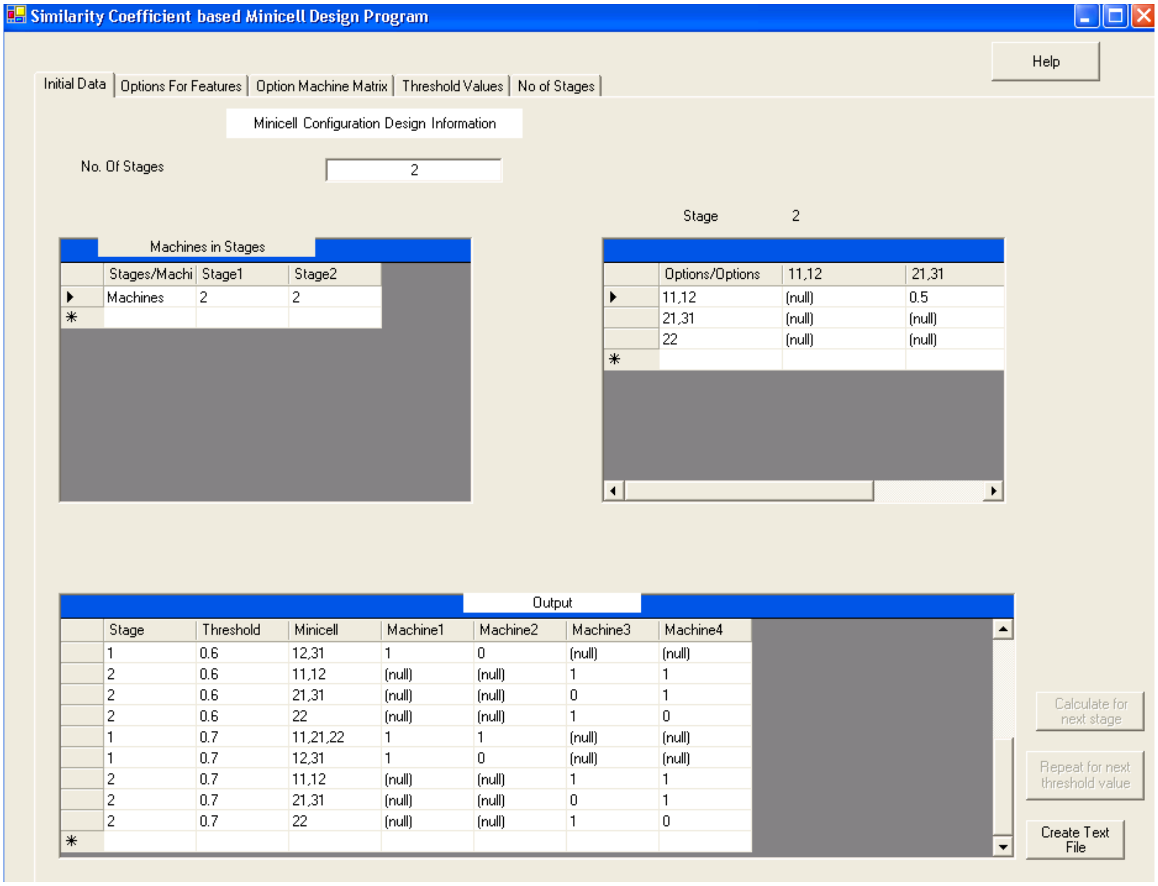

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 1
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,21,22,12,31
11,21,22,12,31 (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0
1 0.4 11,21,22,12,3 1 1 (null) (null)
2 0.4 11,12,21,31 (null) (null) 1 1
2 0.4 22 (null) (null) 1 0
1 0.5 11,21,22,12,3 1 1 (null) (null)
Calculate for next stage


Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage
2
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,12,21,31 22
11,12,21,31 (null) 0.25
22 (null) (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.3 11,21,22,12,3 1 1 (null) (null)
2 0.3 11,12,21,31 (null) (null) 1 1
2 0.3 22 (null) (null) 1 0
1 0.4 11,21,22,12,3 1 1 (null) (null)
2 0.4 11,12,21,31 (null) (null) 1 1
2 0.4 22 (null) (null) 1 0
1 0.5 11,21,22,12,3 1 1 (null) (null)
2 0.5 11,12,21,31 (null) (null) 1 1
2 0.5 22 (null) (null) 1 0
Calculate for next stage
Repeat for next threshold value

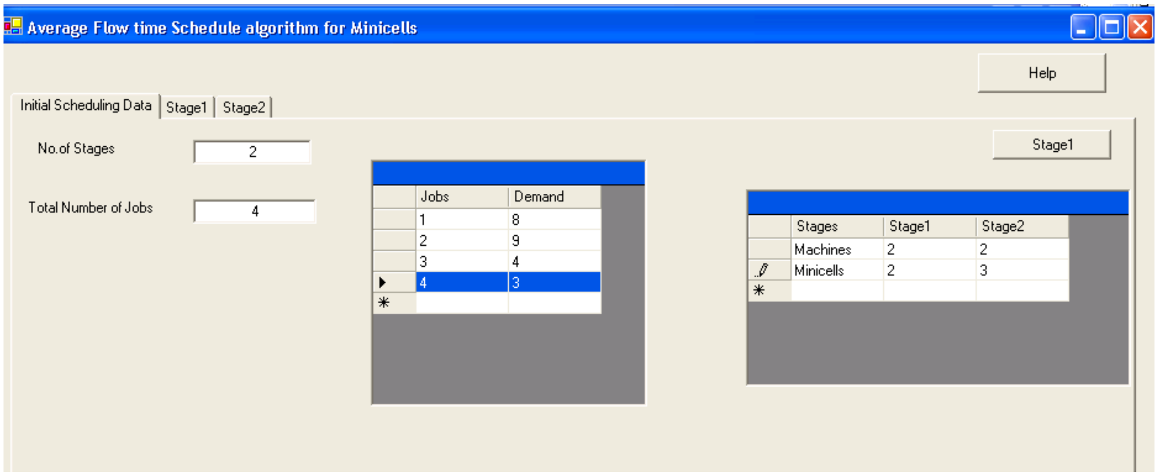

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage
1
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,21,22 12,31
11,21,22 (null) 0.5
12,31 (null) (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
2 0.3 22 (null) (null) 1 0
1 0.4 11,21,22,12,3 1 1 (null) (null)
2 0.4 11,12,21,31 (null) (null) 1 1
2 0.4 22 (null) (null) 1 0
1 0.5 11,21,22,12,3 1 1 (null) (null)
2 0.5 11,12,21,31 (null) (null) 1 1
2 0.5 22 (null) (null) 1 0
1 0.6 11,21,22 1 1 (null) (null)
1 0.6 12,31 1 0 (null) (null)
Calculate for next stage

