DESIGNING CELLULAR MANUFACTURING SYSTEMS WITH TIME VARYING PRODUCT MIX AND RESOURCE AVAILABILITY

by

Elin MacStravic Wicks

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Industrial and Systems Engineering

APPROVED:

Roderick J. Reasor, Ph.D., Chairman

Ralph D. Badinelli, Ph.D.

Michael P. Deisenroth, Ph.D.

Wolter J. Fabrycky, Ph.D.

William G. Sullivan, Ph.D.

June 16, 1995

Blacksburg, Virginia

DESIGNING CELLULAR MANUFACTURING SYSTEMS WITH TIME VARYING PRODUCT MIX AND RESOURCE AVAILABILITY

by

Elin MacStravic Wicks

Roderick J. Reasor, Ph.D., Chairman

Industrial and Systems Engineering

(ABSTRACT)

Cellular manufacturing is a practical application of group technology in which functionally dissimilar machines are grouped together to produce a family of parts. The fundamental problem of cellular manufacturing system design is the identification of part families and machine cell compositions. This problem is commonly referred to as the Part Family / Machine Cell (PF/MC) formation problem. Given a set of parts, processing requirements, and available resources, the objective of the PF/MC formation problem is to obtain a satisfactory partition of parts into families and machines into cells.

The effectiveness of a cellular manufacturing system is sensitive to fluctuations in the demand for products, the product mix, and the availability of resources. This research offers a multi-period formulation of the PF/MC formation problem. It addresses the dynamic nature of the production environment by considering a multi-period forecast of product mix and resource availability during the formation of part families and machine cells. The goal of the multi-period formulation is to obtain a cellular design that performs well with respect to the design objectives over the entire planning horizon.

Design objectives of the multi-period formulation of the PF/MC formation problem are the minimization of intercell material handling costs, the minimization of investment in additional machines, and the minimization of the cost of system re-configuration over the planning horizon. A mathematical model of the problem is developed and a solution procedure is presented based on a genetic algorithm. The advantages of using a genetic algorithm to solve the multi-period PF/MC formation problem include the ease with which alternate design objectives can be incorporated and the ability to generate alternative system designs.

The output of the multi-period PF/MC formation methodology developed in this dissertation is a period by period description of the part families and machine cell compositions. Results are presented of a preliminary investigation of the benefits of using a multi-period model versus one that assumes that the product demand, product mix, and available resources remain constant. In the design problems considered, the multi-period approach to solving the PF/MC formation problem resulted in a cellular design that performed the best overall with respect to the design objectives.

Acknowledgments

The path that has led me to the completion of this dissertation has been long, challenging, and often frustrating. Without the support and guidance of the many people who have touched my life, this culmination of many years of hard work would have been much more difficult to achieve. At this time I would like to thank each of them for their encouragement and support.

I would like to express my thanks and appreciation to my advisor, Dr. Roderick J. Reasor, for his ideas and enthusiasm that helped shape this dissertation, and for his guidance and support throughout its completion. I would also like to acknowledge the other members of my committee, Dr. Ralph D. Badinelli, Dr. Michael P. Deisenroth, Dr. Wolter J. Fabrycky, and Dr. William G. Sullivan for their time and helpful suggestions.

I would also like to thank the faculty of the Industrial Engineering Department of Rutgers University; in particular, Dr. Thomas O. Boucher, Dr. Elsayed A. Elsayed, and Dr. James T. Luxhoj. My decision to pursue a Ph.D. and a career in academia was significantly influenced by their example and encouragement.

My family has been an inexhaustible supply of love and support throughout my entire life. I thank God each and every day for the blessing of having a family as wonderful as mine. To my parents -- Jim and Elin -- thank you for giving me life, love, and for providing me with the best example of what it really means to be a good person. To my brothers and sisters -- Jimmy and Erin; Irene and Pat; Barb and Kit; Larry and Jennifer; Marci and Pat; and Matthew -- thank you so much for being my friends as well as my siblings. Thanks to my adorable nieces -- Samantha, Rachel and Kiera -- whose faces filled with sunshine could brighten up even the darkest day. I would also like to thank my extended family, in particular, Anna and Mike; and Nat and Ellen -- for their endless support and encouragement. Very special thanks go to Al, Claudia, Katie, and Kallie, not only for their support but for providing me with food and shelter during the long nights it took to complete this document.

Last, but most certainly not least, I would like to thank my husband, Matt, for everything that he has given me. Without his love, encouragement, and confidence in me, I never could have come this far. I dedicate this work to you, Matt, for the fond memories of yesterday, the joy of today, and the promise of all that is yet to come.

Table of Contents

List of Illustrations

List of Tables

1.0 Introduction

Group technology is a manufacturing philosophy that advocates identifying similar parts and grouping them together in families to take advantage of their similarities in design and manufacturing. Cellular manufacturing is a practical application of group technology in which functionally dissimilar machines are grouped together to produce a family of parts. Cellular manufacturing has been recognized as one of the most recent technological innovations for improving productivity and competitiveness. By dedicating a machine cell to the production of a part family, many of the efficiencies of mass production can be realized in a less repetitive batch environment. Reductions in set up times, work-in-process inventory levels, and production lead times are some of the benefits associated with cellular manufacturing systems.

1.1 Background

The first phase in the design and implementation of a cellular manufacturing system is the identification of part families and machine cells. This phase is commonly referred to as the Part Family/Machine Cell (PF/MC) formation problem. PF/MC formation is the process of analyzing part and machine populations, grouping parts with similar design features, tooling requirements, or manufacturing routings into families, and grouping the required machines into cells to produce the part families. This problem has captured the attention of researchers for over a decade and numerous solution methodologies have been proposed. Missing from the literature, however, are case studies reporting the usefulness of these techniques in designing an actual cellular manufacturing system. The continued development of new procedures coupled with the lack of success stories, indicates that the PF/MC formation problem has yet to be resolved to the satisfaction of researchers and practitioners alike.

The effectiveness of a cellular manufacturing system is sensitive to fluctuations in product demand, product mix, and resource availability. The majority of existing PF/MC formation models assume each of these factors to be constant. In reality, the demand for a product varies according to its stage in the product life cycle, new products are introduced and the production of older products is discontinued. In addition, resources are continually being replaced due to age and/or obsolescence. It would be advantageous to include these issues in the design of a cellular manufacturing system.

The primary objective of this research is to develop a new PF/MC formation methodology that addresses the ongoing phase-in/phase-out of products and resources. By considering changes in the product mix and available resources during the formation of part families and machine cells, the efficiencies of the cellular manufacturing system can be maintained throughout the planning horizon. A more detailed discussion of the problem and the objectives of the research will be presented in the remaining sections of this chapter.

1.2 Statement of the Problem

The fundamental problem of cellular manufacturing system design is the identification of part families and the composition of machine cells. Given a set of parts, processing requirements, and available resources, the objective of the PF/MC formation problem is to obtain a satisfactory partition of parts into families and machines into cells with respect to one or more design objectives. The system design must be capable of producing the required volume of parts without exceeding resource capacity. The purpose of this section is to outline the issues related to formulating and solving the PF/MC formation problem and to identify the desirable characteristics of a new design methodology.

There are a number of objectives related to the design of a cellular manufacturing system. The following list of objectives was taken from Ballakur and Steudel (1987) and Wemmerlöv and Hyer (1987).

1. Minimize throughput times.
2. Minimize setup times.
3. Minimize inventories.
4. Maximize resource (machine and labor) utilization.
5. Maximize output.
6. Minimize machine relocation costs.
7. Minimize intracell and intercell moves of material.
8. Minimize operating costs.

9. Minimize investment.
10. Minimize the number of cells.
11. Minimize duplication of machines in different cells.
12. Maximize the percentage of operations of a part processed within a single cell.
13. Maximize the number of parts handled by the cells as a percentage of the total number of parts processed through the shop.
14. Minimize job lateness.
15. Obtain a pure flow line structure within the cell.

As indicated by the above list, the cellular manufacturing system design process is a multi-objective task. Some of these objectives are conflicting and will require that tradeoffs be made during the design process. The majority of cell formation techniques focus on a single objective, such as minimizing intercell movement or minimizing capital investment in new machines. Clearly these objectives are conflicting; one can minimize intercell movement by investing in new machines or reduce capital investment requirements at the expense of increased material handling requirements. The new methodology should address the multi-objective nature of the PF/MC formation problem by including multiple objectives in the design evaluation function.

There are a variety of system parameters to be considered during the process of forming part families and machine cells. The following is a list of typical parameters used in the PF/MC formation problem. In general, these parameters are independent of the system design.

1. Part processing requirements.
2. Available resources.
3. Operation times.
4. Set up times.
5. Production volumes.
6. Resource capacities.

Depending on the design objective, certain cost information is also needed. For example, material handling costs and the cost of acquiring additional resources would be necessary to evaluate the trade-off between machine duplication and intercell movement. The new methodology developed to solve the PF/MC formation problem should include all relevant system parameters and costs.

Existing PF/MC formation techniques fail to address the dynamic nature of the production environment. The underlying assumption of these techniques is that the part and machine populations are fixed and that demand is constant. In reality, production volumes will vary from period to period, new products will be introduced, and older products will be phased out. The machine population will also change as older machines are replaced and new technology is acquired. The new methodology should address the dynamic nature of the production environment by incorporating a multi-period forecast of product mix and resource availability. This will require the specification of the system parameters by period.

To obtain a feasible design, certain constraints must be considered during the design process. For example, there must be sufficient resource capacity to produce the required production volumes. Other typical constraints include restrictions placed on capital investment, machine utilization, the number of machines assigned to a cell, and the number of cells formed. The new methodology should include all relevant system constraints.

Once the design objectives, system parameters, and constraints have been identified, a mathematical formulation of the PF/MC formation problem can be developed. In the case of a single objective, such as minimizing intercell transfers, the problem is frequently modeled as a mixed integer program. When multiple objectives are considered, a goal programming formulation is often used. Some researchers, however, have had difficulty solving the problem directly using math programming techniques due to the large number of variables and constraints involved (Shafer and Rogers, 1991). In a recent paper, Venugopal and Narendran (1992) found genetic algorithms to be an effective approach to solving a multi-objective formulation of the PF/MC formation problem. The new methodology uses a genetic algorithm based solution methodology.

In general, it is not easy to identify an optimal (non-dominated) solution to a multi-objective problem with conflicting objectives. In addition, it is not possible to include all relevant factors in a model of the PF/MC formation problem. Intangible criteria and criteria that are not easily measured are difficult to include in the design objective function. Thus, it is essential that the decision maker be presented with a set of satisfactory designs. The alternative designs can then be evaluated with respect to overall system performance. The new methodology should be capable of generating good alternative solutions.

1.3 Research Objectives

The objective of this research is to develop a new Part Family / Machine Cell formation methodology that addresses the dynamic nature of the production environment. This is accomplished by including a multi-period forecast of product mix and resource availability in the set of system parameters. The effectiveness of a cellular manufacturing system (with respect to the design objectives) is sensitive to changes in product mix and resource availability. By incorporating anticipated changes in product mix and resource availability during the design of a cellular manufacturing system, a robust assignment of parts and machines to cells can be made.

The primary objective of this research is to develop a methodology for obtaining a preliminary design for a cellular manufacturing system in the presence of a multi-period forecast of product mix and resource availability. It is intended that the methodology also include the following key features.

1. *The ability to address multiple design objectives.* The PF/MC formation problem is characterized by multiple (and sometimes conflicting) objectives. The new methodology should be capable of including one or more objectives in the design evaluation function as specified by the needs of the designer.

2. *The ability to work with an existing cellular manufacturing system design.* The PF/MC formation problem does not end once the initial design of the system has been decided upon. Unplanned for changes in product mix and/or resource availability need to be evaluated in the context of the existing system. The new methodology should be capable of designing an initial cellular system and investigating improvements to an existing system.

3. *The ability to generate alternative cellular designs.* Given multiple design objectives, it is not usually possible to obtain a solution that is optimal with respect to all objectives. In addition, there are evaluation measures not explicitly considered by the model. The new methodology should be capable of generating alternative designs that can subsequently be evaluated by the designer with respect to additional criteria.

The development of the new methodology for solving the multi-period PF/MC formation problem can be broken down into four major research areas. These areas are described as follows.

1. *Develop a mathematical formulation of the multi-period PF/MC formation problem.* This research area consists of defining the multi-period PF/MC formation problem in terms of the set of design objectives, system parameters, and system constraints.

2. *Develop a genetic algorithm to search for a set of good solutions.* This research area involves developing a chromosomal representation of the solution; creating genetic operators to propel the search for good solutions; and translating the objective function of the mathematical formulation into an evaluation function that can be used by the genetic algorithm.

3. *Validate the genetic algorithm.* This research area involves comparing the solutions obtained by the genetic algorithm to those obtained by existing PF/MC formation techniques for the constant product mix and resource availability case.

4. *Investigate the benefits of the multi-period approach to PF/MC formation.* This research area involves evaluating the performance of designs obtained via the new methodology to designs obtained using single period scenarios. The single period scenarios require periodic adjustments due to changes in the product and resource populations.

The contribution of the multi-period approach to PF/MC formation is the ability to design a cellular manufacturing system that will continue to perform well as the production environment evolves. By addressing the dynamic nature of the production environment in the design phase of the cellular system, many of the benefits associated with cellular manufacturing can be realized throughout the life of the system.

1.4 *Dissertation Overview*

The following four chapters are included in this dissertation document: Literature Review; A Multi-Period Model for PF/MC Formation; Multi-Period v. Single Period PF/MC Formation; and Conclusions, Contributions, and Extensions. The literature review discusses existing approaches to the PF/MC formation problem. A taxonomy of existing techniques is shown and examples of procedures from each category are discussed. The literature review concludes with a brief discussion of the mechanisms of genetic algorithms and their applications.

Chapter 3, "A Multi-Period Model for PF/MC Formation," is a detailed discussion of the modeling approach. The design objectives, system parameters, and system constraints are identified and a mathematical formulation of the problem is developed. The key elements of the mathematical model are addressed in a genetic algorithm based solution methodology. The mechanisms of the genetic algorithm are described and illustrated. The primary outputs and uses of the multi-period PF/MC formation methodology are discussed. This chapter concludes by validating the genetic algorithm approach to solving the PF/MC formation problem. This is accomplished by demonstrating that the genetic algorithm is capable of designing a system that performs at least as well as a system designed by another technique.

The objective of Chapter 4, "Multi-Period v. Single Period PF/MC Formation," is to investigate the potential benefits of using a multi-period approach for designing a cellular manufacturing system. The performance of multi-period cellular designs over the planning horizon are compared to the performance of single period designs over the planning horizon. This dissertation concludes with a discussion of the contributions of the research effort and identifies areas of future research.

2.0 Literature Review

It is the intent of this chapter to review the existing literature concerning PF/MC formation techniques and the application of genetic algorithms. A taxonomy of PF/MC formation techniques is presented in the first section followed by more detailed descriptions of the approaches taken in the subsequent sections. Non-constant demand/mix models are presented along with a model that seeks to improve an existing cellular manufacturing system design. This chapter concludes with a brief tutorial on genetic algorithms and a discussion of the various applications of genetic algorithms.

2.1 A Taxonomy of PF/MC Formation Techniques

The objective of the PF/MC formation problem is to obtain a satisfactory partition of parts into families and machines into cells with respect to one or more design objectives. The literature on the PF/MC formation problem is quite extensive. One of the earliest works in this area was when Burbridge (1963) introduced the production flow analysis approach for solving group technology problems. Since then, many researchers have taken an interest in the PF/MC problem. Numerous formulations and solution strategies have been published in the last decade. Wemmerlöv and Hyer (1986) classified over 70 contributions found in the literature. In a more recent manuscript, Joines *et al.* (1994) review and classify more than 300 methodologies for forming part families and machine cells.

A three-pronged classification scheme can be used to categorize the different strategies used for solving the PF/MC formation problem. This scheme, which is based on the sequence in which the part families and machine cells are formed, classifies existing techniques as belonging to one of three categories: part grouping, machine grouping, or simultaneous machine-part grouping. The part grouping approach forms the part families first. Classification and coding techniques and cluster analysis (based on a part similarity measure) are typically used to form the part families. Machine cells are then formed based on the processing requirements of each part family.

In contrast, the machine grouping approach creates the machine cells first and then allocates each part to a cell. Cluster analysis (based on a machine similarity measure) and graph theoretic techniques have been used to form the machine cells. Parts are then assigned to the cell in which the majority of their processing requirements are met. The advantage of grouping machines first versus parts lies in the computational burden. In most cases, the size of the machine population will be smaller than that of the part population.

The last approach is to form the part families and machine cells simultaneously. This is typically accomplished through the use of cluster analysis or mathematical programming. A few models also use graph theory and network analysis to simultaneously form the part families and machine cells. The focus of current research in PF/MC formation techniques appears to center around the simultaneous grouping approach.

2.2 *The Part Grouping Approach to PF/MC Formation*

The part grouping approach seeks first to identify part families from the population of parts under consideration for cellular manufacture. Once the part families are known, machine cells are created by grouping those machines required to produce each part family. The majority of techniques following this approach use a clustering algorithm to form the part families. Techniques of this category can be farther classified by the information used to obtain the part families. One class of techniques uses part routing information to form the families while the other class of techniques relies upon part design information made available by classification and coding systems. Several techniques from each of these classes will be reviewed in the following sub-sections.

2.2.1 Routing-Based Part Family Formation Techniques

Routing-based part family formation procedures use similarities between the processing requirements of the part population to group the parts into families. Typically a similarity coefficient is defined and used as the input to a clustering algorithm. Once the part families have been formed, the composition of the machine cell is dictated by the processing requirements of the part family assigned to the cell. Techniques of this type vary in terms of the similarity measure used and in the degree to which machine loading and economic data are considered.

Carrie (1973) uses the Jaccard similarity coefficient to quantify the processing similarity between part pairs. This measure appears to have been first proposed by Jaccard (Sokal and Sneath, 1968) and was first used in the context of the PF/MC formation problem by McAuley (1972). The Jaccard similarity coefficient measures the similarity between parts p and q as

$$S_{pq} = \frac{\sum_J a_{pj} \cdot a_{qj}}{\sum_J (a_{pj} + a_{qj}) - \sum_J a_{pj} \cdot a_{qj}}$$

where

$$a_{ij} = \begin{cases} 1 & \text{if part } i \text{ requires processing on machine } j \\ 0 & \text{otherwise} \end{cases}$$

J = set of machines.

This similarity coefficient is valued on the interval [0, 1], where a value of 1 indicates total similarity and a value of 0 indicates total dissimilarity. To illustrate, if part A requires processing on machines {1, 2, 3} and part B requires processing on machines {1, 3, 4}, the similarity coefficient for this part pair is $S_{AB} = 0.5$.

The clustering procedure used by Carrie uses a minimum similarity threshold value to monitor the admittance of a part to a part family. This threshold value is incrementally lowered until all parts have been assigned to a family. Carrie suggests that machine loading be evaluated each time a part is admitted to a family to check that no proposed grouping violates machine availability. However, guidelines for what to do should such a violation occur are not given.