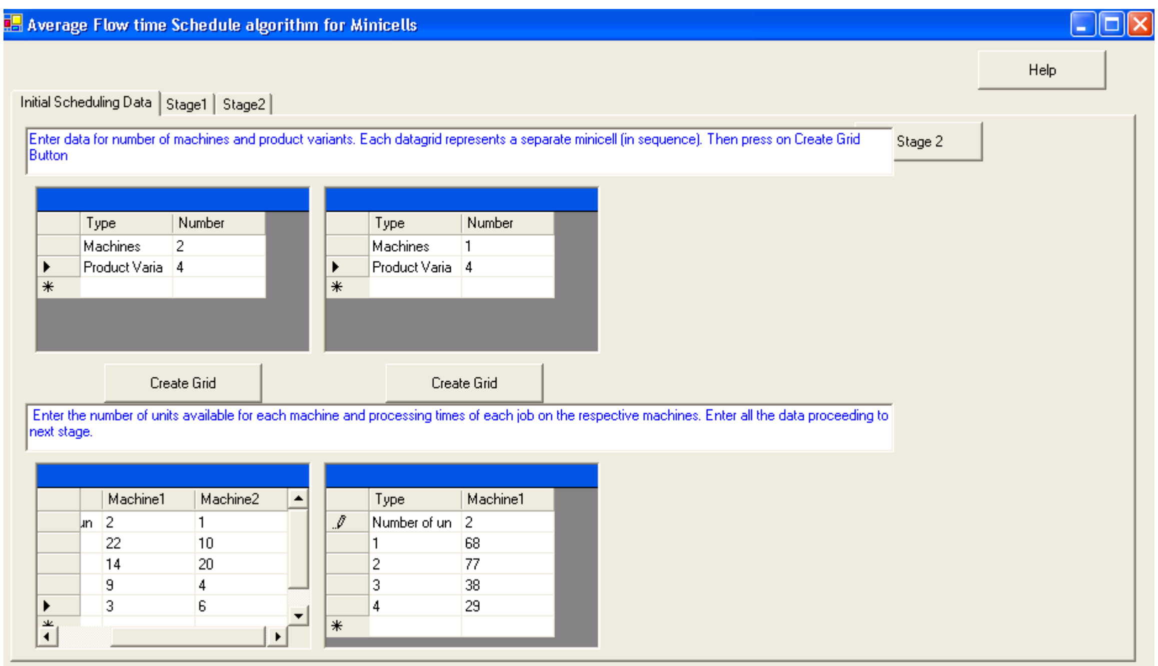

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage 2
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,12 21,31
11,12 (null) 0.5
21,31 (null) (null)
22 (null) (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
2 0.4 22 (null) (null) 1 0
1 0.5 11,21,22,12,3 1 1 (null) (null)
2 0.5 11,12,21,31 (null) (null) 1 1
2 0.5 22 (null) (null) 1 0
1 0.6 11,21,22 1 1 (null) (null)
1 0.6 12,31 1 0 (null) (null)
2 0.6 11,12 (null) (null) 1 1
2 0.6 21,31 (null) (null) 0 1
2 0.6 22 (null) (null) 1 0
Calculate for next stage
Repeat for next threshold value

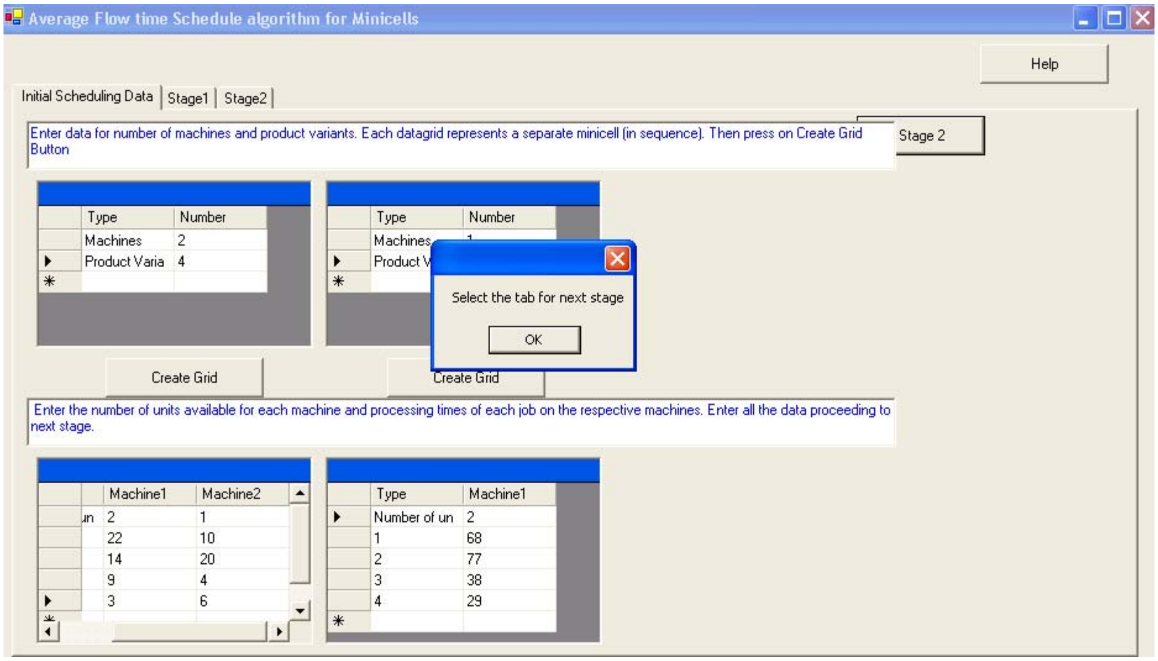

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage
1
Machines in Stages
Stages/Machi
Stage1
Stage2
Machines
2
2
Options/Options
11,21,22
12,31
11,21,22
(null)
0.5
12,31
(null)
(null)
Output
Stage
Threshold
Minicell
Machine1
Machine2
Machine3
Machine4
2 0.5 11,12,21,31 (null) (null) 1 1
2 0.5 22 (null) (null) 1 0
1 0.6 11,21,22 1 1 (null) (null)
1 0.6 12,31 1 0 (null) (null)
2 0.6 11,12 (null) (null) 1 1
2 0.6 21,31 (null) (null) 0 1
2 0.6 22 (null) (null) 1 0
1 0.7 11,21,22 1 1 (null) (null)
1 0.7 12,31 1 0 (null) (null)
Calculate for next stage

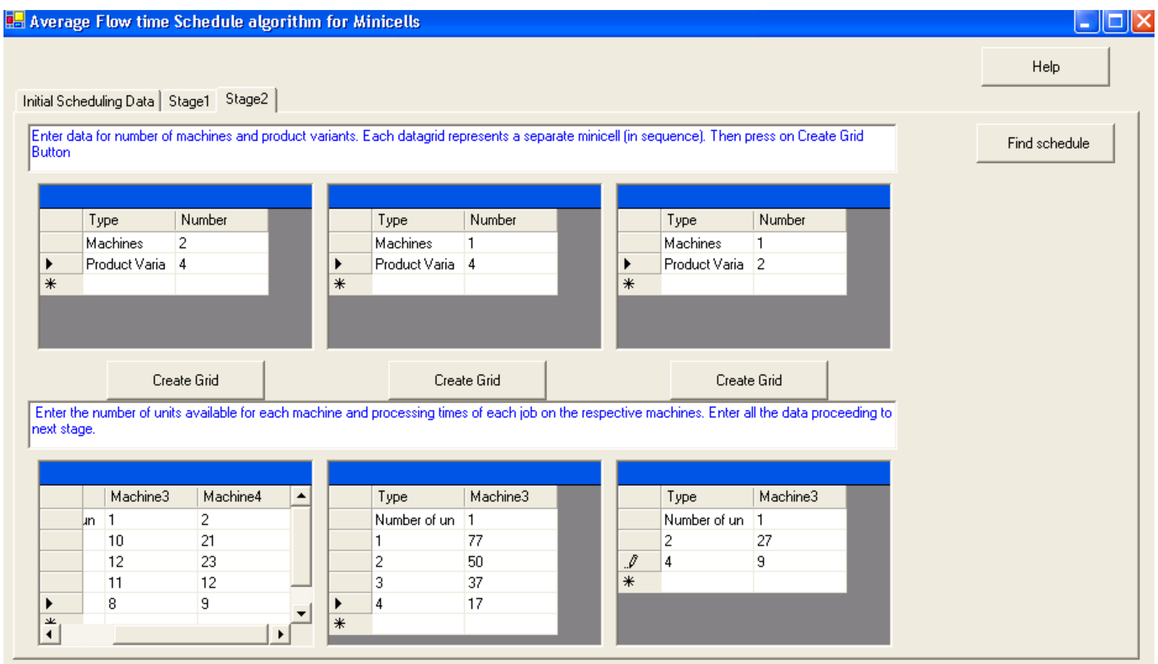

Similarity Coefficient based Minicell Design Program
Help
Initial Data
Options For Features
Option Machine Matrix
Threshold Values
No of Stages
Minicell Configuration Design Information
No. Of Stages
2
Stage
2
Machines in Stages
Stages/Machi Stage1 Stage2
Machines 2 2
Options/Options 11,12 21,31
11,12 (null) 0.5
21,31 (null) (null)
22 (null) (null)
Output
Stage Threshold Minicell Machine1 Machine2 Machine3 Machine4
1 0.6 12,31 1 0 (null) (null)
2 0.6 11,12 (null) (null) 1 1
2 0.6 21,31 (null) (null) 0 1
2 0.6 22 (null) (null) 1 0
1 0.7 11,21,22 1 1 (null) (null)
1 0.7 12,31 1 0 (null) (null)
2 0.7 11,12 (null) (null) 1 1
2 0.7 21,31 (null) (null) 0 1
2 0.7 22 (null) (null) 1 0
Calculate for next stage
Repeat for next threshold value
Create Text File