Choobineh (1988) modifies the Jaccard similarity coefficient to measure similarity in part operation sequence. A similarity coefficient which considers identical sequences of operations of length 1 through L is referred to as a similarity coefficient of order L. The average similarity coefficient of order L between parts p and q, $S_{pq}(L)$, is given by

$$S_{pq}(L) = \frac{1}{L} S_{pq}(1) + \sum_{l=2}^{L} \frac{C_{pq}(l)}{N - l + 1} \qquad L \leq N$$

where

$C_{pq}(l)$ = the number of identical sequences of length l between parts p and q
N = $\min_i (N_i)$ where N_i is the number of operations for part i
$S_{pq}(1)$ = the Jaccard similarity coefficient for parts p and q

Note that the Jaccard coefficient looks only at similarity in sequences of length one. Choobineh's similarity coefficient will have more discriminating value as the order of the similarity coefficient is increased. The pairwise similarity coefficients can be used in a clustering algorithm to form the part families. Choobineh suggests using at least two alternative process plans per part to increase the odds of uncovering natural part families.

To complete the PF/MC formation procedure, Choobineh presents an integer programming model that creates independent cells producing one or more part families. The objective of the model is to minimize total cost in terms of WIP, setup, processing, and machine costs. The system parameters used include part processing times, annual demand, and machine capacities.

Vakharia and Wemmerlöv (1990) also incorporate pairwise similarity between part operation sequences in their similarity coefficient. This similarity coefficient is used along with a flow parameter in a clustering algorithm to form part families. The flow parameter is used to monitor and limit backtracking to help identify part families with desirable flow characteristics. A cell is created for each part family and an unconstrained allocation of machines to cells is done to ensure load feasibility with respect to part processing times, annual demand, machine capacity, and desired machine utilization.

Material flow considerations are also taken into account in the syntactic pattern recognition approach used by Wu, *et al.* (1986). In this approach, a part is represented by a 'grammar' describing its operation sequence. For example, G = {ABCD} indicates that the part must first visit machine type A, then B, then C, and finally D. The minimum spanning tree approach to cluster analysis is used on non-dominated grammars to form part families with similar operation sequences. Material flow restrictions can be incorporated in the definition of domination. For example, in a bi-directional or random access system, the grammar {ABCD} dominates the grammar {BDC}. However, if uni-directional flow is desired, {ABCD} does not dominate {BDC}. Each family is assigned to a separate cell and the cell composition is inferred by the composite grammar of the part family. Machine workload must be analyzed separately to determine if duplicate machines are required. The strength of this approach is that syntactic pattern recognition can be used to assign a new part to a cell. If a single cell cannot accommodate the part, the part is assigned to the cell where the difference between the part grammar and the cell 'language' is a minimum.

2.2.2 Feature-Based Part Family Formation Techniques

Classification and coding systems facilitate the group technology philosophy in both design and manufacture. Part classification and coding is concerned with identifying similarities in the design and manufacture of parts and relating these similarities to a coding system. When a new part is being designed, the classification and coding system can be used to see if a similar part is already in existence, eliminating the need to start the design process from scratch. Similarly, when planning the manufacture of a new part, the classification and coding system can be used to retrieve the process plans of similar parts. Several PF/MC formation techniques use a classification and coding of part features as the basis for forming part families.

Dutta, *et al.* (1985, 1986) create tooling families using a codification of tool and process requirements. They suggest augmenting the feature-based classification and coding scheme with the following six parameters:

1. Type of operation/process
2. Tool material
3. Workpiece material
4. Workpiece size
5. Surface roughness
6. Tolerance ranges.

Five dissimilarity measures are defined to capture the contribution of individual parts and part families to the overall tooling dissimilarity of the part population. An algorithmic procedure is used to iteratively consider the re-allocation of parts to different families with the objective of reducing the overall dissimilarity coefficient.

Han and Ham (1986) present a multi-objective cluster analysis method for use with a classification and coding system. The method is multi-objective because the relative importance of coded features with respect to forming part families is captured by prioritizing the input code. Consider a five feature coding system represented by the following vector.

$$[\,F_1 \mid F_2 \mid F_3 \mid F_4 \mid F_5\,]$$

By rearranging the code sequence to read as

$$[\,F_3 \mid F_2 \mid F_1 \mid F_4 \mid F_5\,],$$

feature three (e.g. part shape) has been identified as the most important feature followed by feature two (e.g. part size), and so on. In addition to prioritizing the input code in this manner, it is also possible to identify a set of significant features such that all of the parts in a part family will have identical code values for those features.

The objective of the cluster analysis is to lexicographically minimize the difference between part codes in the same family. The difference between two part codes is described in terms of a distance function. In this analysis, the absolute Minkowski metric is used to quantify the distance between two parts. The distance between two parts, p and q is given by:

$$d_{pq} = \sum_{K} \left| X_{pk} - X_{qk} \right|$$

where

K = set of codes

k = k^{th} digit

X_{pk} = the value of the k^{th} digit in the code for part p.

There are a variety of other distance functions that could be used. Han and Ham do not address the formation of manufacturing cells to produce the identified part families.

Xu and Wang (1989) use fuzzy mathematics as a clustering technique to form part families based on a coding of part features. The advantage of using fuzzy methodologies is that they allow for uncertainty inherent in part features. Instead of using a 0 or 1 to represent the absence or presence of a feature, a value between 0 and 1 is assigned to represent the degree to which the feature is present in the part. A computer program was developed to perform the cluster analysis to obtain the part families. Also included in the computer program is a fuzzy pattern recognition algorithm that can be used to assign new parts to existing families.

Cluster analysis is not the only solution procedure used in conjunction with a classification and coding system. Kao and Moon (1991) utilize a common three layer feedforward neural network with the backpropagation algorithm to group feature coded parts into part families. A chain type binary coding system is presented for use with the neural network, but the authors claim that the methodology is robust enough to work with any classification and coding system. Kusiak (1983) uses an integer programming formulation with the objective of minimizing the total sum of distances in a part family to the part family median. A sub-gradient algorithm is suggested for solving the integer formulation. In contrast to clustering procedures, where the number of part families formed is an output of the analysis, the number of families to form is a parameter of the integer programming model.

In general, the feature-based part family formation techniques do not capture all of the operational aspects of a creating a cellular manufacturing system. Only Dutta, *et al.* (1985, 1986) explicitly addresses the use of the part families for the creation of manufacturing cells. However, machine capacity and economic factors are ignored.

The usefulness of classification and coding systems goes beyond the PF/MC formation problem. One of the objectives of group technology is to reduce variation proliferation. The use of classification and coding systems can aid this objective in both part design and process planning. Classification and coding systems can simplify the process of introducing a new part into the cellular manufacturing system.

2.3 The Machine Grouping Approach to PF/MC Formation

The machine grouping approach seeks first to partition the population of machines into cells and then to assign each part to a cell. A part family consists of all the parts assigned to the same cell. The majority of the techniques taking this approach to PF/MC formation use cluster analysis to form the machine cells. Alternate solution methodologies include graph theory and mathematical programming. Typically a similarity coefficient is defined to measure the similarity of machines in terms of parts processed. As was the case for the routing-based part family formation techniques, the machine grouping techniques vary in terms of the similarity coefficient and clustering algorithm used and in the degree to which machine loading and economic data are considered.

McAuley (1972) uses the Jaccard similarity coefficient and single linkage cluster analysis to form the machine cells. This method first clusters those elements that are mutually related with the highest possible similarity coefficient, then successively lowers the level of admission by steps of predetermined magnitude. An element having the specified similarity level with any other element in a cluster is admitted to that cluster. An element having the specified similarity level with elements in different clusters will cause the joining of those clusters. The number of machine cells formed depends on the selected threshold value of the similarity coefficient. Once the machine cells have been formed, parts are allocated to cells based on where the majority of the part's processing occurs. A disadvantage of single linkage cluster analysis is that admission to a cluster requires sufficient similarity with only one member of that cluster; it does not consider the overall dissimilarity with other members. In addition, while two clusters may be linked via a common bond with an element, many of the members of the two clusters may be quite far removed from each other in terms of similarity.

Waghodekar and Sahu (1984) present a machine grouping technique known as MACE (MAchine-component CEll formation). The unique feature of this method is that it uses a similarity coefficient of the product type versus one of the additive type, such as the Jaccard coefficient. The product type similarity coefficient, PS_{mn}, is based on the total number of parts processed by each machine m and n and is calculated via the following equation.

$$PS_{mn} = \frac{(\sum_i a_{im} \cdot a_{in})}{(\sum_i a_{im})(\sum_i a_{in})}$$

Recall that $a_{ij} = 1$ if part i requires machine j; 0 otherwise. Machines of close similarity are grouped together and then cells are formed by merging groups to minimize intercell movement. Parts are assigned to cells according to where the majority of a part's processing requirements can be met.

Wei and Kern (1989) define a new similarity coefficient, called the commonality score, and develop a new clustering algorithm for solving the cell formation problem. The commonality score takes into account not only the number of components on which both machines work, but also the components on which the machines do not work. The simple clustering procedure automatically creates the maximum number of cells dictated by the machine-part incidence matrix. A procedure for merging cells to minimize intercell travel and/or to adhere to constraints on cell size is also presented. The advantage of this clustering procedure is that it is linear in complexity.

The three procedures discussed thus far work with the machine-part incidence matrix as the sole input. Steudel and Ballakur (1987) developed a dynamic programming based heuristic to form machine cells that includes part processing times and annual demand. The authors define a new similarity measure, cell bond strength, to quantify pair-wise machine similarity based on part processing requirements. The cell bond strength between machines m and n is defined as

$$CBS_{mn} = \frac{\sum_i (a_{im} \cdot a_{in}) \cdot t_{im}}{\sum_i a_{im} \cdot t_{im}} + \frac{\sum_i (a_{im} \cdot a_{in}) \cdot t_{in}}{\sum_i a_{in} \cdot t_{in}}$$

where t_{ij} is the processing time for part i on machine j weighted by the annual demand for part i. The first component of the cell bond strength measure is the fraction of the total workload of machine m due to the common parts of machines m and n. Likewise, the second component is the corresponding workload fraction for machine n. Thus, the cell bond strength values represent a two-way bond between machines based on workload.

The cell bond strength values are used in a dynamic programming heuristic to form a chain of machines such that the sum of the cell bond strengths are maximized along the length of the chain. This chain is subsequently partitioned into machine cells subject to restrictions on cell size. Although this is the most realistic machine grouping technique discussed thus far in that part processing times and demands were addressed, machine capacity and the minimization of intercell movement were not explicitly considered.

Gupta and Seifoddini (1990) present a similarity coefficient that incorporates not only the processing requirements of parts, but also similarities in operation sequence. The number of times parts visits both machines in a row contributes to the similarity between the machines. The clustering procedure used in the algorithm is complete linkage clustering. The difference between complete linkage clustering and single linkage clustering (used by McAuley, 1972) is the value of the similarity coefficient used to represent a cluster of machines. In single linkage clustering, the highest similarity coefficient among pairs within the group is chosen. Complete linkage clustering takes the opposite approach, representing the similarity of a group of machines by the smallest similarity coefficient found in the group. A quantitative measure is presented to evaluate alternate solutions obtained by varying the threshold value used in the clustering procedure. The technique concludes by computing machine utilization to identify candidates for duplication.

Rajagopalan and Batra (1975) use a graph theoretic technique to solve the cell formation problem. Their approach is to view the production system as a graph whose vertices are the machines and whose edges indicate the relationship between machines. The strength of the relationship between machines is measured via the Jaccard similarity coefficient that accounts for projected product demand. Edges whose value does not exceed a threshold value (signaling a weak relationship between the machines) are not included in the graph. A heuristic graph partitioning procedure is used to form the machine cells with the objective of minimizing intercell moves subject to an upper limit on cell size. Parts are then allocated to the machine cell that can accommodate the longest sequence of consecutive part operations. Machine loading is computed to determine the number of identical machines required by the cell.

Srinivasan *et al.* (1990) model the PF/MC formation problem as an assignment problem. The inputs to the assignment problem are the pairwise similarity measures between machines. Machine similarity is defined as the number of parts processed by both machines. The objective of the model is to maximize the sum of similarity measures of machines within a cell. The output of the assignment model is disjoint machine groups. If part families are not easily identifiable given the formed cells, the assignment model can be re-run using part similarities as the input data. The procedure concludes by investigating the merging cells and their associated part families with the objective of reducing intercell movement. The rule used is to merge two cells if the number of exceptional elements eliminated is greater than the number of voids created by the merger. The model lacks realism in that processing times and machine capacities are not considered.

2.4 The Machine-Part Grouping Approach to PF/MC Formation

The machine-part grouping approach to PF/MC formation seeks to simultaneously identify machine cells and part families. The majority of methodologies taking this approach use an array-based cluster analysis or formulate a mathematical model of the problem and use math programming to solve the problem. A few methodologies also use graph theory and network analysis to identify machine cells and part families. Several techniques using each solution methodology will be reviewed in the following sub-sections.

2.4.1 Array-based Clustering Techniques

Array-based clustering techniques are algorithms that seek to simultaneously form part families and machine cells by re-arranging the rows and columns of the 0-1 machine-part incidence matrix to reveal a block diagonal structure. Part families and their associated machine cells are then easily identified by the blocks of ones appearing in the matrix. In an ideal situation, these techniques will result in a matrix having all ones in the diagonal blocks and all zeroes in the off-diagonal blocks. In general, ones appearing in the off-diagonal blocks represent intercell movement and zeroes in the diagonal blocks represent less than 100% utilization of the machine cell.

McCormick *et al.* (1972) developed the bond energy algorithm to permutate the rows and columns of any non-negative matrix. The bond energy between two columns (rows) is equal to the sum of the products of the elements in the columns (rows). The algorithm seeks to maximize the total bond energy of the matrix. If a block diagonal structure exists in the matrix, the bond energy algorithm will reveal it. Slagle *et al.*

(1975) developed a similar algorithm based on the concept of the shortest spanning path of a graph. Bhat and Haupt (1976) developed a matching algorithm that outperforms the bond energy algorithm and shortest spanning path in terms of computational efficiency.

King (1980) developed the Rank Order Clustering (ROC) algorithm to generate diagonalized groupings of the ones in the incidence matrix. The elements of a row (column) are read as a binary word. The algorithm iteratively arranges the rows and columns in decreasing rank order until the block diagonal form emerges. Recognizing the computational difficulty in reading the elements as binary words when the matrix is large, King and Nakornchai (1982) developed the ROC2 algorithm to address this shortcoming. The ROC2 algorithm begins by identifying all rows with ones in the right-most column of the matrix and moving them to the top while maintaining their relative order. Columns are then similarly sorted by moving those with ones in the top-most position to the left of the matrix. This procedure continues until the matrix does not change from iteration to iteration. Chan and Milner (1982) developed a very similar procedure known as the Direct Clustering Algorithm.

The ROC and ROC2 algorithms have been criticized in the literature because the solutions obtained by these algorithms are dependent on the arbitrary initial organization of the machine-part incidence matrix. In addition, the identification of part families and machine cells is difficult in the presence of exceptional elements. To obtain a pure block diagonal structure, ROC and ROC2 require that the user identify and eliminate exceptional elements (either by duplicating machines or removing parts from the matrix) prior to using the algorithm. Similarly, the results obtained using the bond energy and the shortest spanning path algorithms must be modified manually if the block diagonal structure does not emerge due to the existence of exceptional elements. Several authors have presented techniques that overcome these deficiencies.

Boe and Cheng (1991) present a 'close neighbor' algorithm for revealing a block diagonal structure in the incidence matrix. This algorithm is indifferent to the original arrangement of the matrix and will result in the desired block diagonal form in a single pass. Khator and Irani (1987) introduce a heuristic procedure, known as the Occupancy Value method, for identifying the part family and machine cell clusters. The unique feature of their method is that it progressively develops the block diagonal form beginning from the upper left corner of the matrix. A drawback to this heuristic is that a good seed machine must be selected. Neither of these two procedures require the pre-identification of exceptional elements.

Askin *et al.* (1991) propose a Hamiltonian path heuristic to rearrange the machine-part incidence matrix to obtain a near block diagonal structure. This method differs from other reordering techniques in that distance measures are used in the reordering process. The Jaccard similarity coefficient, modified for part grouping, is converted to a distance measure to form a between part distance matrix and a between machine distance matrix. The value of the Jaccard coefficient for a pair of machines is equal to the ratio of the number of components processed by both machines to the number of components processed by at least one of the machines. The two stage heuristic consist of first solving the traveling salesman problem (stage 1) and then using the resulting tour as the initial tour in the Hamiltonian path problem (stage 2). This procedure is carried out separately for the set of machines and the set of parts. The resulting ordering of machines and parts is used to reorder the machine-part incidence matrix. The solution obtained is independent of the initial arrangement of the matrix.

Kusiak and Chow (1987) present two algorithms for identifying part families and machine cells via re-arranging the machine-part incidence matrix. The first algorithm, known in the literature as the Cluster Identification Algorithm (CIA), is a 'line drawing' approach which identifies clusters in the matrix in an iterative fashion. This algorithm is reported to be very computationally efficient when the block diagonal structure is embedded in the matrix. In the more general case where exceptional elements are present, the authors propose the use of their Cost Analysis Algorithm, which is an extension of the CIA. Given the additional input of part subcontracting cost, this algorithm will identify independent part families and machine cells and a set of parts to be subcontracted with the minimum sum of subcontracting costs.

In general, array-based clustering techniques over simplify the cell formation problem. These techniques are based solely upon the 0-1 machine-part incidence matrix and therefore ignore many important aspects of the problem (e.g. processing time, machine capacity, and costs). By ignoring production volume, each component is implicitly considered equally important. However, these techniques could be used to obtain a quick, preliminary cellular design.

2.4.2 Math Programming Techniques

Askin and Chiu (1990) formulate the part family-machine cell formation problem as a mixed integer program with binary variables. The objective function is the minimization of four costs: machine overhead, cell overhead, family tooling, and intercell material handling costs. The inputs to the program include setup costs, raw material costs, inventory holding costs, processing times, setup times, part demands, material handling costs, and machine capacities. Because of the large number of integer and binary variables in the problem formulation, the authors present an alternate four-step formulation. In Step 1, batch sizing decisions are made considering setup costs and inventory costs for cycle inventory and WIP. Step 2 allocates part operations to specific machines subject to machine capacity. Step 3 employs a graph partitioning procedure to assign machines to cells. Step 4 investigates the cost trade-off in reducing intercell movement by duplicating machines.

Boctor (1991) formulates the cell formation problem as a mixed integer linear program. The program allows the designer to place restrictions on the number of cells formed and their sizes. The objective of the procedure is to minimize the number of exceptional parts. Inputs to the procedure include the machine-part incidence matrix, the maximum and/or minimum number of machines per cell, and the number of cells to be formed.

Rajamani *et al.* (1990) formulate three integer programming models to address the cell formation problem in the presence of alternative process plans. Model I simply assigns machines to parts whereas Model II assigns machines to known part families to form cells. Model III forms part families and machine cells simultaneously. All three models consider alternative process plans, part demand, processing times, and resource constraints. The objective in each model is the minimization of capital investment subject to the formation of independent cells.