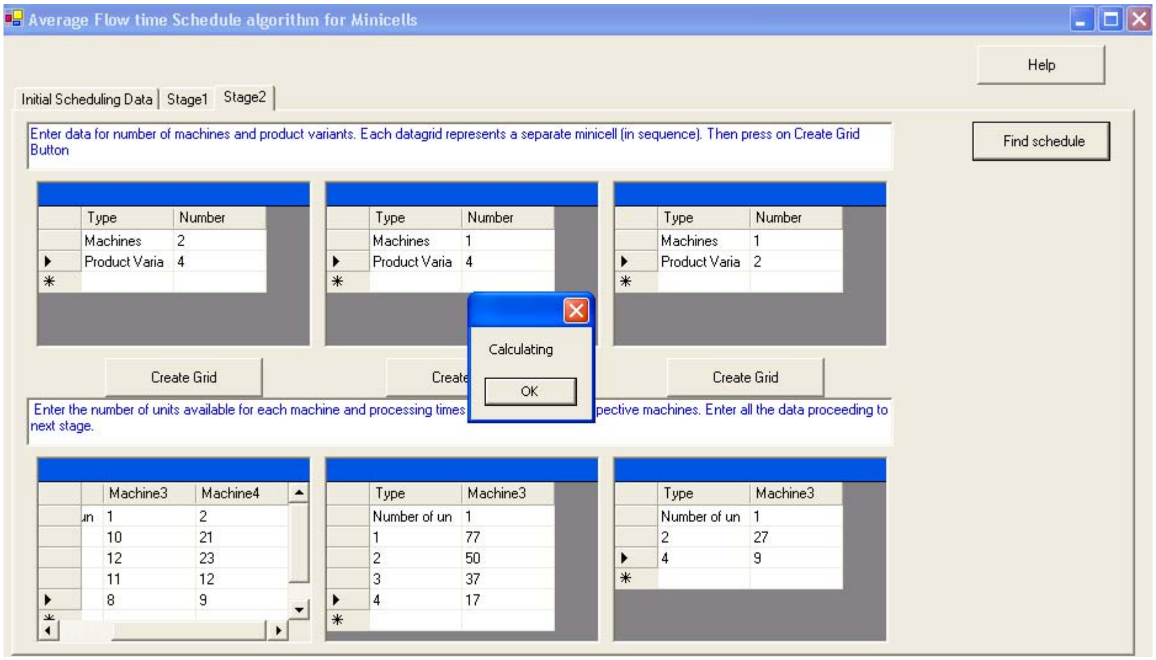
Scheduling Stage



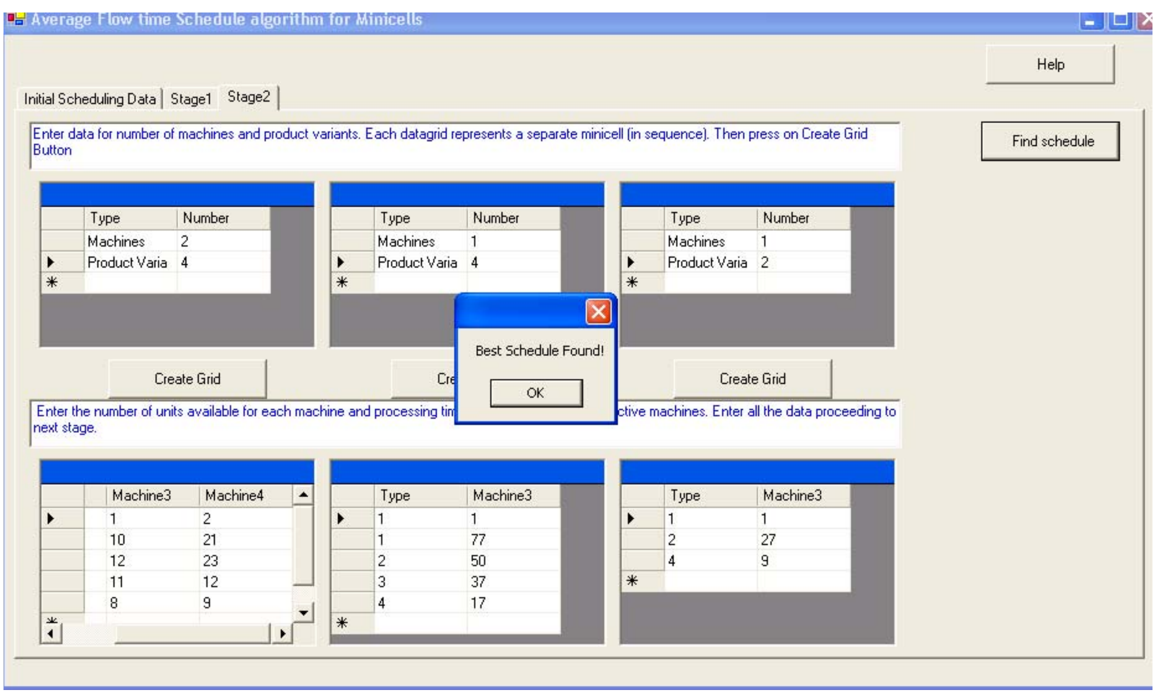



Average Flow time Schedule algorithm for Minicells
Help
Initial Scheduling Data
Stage1
Stage2
Stage 2
Enter data for number of machines and product variants. Each datagrid represents a separate minicell (in sequence). Then press on Create Grid Button
Type
Number
Machines
2
Product Varia
4
Select the tab for next stage
OK
Create Grid
Create Grid
Enter the number of units available for each machine and processing times of each job on the respective machines. Enter all the data proceeding to next stage.
Machine1
Machine2
Number of un


Average Flow time Schedule algorithm for Minicells
Help
Initial Scheduling Data
Stage1
Stage2
Find schedule
Enter data for number of machines and product variants. Each datagrid represents a separate minicell (in sequence). Then press on Create Grid Button
Type
Number
Machines
Product Varia
Create Grid
Create Grid
Create Grid
Enter the number of units available for each machine and processing times of each job on the respective machines. Enter all the data proceeding to next stage.
Machine3
Machine4
Number of un



Average Flow time Schedule algorithm for Minicells
Help
Initial Scheduling Data
Stage1
Stage2
Find schedule
Enter data for number of machines and product variants. Each datagrid represents a separate minicell (in sequence). Then press on Create Grid Button
Calculating
OK
Create Grid
Create Grid
Enter the number of units available for each machine and processing times
pective machines. Enter all the data proceeding to next stage.



Average Flow time Schedule algorithm for Minicells
Help
Initial Scheduling Data
Stage1
Stage2
Find schedule
Enter data for number of machines and product variants. Each datagrid represents a separate minicell (in sequence). Then press on Create Grid Button
Best Schedule Found!
OK
Create Grid
Create Grid
Enter the number of units available for each machine and processing tim
ctive machines. Enter all the data proceeding to next stage.

Product Variants Demand for Tested problems

The product variant demand used in the four tested problems for calculating machine requirements and average flow times are given below.

a) *27 Product Variants*

In this problem, same demand values are used for machine capacity and average flow time calculations. The demand values are shown below.

Product Variant	Demand
1	8
2	3
3	3
4	2
5	4
6	3
7	7
8	8
9	7
10	5
11	2
12	2
13	10
14	4
15	1
16	10
17	8
18	9
19	10
20	2
21	5
22	3
23	4
24	2
25	6
26	6
27	4

b) *12 Product Variants*

In this problem also, same demand values were used for machine capacity and average flow time calculations. The demand table can be seen below.

Product Variants	Demand
1	8
2	5
3	5
4	12
5	4
6	13
7	5
8	2
9	8
10	10
11	14
12	5

c) *18 Product Variants*

This problem was tested by assigning separate demand values for machine capacity and average flow time calculations. The demand table is given below.

Product Varaints	Demand for Machine Capacity	Demand for Average Flow Time
1	15	10
2	30	28
3	22	19
4	45	47
5	12	10
6	32	35
7	28	22
8	25	37
9	28	17
10	25	30
11	20	20
12	8	9
13	15	19
14	12	8
15	2	2
16	10	8
17	12	8
18	14	13

d) *8 Product Variants*

This problem was also tested by varying the demand assignments to machine capacity and average flow time calculations. The demand tabled used in the calculations is given below.