Nagi *et al.* (1990) incorporate the problem of selecting among alternative process plans in their two stage algorithm for solving the cell formation problem. The first stage is formulated as a linear program. Given an initial partition of machines into cells, the linear program assigns part demand to one or more alternative routings with the objective of minimizing intercell movement. The second stage of the algorithm attempts to further reduce intercell movement by merging cells subject to restrictions on cell size. The stages are solved iteratively until convergence is achieved. This model includes part processing times and machine capacities but does not allow for machine duplication. The solution obtained is dependent on the initial partition of machines, but the authors outline a method for arriving at a good initial partition.

Sankaran (1990) addresses the multi-objective nature of the cell formation problem by formulating the problem as a goal program. In Sankaran's formulation, goals are set with respect to:

1. Operating costs
2. Capital investment costs
3. Machine similarity (with respect to part families)
4. Tool similarity (with respect to machine cells)
5. Machine capacities / utilization
6. Material handling capacities.

The objective of the goal programming problem is to minimize the weighted sum of deviations away from the desired goal targets. Inputs to this model include operating costs, machine investment costs, part demand and processing times, machine capacities, and material handling capacities.

Shafer and Rogers (1991) also present a goal programming formulation for the cell formation problem. Four objectives are considered in their formulation:

1. Minimizing set up time
2. Minimizing intercell movements
3. Minimizing investment in new machines
4. Maintaining acceptable machine utilization levels.

With respect to set up times, the authors recognize that, in some cases, part set up time will be dependent on the sequence in which parts are processed and incorporate this in their model. Thus the results of the goal programming formulation would include a processing sequence for parts within a cell in addition to the formation of part families and machine cells. The limitation of the authors' formulation is that it cannot be solved directly. A two stage heuristic is presented that attempts to address three of the four aforementioned goals. The first stage is formulated as a linear program and forms the part families and machine cells with the objective of minimizing the cost of new machines and the cost of intercell movement. The second stage determines a part processing sequence for each cell such that set up times are minimized.

Shtub (1989) formulates the cell formation problem as a generalized assignment problem. This model considers alternative process plans, processing times, machine capacities, and processing costs. The objective of the model is to minimize total processing cost subject to available machine capacities. Missing from this model is the inclusion of part demand and penalties for intercell movement.

Jain *et al.* (1991) address the combined problem of cell formation and machine tool provisioning. Modeled as a 0-1 integer programming problem, the objective of the formulation is to minimize overall system cost. Overall system cost is defined as the sum of processing costs, tool costs, and the cost of machines. Inputs to the model include part demand, part tooling requirements and processing times, machine capacities, processing costs, tool lives and costs, and machine costs. The outputs of the model are the assignment of parts to cells and the cell compositions in terms of the number of machines and copies of tools required to process the part family. The model also allows for alternative process plans.

Gunasingh and Lashkari (1989) present a two stage approach to the cell formation problem in which the machine allocation problem and the part allocation problem are formulated as linear 0-1 integer programs. The machine allocation problem is solved first with the objective of maximizing the sum of similarity indices among machines in a cell. Pair-wise machine similarity is expressed in terms of their capability to process a set of parts requiring both machines. Given the resulting machine cells, the parts allocation problem is solved with the objective of maximizing the sum of the compatibility indices of all parts assigned to cells. The compatibility index of a part with respect to a machine cell is a measure of the cell's ability to process the part. The inputs to this model are the part processing requirements and available machines and tools.

In a later paper, Lashkari and Gunasingh (1991) expand the above model to include alternative process plans and processing times. A non-linear 0-1 integer programming formulation is presented and then decomposed into two linear 0-1 integer programming problems that can be solved iteratively. The objective of both sub-problems is the maximization of a similarity index. The authors define two such indices: similarity in terms of tooling requirements and similarity based upon machine utilization. Given an initial partition of parts into families, the first sub-problem is solved to obtain the best allocation of machines into cells. Given the composition of the cells, the second sub-problem is solved to obtain the best allocation of parts to cells. This procedure is repeated until the solutions converge.

Leung *et al.* (1993) formulate the cell formation problem under material handling constraints as a linear integer program with the objective of minimizing the sum of machine processing cost, tool cost, and material handling cost. In this model, the location of machine work centers is known and the problem consists of allocating tools and parts to the work centers. The inputs to this model are part processing requirements and demand, machine capacities and magazine capacities, distances between work centers, tool life, and material handling capacity and cost.

2.4.3 Graph Theory and Network Analysis Techniques

Kumar *et al.* (1986) formulate the cell formation problem as a *k*-decomposition problem. An undirected graph is constructed in which parts and machines are represented by vertices and the edges represent the requirement of machines for each part. The edges are weighted by the volume of products to be processed. The authors present a heuristic procedure for decomposing the graph into *k* subgraphs with the objective of minimizing intercell moves by minimizing the sum of the edges cut. A second heuristic procedure is present to improve upon an existing *k*-decomposition. In both heuristic procedures, the problem is modeled as a linear transportation problem in which the linear coefficients correspond to the *k* largest eigenvalues (and associated eigenvectors) of the weighted node-to-node connection matrix.

Al-Qattan (1990) uses a method based on network analysis to form machine cells and part families. The method is based on branching from a seed machine and bounding on a completed part. The machine-part incidence matrix is the only input to this method. The objective of the method is to minimize the duplication of machines in the formation of independent cells. The algorithm begins by identifying the set of bottleneck machines. The seed machine is chosen as the one processing the smallest number of jobs. A branch is created from the seed machine for each part processed by the machine. All parts not requiring another machine are bounded, else branches are created for all other required machines. All machines belonging to the bottleneck group are bounded. Branching continues until all nodes are bounded. The group of machines and parts in the tree form a machine cell. A new seed machine is selected from the set of unassigned machines and the algorithm is repeated to form another cell. This process continues until all machines and parts have been assigned to a cell. Alternative solutions can be obtained by selecting different seed machines and parts and subsequently compared by incorporating cost information.

2.5 Non-constant Product Demand and Product Mix

A cellular manufacturing system tends to perform best under the product mix and demand for which the system was designed. When changes in demand/mix occur, many of the benefits of cellular manufacturing can be lost. Ideally, these changes should be estimated and incorporated in the initial design of the system. Two algorithms have been developed that include non-constant demand/mix in the PF/MC formation problem.

Seifoddini (1990) addresses the uncertain nature of product mix in his probabilistic PF/MC formation technique. Given a set of possible product mixes and the probabilities of occurrence, the algorithm seeks to create part families and machine cells such that the system will perform well (although not optimally) with respect to the design objective, regardless of which product mix is being used. The design objective in this model was to minimize the expected intercellular handling cost of the system.

The inputs to the algorithm are the set of product mixes to be considered and the associated probabilities of occurrence. The algorithm begins by solving the PF/MC formation problem for each product mix with the objective of minimizing intercellular material handling costs. Any appropriate PF/MC formation technique could be used to obtain the individual designs. The next step is to evaluate each system design with respect to all product mixes. Let

L = number of product mixes (and, therefore, cell designs)
ICM_{nj} = intercellular material handling cost for cell design n under product mix j
P_j = probability of having product mix j
$EIMC_n$ = expected intercellular material handling cost for cell design n

For each system design, the intercellular material handling costs under different product mixes are computed and used to determined the expected intercellular handling cost as follows:

$$EIMC_n = \sum_{j=1}^{L} IMC_{nj} \cdot P_j$$

The best design is then identified as the system with the smallest expected intercellular handling cost.

The probabilistic PF/MC formation technique provides the opportunity for including the uncertainty associated with product mix in the cell design process so that a robust assignment of parts and machines to cells can be made. The primary drawback of this technique is that only the optimal designs for each product mix are considered. It is possible that a system design exists with a lower expected intercellular material handling cost over all product mixes, even though it is not optimal with respect to any individual product mix.

Harhalakis *et al.* (1994) also include product demand variations in their PF/MC formation technique. Unlike Seifoddini, who looked at multiple product mixes for a single period, Harhalakis *et al.* focus on product demand changes over a system design horizon, which is broken into elementary time periods. The objective of their algorithm is to arrive at a cellular design that minimizes the expected intercell material handling cost over the entire design horizon.

The algorithm begins by mapping the forecast of product demand to a set of feasible production volumes, given resource capacity constraints. If sufficient capacity exists to produce all products at their demand level, then the product demands are the feasible production volumes. Otherwise, the projected demands are used in a linear program that finds a set of feasible production volumes (product mix) such that profit is maximized. Given several product mixes, a procedure for calculating the joint probabilities for every feasible production mix is presented. The joint probabilities are used to evaluate the mean production volume for each product. This step is repeated for each elementary time period.

The mean production volumes are used in a cell formation technique (originally presented by Harhalakis *et al.*, 1990) that seeks to minimize the intercell traffic over the entire design horizon. This technique begins by placing each machine in a separate cell and computing the resulting intercell traffic. Cells are then systematically aggregated until either no feasible aggregation can be made (due to cell size constraints) or the traffic between cells is equal to zero.

This cell formation technique incorporates both the uncertainty associated with demand in a single period and changes in demand over multiple periods. The objective of the algorithm is to minimize the mean-value of intercell traffic over the entire design horizon given uncertain demand. Alternate product routings were not considered; however, the authors emphasized the need to incorporate alternate routings in cellular manufacturing system design under time varying demand.

Once a cellular system has been designed and implemented, unplanned for changes in product mix and/or demand may result in poor system performance. Under these circumstances, it is necessary to reevaluate the cellular system design and make improvements when necessary. Vakharia and Kaku (1993) developed a system redesign methodology to handle long-term demand changes. The methodology consists of reallocating parts that are either new or whose demand has changed among the existing cells. The machine population of each cell remains unchanged as does the assignment of parts whose demand has remained the same. The objective of the reallocation process is to minimize both the cost of intercell traffic and the cost of acquiring additional equipment.

In the same paper, Vakharia and Kaku present the results of a study aimed at assessing the impact of long-term demand changes on a cellular manufacturing system. For several initial system designs, the performance of the system under changing product volumes and product mixes was measured with respect to total intercell moves, the acquisition of additional equipment, and the total cost (intercell moves and additional equipment). They found that, in general, product volume changes increased duplication of equipment. In designs having low routing flexibility (small, highly specialized cells), changes in part mix increased intercell traffic more than changes in product volume. In designs having high routing flexibility (fewer, larger cells), changes in product volume increased intercell traffic more than changes in product mix. Overall, they found the most robust designs to be those having fewer cells.

2.6 Genetic Algorithms

Genetic algorithms are a class of global optimization techniques that utilize concepts from the field of biological genetics. Developed by Holland (1975), genetic algorithms combine survival of the fittest among solution structures with a structured, randomized search strategy in which new, strong solutions displace weak solutions in successive iterations. The following sections give an overview of how a genetic algorithm works; the control parameters; a simple example; and various applications of genetic algorithms.

2.6.1 The Genetic Algorithm Procedure

Genetic algorithms work by maintaining a population of possible solutions to the problem. Each solution is coded and represented by a string. The algorithm begins with an initial population of solutions whose fitness is measured via an evaluation function. The next population of solutions is obtained by first selecting parent structures in proportion to their fitness. By selecting fit parents, it is hoped that desirable solution characteristics will be repeated in future generations while undesirable characteristics die out. The parent structures are altered via genetic operators such as crossover and mutation to produce children. The fitness of the children is measured and the new population replaces the old population. This procedure repeats until a stopping criterion is met (e.g., convergence or maximum number of generations). The following is a high level pseudo-code of the genetic algorithm procedure.

```
Initialize Population → P(0)
Evaluate P(0)
Set t = 1
Loop
        Select P(t) from P(t-1)
        Reproduce P(t) via crossover and mutation
        Evaluate P(t)
        t = t + 1
Until (stopping rule)
```

The key elements of genetic algorithms are three operators: reproduction, crossover and mutation. The purpose of the reproduction operator is to select structures from the current population to be parents for the next generation. Structures are selected according to their fitness function values; structures with higher values have a higher probability of contributing one or more offspring (survival of the fittest).

The crossover operator is used to create offspring from the parent structures. A subset of the two parent structures is exchanged (crossed-over) to form two new structures. The mutation operator arbitrarily alters the value of one or more parameters of a selected structure. The crossover and mutation operators serve to introduce new information into the population and to avoid premature convergence.

Many biological terms are used in the genetic algorithm literature. A set of potential solutions are called a population, and the population at any given time is called a generation. Individual solutions in the population are represented by a series of characters (genes) and are known as chromosomes. The following table (Goldberg, 1989) summarizes the correspondence between natural and artificial terminology.

TABLE 2-1
Correspondence Between Natural and Genetic Algorithm Terminology

Natural	Genetic Algorithm
population	set of strings
chromosome	string
gene	feature, character, parameter
allele	parameter value
locus	string position

There are four major differences between genetic algorithms and traditional optimization techniques (Goldberg, 1989).

1. Genetic algorithms work with the coding of the parameter set, not the parameters themselves.
2. Genetic algorithms work from a population of points, not a single point.
3. Genetic algorithms use payoff (objective function) information to guide

the search, not derivatives or other auxiliary knowledge.

4. Genetic algorithms use probabilistic transition rules, not deterministic rules.

Genetic algorithms are implicit parallel processors in that they analyze and modify a population of solutions instead of a single point in the search space. The use of information from a population of points combined with a probabilistic search strategy enables genetic algorithms to come out of local optima and explore regions of the solution space that are likely to contain improvements.

2.6.2 Control Parameters for Genetic Algorithms

The key elements of genetic algorithms are the operators of reproduction, crossover, and mutation. The effectiveness of the genetic algorithm will depend on how these operators are applied to produce new generations of solutions. This section describes some of the work that has been done to optimize the control parameters of genetic algorithms.

Grefenstette (1986) characterizes a particular subclass of genetic algorithms by the following six parameters.

1. Population Size (N)
2. Crossover Rate (C)
3. Mutation Rate (M)
4. Generation Gap (G)
5. Scaling Window (W)
6. Selection Strategy (S)

Thus, a particular genetic algorithm in this subclass can be identified using the notation:

$$GA(N, C, M, G, W, S).$$

Population size refers to the number of candidate solutions initially created and maintained. In most applications of genetic algorithms, the population size remains constant. If the population is too small, the algorithm will perform poorly due to an insufficient sample size. A large population allows for a better search but increases the number of computations required and the time to convergence.

The crossover rate refers to the frequency with which the crossover operator is applied to create new structures. A low crossover rate may stagnate the search due to a low exploration rate. High crossover rates quickly introduce new information, however good solution structures may be lost before they can be improved.

The mutation rate refers to the probability that an individual parameter value will arbitrarily be changed. Mutation prevents the value of any parameter from forever remaining unchanged. As the mutation rate increases, the search becomes more random and the probability of a good solution remaining intact decreases.

The generation gap refers to the proportion of the current population that gets replaced by the new population. A value of G = 1.0 indicates that the entire population is replaced during each generation while a value of G = 0.5 means that half of the current population survives to the next generation.

When maximizing a function, it is common to scale the performance of a particular solution before comparing it with other solutions. A common linear scaling procedure is:

$$f_s(x_i) = f(x_i) - f_{min}$$

where f_{min} is the smallest possible value that $f(x)$ can take on in a given search space. If f_{min} is not known, it is common to use the minimum value observed thus far. Grefenstette (1986) defines the scaling window in terms of three ways of obtaining f_{min}. If W = 0, f_{min} is set to the smallest fitness value observed in the initial population. If W = 7, no scaling is performed. If 1 < W < 7, then f_{min} is defined as the smallest fitness value observed in the last W generations. The scaling window can help control premature convergence of the genetic algorithm.

The last control parameter is the selection strategy. Grefenstette's experiment explored two selection strategies. If S = P, then a pure selection strategy is performed where each structure in the current population receives copies in the mating pool in proportion to its relative fitness. When S = E, an elitist strategy is being performed. The elitist strategy begins with pure selection, but adds the stipulation that the current best structure survives intact to the next generation. This strategy ensures that the current best structure does not disappear due to random chance or recombination operations (crossover and mutation).

De Jong (1975) suggests the following parameter settings which have been used in a number of genetic algorithm applications.

$$GA_D = GA(50, 0.6, 0.001, 1.0, 7, E)$$

Grefenstette (1986) performed an experiment to identify optimal control parameters for this subclass of genetic algorithms with respect to two performance measures. The first measure was on-line performance which is defined as the average performance of all structures evaluated during the course of the search. The second measure was off-line performance which is defined as the best performance achieved in a predefined time interval. The experiment identified the following optimal control parameters with respect to on-line (GA_{G1}) and off-line (GA_{G2}) performance.

$$GA_{G1} = GA(30, 0.95, 0.01, 1.0, 1, E)$$
$$GA_{G2} = GA(80, 0.45, 0.01, 0.9, 1, P)$$

These parameter values can be used as a starting point for new genetic algorithms. Through experimentation, the parameter values can be tailored for a specific application.

2.6.3 A Simple Genetic Algorithm

Consider the problem of maximizing the function $f(x) = x^3$. Furthermore, let x be restricted to take on values between zero and 63. The following paragraphs describe a simple genetic algorithm that can be used to solve this problem.

Genetic algorithms work with a string representation of the parameters. In this example there is only one parameter, x, which can be represented by a six digit binary string (e.g., 010110). To begin the genetic algorithm, an initial population of solutions is needed. A population of six strings can be obtained by flipping a fair coin 36 times, where heads = 1 and tails = 0. The initial population is then evaluated by decoding the values of x and using these values in the fitness function $f(x) = x^3$. This binary string representation can be decoded to obtain the base 10 value of x in the following manner:

$$x = 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 22.$$

The corresponding fitness value is $f(x) = 10{,}648$. Table 2-2 summarizes the initial population of strings and their fitness values.

TABLE 2-2
Initial Population and Fitness Values

String No.	Initial Population	x_i value	Fitness Value $f_i = x_i^3$	Proportion of Total $f_i / \Sigma_i f_i$	Expected Copies f_i / f_{avg}	Actual Copies
1	1 0 0 1 1 0	38	54,872	0.308	1.849	2
2	0 1 1 0 0 1	25	15,625	0.088	0.527	1
3	1 0 1 0 0 1	41	68,921	0.387	2.323	2
4	0 1 0 0 1 1	19	6,859	0.050	0.299	0
5	0 0 1 1 1 0	14	2,744	0.015	0.092	0
6	0 1 1 1 1 0	30	27,000	0.152	0.910	1
Sum			178,021	1.000	6.000	6
Average			29,670	0.167	1.000	1
Maximum			68,921	0.387	2.323	2

Once the initial population has been generated and its fitness evaluated, parent structures are selected to join the mating pool in proportion to their relative fitness, $f_i / \Sigma_i f_i$. The number of times a structure is copied and added to the mating pool should be approximately proportional to the structure's fitness relative to the rest of the population. The actual number of copies of each structure in the mating pool was obtained by forming a probability density function for the proportion of total fitness and generating six random numbers between 0 and 1. Each time a random number was generated, the corresponding structure was copied and added to the mating pool.