Product Variants	Demand for Machine Capacity	Demand for Average Flow time
1	12	13
2	3	3
3	6	7
4	2	2
5	10	11
6	8	10
7	8	6
8	6	7

REFERENCES

1. Allahverdi, A., Aldowaisan, T., (2002), "New heuristics to minimize total completion time in m-machine flow shops", *International Journal of Production Economics*, Vol. 77, pp. 71-83.
2. Askin, R. G., Creswell, S. H., Goldberg, J. B. and Vakharia, A. J., (1991), "Hamiltonian path approach to reordering part-machine matrix for cellular manufacturing", *International Journal of Production Research*, Vol. 29, pp. 1081-1100.
3. Babiarz, P., Piotrowski, M., Pomianek, B., and Wawrzynkiewicz, P., "Prerequisites for successful introduction of mass customization strategy in central Europe" , http://www.robertfreund.de/download/ Maciej_Piotrowski %20Ohrid%202004.pdf %20Ohrid%202004.pdf, accessed on 6/21/2007
4. Badurdeen, F. F., and Masel, D. T., and Thuramalla, S., (2006), "A Modular Minicell Configuration for Mass Customization Manufacturing", Group Technology & Cellular Manufacturing International Conference, July 2-5, Groningen, Netherlands.
5. Badurdeen, F., Masel, D. and Thuramalla, S. (2007), "Increasing Mass Customization Manufacturing Flexibility using Minicells", 40th CIRP International Conference, May 30-June1, Liverpool, England.
6. Badurdeen, F.F., (2005), "Minicell Configuration for Mass Customization Manufacturing", PhD Dissertation, Ohio University, Athens, Ohio, USA.
7. Badurdeen, F.F., Class notes, ME 512: Manufacturing Systems, Fall 2005 semester, University of Kentucky, USA.
8. Badurdeen, F.F., Masel, D. and Thuramalla, S. and (2006), "Multi-Objective Algorithm to Design Minicells for Mass Customization Manufacturing", Industrial Engineering & Research Conference, May 20-24, Orlando, Florida, USA.
9. Baker, K. R., (1974), "Introduction to Sequencing and Scheduling", Wiley, New York.
10. Baldwin, C.Y., and Clark, K.B., (1997), "Managing in an age of modularity",

Harvard Business Review, Sep/Oct97, Vol. 75, Issue 5, pp. 84–93.

11. Berman, B., (2002), “Should your firm adopt a mass customization strategy?”, Business Horizons, Vol. 45, Issue 4 , July-August 2002, pp. 51-60.
12. Blecker, T., Graf, G.(2003), “Multi-Agent Systems in Internet based Production Environments-an enabling infrastructure for Mass Customization”, Proceedings of the 2nd Interdisciplinary world congress on mass customization and personalization MCPC, Munich, October 6-8,S 1-21
13. Burbridge, J.L., (1963), “Production Flow Analysis”, *Production Engineer*, December, pp. 742-752.
14. Campbell, H.G., Dudek, R.A., and Smith, M.L., (1970), “A Heuristic Algorithm for the n Job m Machine Sequencing Problem”, *Management Science*, Vol. 16 Issue 10, pp. 630- 637.
15. Carrie, A. S., (1973), “Numerical taxonomy applied to group technology and plant layout, *International Journal of Production Research*, Vol. 11, pp. 399-416.
16. Chadalavada, P.K., (2006), “Minimum flow time schedule genetic algorithm for mass customization manufacturing using minicells”, Masters Thesis, University Of Kentucky, Kentucky, USA.
17. Chan, H. M. and Milner, D. A.,(1982), “ Direct clustering algorithm for group formation in cellular manufacturing”, *Journal of Manufacturing Systems*, Vol. 1, No. 1, pp. 65-74.
18. Chandrasekharan, M. P. and Rajagopalan, R., (1986), “An ideal seed non-hierarchical clustering algorithm for cellular manufacturing”, *International Journal of Production Research*, Vol. 24, No.2, pp. 451-464.
19. Chandrasekharan, M. P. and Rajagopalan, R., (1986), “MODROC: an extension of rank order clustering for group technology”, *International Journal of Production Research*, Vol. 24,No. 2, pp. 1221-1233
20. Chandrasekharan, M. P. and Rajagopalan, R., (1987), “ZODIAC - An algorithm for concurrent formation of part families and machine cells”, *International Journal of Production Research*, Vol.25, No.6, pp. 835-850.
21. Choobineh, F., (1988), “A framework for the design of cellular manufacturing

systems", *International Journal of Production Research*, Vol. 26, pp. 1161-1172.

22. Chu, C.H., (1989), "Clustering analysis in manufacturing cell formation", Omega, *International Journal of Management Science*, Vol. 17, pp. 289-295.
23. Dannenbring, D.G., (1977), "An evaluation of flow shop sequencing heuristics", *Management Science*, Vol. 23, No.1, July 1977, pp. 1174-1182.
24. Davis, S. M., (1987), "Future Perfect", Addison-Wesley publishing company.
25. Drolet, J, Abdulnour, G and Rheault, M, (1996), "The cellular manufacturing evolution", *Computers & Industrial Engineering*, Vol.31, No. 1-2, pp. 139-42.
26. Duray, R., (1997), "Mass customization Configurations: An empirical investigation of manufacturing practices of customization", Dissertation, The Ohio State University.
27. Duray, R., (2004), "Mass customizers' use of inventory, planning techniques and channel management", *Production Planning & Control*, Vol. 15, No.4, June 2004, pp. 412-421.
28. Duray, R., Ward, P.T., Milligan G.W., Berry, L.W (2000), "Approaches to mass customization: configurations and empirical validation", *Journal of Operations Management*, Issue 18, pp. 605–625.
29. EI-Essawy, I. F. K. and Torrance, J., (1972), "Component flow analysis- an effective approach to production systems' design", *Production Engineer*, Vol.51, pp. 165.
30. Feitzinger, E. and Lee, H.L., (1997), "Mass Customization at Hewlett-Packard: The Power of Postponement", *Harvard Business Review*, January-February, pp.116-121.
31. Feitzinger, E., Lee, H. L., (1997), "Mass Customization at Hewlett-Packard: The Power of Postponement", *Harvard Business Review*, Jan/Feb97, Vol. 75, Issue 1, pp.116-121.
32. Framinan, J. M., Leisten, R. and Rajendran C. ,(2003), "Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idle time or flow time in the static permutation flow shop sequencing problem", *International Journal of Production Research*, Vol. 41, No.1,pp. 121-148.

33. Framinan, J.M and Leisten R, (2003), “An efficient constructive heuristic for flow time minimization in permutation flow shops”, *OMEGA*, 31, pp. 311-317.
34. Framinan, J.M., Leisten, R. and Usano R.R., (2002), “Efficient heuristic for flowshop sequencing with the objectives of makespan and flow time minimisation”, *European Journal of Operational Research*, Vol. 141, pp. 559-569.
35. Framinan, J.M., Usano, R.R, and Leisten R, (2005), “Comparison of heuristics for flow time minimization in permutation flow shops”, *Computers & Operations Research*, Vol.32, No. 5, pp. 1237–1254.
36. Garey, M.R., and Johnson, D.S., (1976), “Strong NP-Completeness Results: Motivation, Examples, and Implications”, *J. Assoc. Comp. Mach.*, 25 (1978), pp. 499-508.
37. Garey, M.R., Johnson, D.S., and Sethi, R., (1976), “The complexity of flow shop and job shop scheduling”, *Mathematics of Operations Research*, 1, pp. 117-129.
38. Gelders, L.F. and Sambandam, N., (1978), “Four simple heuristics for scheduling a flow shop", *International Journal of Production Research*, Vol. 16, pp.221- 231.
39. Gilmore, J.H. and Pine, B.J., (1997), “Four Faces of Mass Customization”, *Harvard Business Review*, Jan-Feb 1997, pp. 91-101.
40. Greene, T.J., and Sadowski, R.P., (1983), "Cellular Manufacturing Control”, *Journal of Manufacturing Systems*, Vol. 2, No. 2, pp. 137-146.
41. Gupta, J.N.D., (1971), “A functional heuristic algorithm for the flow-shop scheduling problem”, Operational Research Quarterly, Vol. 22, pp.39-47.
42. Gupta, J.N.D., (1972), “Heuristic algorithms for multi-stage flow shop scheduling problem”, AIIE Transactions, Vol. 4, pp 11-18.
43. Gupta, T. and Seifoddini, H., (1990), “Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system”, *International Journal of Production Research*, Vol. 28, pp. 1247-1269.
44. Gupta, T. and Seifoddini, H., (1990), “Production data based similarity

coefficient for machine component grouping decisions in the design of cellular manufacturing systems", *International Journal of Production Research*, Vol. 28, pp. 1247-1269.

45. Ho, J.C. and Chang, Y.L., (1991), "A new heuristic for the n-job, m-machine flow shop problem", *European Journal of Operational Research*, Vol. 52, pp. 194-202.
46. Ho, J.C., (1995), "Flowshop sequencing with mean flow time objective", *European Journal of Operational Research*", Vol. 81, pp. 57-578.
47. http://personal.ashland.edu/~rjacobs/m503jit.html, accessed on 05/14/2007.
48. Hyer, N. and Wemmerlov, U., (2002), "Reorganizing the Factory: Competing Through Cellular Manufacturing", Productivity Press, Portland, OR.
49. Hyer, N.L. and Brown, K.A., (2003), "Work Cells with Staying Power: Lessons from Process-Complete Operations", *California Management Review*, Vol. 46, No.1, pp. 27-52.
50. Ignall, E. and Schrage, L., (1965),"Application of the branch-and- bound technique to some flow shop scheduling problems", *Operations Research*, Vol. 13, pp. 400-412.
51. Jiao, J. and Tseng, M.M., (2004), "Customizability analysis in design for mass customization", *Computer Aided Design*, Vol. 36, No. 8, pp. 745–757.
52. Jiao, J., Zhang, L., Pokharel, S., (2007), "Process platform planning for variety coordination from design to production in mass customization manufacturing", IEEE Transactions on Engineering Management, Vol. 54, No. 1, Feb. 2007, pp. 112-29.
53. Johnson, S.M., (1954), "Optimal two-and three-stage production schedules", *Naval Research Logistics* ,Vol. 1, pp.61-68.
54. Kaparthi, S. and Suresh, N. C., (1992), "Machine-component cell formation in group technology: a neural network approach", *International Journal of Production Research*, Vol.30, pp.1353-1367.
55. Kaparthi, S. and Suresh, N. C., (1993), "An improved neural network leader algorithm for part machine grouping in group technology", *European Journal of Operational Research*, Vol. 69, pp. 342-356.

56. King, J. R. and Nakornchai, V., (1982), "Machine component group formation in Group Technology- review and extension", *International Journal of Production Research*, Vol. 20, No. 2, pp. 117-133.
57. King, J.R. and Nakronchai, V., (1982), "Machine-Component Group Formation in Group Technology: Review and Extension", *International Journal of Production Research*, Vol.20, No.2, pp. 117-133.
58. King, J.R., (1980), "Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm formation in group technology", *International Journal of Production Research*, Vol. 18, No. 2, pp. 213-232.
59. Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G and Van Brussel, H., (1999), "Reconfigurable Manufacturing Systems", CIRP Annals, Vol. 48, No.2.
60. Kotha, S., (1995), "Mass Customization: Implementing the Emerging Paradigm for Competitive Advantage", *Strategic Management Journal*, Vol.16, pp.21-42.
61. Krishnapillai, R. and Zeid, A., (2006), "Mapping product design specification for mass customization", *Journal of Intelligent Manufacturing*, Vol. 17, February, 2006, pp. 29-43.
62. Kusiak, A. and Chow, W. S., (1987), "Efficient Solving of the group technology problem", *Journal of Manufacturing Systems*, Vol. 6, pp. 117-124.
63. Lampel, J. and Mintzberg, H., (1996), "Customizing Customization", *Sloan Management Review*, Fall 1996, pp. 21-29.
64. Liu, J., Reeves, CR., (2001), "Constructive and composite heuristic solutions to the $P//\sum C_i$ scheduling problem", *European Journal of Operational Research*, 132, pp.439–52.
65. MacCarthy, B., Brabazon, P., Bramham, J., (2003), "Fundamental modes of Operations for Mass Customization", *International Journal of Production Economics*, Vol. 8, pp. 289-304.
66. Mahesh, O. and Srinivasan, G., (2002), "Incremental cell formation considering alternative machines", *International Journal of Production Research*, Vol. 40, No.14, pp. 3291-3310.