Once the mating pool has been established, structures are randomly paired for mating. Each pair of parent structures undergoes crossover (with a probability equal to the crossover rate) in the following manner. First, an integer position k along the structure length (l) is selected at random ($1 \leq k \leq l - 1$). Two children structures are then created by exchanging the characters beyond the crossover point, inclusively.

The last operator to employ before the new generation is finalized is the mutation operator. Mutation is performed on a bit-by-bit basis. For this example, the probability of mutation is taken to be 0.001 (De Jong, 1975). Simulating the mutation operator resulted in no bit positions being changes from 0 to 1 or vice versa. Note that with 36 bit positions in the population, only (0.001)(36) = 0.036 bits are expected to undergo mutation in each generation. The new generation, P(1), is shown in Table 2-3.

TABLE 2-3
Mating Pool, Crossover Sites, and New Population

Mating Pool After Reproduction	Mate	Crossover Site	New Population	x_i value	Fitness Value $f_i = x_i^3$
1 0 0 1 1|0	2	5	1 0 0 1 1 1	39	59,319
0 1 1 0 0|1	1	5	0 1 1 0 0 0	24	13,824
1|0 0 1 1 0	4	1	1 0 1 0 0 1	41	68,921
1|0 1 0 0 1	3	1	1 0 0 1 1 0	38	54,872
1 0 1 0|0 1	6	4	1 0 1 0 1 0	42	74,088
0 1 1 1|1 0	5	4	0 1 1 1 0 1	29	24,389
Sum					295,413
Average					49,236
Maximum					74,088

In one generation, the average fitness of the population increased from 29,670 to 49,236. The maximum fitness improved from 68,921 to 74,088 in the same time period. By continuing the operations of reproduction, crossover, and mutation to create successive generations of solutions, the average fitness (and maximum fitness) should approach the optimum value of $f(63) = 250,047$.

2.6.4 Applications of Genetic Algorithms

Genetic algorithms have been successfully applied to a wide variety of problems in the physical and social sciences as well as engineering. Davis (1985) and Biegel and Davern (1990) outline job shop scheduling procedures using genetic algorithms. Goldberg (1983) applied genetic algorithms to the optimization of pipeline systems. Other applications include the traveling salesman problem (Grefenstette, *et al.*, 1985), the multi-vehicle routing problem (Nygard and Kadaba, 1990), and communications networks (Davis and Coombs, 1987).

The literature contained two examples of the use of genetic algorithms to solve the PF/MC formation problem. Venugopal and Narendran (1992) developed a bi-criteria mathematical formulation of the problem and a genetic algorithm to solve it. The objectives considered in their model were the minimization of the volume of intercell moves and the minimization of total within cell load variation.

Two independent sub-populations of solutions were maintained by the genetic algorithm, one sub-population for each objective. Each potential solution was represented by a *machine-cell chromosome*. The length of the chromosome was equal to the size of the machine population. The string position corresponded to the machine number and the value in each position indicated the cell to which the machine had been assigned. The initial population for each objective was randomly generated. The creation of successive populations was carried out separately for each sub-population. Strings were selected for mating according to their fitness function value and the genetic operators of crossover and mutation were used to create new solutions.

This approach lacks a formalized procedure for evaluating the trade-off between the objectives. In essence, the approach taken was to solve two, single objective problems with the idea that a single solution that performed satisfactorily with respect to each objective could be found. If a single solution that is optimal with respect to both objectives cannot be found, the 'best' solution must be determined subjectively.

Joines *et al.* (1995) developed an integer programming formulation of the the PF/MC formation problem and a genetic algorithm to solve it. The objective of their model was to maximize grouping efficacy. The algorithm was tested on data sets from the literature and was found to be an effective tool for solving the PF/MC formation problem.

2.7 Summary

Numerous analytical methods for solving the PF/MC formation problem have been developed. The primary input to these approaches is part processing information. A review of the relevant literature finds existing models to be lacking in two areas. The first area concerns the modeling of the system. There are many relevant factors of the production system to be included in the model. Current models address only a subset of these factors. In addition, the majority of existing cell formation techniques address a single objective only.

The second deficient area is the problem of changes in the part and machine populations. The majority of existing models consider the part population to be constant and thus do not include the phase in/phase out of products in the design of the system. No existing techniques explicitly consider changes in resource capacity on a period by period basis.

Due to the complexity of the cell formation problem and its mathematical model, traditional optimization techniques may not be an efficient means for solving the problem. Genetic algorithms have proven to be a useful technique for solving complex problems, including structural design, vehicle routing, job shop scheduling, and neural network design. The application of genetic algorithms to the optimization of the cellular design problem seems promising.

3.0 A Multi-Period Model for PF/MC Formation

The objective of this dissertation is to develop a new cellular manufacturing system design methodology for the formation of part families and machine cells. It is necessary that the design methodology be capable of addressing multiple design objectives and the dynamic nature of the production environment. The design methodology developed includes multiple objectives in the design evaluation function and a forecast of product mix and resource availability in the set of system parameters. The result of applying this methodology is a preliminary cellular system design that performs well with respect to the design objectives over the entire planning horizon.

This chapter begins with a mathematical formulation of the multi-period PF/MC formation problem. The design objectives are identified; the system parameters and constraints are defined; and a mixed-integer formulation of the problem is developed. Next, a solution methodology based on a genetic algorithm is presented. The primary outputs of the methodology and its uses are discussed. The results of the genetic algorithm are then validated for the single period case and a simple multi-period case.

3.1 The Multi-Period PF/MC Formation Problem Formulation

In this section, a mathematical formulation of the multi-period PF/MC formation problem is presented. Before a solution technique can be developed, the problem must be clearly defined in terms of the design objective, system parameters, and system constraints. The following assumptions were made in the development of the model.

1. The subset of parts to be produced in the cellular system has been identified.
2. Process plan selection has already taken place.
3. Parts have fixed operation sequences with known processing times.
4. The production volumes for parts are known for all periods in the planning horizon.
5. Parts are to be produced at their specified volumes.
6. The cost of transporting parts between cells is known.

7. The acquisition cost of each resource is known.
8. The capacity of each resource is known.
9. The number of available resources is known.
10. The cost of moving machines between cells is known.
11. There is no cost associated with rearranging part families.

The following sections describe the elements of the problem formulation (objectives, parameters, and constraints) and the resulting mathematical model.

3.1.1 The Design Objectives

As was indicated in Chapter 1, there are numerous objectives to be met when designing a cellular manufacturing system. However, it would be difficult to explicitly consider all of these objectives in the PF/MC formation problem. In this dissertation, the objectives included in the problem formulation have been limited to those for which design performance can be measured (or estimated) in the early phase of preliminary system design. Three objectives have been included in the multi-period formulation of the PF/MC formation problem. These objectives are:

1. minimize intercell transfers of parts;
2. minimize duplication of machines; and
3. minimize between-period reconfiguration of cells.

The minimization of intercell transfers is, perhaps, one of the most important objectives in cellular manufacturing system design. Intercell transfers occur when a part requires processing on a machine that is not located in the part's primary cell. Intercell transfers decrease the efficiency of the cellular system by increasing material handling requirements, lengthening flow lines, and complicating production control. Thus, minimizing intercell transfers is fundamental to achieving many of the benefits associated with cellular manufacturing.

The importance of minimizing intercell transfers is reflected in the objectives used by existing PF/MC formation techniques. Two common objectives are the minimization of intercell transfers and the minimization of machine duplication. Many models consider only one of these objectives. In models where minimizing intercell transfers is the sole objective, a cellular system is designed without explicitly considering the duplication of machines. Models seeking to minimize machine duplication frequently seek to form independent cells (i.e., no intercell transfers are allowed) at a minimal cost.

Clearly these objectives are conflicting. Intercell moves can be minimized by duplicating machines and machine duplication can be minimized at the expense of increased intercell transfers. The formulation of the PF/MC formation problem presented in this dissertation allows for a tradeoff between intercell transfers and machine duplication by including both objectives in the design evaluation function.

The last objective, minimize system re-configuration, is particular to the multi-period formulation of the PF/MC formation problem. System re-configuration is defined as changing the composition of the machine cells by moving machines from one cell to another in subsequent periods. In the presence of non-constant product mix and resource availability, the best cellular design in one period (with respect to the first two objectives) may not be an efficient cellular design for subsequent periods. By re-configuring the machine cells, the cellular system can continue to operate efficiently as the product mix and resource population change.

While system re-configuration can help maintain efficiencies, it is not without its drawbacks. Moving machines from cell to cell requires effort and can lead to the disruption of production. The designer must now consider the tradeoffs between system re-configuration and increased intercell transfers and/or increased machine duplication.

Including the minimization of system-reconfiguration as a design objective helps guides the search for cellular designs that are low in intercell transfers and machine duplication without requiring extensive changes to the system over the planning horizon. It is very unlikely that a single solution will be optimal with respect to all of the individual objectives. As a result, tradeoffs must be made to find a solution that performs satisfactorily with respect to all three objectives.

To facilitate the integration of these objectives and to formalize the tradeoffs being made, system performance with respect to each objective will be measured in dollars. In cost terminology, the three objectives can be re-written as:

1. minimize intercell material handling costs;
2. minimize capital investment in machines; and
3. minimize machine relocation costs.

Material handling cost is defined as the cost of transferring parts between cells. Capital investment is defined as the cost of acquiring additional machines. Relocation cost is defined as the cost of physically moving equipment as well as any cost associated with the disruption of production.

Now the tradeoffs being made between the design objectives can be easily interpreted. A machine will be duplicated only if the reduction in material handling cost is greater than the acquisition cost of the machine. Machines will be relocated in response to a changing product mix only if the cost of doing so is justified by a decrease in material handling costs and/or machine duplication costs. The overall objective of the multi-period PF/MC formation problem is to minimize total system cost.

3.1.2 System Parameters and Constraints

Evaluating the performance of a proposed cellular design with respect to the design objectives requires the specification of several system parameters. The values of the following parameters must be supplied for each period in the planning horizon.

1. **Product Mix** - The product mix is defined as the set of parts to be produced within the cellular system. The product mix may vary from period to period as new parts are introduced and older parts are discontinued.

2. **Production Volume** - Production volume is defined as the quantity of each part in the product mix to be produced each period. It is expected that the production volume for an individual part will vary across periods according to its stage in the production life cycle. Production volumes are used to determine minimum resource capacity requirements and the number of intercell transfers.

3. **Operation Sequence** - The operation sequence for a part is defined as an ordered list of the machine types the part must visit to complete its processing. The operation sequence should include any non-consecutive operations on the same machine type. Operation sequences are used to obtain a more realistic estimate of intercell transfers than is provided by using the binary machine-part incidence matrix.

4. **Processing Time** - Processing time is defined as the time required to perform an operation on a part. The processing time should include the set-up time in addition to the actual operation time. Part processing times need to be specified for all machine types in the operation sequence. In the case of multiple operations on the same machine type, the sum of the corresponding processing times for each operation should be used. Processing times are used to determine minimum resource capacity requirements.

The values of the following system parameters must be supplied and are assumed to remain constant over the planning horizon.

5. **Available Resources** - The available resources are defined as the set of machines that will be used to form the manufacturing cells. The number of each type of machine that is available for production at the start of the planning horizon must be specified.
6. **Resource Capacity** - The capacity of a resource is defined as the amount of time a machine is available for production in each period.

The following system parameters represent cost data used to evaluate the system design with respect to the objective of minimizing total system cost. These costs are specified on a period by period basis.

7. **Resource Acquisition Cost** - Resource acquisition cost is defined as the total cost of purchasing and installing an additional unit of a machine type. In each period, the resource acquisition cost should reflect any estimated price changes over the planning horizon.
8. **Material Handling Cost** - Material handling cost is defined as the per unit cost of transferring a part between two cells. Material handling costs may vary from part to part and should reflect any individual handling needs.
9. **Machine Relocation Cost** - Machine relocation cost is defined as the cost of removing a machine from one cell and installing it in another. Relocation costs may differ between machines due to the type of equipment involved and may change over time.

The multi-period PF/MC formation problem is a constrained optimization problem. The following constraints are placed on candidate solutions.

1. **Every part must be assigned to a primary (parent) cell.** It is assumed that every part defined in the product mix is to be produced by the cellular system. A parent cell designation is needed for all parts as a basis for estimating intercell transfers. Furthermore, each part is assigned to exactly one part family in periods for which demand for the part exists. However, the composition of the family and the cell to which the family is assigned may change with each period.

2. **There must be sufficient resource capacity to produce the product mix at the specified production volumes.** This model does not consider sub-contracting or backorders to compensate for insufficient capacity. Instead, additional machines will be acquired if necessary.

3. **Within each cell, there must be sufficient capacity of all machine types assigned to the cell to produce the members of the associated part family at their specified production volumes.** A part will not be transferred to another cell for processing if the required machine type exists in the parent cell. For example, suppose that machine type X has been assigned to cell Y. Part family Y consists of parts A, B, F, and H. Parts B, F, and H all require processing on machine type X. This constraint ensures that there exists sufficient capacity of machine type X in cell Y to process the combined production volumes of parts B, F, and H.

4. **Each machine cell must contain a pre-specified minimum number of unique machine types.** Given the objectives of minimizing material handling, capital investment, and machine relocation costs, the "optimal" solution is to place all machines in a single cell. A lower bound on the number of unique machine types is necessary to guard against the formation of empty cells. As a general rule-of-thumb, the lower bound should not be below the smallest number of unique machine types required by any part in the product mix.

5. **Each part family must contain a pre-specified minimum number of parts.** This constraint prevents the formation of a machine cell that has no parts to produce.

Supplementary constraints include capital investment budgets and upper limits on the size of part families and machine cells. Realistically, there is likely to be a limited amount of funds available for acquiring additional equipment in each period. However, care must be taken to ensure that a feasible solution will exist given a constraint on investment and the necessary constraints on capacity.

3.1.3 The Mathematical Model

In this section, a mixed-integer formulation of the multi-period PF/MC formation problem is presented. The primary purpose of this model is to serve as a blueprint for the development of the genetic algorithm to solve the problem. The following notation is used to develop the mathematical representation of the objective function and design constraints.

The indexing set used to differentiate between parts, machines, cells, and periods is as follows.

N = set of parts
i = index of parts, $i = 1, 2, ..., N$
M = set of machines
j = index of machines, $j = 1, 2, ..., M$
C = set of cells
k = index of cells, $k = 1, 2, ..., C$
P = set of periods
l = index of periods, $l = 1, 2, ..., P$

The notation being used for the system parameters described in the previous section is as follows.

D_{il} = demand (production volume) for part i in period l
S_{il} = number of processing operations for part i in period l
$O(i,r,l)$ = machine type required by the r^{th} operation on part i in period l
T_{ijl} = processing time of part i on machine type j in period l
M_j = number of type j machines available at start of planning horizon

C_j = capacity of machine type j

P_{jl} = cost of acquiring a type j machine in period l

H_{il} = intercell per unit material handling cost for part i in period l

R_{jl} = cost of relocating machine type j in period l

LM = minimum number of machines per cell

LP = minimum number of parts per family

A = a large number

The primary decision variables of the multi-period PF/MC formation problem are the assignment of parts to cells (families) and the assignment of machine types to cells. In addition, given that a machine type has been assigned to a cell, the number of units of the machine type assigned to the cell must also be determined. Let

$$x_{ikl} = \begin{cases} 1 & \text{if part type } i \text{ is assigned to cell } k \text{ during period } l \\ 0 & \text{otherwise} \end{cases}$$

$$y_{jkl} = \begin{cases} 1 & \text{if machine type } j \text{ is assigned to cell } k \text{ during period } l \\ 0 & \text{otherwise} \end{cases}$$

n_{jkl} = number of type j machines assigned to cell k in period l

The following notation is introduced to simplify the expression of the objective function. Let

q_{il} = number of intercell transfers that occur during the production of part i during period l

b_{jl} = number of additional type j machines acquired at the beginning of period l

u_{jl} = number of type j machines that are relocated between period l-1 and period l

Using this notation, the objective function and system constraints can now be written in equation form. The mathematical model of the multi-period PF/MC formation problem is shown next.

$$MIN\ \ Z = \sum_{l=1}^{P} \left[\sum_{i=1}^{N} (H_{il} \cdot D_{il} \cdot q_{il}) + \sum_{j=1}^{M} (P_{jl} \cdot b_{jl}) + \sum_{j=1}^{M} (R_{jl} \cdot u_{jl}) \right]$$

where

$$q_{il} = \sum_{k=1}^{C} x_{ikl} \left[\sum_{r=1}^{S_{il}-1} (1 - y_{O(i,r,l)kl} \cdot y_{O(i,r+1,l)kl}) \right] \qquad \forall\, i, l \qquad [O1]$$

$$b_{jl} = \max \left\{ 0, \sum_{k=1}^{C} n_{jkl} - M_j - \sum_{s=1}^{l-1} b_{js} \right\} \qquad \forall\, j, l \qquad [O2]$$

$$u_{jl} = \sum_{k=1}^{C} \left[\max \{ 0, n_{jkl} - n_{jk(l-1)} \} \right] - b_{jl} \qquad \forall\, j, l \qquad [O3]$$

Subject to:

$$\sum_{k=1}^{C} x_{ikl} = \min \{ 1, D_{il} \} \qquad \forall\, i, l \qquad [C1]$$

$$\sum_{i=1}^{N} D_{il} \cdot T_{ijl} \cdot x_{ikl} \cdot y_{jkl} \le C_j \cdot n_{jkl} \qquad \forall\, j, k, l \qquad [C2]$$

$$\sum_{i=1}^{N} D_{il} \cdot T_{ijl} \le C_j \sum_{k=1}^{C} n_{jkl} \qquad \forall\, j, l \qquad [C3]$$

$$\sum_{j=1}^{M} y_{jkl} \ge LM \qquad \forall\, k, l \qquad [C4]$$

$$\sum_{i=1}^{N} x_{ikl} \ge LP \qquad \forall\, k, l \qquad [C5]$$

$n_{jkl} \leq y_{jkl} \cdot A$	$\forall\, j, k, l$	[C6]
$x_{ikl}, y_{jkl} = \{0, 1\}$	$\forall\, i, j, k, l$	[C7]
$n_{jkl} \geq 0$, integer	$\forall\, i, j, k, l$	[C8]

The overall objective of the multi-period PF/MC formation problem is to minimize total system cost. Total system cost consists of material handling cost, capital investment, and relocation costs. The first term in the objective function is the material handling cost for the proposed design. Total material handling cost is obtained by multiplying the number of intercell transfers for each part by the per unit material handling cost and the production volume. Equation [O1] shows the computation of the number of intercell transfers per part, per period. An intercell transfer occurs whenever either machine required by consecutive operations is not located in the cell to which the part has been assigned (the primary cell). The scheduling of operations occurring outside the primary cell is not addressed by this formulation. As a result, the actual number of intecell transfers may be less than the number given by equation [O1]. This will happen when consecutive operations that take place outside the primary cell are performed in the same external cell.

The second term in the objective function is the investment in additional machines. Total capital investment is obtained by multiplying the number of each machine type required by the acquisition cost. Equation [O2] shows the computation of the number of machines acquired for each period in the planning horizon. The last term in the objective function is relocation cost. Total relocation cost is obtained by multiplying the number of each machine type relocated by the cost of relocating the machine. Equation [O3] shows the computation of the number of machines relocated for each period in the planning horizon.

Constraint [C1] ensures that each part is assigned to exactly one primary cell and that a part is only assigned to a cell for periods in which demand exists for the part. Constraint [C2] is the within-cell capacity constraint. The total capacity of each machine type assigned to a cell must be sufficient to process the part family assigned to the cell. Constraint [C3] checks the entire system capacity. Both of these constraints are needed because it is possible to satisfy constraint [C2] without satisfying constraint [C3]. This is because constraint [C2] provides within-cell capacity for the part family without considering the capacity needed to perform operations outside of the primary cell. Thus, constraint [C3] is needed to supplement constraint [C2] to ensure that the total system capacity is sufficient to produce all parts at their required volumes.

Constraints [C4] and [C5] enforce lower limits on the number of machines and the number of parts assigned to each cell. Constraint [C6] ensures that the number of units of a given machine type in a cell is equal to zero unless the machine type has been assigned to the cell. Finally, the values of the decision variables are restricted by constraints [C7] and [C8].

3.2 A Genetic Algorithm Based Solution Methodology

The multi-period PF/MC formation solution procedure described in this dissertation belongs to the machine-grouping category of cell formation techniques. In this category, the machines cells are formed first, followed by the assignment of parts to cells. The assignment of machines to cells over the planning horizon is made via a genetic algorithm. Embedded in the algorithm is a heuristic for assigning parts to cells. The mathematical formulation described in the previous section was used as a blueprint for creating the genetic algorithm to solve the multi-period formulation of the PF/MC formation problem.

Genetic algorithms work by maintaining and manipulating a population of potential solutions to the problem. A "survival of the fittest" strategy is employed, in which new, strong solutions displace weaker solutions in the population. The new solutions are obtained by combining information contained in current solutions via genetic operators such as crossover and mutation. Candidate solutions are evaluated with respect to the design evaluation function. The following is a summary of the genetic algorithm used to solve the PF/MC formation problem.

Step 0. Initialize the generation counter, $g = 0$.

Step 1. Randomly generate an initial population of machine assignment solutions, $MA(g)$.

Step 2. For each potential machine assignment solution, apply the part assignment heuristic to form the part family for each cell.

Step 3. Evaluate the fitness of each solution with respect to the design evaluation function.

Step 4. Select individuals from the current population to become parents of the next generation according to their fitness values.

Step 5. Randomly mate parent solutions and create children by applying the genetic operators of crossover and mutation.

Step 6. Evaluate the fitness of each child solution with respect to the design evaluation function.

Step 7. Increment the generation counter, $g = g + 1$.

Step 8. Form the new generation, $MA(g)$, by replacing the weak solutions in population $MA(g - 1)$ with more fit child solutions.

Step 9. If the termination criterion has not been met, go to step 4.

The following sections describe the chromosomal representation of the machine assignment problem and the steps of the algorithm in more detail.

3.2.1 The Chromosomal Representation of Solutions

Unlike more traditional optimization techniques, genetic algorithms work with a coding of the decision variables, not the variables themselves. A chromosomal representation of candidate solutions is required for manipulation by the genetic algorithm in its search for the best solution. The chromosomal structure needs to code the key features of the problem in such a way that good solution characteristics are repeated in subsequent generations while undesirable characteristics die out. For the multi-period PF/MC formation problem, the key feature to be coded in the chromosomal structure is the assignment of machines to cells for each period in the planning horizon. The resulting machine assignment will largely determine the assignment of parts to cells, which in turn determines the number of each machine type assigned to the cells.

The chromosome used in the multi-period PF/MC formation genetic algorithm is divided into periods, since each period in the planning horizon may have a different cellular machine composition due to machine relocation.

[Period 1 Assignment][Period 2 Assignment] ... [Period P Assignment]

Within each period, the chromosome is further divided into cells, representing the composition of each cell in terms of machine type.

[Period l Assignment]
[Cell 1][Cell 2]...[Cell C]

In each cell substring, the string position corresponds to a machine type, and the value in the position (either 1 or 0) indicates whether or not the machine type has been assigned to that cell.

[Cell k]
$[y_{1kl} \;\; y_{2kl} \cdots y_{Mkl}]$

where

$$y_{jkl} = \begin{cases} 1 & \text{if machine type } j \text{ has been assigned to cell } k \text{ in period } l \\ 0 & \text{otherwise} \end{cases}$$

The length of the complete chromosome is $M \times C \times P$, where M is the number of machine types, C is the number of cells, and P is the number of periods in the planning horizon.

For example, consider the following coded machine assignment for a two period, three cell, and seven machine type problem.

[1100111] [0010101] [0011000] | [1010111] [1010100] [1111001]

Table 3-1 shows the decoded machine composition of the cells in each period for this chromosomal representation of a potential solution to the multi-period PF/MC formation problem.

TABLE 3-1
Decoded Machine Assignment Chromosome

	Machine Assignments	
Cell No.	Period 1	Period 2
1	A B E F G	A C E F G
2	C E G	A C E
3	C D	A B C D G

3.2.2 The Part Assignment Heuristic

This section describes the heuristic used within the genetic algorithm to assign parts to cells for each candidate machine assignment in the population. The objective of the part assignment heuristic is to minimize the number of intercell transfers. Given a candidate machine assignment solution, the first step of the heuristic is to compute the number of intercell transfers that would result if part i is assigned to cell k. Cells that do not contain any of the machines in the part's operation sequence are not candidates for assignment. Each part is then assigned to the cell that results in the minimum number of intercell transfers. In the case of a tie, the part is assigned to the cell where the majority of its processing (in terms of operation time) takes place. It should be easier to schedule a shorter operation in another cell. In the event that a tie still exists (e.g., two cells contain all of the machines required by the part), the tie is randomly broken. The following example illustrates the procedure of assigning parts to cells.

Consider the assignment of 14 parts to the machine cells shown previously in Table 3-1. The operation sequences for these parts are shown in Table 3-2. The number of intercell transfers for assigning a given part to each cell is also shown. Consider the period 1 assignment of part 2, which requires processing on machines A and E. The assignment of part 2 to cell 1 would require no intercell transfers. If part 2 is assigned to cell 2, one intercell transfer would be required. Cell 3 is not considered for assignment since it contains neither machine A nor E. Thus, part 2 is assigned to cell 1 to minimize intercell transfers.

In the case of part 9, the same number of intercell transfers occurs if it is assigned to cell 1, 2, or 3 in period 1. This tie is broken by considering the processing time in each cell. As a result, part 9 is assigned to cell 1.

Now consider the assignment of part 7 in period 1. Zero intercell transfers are required if part 7 is assigned to cell 1 or cell 3. To break this tie, the processing time in each cell is computed. In this case, the processing time is the same in each cell because the machines required by part 7 have been assigned to both cells. The tie between cell 1 and cell 3 is randomly broken, resulting in the assignment of part 7 to cell 3.

The final assignment of parts to cells in periods 1 and 2 obtained using the part assignment heuristic is shown in Table 3-3.

TABLE 3 - 2
Part Assignment Heuristic Results for Example Problem

Part	Operation	PERIOD 1							PERIOD 2						
		Intercell Transfers			Processing Time			Assigned	Intercell Transfers			Processing Time			Assigned
No.	Sequence	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3	Cell	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3	Cell
1	B - F - G	0	2	-				1	1	-	2				1
2	A - E	0	1	-				1	0	0	1	2637	2637		2*
3	D	-	-	0				3	-	-	0				3
4	B - G	0	1	-				1	1	-	0				3
5	A - B - G	0	2	-				1	2	2	0				3
6	B - F - G	0	2	-				1	1	-	2				1
7	B - D	1	-	1	1456		1456	3*	-	-	0				3
8	A - E - F	0	2	-				1	0	1	2				1
9	A - D - G	2	2	2	6656	4056	1040	1	2	2	0				3
10	C - G	1	0	1				2	0	1	0	9994		9994	1*
11	C - D	-	1	0				3	1	1	0				3
12	C - D	-	1	0				3	1	1	0				3
13	C	-	0	0		1664	1664	2*	0	0	0	998	998	998	2*
14	A - E	0	1	-				1	0	0	1	2256	2256		2*

* tie was randomly broken

TABLE 3-3
Part Assignment Results

Cell No.	Part Assignments	
	Period 1	Period 2
1	1, 2, 4, 5, 6, 8, 9, 14	1, 6, 8, 10
2	10, 13	2, 13, 14
3	3, 7, 11, 12	3, 4, 5, 7, 9, 11, 12

Recall the constraint on the minimum number of parts assigned to a family. Immediately after the part assignment heuristic is invoked, the number of parts assigned to each cell is counted. If a cell has too few parts assigned to it, parts are randomly chosen from other cells until the constraint is satisfied.

3.2.3 The Evaluation Function

The purpose of the evaluation function is to measure the fitness of candidate solutions in the population with respect to the design objectives. The fitness values are used to select parent solutions used to create the next generation of solutions. The specific form of the evaluation function depends on the set of design objectives being considered. As discussed earlier, the design objectives of the multi-period PF/MC formation procedure are the minimization of intercell movement, machine duplication, and system reconfiguration. The corresponding definition of the fitness of a potential solution is then the sum of material handling costs, machine acquisition costs, and machine relocation costs. Using this definition of fitness, solutions having lower values are more fit than solution with higher values.

In general, genetic algorithms use payoff information to guide the search for good solutions. In other words, the genetic algorithm seeks to maximize fitness. As a result, the minimize cost objective function needs to be transformed into a maximization function. This is accomplished via a normalization process that converts the smallest observed raw fitness value into the largest transformed fitness value. Once the fitness values have been transformed, they are scaled to prevent premature population convergence. The following sections illustrate the computation of raw fitness values, the transformation process, and the scaling process.

3.2.3.1 Material Handling Cost

One of the objectives of the design process is to minimize material handling costs. Additional material handling is required when a part requires processing on a machine that is not located in the cell to which the part has been assigned. This dissertation is concerned only with material handling costs associated with intercell transfers; within cell material handling costs are not considered. Intercell material handling cost may vary from part to part, depending on each part's individual handling needs.

The material handling cost associated with a part is computed by multiplying the number of intercell transfers required to process the part by the per unit material handling cost for the part. The multi-period PF/MC formation procedure considers the operation sequence and the demand of each part when computing the number of intercell transfers. The operation sequence is important, since the transfer of a part to another cell for an intermediate processing operation has a corresponding transfer back to the primary cell to complete its processing. Thus, a total of two intercell transfers are made and should be reflected in the calculation of material handling cost. The following example illustrates the computation of material handling cost.

According to Table 3-3, part 9 belongs to the part family assigned to cell 1. Cell 1 is comprised of machine types A, B, C, E, F, and G. Part 9 requires processing on machine types A, D, and G (in that order). The intercell material handling cost for part 9 is \$1/unit and the demand for part 9 in the first period is 1,040 units. Given this information, material handling cost for Part X can be computed as \$2,080. Processing of part 9 begins in cell 1 on machine type A. However, the second operation requires machine type D, which is not located in the parent cell. As a result, part 9 must be transferred to another cell for processing on machine type D at a cost of (\$1/unit)(1,040 units) = \$1,040. When the processing on machine type D is complete, part 9 must be transferred back to cell 1 at an additional cost of \$1,040 for the completion of its processing on machine type G. Thus, the total material handling cost for part 9 is \$2,080.

Table 3-4 shows the material handling cost for the candidate machine assignment of Table 3-1 and its corresponding part assignment of Table 3-3. This configuration results in a material handling cost of \$3,120 in period 1 and \$694 in period 2. Thus, the contribution of material handling cost to the total fitness of the solution is \$3,814.

TABLE 3-4
Material Handling Costs for Example Problem

Part No.	Per Unit Material Handling Cost	PERIOD 1 Demand	PERIOD 1 Number of Intercell Transfers	PERIOD 1 Material Handling Cost	PERIOD 2 Demand	PERIOD 2 Number of Intercell Transfers	PERIOD 2 Material Handling Cost
1	$ 1	1,040	0	$ 0	347	1	$ 347
2	1	1,040	0	0	1,388	0	0
3	1	1,040	0	0	1,040	0	0
4	1	1,040	0	0	1,388	0	0
5	1	1,040	0	0	1,010	0	0
6	1	1,040	0	0	347	1	347
7	1	1,040	1	1,040	1,560	0	0
8	1	1,040	0	0	347	0	0
9	1	1,040	2	2,080	1,388	0	0
10	1	1,040	0	0	1,388	0	0
11	1	1,040	0	0	1,560	0	0
12	1	1,040	0	0	347	0	0
13	1	1,040	0	0	624	0	0
14	1	1,040	0	0	347	0	0
			Total	$ 3,120		Total	$ 694

3.2.3.2 Investment in Duplicate Machines

One strategy for minimizing intercell transfers is to duplicate machines in multiple cells. However, duplicating a machine requires a capital investment. Given an overall design objective of minimizing system cost, duplicating a machine is only desirable if the savings in material handling cost is greater than the acquisition cost of the machine. Thus, a component of the fitness evaluation function used by the genetic algorithm is the duplicate machine acquisition cost for each candidate machine assignment solution.

The first step in computing the machine acquisition cost for an individual solution is to determine how many of each machine is required by each cell. Recall that the coded machine assignment solution (chromosome) consists of a string of 1's and 0's. The coded solution, therefore, only indicates the presence of a machine type in a cell, not how many of that machine type are required. The required number of machines is determined by part processing times, part demand, and the machine capacity. At this time, the potential solution is modified to satisfy the constraints on system capacity and within cell capacity.

Given the required number of each machine type, machine acquisition cost is computed by comparing the required number of machines to the number of machines available.

The matrix of processing times for the 14 parts in the example problem is shown in Figure 3-1. These times are used in conjunction with part demand to determine the total workload on each machine. Table 3-5 shows the minimum number of machines required to satisfy the constraint on system capacity.

PARTS

M		1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	A		0.7			2.4			2.4	2.5					2.7
C	B	0.5			0.6	0.9	2.1	1.4							
H	C										2.5	3.0	0.7	1.6	
I	D			3.1				1.4		1.0		0.6	1.0		
N	E		1.2						4.4						3.8
E	F	0.5					4.6		2.3						
S	G	0.6			4.7	3.6	1.5			3.9	4.7				

FIGURE 3-1. Matrix of Processing Times for Example Problem

TABLE 3-5
Machine Data and Total Workload for Example Problem

Machine	Acquisition	Capacity	Total Workload (hrs)		Minimum Number Required	
Type	Cost	(hrs)	Period 1	Period 2	Period 1	Period 2
A	$ 1,200	8,320	11,128	8,635	2	2
B	3,000	8,320	5,720	4,828	1	1
C	2,500	8,320	8,112	9,391	1	2
D	800	8,320	7,384	8,079	1	1
E	3,000	8,320	9,776	4,511	2	1
F	1,000	8,320	7,696	2,568	1	1
G	3,000	8,320	19,760	22,825	3	3

Using the machine cells in Table 3-1 and the part families in Table 3-3, the following table of within cell machine workload. Table 3-6 shows the number of machines required in each cell to satisfy the within cell capacity constraint. The total number of machines required is compared to the number of machines available to determine machine acquisition requirements.

TABLE 3-6
Investment Requirements for Example Problem

PERIOD 1										
Machine Type	Within Cell Workload (hrs)			Number of Machines Required in Cell			Total Number Required	Current Number Available	Number to be Acquired	Total Investment Cost
	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3				
A	10,400	-	-	2	-	-	2	2	0	$ 0
B	4,264	-	-	1	-	-	1	1	0	0
C	-	4,264	3,848	-	1	1	2	1	1	2,500
D	-	-	6,344	-	-	1	1	1	0	0
E	9,776	0	-	2	1	-	3	2	1	3,000
F	7,696	-	-	1	-	-	1	1	0	0
G	14,872	4,888	-	2	1	-	3	3	0	0
									Total	$ 5,500

PERIOD 2										
Machine Type	Within Cell Workload (hrs)			Number of Machines Required in Cell			Total Number Required	Current Number Available	Number to be Acquired	Total Investment Cost
	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3				
A	833	1,909	5,894	1	1	1	3	2	1	$ 1,200
B	-	-	3,926	-	-	1	1	1	0	0
C	3,470	998	4,923	1	1	1	3	2	1	2,500
D	-	-	6,519	-	-	1	1	1	0	0
E	1,527	2,984	-	1	1	-	2	3	0	0
F	2,568	-	-	1	-	-	1	1	0	0
G	7,252	-	15,573	1	-	2	3	3	0	0
									Total	$ 3,700

The current number of machines available in period 1 is an input to the problem. The current number of machines available in subsequent periods is equal to the number available in the first period plus any machines acquired in previous periods. Based on the distribution of the machines to cells and the workloads imposed by the part families, additional units of machine types C and E are required in period 1. Additional units of machine types A and C are required in period 2.

In this example problem, the number of machines assigned in each period is greater than or equal to the minimum number of machines required to satisfy the system capacity constraint. Had this not been the case, additional capacity would have been acquired and assigned to a cell. If additional capacity is needed, it is assigned to cells with the objective of reducing material handling costs.

The total investment cost for this candidate solution is obtained by multiplying the number of machines to be acquired by the corresponding acquisition cost. Capital investments of $5,500 in period 1 and $3,700 in period 2 are required. Thus, the contribution of investment cost to the total fitness of the solution is $9,200.

3.2.3.3 Relocation Cost

In the presence of changing part mix and part demand, the cellular design used in period l may not perform well with respect to the objective of minimizing material handling cost in period $l + 1$. To overcome this possible problem, the multi-period PF/MC formation algorithm investigates the benefits of reassigning machines (and parts) to different cells in subsequent periods. In this way, the efficiencies of the cellular system can be maintained over the entire planning horizon.

Relocation cost, as defined in this dissertation, consists of all costs associated with moving a machine from one cell to another. This cost should include the cost of the physical relocation as well as any costs associated with the disruption of production, if applicable. While this cost may not be easy to estimate, it should reflect the tradeoff between increased material handling and the duplication of machines.

Within the genetic algorithm, the total relocation cost for period l is computed by comparing the number and types of machines assigned to cells in period l to the number and types in period l - 1. The procedure for computing relocation cost will be demonstrated using the example problem. The final composition of the machine cells, adjusted to show the number of machines, is shown in Table 3-7. The number of machines relocated is determined by analyzing the difference in machine assignments between period 1 and period 2.

TABLE 3-7
Adjusted Machine Assignment for Example Problem

Cell No.	Machine Assignments	
	Period 1	Period 2
1	A A B E E F G G	A C E E F G
2	C E G	A C E
3	C D	A B C D G G

In can be seen by looking at Table 3-7 that a number of machines are relocated between the two periods. In the second period, cells 2 and 3 each have a unit of machine type A that was not present during period 1. One unit of machine type A was purchased while the other unit was removed from cell 1 and placed in a new cell. Cell 3 has two units of machine type G during period 2; one unit was removed from cell 1 while the other was removed from cell 2. In addition, machine type B was relocated from cell 1 in period 1 to cell 3 in period 2.