67. McAuley, J., (1972), “Machine grouping for efficient production”, *Production Engineer*, Vol. 51, pp. 53-57.
68. Miyazaki, S., and Nishiyama, N., (1980), “Analysis for minimizing weighted mean flow-time in flow-shop scheduling", *Journal of the Operations Research Society of Japan*, Vol.23, pp. 118-132.
69. Miyazaki, S., Nishiyama, N. and Hashimoto, F., (1978), “An adjacent pair wise approach to the mean flow time scheduling problem”, *Journal of Operational Research Society Japan*, Vol. 21, pp. 287-299.
70. Moser, K., (2007), “Mass Customization Strategies- Development of a competence-based framework for identifying different mass customization strategies”, PhD Dissertation, TUM Business School.
71. Nair, G. J. and Narendran, T. T., (1998), “CASE: a clustering algorithm for cell formation with sequence data”, *International Journal of Production Research*, Vol. 36, No.1, pp. 157- 179.
72. Nair, G. J. and Narendran, T. T., (1999), “ACCORD: a bi-criterion algorithm for cell formation using ordinal and ratio-level data”, *International Journal of Production Research*, Vol.37, No.3, pp. 539-556.
73. Nawaz, M., Enscore, E.E., and Ham, I., (1983), “A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem”, *OMEGA*, 11, pp. 91-95.
74. Nearchou, A.C., (2004), “A Novel Metaheuristic Approach for the Flow shop Scheduling Problem”, *Engineering Applications of Artificial Intelligence*, Vol.17, No.3, pp. 289-300.
75. Palmer, D.S., (1965), “Sequencing jobs through a multi-stage process in the minimum total time –A quick method of obtaining a near Optimum”, Operational Research Quarterly, Vol. 16, pp. 101-107.
76. Piller, F.T., and Stotko, C.M., (2002), “Mass Customization: Four Approaches to Deliver Customized Products and Services with Mass Production Efficiency”, Proceedings to the 2002 IEEE International Engineering Management Conference, Managing Technology for the New Economy, 18 - 20 August 2002, St. John's College, Cambridge, UK, pp. 773 – 778.
77. Pine, B.J., (1993), “Mass Customization: The New Frontier in Business

Competition", Harvard Business School Press, Boston, MA.

78. Pine, B.J., (2007), Interview with Frank Piller as reported in the Mass Customization & Open Innovation Newsletter, Vol. 10, No. 1, pp.4-7.
79. Pine, B.J., Gilmore, J.H., (1999), "The Experience Economy: Work Is Theatre & Every Business a Stage", Harvard Business School Press.
80. Pine, B.J., Peppers, D., Rogers, M. (1995), "Do you want to keep your customers forever? ", *Harvard Business Review*, Vol.73, Issue 2, pp.103–114.
81. Pine, B.J., Victor, B and Boynton, A.C., (1993), "Making Mass Customization Work", *Harvard Business Review*, September – October, 1993, pp. 108-116.
82. Pinedo, M., (1995), "Scheduling: Theory, algorithms, and Systems", Prentice Hall, Englewood Cliffs, New Jersey.
83. Pugazhendhi, S, Thiagarajan, S., Rajendran, C., and Anantharaman, N., (2004), "Relative performance evaluation of permutation and non-permutation schedules in flow line-based manufacturing systems with flow time objective", *International Journal of Advanced Manufacturing Technology*, June 2004, Vol. 23 Issue 11/12, pp. 820-830.
84. Qiao, G., Lu, R., McLean, C. (2006), "Flexible manufacturing systems for mass customization manufacturing", *International Journal of Mass Customization*, Vol. 1, No. 2-3, pp. 374-93
85. Qiao, G., Lu, R.F., McLean, C. (2004), "Process control and logistics management for mass customization manufacturing", IIE Annual Conference and Exhibition 2004, pp. 1517-1522
86. Qiao, G., Lu, R.F., Riddick, F. (2006)," Flexible Modeling and Simulation for Mass customization manufacturing", Proceedings of the 2003 IIE Annual Conference, 2003, Portland, OR
87. Qiao, G., McLean, C., Riddick, F. (2002), "Simulation system modeling for mass customization manufacturing", Proceedings of the 2002 Winter Simulation Conference (Cat. No.02CH37393), Vol.2, pt. 2, pp. 2031-2036
88. Rajagopalan, R. and Batra, J. L., (1975), "Design of cellular production system: a graph theoretic approach", *International Journal of Production Research*, Vol. 13, pp. 567-579.

89. Rajendran C. and Chaudhuri, D., (1991), "An efficient heuristic approach to the scheduling of jobs in a flowshop", *European Journal of Operational Research*, Vol.61, pp. 318-325.
90. Rajendran, C., (1993), "Heuristic algorithm for scheduling in a flow shop to minimize total flow time", *International Journal of Production Economics*, Vol. 29, pp. 65-73.
91. Rajendran, C., Ziegler, H., (1997), "An efficient heuristic for scheduling in a flow shop to minimize total weighted flow time of jobs", *European Journal of Operational Research*, 103, pp.129–38.
92. Ramabhatta, V. and Nagi, R.,(1998), "An Integrated formulation of manufacturing cell formation with capacity planning and multiple routings", Annals of Operations Research, Vol. 77, pp. 79-95
93. Reisman, A., Kumar, A., Motwani, J., (1997), "Flow shop Scheduling/Sequencing Research: A Statistical Review of the Literature, 1952-1994", IEEE transactions on Engineering Management, Vol.44, No. 3, August 1997, pp. 316-329.
94. Schonfeld, Erick (1998), "The customized, digitized, have-it–your-way economy", Fortune (28 September), pp.114-121,124.
95. Seifoddini, H., Wolfe, M.P. (1986), "Application of the similarity coefficient method in machine cells formation in group technology", Trans.Am. Inst.Ind.Engrs , pp.271-277
96. Seifoddini, H., Wolfe, M.P. (1987), "Selection of the Threshold value based on material handling cost in group technology applications", Trans.Am. Inst.Ind.Engrs, pp.266-260
97. Seifoddini, H.K., (1989), "Single Linkage Versus Average Linkage Clustering In Machine Cells Formation Applications", *Computers Industrial Engineering*, Vol16, No.3, pp. 419-426.
98. Selim, H.M., Askin, R.G. and Vakharia, A.J., (1998), "Cell formation in group technology: Review, evaluation and directions for future research", *Computers Industrial Engineering*, Vol. 34, No.1, pp.3-20.
99. Selvam, R.P. and Balasubramanian, K.N., (1985), "Algorithmic grouping of

operation sequences", *Engineering costs and Production Economics*, Vol.9, pp.125-134.