Table 3-8 gives the relocation cost associated with each machine type. Total relocation cost is computed by multiplying the number of units of each machine type relocated by the cost of relocation. There are no relocation costs associated with new machines in the period in which they are acquired. It is assumed that any installation and setup costs are included in the acquisition cost. As a result of the machine relocations described in the previous paragraph, the contribution of relocation cost to total fitness is

$$\text{Total Relocation Cost} = \$600 + \$1{,}500 + 2 \times \$1{,}500 = \$5{,}100.$$

TABLE 3-8
Machine Relocation Costs for Example Problem

Machine Type	A	B	C	D	E	F	G
Relocation Cost	\$600	\$1,500	\$1,250	\$400	\$1,500	\$500	\$1,500

3.2.3.4 *Total Fitness*

The fitness of an individual solution dictates the number of copies of that solution in the mating pool. The more copies an individual receives, the greater the probability that the characteristics of the solution will be repeated in subsequent generations. The total raw fitness of each solution is obtained by adding together the computed material handling cost, machine acquisition cost, and relocation cost. For the example problem,

Raw Fitness = $3,814 + $9,200 + $5,100 = $18,114.

In accordance with the overall objective of minimizing cost, a low raw fitness value means a more fit individual. However, genetic algorithms work with maximization functions. To accomodate this requirement, the raw fitness scores need to be transformed for use by the genetic algorithm.

The transformation is accomplished via a normalization process that reverses the order of the raw fitness scores such that the smallest raw fitness score becomes the largest transformed fitness score. This result is achieved by dividing the minimum raw fitness score of the population by the raw fitness score of the individual solution. Let

$F(t)$ = raw fitness value of population member t
$F'(t)$ = transformed fitness value of population member t
MP = minimum raw fitness value in the current population

Then,

$$F'(t) = MP / F(t)$$

If the minimum raw fitness value equals zero, the raw fitness score is set to 0.1 The transformed fitness values are defined on the interval (0,1]. The transformation procedure is illustrated in the following example.

Table 3-9 shows a population of 30 candidate solutions to the two period, three cell, seven machine, and 14 part example problem. For ease of illustration, the {0, 1} chromosomal representation, which indicates the absence or presence of a machine type, has been modified to show the actual number of each machine type assigned to each cell based on capacity requirements. The corresponding assignment of parts to cells for each solution is given in Table 3-10.

TABLE 3-9
Current Population of Machine Assignment Chromosomes

	Modified Machine Assignment Chromosome								
	Period 1			Period 2					
t	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3	*F(t)*	*F'(t)*	*F''(t)*
1	0100012	2001201	1010100	1010101	1110102	0001011	29,086	0.64928	0.68989
2	1000100	1001001	0110112	1020102	1110010	0001101	25,144	0.75107	0.75331
3	0001001	1000210	1110002	0010001	1101102	1010111	21,821	0.86545	0.82457
4	1000101	1000001	0111113	1020011	0101102	1010100	26,506	0.71248	0.72926
5	2011202	1100102	0100010	1001002	1110000	0010111	29,751	0.63477	0.68084
6	0100001	1000100	2111213	1020103	1001101	0100010	29,906	0.63148	0.67879
7	1010010	0100102	2001202	1010112	1000002	0111100	33,578	0.56242	0.63577
8	2110202	1001001	0000111	1000101	0010002	1111010	26,017	0.72587	0.73761
9	1000011	0001101	2110202	1120103	0001011	1000100	30,626	0.61663	0.66954
10	0011100	2100212	1000001	1010102	0101110	1010001	25,032	0.75443	0.75540
11	1000111	1000101	0111003	1100112	0021002	1000001	23,052	0.81923	0.79578
12	1001000	2111103	0000111	1000101	1011100	0110013	23,465	0.80482	0.78680
13	2110212	0101000	1001001	2020103	0101100	1000011	23,712	0.79643	0.78157
14	1001101	1110002	0000110	0000101	2101100	1020012	26,128	0.72279	0.73569
15	0001001	1111103	1000211	0010011	1110103	1001100	21,176	0.89181	0.84100
16	1011102	0000011	1100110	1000101	1001011	0120103	23,968	0.78793	0.77627
17	2100002	0010111	0001100	1000001	0001101	1120113	30,893	0.61130	0.66622
18	2100203	0011100	1000011	2111102	1000010	1010101	26,165	0.72177	0.73505
19	0010101	1100010	2001202	1010001	1000111	0111102	24,712	0.76420	0.76149
20	1010112	1000101	0101000	0110000	1001111	1021003	29,422	0.64187	0.68527
21	1100012	0001100	1010201	1101102	1010111	0010001	23,117	0.81693	0.79434
22	1000111	1111103	0000101	1021003	1000111	0110100	29,432	0.64165	0.68513
23	1001101	0100112	1010001	0011000	1001101	1110113	20,160	0.93676	0.86900
24	2111212	0001001	1000101	1120003	0000111	1001101	22,488	0.83978	0.80858
25	2100200	0000111	1011002	1010001	1111110	1000102	27,118	0.69640	0.71925
26	1101010	0010002	1000201	1111002	1000111	0010101	19,864	0.95071	0.87770
27	2001201	0110102	0000011	0010010	1000101	1121003	27,865	0.67773	0.70761
28	0001101	2100202	0010011	0020011	1101102	1000100	22,978	0.82187	0.79742
29	1000001	0101112	1010201	1100112	1011001	0010101	21,900	0.86233	0.82263
30	0000111	1110002	1001101	1000100	1100012	0021102	18,885	1.00000	0.90841
						Minimum	18,885	0.56242	0.63577
						Maximum	33,578	1.00000	0.90841
						Average	25,446	0.75701	0.75701

TABLE 3-10
Assignment of Parts to Current Population of Machine Cell Assignments

t	Period 1 Assignment														Period 2 Assignment													
	Part Number														Part Number													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	2	2	1	1	1	1	2	2	1	1	1	3	2	3	1	3	2	2	3	3	1	3	1	2	2	1	1
2	3	1	2	3	3	3	2	3	2	3	3	3	3	1	2	1	3	1	2	2	1	1	3	1	1	1	2	1
3	1	2	1	3	3	2	1	2	1	3	1	1	3	2	3	2	2	2	2	3	2	3	2	1	3	3	1	3
4	3	2	3	3	3	3	3	3	3	3	3	3	3	1	1	3	2	2	2	1	2	3	2	1	1	1	3	3
5	3	1	1	2	2	3	1	1	1	1	1	1	1	1	3	2	1	1	2	3	1	3	1	3	2	2	3	2
6	3	2	1	3	3	3	3	3	3	3	3	3	3	3	3	1	2	1	1	3	2	1	2	1	1	1	1	1
7	1	3	3	2	2	3	3	3	3	3	3	3	1	3	1	1	3	2	1	1	3	1	2	1	3	3	3	1
8	3	1	2	1	1	3	2	1	2	1	1	1	1	1	3	1	3	2	3	3	3	1	3	2	3	3	2	1
9	1	3	2	3	3	1	2	3	2	3	3	3	3	3	2	3	2	1	1	2	1	1	2	1	1	1	1	1
10	2	2	1	2	2	2	1	2	1	1	1	3	1	2	2	1	2	1	1	2	2	2	1	3	1	1	3	1
11	1	2	3	3	3	1	3	1	3	3	3	3	3	2	1	1	2	1	1	1	1	3	2	2	2	2	2	1
12	3	2	2	2	2	3	2	3	2	2	1	2	2	2	3	1	2	3	3	3	2	1	2	3	2	2	2	1
13	1	1	2	1	1	1	2	1	3	1	1	1	1	1	3	1	2	1	1	3	2	1	1	1	1	1	1	1
14	1	1	1	2	2	1	1	3	1	2	1	1	2	1	3	2	2	3	2	3	2	2	2	3	3	3	1	2
15	3	3	1	2	2	3	2	3	2	2	2	2	2	3	1	2	3	2	2	1	1	2	3	2	1	1	2	3
16	2	3	1	1	3	2	1	3	1	1	1	1	1	1	2	1	2	3	3	2	3	1	2	3	3	3	3	1
17	2	1	3	1	1	2	1	2	1	2	1	1	2	1	3	1	2	3	3	3	3	3	2	3	3	3	3	3
18	3	2	2	1	1	3	1	1	1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	3	2	1	3	1
19	2	3	3	3	2	2	3	3	3	1	3	3	1	3	2	2	3	3	3	2	3	2	3	1	3	3	1	2
20	1	2	3	1	1	1	3	1	2	1	1	1	1	2	2	2	3	3	3	2	3	2	2	3	3	3	1	2
21	1	3	2	1	1	1	1	3	2	3	2	1	3	3	2	2	1	1	1	2	1	2	3	2	1	1	2	1
22	1	1	2	2	2	1	2	1	2	2	2	2	2	3	2	2	1	1	1	2	1	2	1	1	1	1	3	2
23	2	1	1	2	2	2	1	2	1	3	1	1	3	1	3	2	2	3	3	3	3	3	2	3	1	1	1	2
24	1	1	1	3	1	1	1	1	2	1	1	1	1	1	2	3	3	1	1	2	1	2	3	1	1	1	1	3
25	2	1	3	1	1	2	1	1	3	3	3	3	3	1	2	3	2	3	2	2	2	2	2	3	1	2	2	3
26	1	3	1	2	1	1	1	3	1	2	1	1	2	3	2	2	1	1	1	2	1	2	1	3	1	1	3	2
27	3	1	1	2	2	3	1	1	1	2	1	1	2	1	1	2	3	3	3	2	3	2	3	3	3	3	1	2
28	3	2	1	2	2	3	3	2	1	3	3	3	3	2	1	3	2	2	2	1	2	3	2	1	1	1	1	3
29	2	3	2	2	2	2	2	3	2	1	3	3	3	3	1	1	2	1	1	1	1	1	2	3	2	2	2	1
30	1	3	3	2	2	1	1	1	3	2	1	1	2	3	2	1	3	2	2	2	1	1	3	3	3	3	3	1

The minimum raw fitness value for this population is \$18,885 (member 30). The transformed fitness values, $F'(t)$, shown in the second to last column of Table 3-9 were obtained by dividing \$18,885 by each individuals raw fitness value, $F(t)$. For example, the transformed fitness of population member 1 is given by

$$F'(1) = \$18{,}885 \,/\, \$29{,}086 = 0.64928$$

As a result of this normalization process, low raw fitness values have a corresponding high transformed fitness value. Notice that population member 30, who had the lowest raw fitness score, has the maximum transformed fitness value.

Once the raw fitness values have been transformed, they undergo a scaling process. The objective of the scaling process in early generations is to help prevent extraordinary individuals from taking over a population too quickly, which can lead to premature convergence on a local optima. In later generations, the scaling process helps differentiate between good solutions and mediocre solutions. It is the scaled fitness values that are used in the mating pool selection process.

A linear scaling procedure is used in the genetic algorithm. Scaled fitness values, $F''(t)$, are calculated from the transformed fitness values using a linear equation.

$$F''(t) = aF'(t) + b$$

In each generation, the coefficients a and b are chosen such that the best individual receives a pre-specified number of copies in the mating pool (usually two), while an average population member receives one copy. The last column of Table 3-9 shows the scaled fitness value for each member of the population.

3.2.4 The Genetic Operators

Once the current population of solutions has been evaluated, the next step of the genetic algorithm is to create new candidate solutions that have the potential of replacing members of the current population in the next generation. This is achieved by selecting good solutions from the current population and combining their information in an attempt to find better solutions. The search for better solutions is accomplished by applying the genetic operators of reproduction, crossover, and mutation.

The reproduction operator is used to select individuals from the current population to become parents of the next generation. Once the parents have been selected, children are created via the crossover and mutation operators. The crossover operator produces children by exchanging information contained in the parent solutions such that the children contain features from each parent. The mutation operator makes small random changes in a parent solution to produce a single child. The application of these operators to the search of good solutions to the multi-period PF/MC formation problem is described in the following sections.

3.2.4.1 The Reproduction Operator

The purpose of the reproduction operator is to select parents for the next generation of solutions. Parents are selected according to their scaled fitness values. A variation of the remainder stochastic sampling without replacement policy is used in the genetic algorithm. Under this policy, the expected number of copies is computed for each member of the population. The expected number of copies, $E(t)$, for population member t is given by

$$E(t) = PS \times F''(t) \,/\, \Sigma_t F''(t)$$

where PS is the size of the population.

Each member of the population is copied into the mating pool according to the integer portion of the expected number of copies. The fractional portions are normalized to sum to one and are treated as probabilities during the selection process to fill out the rest of the mating pool. A uniform number between zero and one is generated and the corresponding solution receives an additional copy in the mating pool. The probability of the solution receiving another copy is set equal to zero. This process continues, with each population member receiving at most one additional copy, until the mating pool is full. In this disseration, the size of the mating pool is equal to the population size. Once the mating pool has been established, the members are randomly mated, and children are created by applying the crossover and mutation operators. The selection and mating procedures are demonstrated in the following example.

The expected number of copies for the members of the population shown in Table 3-9 are given in Table 3-11. Based on the integer portion of the expected number of copies, 13 copies have been identified as members of the mating pool. The remaining 17 members of the mating pool are selected based on the fractional portion of their expected number of copies. The actual number of copies each individual received in this example are also shown in Table 3-11.

TABLE 3-11
Selection of Parents to Form Mating Pool

t	Scaled Fitness $F''(t)$	Expected Number of Copies $E(t)$	Actual Number of Copies
1	0.68989	0.91133	1
2	0.75331	0.99512	0
3	0.82457	1.08926	1
4	0.72926	0.96335	1
5	0.68084	0.89939	1
6	0.67879	0.89668	1
7	0.63577	0.83984	1
8	0.73761	0.97437	1
9	0.66954	0.88446	0
10	0.75540	0.99788	1
11	0.79578	1.05122	1
12	0.78680	1.03935	1
13	0.78157	1.03245	2
14	0.73569	0.97184	1
15	0.84100	1.11095	1
16	0.77627	1.02545	1
17	0.66622	0.88008	1
18	0.73505	0.97099	0
19	0.76149	1.00592	1
20	0.68527	0.90523	1
21	0.79434	1.04932	1
22	0.68513	0.90505	1
23	0.86900	1.14795	2
24	0.80858	1.06813	2
25	0.71925	0.95012	1
26	0.87770	1.15943	1
27	0.70761	0.93475	1
28	0.79742	1.05339	1
29	0.82263	1.08669	1
30	0.90841	1.20000	1

Once the mating pool has been established, its members are randomly mated. Table 3-12 shows the mating pool and the random selection of mates. Column p is the mating pool chromosome number while column t refers to the chromosome number in the current population.

TABLE 3-12
The Mating Pool

p	t	Machine Assignment Chromosome (unmodified)						Mate (p)
		Period 1			Period 2			
		Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3	
1	1	0100011	1001101	1010100	1010101	1110101	0001011	6
2	3	0001001	1000110	1110001	0010001	1101101	1010111	16
3	4	1000101	1000001	0111111	1010011	0101101	1010100	13
4	5	1011101	1100101	0100010	1001001	1110000	0010111	10
5	6	0100001	1000100	1111111	1010101	1001101	0100010	11
6	7	1010010	0100101	1001101	1010111	1000001	0111100	1
7	8	1110101	1001001	0000111	1000101	0010001	1111010	22
8	10	0011100	1100111	1000001	1010101	0101110	1010001	23
9	11	1000111	1000101	0111001	1100111	0011001	1000001	18
10	12	1001000	1111101	0000111	1000101	1011100	0110011	4
11	13	1110111	0101000	1001001	1010101	0101100	1000011	5
12	13	1110111	0101000	1001001	1010101	0101100	1000011	26
13	14	1001101	1110001	0000110	0000101	1101100	1010011	3
14	15	0001001	1111101	1000111	0010011	1110101	1001100	28
15	16	1011101	0000011	1100110	1000101	1001011	0110101	30
16	17	1100001	0010111	0001100	1000001	0001101	1110111	2
17	19	0010101	1100010	1001101	1010001	1000111	0111101	29
18	20	1010111	1000101	0101000	0110000	1001111	1011001	9
19	21	1100011	0001100	1010101	1101101	1010111	0010001	20
20	22	1000111	1111101	0000101	1011001	1000111	0110100	19
21	23	1001101	0100111	1010001	0011000	1001101	1110111	24
22	23	1001101	0100111	1010001	0011000	1001101	1110111	7
23	24	1111111	0001001	1000101	1110001	0000111	1001101	8
24	24	1111111	0001001	1000101	1110001	0000111	1001101	21
25	25	1100100	0000111	1011001	1010001	1111110	1000101	27
26	26	1101010	0010001	1000101	1111001	1000111	0010101	12
27	27	1001101	0110101	0000011	0010010	1000101	1111001	25
28	28	0001101	1100101	0010011	0010011	1101101	1000100	14
29	29	1000001	0101111	1010101	1100111	1011001	0010101	17
30	30	0000111	1110001	1001101	1000100	1100011	0011101	15

3.2.4.2 The Crossover Operator

The crossover operator creates new potential solutions (children) by exchanging portions of the parent solutions. In this way, each child retains some of the features of each parent solution. The genetic algorithm used to solve the multi-period PF/MC formation problem employs the standard single-point and two-point crossover operators as well as two problem specific crossover operators, the cell-swap operator and the period-swap operator.

The single point crossover operator randomly generates a single crossover point along the length of the chromosome. This crossover point divides each of the parent chromosomes into two segments. The children are created by swapping the second segments of the parents. The first child consists of segment 1 of parent 1 and segment 2 of parent 2. The second child consists of the remaining segments: segment 1 of parent 2 and segment 2 of parent 1. The single-point crossover operator is illustrated in Figure 3-2 (a).

The two-point crossover operator randomly generates two crossover points along the length of the chromosome, dividing each parent chromosome into three segments. The two children are created by exchanging segment 2 between the parents. This operator is illustrated in Figure 3-2 (b).

The problem specific cell-swap crossover operator randomly chooses a cell number and a period number. The children are created by exchanging the machine assignments in the selected cell within the selected period between the parents. The period-swap crossover operator randomly selects a period in the planning horizon and creates the children by exchanging the machine assignments for the entire selected period between the parents. These operators are very similar to the two-point crossover operator. The difference is that the crossover points in the cell-swap operator will always correspond to the beginning and the end of a cell subdivision of the chromosome in a particular period and the crossover points in the period-swap operator always correspond to the beginning and the end of a single period subdivision of the chromosome. The cell-swap and period-swap operators are shown in Figure 3-2 (c) and Figure 3-2 (d), respectively.

Parent 1:	1 0 0 0 1 0 1 1 1 0 | 0 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0
Parent 2:	0 1 1 0 1 1 1 1 0 0 | 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 0 0 0 1 0 0 1 0 1 0 1
Child 1:	1 0 0 0 1 0 1 1 1 0 | 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0 1 1 0 0 0 0 1 0 0 1 0 1 0 1
Child 2:	0 1 1 0 1 1 1 1 0 0 | 0 1 1 1 0 0 1 1 0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0

(a) Single-point crossover operator (crossover after position 10).