100.Shafer, S.M. and Meredith, J.R., (1990), "A comparison of selected manufacturing cell formation techniques", *International Journal of Production Research*, Vol.28, pp. 661-673.

101.Shafer, S.M. and Rogers, D.F., (1991), "A goal programming approach to the cell formation problem", *Journal of Operations Management*, Vol. 10, No.1, pp.28-43.

102.Shafer, S.M., Suresh, N.C., Kay ,J.M., (1998), " Part-Machine-Labor Grouping: The Problem and Solution Methods In Group Technology and Cellular Manufacturing State-of-the-art Synthesis of Research and Practice", Kluwer Academic Publishers.

103.Shafer, S.M., and Rogers , D.F., (1993), " Similarity and distance measures for cellular manufacturing", Part 1, A survey, *International Journal of Production Research*, Vol. 31, No.5, pp.1133-1142.

104.Shambu, G., Suresh, N.C., Pegels, C.C., (1996), "Performance Evaluation of Cellular Manufacturing Systems: A Taxonomy and Review of Research", *International Journal of Operations and Production Management*, Vol. 16, No.8, pp.81-103.

105.Singh, N., (1993), "Design of cellular manufacturing systems: An invited review", *European Journal of Operational Research*, Vol. 69, pp.284-291.

106.Srinivasan, G. and Narendran, T. T., (1991), "GRAFICS - A non-hierarchical clustering algorithm for group technology", *International Journal of Production Research*, Vol. 29, No. 3, pp.463-478.

107.Srinivasan, G.,(1994), "A clustering algorithm for machine cell formation in group technology using minimum spanning trees", *International Journal of Production Research*, Vol. 32, No.9 ,pp. 2149-2158.

108.Suresh, N.C., Slomp, J., and Kaparthi, S., (1995), "The capacitated cell formation Problem: A new hierarchical methodology", *International Journal of Production Research*, Vol. 33, pp.1761-1784.

109.Urbani, A., Molinari-Tosatti, L., and Bosani, R., (2003), "Flexibility and

Reconfigurability for Mass Customization", The Customer Centric Enterprise, Springer-Verlag, Berlin, Germany, pp. 349-360.

110. Vakharia, A. J. and Wemmerlov, U., (1990), "A comparative investigation of hierarchical clustering techniques and dissimilarity measures applied to the cell formation problem", *Journal of Operations Management*, Vol. 13, pp. 117-138.

111. Vakharia, A.J. and Wemmerlov, U., (1990), "Designing a cellular manufacturing system: A materials flow approach based on operations sequences", IIE Transactions, Vol.22, pp.84-97.

112. Wemmerlov, U. and Hyer N.L.,(1986), "Procedures for the part family/machine group identification problem in cellular manufacturing", *Journal of Operations Management*, Vol. 6, No. 2, pp. 125-147.

113. Woo, D.S., Yim, H.S., (1998) "A heuristic algorithm for mean flow time objective in flow shop scheduling", *Computers and Operations Research*, Vol. 25, pp. 175–82.

114. www.dell.com accessed on 05/14/2007

115. www.toyota.com accessed on 5/15/2007

116. Zipkin, P., (2001), "The limits of mass customization", *Sloan Management Review*, 42/3(spring): pp.81-87.

VITA

I was born in the state of Andhra Pradesh in India on the eleventh day of the tenth month of the year nineteen hundred and eighty two (11/10/1982). I completed my secondary school education at Keyes High School, Hyderabad and finished my pre-college from St.Francis Junior College, Hyderabad. I then pursued my Bachelors in Mechanical Engineering at Vasavi College of Engineering and Technology, affiliated to Osmania University, Hyderabad. I was awarded merit certificates for my outstanding performance during my schooling and also received Gold Medal for my outstanding performance during my Bachelors. I then joined University of Kentucky, to pursue my Masters in Mechanical Engineering in August 2001. I served as a Teaching and Research assistant during the course of my MS. A list of publications during my MS is listed below:

Conference Publications and Presentations:

1. Haritha Metta, Arvind Goyal, Brandon Stump, Kundana Inala, Smitha Thuramalla, Fazleena Badurdeen "A Simple Simulation Model to Demonstrate Mass Customization Strategies", 2007, (To be presented), Third World Conference on Mass Customization & Personalization, October 07-11, 2007, Boston, MA.

2. Fazleena Badurdeen, Bader Meriden, Smitha Thuramalla, "Performance Analysis of Minicell-based Manufacturing System for Mass Customization", (To be presented), Third World Conference on Mass Customization & Personalization, October 07-11, 2007, Boston, MA.

3. Fazleena Badurdeen, Dale Masel, Smitha Thuramalla, "Increasing Mass Customization Manufacturing Flexibility using Minicells", 40th CIRP International Seminar on Manufacturing Systems, May 30-June1, 2007, Liverpool, England.

4. Fazleena Badurdeen, Dale Masel, Smitha Thuramalla, "A Modular Manufacturing

System for Mass Customization using Minicells", International Conference on Group Technology & Cellular Manufacturing, GT / CM 2006, July 3-5, 2006, Groningen, Netherlands.

5. Fazleena Badurdeen, Dale Masel, Smitha Thuramalla, "Multi-objective Genetic Algorithm to design Minicells for Mass Customization Manufacturing", Industrial Engineering & Research Conference, May 20-24, 2006, Orlando, Florida, USA.

Smitha Thuramalla