Parent 1:	0 0 0 1 1 0 0 0 0 1 0 0 0 | 1 1 1 0 0 1 1 1 1 0 0 0 | 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 0 0
Parent 2:	1 0 0 0 1 0 1 1 1 0 0 1 1 | 1 0 0 1 1 0 0 0 1 0 1 0 | 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0
Child 1:	0 0 0 1 1 0 0 0 0 1 0 0 0 | 1 0 0 1 1 0 0 0 1 0 1 0 | 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 0 0
Child 2:	1 0 0 0 1 0 1 1 1 0 0 1 1 | 1 1 1 0 0 1 1 1 1 0 0 0 | 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 1 0

(b) Two-point crossover operator (crossover between positions 13 and 26).

Parent 1:	0 0 1 0 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 | 1 1 1 1 0 1 1 | 0 1 0 0 1 0 0
Parent 2:	0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 1 1 0 1 0 | 1 0 0 0 1 0 1 | 0 1 0 0 1 0 1
Child 1:	0 0 1 0 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 0 1 0 1 | 1 0 0 0 1 0 1 | 0 1 0 0 1 0 0
Child 2:	0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 1 1 0 1 0 | 1 1 1 1 0 1 1 | 0 1 0 0 1 0 1

(c) Cell-swap crossover operator (swap cell 1 of period 2).

Parent 1:	1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 | 1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 0
Parent 2:	0 1 1 0 1 1 1 1 0 0 1 1 0 1 1 0 0 0 0 0 1 | 1 0 1 1 1 1 0 1 1 0 0 0 0 1 0 0 1 0 1 0 1
Child 1:	0 1 1 0 1 1 1 1 0 0 1 1 0 1 1 0 0 0 0 0 1 | 1 0 1 0 1 0 1 0 1 0 1 1 0 1 1 0 0 0 1 1 0
Child 2:	1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 | 1 0 1 1 1 1 0 1 1 0 0 0 0 1 0 0 1 0 1 0 1

(d) Period-swap crossover operator (swap period 1).

FIGURE 3-2. The Crossover Operators.

The crossover operator is not applied to every pair of parent solutions; instead, crossover occurs according to a specified probability known as the crossover rate. The crossover rate being used in the multi-period PF/MC formation genetic algorithm is 0.6, meaning that 60% of all paired parent solutions produce children via crossover. If crossover is to occur, one of the four crossover operators described above is randomly selected. If crossover does not occur, the children are exact duplicates of the parents, subject to the subsequent application of the mutation operator. In the example mating pool, crossover occurred in only six out of the 15 pairs.

3.2.4.3 The Mutation Operator

The purpose of the mutation operator is to rejuvenate the search and extend it into previously unexplored areas of the solution space. Mutation prevents the value of any parameter from remaining unchanged forever. The probability that a parameter will mutate from parent to child is know as the mutation rate. In this dissertation, the mutation rate is 0.01, meaning that only one parameter out of every 100 examined is expected to change its value from parent to child.

Recall that the individual characters (parameters) that make up the machine assignment chromosome are 1's and 0's, indicating the presence or absence of a machine type in a particular cell in a specific period. When mutation occurs, parent values of 1 are changed to 0 in the child chromosome and parent values of 0 are changed to 1. For example, consider the following parent chromosome and resulting child chromosome.

Parent: 1 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 $\underline{1}$ 0 1 0 0 0 0 1 1 1

Child: 1 1 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 $\underline{0}$ 0 1 0 0 0 0 1 1 1

Mutation only occurred at string position 33. As a result, the parent value of 1 was changed to 0 in the child. For the example problem, the result of the mutation is decoded as follows. Machine type 5 is present in cell 2 during period 2 in the parent solution. As a result of the mutation operator, machine type 5 is not present in cell 2 during period 2 in the child solution. All other machine assignments are the same. In the example population, mutation occurred in fifteen string positions, altering the structure of 11 child chromosomes (mutation occurred twice in four of the child chromosomes).

3.2.5 Selecting the New Population of Solutions

After the genetic operators have been applied, the resulting children are evaluated with respect to the evaluation function. Table 3-13 shows the entire child population and its fitness values for the example problem. The next step of the genetic algorithm is to select the next generation of solutions out of the population of child solutions and the population of parent solutions. Children will replace parents solutions in the new population according to a replacement strategy. The replacement strategy is as follows.

TABLE 3-13
The Child Population of Machine Assignment Solutions

c	Modified Machine Assignment Choromosome						F(c)	F'(c)	F''(c)
	Period 1			Period 2					
	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3			
1	0100012	2001201	1010100	1010101	1110002	0001011	29,086	0.53411	0.60488
2	0001001	1000210	1110002	0010001	1101102	1010111	21,821	0.71193	0.66961
3	2001201	1110002	0000110	1020011	0101102	1010100	29,196	0.53209	0.60415
4	2011202	1100102	0100010	1001002	1110000	0010111	29,751	0.52217	0.60054
5	0100001	1000100	2111213	1120103	1001101	0000011	27,998	0.55486	0.61244
6	1110010	0100102	2001201	1021112	1110102	0110100	34,121	0.45529	0.57620
7	2110202	1001001	0000111	1000101	0010002	1111010	26,017	0.59711	0.62782
8	0011100	1100112	1000001	1010102	0101110	1010001	20,832	0.74573	0.68192
9	1000211	1000101	0111003	1100112	0021002	1000001	26,745	0.58086	0.62190
10	1001000	1111103	0000111	1000101	1011100	0110013	22,265	0.69773	0.66444
11	2110212	0101000	1001001	1020102	0101101	1100010	21,314	0.72886	0.67578
12	1101010	0010002	1000201	2020103	0101100	1000011	33,802	0.45959	0.57776
13	1000201	1000001	0111113	0010111	2101102	1010011	27,696	0.56091	0.61464
14	0001001	1111103	1000211	0010011	1110103	1001100	22,216	0.69927	0.66500
15	1011102	0000111	1100110	1000101	1001011	0120103	26,968	0.57605	0.62015
16	2100002	0010111	0001100	1010001	0001101	1110113	29,505	0.52652	0.60212
17	0010101	1100010	2001202	1010001	1000111	1121103	28,161	0.55165	0.61127
18	2010212	1000102	0101000	0110000	1001111	1021003	35,816	0.43374	0.56835
19	1100012	0001110	1010101	1101102	1010110	0010001	15,535	1.00000	0.77447
20	1000111	1111103	0100100	1021003	1000111	0110100	28,279	0.54935	0.61043
21	1001101	0100112	1010001	0011000	1001101	1110113	20,160	0.77059	0.69096
22	1001101	0100112	1010001	0011000	1011101	1110113	22,660	0.68557	0.66002
23	2111213	0001001	1000101	1121003	0000111	1001101	17,981	0.86397	0.72496
24	2111212	0001001	1000101	1110000	0000111	1021103	25,479	0.60972	0.63241
25	2100200	0000111	1011002	0010010	1000101	1121003	22,775	0.68211	0.65876
26	2110212	0101000	1001001	1111002	1000111	0010101	17,774	0.87403	0.72862
27	1001101	1110102	0000011	1010001	1111112	1000101	18,808	0.82598	0.71113
28	0001101	2100202	0010011	0020011	1101102	1000100	22,978	0.67608	0.65656
29	1001000	0101112	1010201	1100112	1021002	0010101	29,128	0.53334	0.60460
30	0000111	1110002	1001101	1000100	1100012	0021102	18,885	0.82261	0.70990
						Minimum	15,535	0.43374	0.56835
						Maximum	35,816	1.00000	0.77447
						Average	25,125	0.64539	0.64539

1. Order the adult (current) population, $A(PS)$, from least fit to most fit and the child population, $C(PS)$, from most fit to least fit.

2. Set child pointer, c, to $C(1)$ and adult pointer, t, to $PP(1)$. Initialize the new population counter, $NPC = 0$.

3. Compare the most fit child to the least fit adult.

   a. If the child is more fit than the adult, admit the child to the new population and discard the adult from further comparisons. Set $c = c + 1$, $t = t + 1$, and $NPC = NPC + 1$.

   b. If the child is less fit than the adult, admit the child to the new population with probability X (currently, X = 1 / PS). If the child is admitted, discard the adult and set $c = c + 1$, $t = t + 1$, and $NPC = NPC + 1$; else, discard the child and set $c = c + 1$.

4. Repeat step 3 until all children have been considered for admittance to the new population.

5. Fill the remaining slots in the new population with individuals from the adult population, beginning with the most fit member, $A(PS)$ and continuing until the new population is full.

The goal of the replacement strategy is to create generations of solutions that, on average, outperform the previous generation. This is accomplished by restricting admittance to the new population to only those children that are better than members of the current population. The exception to this rule is step 2(b) in which some less fit children are allowed to survive and become part of the next generation. The purpose of occasionally admitting less fit individuals is to help the algorithm come out of local optima. Applying this replacement strategy to the example problem yields the new generation of solutions shown in Table 3-14. Notice that the minimum raw fitness score had decreased from 18,885 to 15,535. In addition, the average raw fitness score in the old population was 25,466 while the average raw fitness score in the new population is 21,583.

TABLE 3-14

The New Population of Machine Assignment Solutioms

t	Modified Machine Assignment Choromosome						F(t)	F'(t)	F"(t)
	Period 1			Period 2					
	Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3			
1	1100012	0001110	1010101	1101102	1010110	0010001	15,535	1.00000	0.87440
2	2110212	0101000	1001001	1111002	1000111	0010101	17,774	0.87403	0.80674
3	2111213	0001001	1000101	1121003	0000111	1001101	17,981	0.86397	0.80133
4	1001101	1110102	0000011	1010001	1111112	1000101	18,808	0.82598	0.78093
5	0000111	1110002	1001101	1000100	1100012	0021102	18,885	0.82261	0.77912
6	1001101	0100112	1010001	0011000	1001101	1110113	20,160	0.77059	0.75118
7	0011100	1100112	1000001	1010102	0101110	1010001	20,832	0.74573	0.73783
8	2110212	0101000	1001001	1020102	0101101	1100010	21,314	0.72886	0.72877
9	0001001	1000210	1110002	0010001	1101102	1010111	21,821	0.71193	0.71968
10	0001001	1111103	1000211	0010011	1110103	1001100	22,216	0.69927	0.71288
11	1001000	1111103	0000111	1000101	1011100	0110013	22,265	0.69773	0.71205
12	1001101	0100112	1010001	0011000	1011101	1110113	22,660	0.68557	0.70552
13	2100200	0000111	1011002	0010010	1000101	1121003	22,775	0.68211	0.70366
14	0001101	2100202	0010011	0020011	1101102	1000100	22,978	0.67608	0.70042
15	0000111	1110002	1001101	1000100	1100012	0021102	18,885	0.82261	0.77912
16	1101010	0010002	1000201	1111002	1000111	0010101	19,864	0.78207	0.75735
17	1001101	0100112	1010001	0011000	1001101	1110113	20,160	0.77059	0.75118
18	0001001	1111103	1000211	0010011	1110103	1001100	21,176	0.73361	0.73132
19	0001001	1000210	1110002	0010001	1101102	1010111	21,821	0.71193	0.71968
20	1000001	0101112	1010201	1100112	1011001	0010101	21,900	0.70936	0.71830
21	2111212	0001001	1000101	1120003	0000111	1001101	22,488	0.69081	0.70833
22	0001101	2100202	0010011	0020011	1101102	1000100	22,978	0.67608	0.70042
23	1000111	1000101	0111003	1100112	0021002	1000001	23,052	0.67391	0.69926
24	1100012	0001100	1010201	1101102	1010111	0010001	23,117	0.67202	0.69824
25	1001000	2111103	0000111	1000101	1011100	0110013	23,465	0.66205	0.69289
26	2110212	0101000	1001001	2020103	0101100	1000011	23,712	0.65515	0.68918
27	1011102	0000011	1100110	1000101	1001011	0120103	23,968	0.64816	0.68542
28	0010101	1100010	2001202	1010001	1000111	0111102	24,712	0.62864	0.67494
29	0011100	2100212	1000001	1010102	0101110	1010001	25,032	0.62061	0.67063
30	1000100	1001001	0110112	1020102	1110010	0001101	25,144	0.61784	0.66914
						Minimum	15,535	0.61784	0.66914
						Maximum	25,144	1.00000	0.87440
						Average	21,583	0.72866	0.72866

3.2.6 Terminating the Genetic Algorithm

The genetic algorithm continues to create new generations until a criterion for termination is met. A single criterion or set of criteria can be used to halt the genetic algorithm. In this dissertation, three termination criteria are used in conjunction.

The first termination criterion is a test of population convergence. If the fitness of the best solution equals the fitness of the worst solution, the algorithm will terminate. Once the population has converged, the search stagnates since the crossover operator ceases to create new solutions. The second termination criterion monitors improvement from generation to generation. If there is no improvement in the best solution found for a pre-specified number of generations, the algorithm will terminate.

The last termination criterion is the maximum number of generations rule. Under this rule, the algorithm stops when a specified number of generations have been created and evaluated. The best solution is then taken to be the most fit solution found in the last generation.

3.3 Implementation and Uses of the Multi-Period Model

The genetic algorithm developed to solve the multi-period PF/MC formation problem was programmed in BASIC. The set of system parameters described in Section 3.1 are the required inputs to the program. Due to the probabilistic nature of genetic algorithms, it is standard practice to execute the algorithm several times and select the best solution (or set of solutions) for further analysis and potential implementation.

The CPU time required to solve the multi-period PF/MC formation problem will vary with the size of the problem and the processor used. Table 3-15 summarizes observed execution times using a 486, 66 MHz PC with 8 MB of RAM.

TABLE 3-15
Genetic Algorithm Execution Times

Design Problem	Number of Machines	Number of Parts	Number of Periods	Average Execution Time (1 Run)
1	19	30	2	20 min.
2	11	25	3	39 min.
3	7	22	5	21 min.

The multi-period PF/MC formation methodology provides the system designer with a variety of useful information. The primary output of the model is the period by period assignment of parts to families and machines to cells over the planning horizon. By identifying how and when the system is going to change and what resources will need to be acquired, plans for these changes and acquisitions can be made in advance.

The multi-period PF/MC formation methodology can be used to either create a new system or to investigate improvements to an existing system. When unanticipated changes occur in the product mix and/or resource population, the system may begin to perform less efficiently. By including the current system configuration in the set of system parameters, potential re-configuration of the system to regain operating efficiencies can be investigated.

Finally, the use of a genetic algorithm to solve the PF/MC formation problem facilitates the generation of alternative system designs. These alternative designs can be further evaluated in terms of secondary design objectives to determine the best partition of parts into families and machines into cells.

3.4 Validation of the Genetic Algorithm Solution Approach

The multi-period PF/MC formation problem formulation is unique in its treatment of the dynamic production environment. Although the performance of the multi-period algorithm cannot be evaluated with respect to other multi-period approaches (no comparable techniques have been found), the multi-period algorithm reduces to the common single period formulation (constant product mix, demand, and resource availability) when the number of periods in the planning horizon is set equal to one. Thus, it is possible to evaluate the performance of the genetic algorithm approach with respect to existing PF/MC formation strategies for the single period case.

The validation of the genetic algorithm approach was accomplished by comparing the cellular designs obtained by the genetic algorithm to those obtained by published PF/MC formation techniques. Several test problems of various sizes and design objectives were selected from the literature. When necessary, the evaluation function of the genetic algorithm was modified to accommodate the objective of the specific test problem. For example, not all of the selected problems considered operation sequence in the computation of intercell transfers. In other examples, machine duplication as a means of reducing intercell transfers was not considered.

The ability of the genetic algorithm to solve a multi-period problem was tested by repeating the data for the single period problem in subsequent periods. The results of the validation process are presented in detail for three of the test problems.

3.4.1 Validation Problem 1

The first validation problem is the well known Burbridge problem. This 43 part, 16 machine problem originated with Burbridge (1973) and has been used by several researchers to demonstrate methods of PF/MC formation. Figure 3-3 shows the original binary machine-part incidence matrix.

Boctor (1991) developed a linear formulation of the PF/MC formation problem. The objective of the formulation is to minimize the number of operations that take place outside of a part's primary cell (minimize intercell transfers). Boctor's formulation included a constraint on the maximum number of machines allowed in each cell. Figure 3-4 shows Boctor's four cell solution with a maximum number of five machines per cell. There are 26 intercell transfers associated with this solution.

Before using the genetic algorithm to solve this single period problem, two modifications were necessary. First, Boctor's formulation does not take into account each part's operation sequence when forming the cells. As a result, the intercell transfer counter of the genetic algorithm was modified to ignore operation sequence. Secondly, Boctor's formulation does not consider machine duplication in an effort to reduce intercell transfers. To model this constraint, the machine acquisition costs used in the genetic algorithm were set sufficiently high with respect to material handling cost. This had the effect of discouraging machine duplication as a means of reducing transfers.

The genetic algorithm found two solutions to the Burbridge problem. Both designs result in 26 intercell transfers. The first design is identical to Boctor's solution. The second design, shown in Figure 3-5, has the same machine assignment with a slightly different part assignment. In the second solution, parts 11 and 20 are assigned to cell 3, instead of cell 4 as in the first solution. This assignment of parts has no effect on the number of intercell transfers. However, it does illustrate the ability of the genetic algorithm to find alternative solutions to a problem.

MACHINES

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1						1	1	1		1						
	2		1				1		1	1					1		1
	3								1			1		1			
	4									1							
	5				1	1										1	
	6						1								1		
	7			1			1										1
	8					1	1		1								
	9				1	1			1			1					
	10		1							1							1
	11								1				1				
	12						1		1		1						
	13						1	1			1						
	14				1	1	1									1	
	15					1			1								
	16					1											
	17			1			1								1		
P	18									1							1
A	19				1	1	1		1							1	
R	20								1			1					
T	21				1	1			1							1	
S	22												1				
	23				1	1	1		1								
	24								1			1	1	1			
	25							1			1						
	26										1						
	27								1			1	1				
	28		1						1	1							
	29				1	1											
	30											1	1				
	31								1		1						
	32		1				1			1							1
	33					1	1									1	
	34			1			1										
	35			1											1		
	36			1													
	37	1	1				1		1	1							1
	38		1						1	1							1
	39						1				1						
	40		1				1			1							
	41					1			1							1	
	42	1	1				1			1							1
	43					1	1		1							1	

FIGURE 3-3. Binary Machine-Part Incidence Matrix for Validation Problem 1

MACHINE CELLS

PART FAMILIES	1	2	9	16	3	6	7	10	14	4	5	8	15	11	12	13
2		1	1	1		1			1			1				
4			1													
10		1	1	1												
18			1	1												
28		1	1									1				
32		1	1	1		1										
37	1	1	1	1		1						1				
38		1	1	1								1				
40		1	1			1										
42	1	1	1	1		1										
1						1	1	1				1				
6						1			1							
7				1	1	1										
12						1		1				1				
13						1	1	1								
17					1	1			1							
25							1	1								
26								1								
34					1	1										
35					1				1							
36					1											
39						1		1								
5										1	1		1			
8						1					1	1				
9										1	1	1		1		
14						1				1	1		1			
15											1	1				
16											1					
19						1				1	1	1	1			
21										1	1	1	1			
23						1				1	1	1				
29										1	1					
31								1				1				
33						1					1		1			
41											1	1	1			
43						1					1	1	1			
3												1		1		1
11												1			1	
20												1		1		
22															1	
24												1		1	1	1
27												1		1	1	
30														1	1	

FIGURE 3-4. Published Solution to Validation Problem 1

MACHINE CELLS

PART FAMILIES	1	2	9	16	3	6	7	10	14	4	5	8	15	11	12	13
2		1	1	1		1			1			1				
4			1													
10		1	1	1												
18			1	1												
28		1	1									1				
32		1	1	1		1										
37	1	1	1	1		1						1				
38		1	1	1								1				
40		1	1			1										
42	1	1	1	1		1										
1						1	1	1				1				
6						1			1							
7				1	1	1										
12						1		1				1				
13						1	1	1								
17					1	1			1							
25							1	1								
26								1								
34					1	1										
35					1				1							
36					1											
39						1		1								
5										1	1		1			
8						1					1	1				
9										1	1	1		1		
11												1			1	
14						1				1	1		1			
15											1	1				
16											1					
19						1				1	1	1	1			
20												1		1		
21										1	1	1	1			
23						1				1	1	1				
29										1	1					
31								1				1				
33						1					1		1			
41											1	1	1			
43						1					1	1	1			
3												1		1		1
22															1	
24												1		1	1	1
27												1		1	1	
30														1	1	

FIGURE 3-5. Genetic Algorithm Solution to Validation Problem 1

Boctor (1991) also presented a two cell solution to the Burbridge problem. For this solution, the maximum number of machines per cell was set to nine. The reported solution required 13 intercell transfers. The genetic algorithm also found a two cell solution requiring only 13 intercell transfers. The genetic algorithm solution differed from Boctor's by one part assignment.

3.4.2 Validation Problem 2

The second validation problem concerns a production system consisting of 24 parts and 14 machine types. The machine-part incidence matrix with workloads imposed on machines by each part is given in Figure 3-6. This problem was originally an example used by King (1980) to demonstrate the ROC algorithm. The machine workloads were added by Dahel and Smith (1993) to demonstrate their 0-1 integer programming formulation of the PF/MC formation problem. Based on their values of part demand and processing time (not given in the article), the required number of each machine type (shown beneath the machine type number in Figure 3-6), was computed.

The objective of Dahel and Smith's PF/MC formulation is the minimization of intercell moves without duplicating machines. The integer programming model includes parameters such as part demand, processing times, machine capacities, and the number of machines available. Constraints are placed on the number of machines per cell (minimum and maximum). Dahel and Smith's four cell solution to this problem is shown in Figure 3-7. The number of machines in each cell was restricted in the range of four to fourteen. This solution requires no intercell transfers.

To discourage the duplication of machines in the genetic algorithm's search for a solution to this problem, the value of the machine acquisition cost parameter was set high relative to material handling cost. The solution obtained by the genetic algorithm is displayed in Figure 3-8. This solution also consists of independents cells requiring no intercell transfers; however, the composition of part families and machine cells is quite different than those obtained by Dahel and Smith.

MACHINES
(Number Available)

PARTS	1 (1)	2 (2)	3 (2)	4 (3)	5 (3)	6 (4)	7 (2)	8 (4)	9 (2)	10 (1)	11 (2)	12 (1)	13 (2)	14 (2)
1				.38	.52		.54							
2				.21	.45		.15							
3		.58	.44							.28	.32			
4		.77	.36								.40			
5								.36	.26					
6	.37												.38	
7	.28						.48					.26	.40	
8												.44	.36	
9						.60		.42	.33					.32
10						.28		.38						
11						.34								.36
12						.48		.36	.37					
13									.42					.27
14						.30		.40						
15						.45		.42	.25					.44
16						.55		.54						
17				.40	.56									
18													.25	
19				.28										
20				.60	.38									
21			.37								.29			
22						.52		.35						
23				.35	.40								.42	
24										.45	.38			

FIGURE 3-6. Machine-Part Incidence Matrix with Processing Times for Validation Problem 2

MACHINE CELLS
(Number Assigned)

PART FAMILIES

	2 (1)	3 (1)	4 (1)	5 (1)	7 (1)	11 (1)	6 (2)	8 (2)	9 (1)	14 (2)	1 (1)	2 (1)	3 (1)	4 (1)	5 (1)	7 (1)	10 (1)	11 (1)	12 (1)	13 (2)	4 (1)	5 (1)	6 (2)	8 (2)	9 (1)
1			1	1	1																				
2			1	1	1																				
4	1	1				1																			
19			1																						
21		1				1																			
9							1	1	1	1															
11							1			1															
13									1	1															
15							1	1	1	1															
16							1	1																	
3												1	1				1	1							
6											1									1					
7											1					1			1	1					
8																			1	1					
18																				1					
20														1	1										
23														1	1					1					
24																	1	1							
5																								1	1
10																							1	1	
12																							1	1	1
14																							1	1	
17																					1	1			
22																							1	1	

FIGURE 3-7. Published Solution to Validation Problem 2

MACHINES
(Number Assigned)

PART FAMILIES	4	5	6	8	9	14	2	3	10	11	2	3	6	8	11	1	4	5	7	12	13
	(2)	(1)	(2)	(2)	(2)	(2)	(1)	(1)	(1)	(1)	(1)	(1)	(2)	(2)	(1)	(1)	(1)	(2)	(2)	(1)	(2)
5				1	1																
9			1	1	1	1															
11			1			1															
12			1	1	1																
13					1	1															
15			1	1	1	1															
17	1	1																			
19	1																				
20	1	1																			
3							1	1	1	1											
21								1		1											
24									1	1											
4											1	1			1						
10													1	1							
14													1	1							
16													1	1							
22													1	1							
1																	1	1	1		
2																	1	1	1		
6																1					1
7																1			1	1	1
8																				1	1
18																					1
23																	1	1			1

FIGURE 3-8. Genetic Algorithm Solution to Validation Problem 2

3.4.3 Validation Problem 3

The third validation problem tests the ability of the genetic algorithm to solve the multi-period PF/MC formation problem. A multi-period data set was created by duplicating the single period data set for each period in the planning horizon. The expected result for this test is the exact same cellular design in each period (no relocation of machines or reassignment of parts should occur).

The single period data set shown in Figure 3-9 was duplicated for a five period planning horizon. The genetic algorithm was able to find the optimal solution for this problem (shown in Figure 3-10). The recommended composition of machine cells and part families remains the same across the planning horizon. The number in parentheses next to the machine type indicates how many units of the machine type have been assigned to the cell to fulfill capacity constraints.

PARTS

M			2	3	4	5	6	7	8	9	10	11	12	13	14	Capacity
A	A		0.7			2.4			2.4	2.5					2.7	832
C	B	0.5			0.6	0.9	2.1	1.4								832
H	C										2.5	3.0	0.7	1.6		832
I	D			3.1				1.4		1.0		0.6	1.0			832
N	E		1.2						4.4						3.8	832
E	F	0.5					4.6		2.3							832
S	G	0.6			4.7	3.6	1.5			3.9	4.7					832
Demand		104	104	104	104	104	104	104	104	104	104	104	104	104	104	

FIGURE 3-9. Processing Times, Capacities, and Demands for Validation Problem 3

FAMILIES

			10	11	12	13	1	3	4	5	6	7	8	9	2	14
	3	(1)	1	1	1	1										
	4	(1)		1	1											
	7	(1)	1													
C	1	(1)								1			1	1		
E	2	(1)					1		1	1	1	1				
L	4	(1)						1				1		1		
L	5	(1)											1			
S	6	(1)					1				1		1			
	7	(2)					1		1	1	1			1		
	1	(1)													1	1
	5	(1)													1	1

FIGURE 3-10. Genetic Algorithm Soltuion to Validation Problem 3

3.4.4 Additional Validation Problems

The genetic algorithm was tested on quite a few data sets and solutions found in the literature. Not all problems could be used, however, due to a lack of data or due to the nature of the reported solution technique. For example, it is difficult to compare the genetic algorithm results to those obtained using the ROC algorithm, in that the ROC algorithm requires visual identification of bottleneck machines and/or exceptional parts. However, the genetic algorithm was executed using the duplication of machines recommended by King (1980) for the Burbridge problem and a comparable solution was found.

The genetic algorithm was also tested using the following data sets:

1. Shafer and Rogers (1991) - 12 parts and 6 machines;
2. Venugopal and Narendran (1992) - 30 parts and 15 machines;
3. Harhalakis *et al.* (1990) - 20 parts and 20 machines;
4. Nagi, *et al.* (1990) - 20 parts and 20 machines; and
5. Boctor (1991) - 30 parts and 16 machines.

In each of these tests, an identical solution to the one reported was found. Based on these results, it seems reasonable to conclude that the genetic algorithm presented in this dissertation is capable of providing good solutions to the PF/MC formation problem. In addition, the ability of the genetic algorithm approach to provide alternative solutions has been demonstrated.

3.5 Summary

In this chapter, a model for identifying part families and machine cells in the presence of changes in the product mix and the availability of resources was developed. The objectives of the model were defined and the system parameters and constraints identified. A solution procedure based on a genetic algorithm was illustrated. Finally, the ability of the genetic algorithm to find good solutions to the standard, constant demand, constant resource PF/MC formation problem was demonstrated. Chapter 4 investigates the advantages of using the multi-period PF/MC formation methodology over a single period design approach.

4.0 Multi-Period v. Single Period PF/MC Formation

The multi-period approach to obtaining a preliminary design of a cellular manufacturing system is unique in its treatment of the changing production environment. The need for a multi-period approach to PF/MC formation arises from a changing part population (in terms of product mix and production volumes) and a changing machine population. A cellular design obtained under the assumptions of constant demand and resource availability would required periodic adjustments as the part population and machine population change. The objective of this chapter is to investigate the possible advantages of the multi-period approach over a single period approach in a dynamic production environment.

The investigation of the advantages of using a multi-period approach is made by comparing the cellular design obtained using the multi-period approach to the cellular designs obtained by two single period approaches. In this comparison, the benefit is defined as the difference in system costs (material handling, investment in machines, and machine relocation) over the planning horizon. The two single period approaches are described next.

1. **Fixed Family / Fixed Cell Approach** - This approach begins by solving the single period PF/MC formation problem using the product mix and available resources for the first period in the planning horizon. The resulting part families and machine cells remain fixed through the remainder of the planning horizon. The re-allocation of parts to families and the relocation of machines between cells are not considered. When new parts are introduced to the system, a fixed cell assignment will be made to minimize intercell transfers. Additional machines will be acquired as necessary to meet capacity requirements and to replace retired machines.

2. **Optimal By Period Approach** - This approach begins in the same manner as the fixed machine cells and fixed part families approach. The initial cellular design is obtained by solving the single period PF/MC formation problem using the first period's product mix and available resources. For subsequent periods in the

planning horizon, the single period model is solved considering only material handling cost and investment cost. The cost of relocating the machines between periods is then computed and included in the total system cost of this approach.

Solutions to the single period approaches are obtained using a modified version of the multi-period genetic algorithm. In the Fixed Cell / Fixed Family approach, the genetic algorithm is executed with the planning horizon set equal to one. For subsequent periods, the part assignment heuristic is used to assign new parts to cells. The workloads of machines are computed and the required number of each type are assigned to the cells such that the system capacity and within cell capacity constraints are met. The relocation of underutilized machines is not considered.

For the Optimal By Period approach, the multi-period genetic algorithm is executed separately for each period with the planning horizon set equal to one. The acquisition of machines in period l are added to the number of available machines in period $l + 1$. A special purpose sub-routine is used to calculate the relocation costs associated with re-configuring the system at the beginning of each period.

In each of the following three sections in this chapter, a hypothetical cellular manufacturing system design problem is presented. For each design problem, the solutions obtained by the multi-period approach and the two single period approaches are compared.

4.1 Design Problem 1

The first design problem consists of 30 parts to be produced on a set of 19 unique machine types. The machine-part incidence matrix (with processing times) for this problem is given in Figure 4-1. The planning horizon is two periods. The operation sequence and period demands for the parts are listed in Table 4-1. Notice that only 24 parts are to be produced in the first period. The remaining six parts are introduced to the production system during the second period. Relevant resource related data is listed in Table 4-2.

MACHINES

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

FIGURE 4-1. Machine-Part Incidence Matrix with Processing Times for Design Problem 1

TABLE 4-1
Part Data for Design Problem 1

Part No.	M/H Cost per Unit	Operation Sequence	Demand	
			Period 1	Period 2
1	$ 1	8-12-17	100	200
2	1	12-11-17- 9	400	800
3	1	15-2-12	0	500
4	1	16-8-9	600	400
5	1	18-16-19-4	0	900
6	1	17-5-1-3-2	0	700
7	1	19-5-12-8	0	400
8	1	10-7-14-17-11-18-13	800	300
9	1	7-3-2-15-14-17	800	500
10	1	5-10-2	900	200
11	1	18-16-6	300	300
12	1	3-10-9-19-6-15	900	800
13	1	10-15-8-2	1,000	200
14	1	10-3-15-12-6-11-1	0	700
15	1	5-8-16-14-18-2-6	600	400
16	1	6-12-10-5-3	500	300
17	1	4-9-6-17	400	800
18	1	6-9-15	0	200
19	1	19-4-9-7-14-11-12	400	800
20	1	2-12-8	800	300
21	1	9-12-4-5-8-3	400	700
22	1	9-7-4-8-5-12-19	1,000	600
23	1	6-3-1-14	900	900
24	1	6-16-5-2-7-8-4	800	100
25	1	18-14-13-6	700	800
26	1	16-7-9-13-17-12	600	800
27	1	4-14-2	800	500
28	1	10-17-18-2	400	1,000
29	1	7-1-15	200	500
30	1	8-12-11-1-3	100	100

TABLE 4-2
Resource Data for Design Problem 1

Machine Type	Acquisition Cost	Relocation Cost	Capacity	Number Available Period 1	Planned Acquisitions Period 2
1	$ 6,000	$ 3,000	10,000	1	1
2	8,000	4,000	12,000	3	
3	3,000	1,500	10,000	3	
4	4,000	2,000	14,000	2	
5	9,000	4,500	12,000	2	
6	9,000	4,500	14,000	2	
7	7,000	3,500	12,000	3	
8	6,000	3,000	14,000	2	
9	4,000	2,000	10,000	3	1
10	3,000	1,500	12,000	2	
11	9,000	4,500	12,000	2	
12	6,000	3,000	10,000	3	
13	3,000	1,500	13,000	2	
14	7,000	3,500	12,000	3	
15	8,000	4,000	14,000	1	1
16	8,000	4,000	12,000	2	
17	7,000	3,500	10,000	3	
18	3,000	1,500	12,000	1	
19	9,000	4,500	13,000	1	

The following constraints (in addition to system and within cell capacity constraints) were placed on the design of the cellular manufacturing system.

1. Three machine cells and part families are to be formed.
2. Each machine cell must contain a minimum of four machine types.
3. Each part family must have at least three members.

It is assumed that the values of the system parameters listed in Tables 4-1 and 4-2 remain constant over the two period planning horizon, with the exception of part demand. It is also assumed that the per unit material handling cost is $1 / unit for all parts.

The cellular system designs obtained by the Multi-Period approach and the single period approaches can be found in Appendix A.1. Table 4-3 summarizes the performance results for the three design strategies. The Optimum By Period approach performed the best with respect to total material handling cost and machine acquisition cost. However, this approach performed the worst with respect to machine relocation cost. This is to be expected since the design for each period was optimized in isolation, without regard to system re-configuration requirements between periods.

The Fixed Cell / Fixed Family approach performed the worst with respect to machine acquisition costs. This result is attributed to the within cell capacity constraint. Without the ability to relocate machines or reassign parts, this approach is unable to make use of under-utilized machines in other cells and must instead purchase extra capacity when needed.

With respect to total system cost, the Multi-Period approach performed the best. The total system cost for the Fixed Cell / Fixed Family approach and the optimum by period approach are 29.8% and 26.7% higher than the multi-period approach, respectively. This result becomes even more dramatic if the $18,000 of planned investment is subtracted from the total system cost (46.8% and 42.0%).

TABLE 4-3
Comparison of Results for Design Problem 1

		Multi-Period Approach	Fixed Cell / Fixed Family Approach	Optimum by Period Approach
Material Handling Cost	Period 1	$ 7,300	$ 7,700	$ 7,700
	Period 2	9,400	8,500	5,500
	Sub-Total	$ 16,700	$ 16,200	$ 13,200
Machine Acquisition Cost	Period 1	$ 15,000	$ 0	$ 0
	Period 2	12,000	48,000	18,000
	Sub-Total	$ 27,000	$ 48,000	$ 18,000
Machine Relocation Cost	Period 1	N/A	N/A	N/A
	Period 2	$ 6,000	N/A	$ 31,500
	Sub-Total	$ 6,000	N/A	$ 31,500
Total System Cost	Period 1	$ 22,300	$ 7,700	$ 7,700
	Period 2	27,400	56,500	55,000
	TOTAL	$ 49,470	$ 64,200	$ 62,700

4.2 Design Problem 2

The second problem used to evaluate the benefits of using a multi-period approach versus a single period approach consisted of 11 machine types used to produce 25 parts. Figure 4-2 is the machine part incidence matrix, with processing times, for this problem. Part population data for a three period planning horizon are given in Table 4-4. Data for the machine population (including a schedule for planned investments) are given in Table 4-5.

MACHINES

		1	2	3	4	5	6	7	8	9	10	11
	1	5								2	1	
	2					6			4			
	3	1	3									4
	4			1			6				1	
	5		3			1				4		
	6					4			6		5	
	7					6	3				2	
	8				4					6		1
	9						2				6	3
	10			2								4
P	11	3		6	4							
A	12							3		1		
R	13	4		6		2						
T	14							1	3		3	
S	15			3	1					2		
	16				3						6	
	17					3	6					
	18	2					3				3	
	19			4		3	1					
	20				6					4		3
	21							2	1			
	22		3								5	2
	23						6			1	3	
	24		4					2				
	25		5				2	6				

FIGURE 4-2. Machine-Part Incidence Matrix and Processing Times for Design Problem 2

TABLE 4-4
Part Data for Design Problem 2

Part No.	M/H Cost per Unit	Operation Sequence	Demand		
			Period 1	Period 2	Period 3
1	$ 5	10-1-9	300	200	500
2	5	5-8	700	600	500
3	5	1-2-11	0	600	400
4	5	3-10-6	0	700	800
5	5	2-5-9	800	600	1000
6	5	5-10-8	600	300	0
7	5	6-5-10	0	900	800
8	5	4-9-11	400	800	200
9	5	6-10-11	300	200	600
10	5	3-11	400	1000	500
11	5	3-1-4	0	0	200
12	5	7-9	700	700	1000
13	5	3-1-5	100	600	800
14	5	7-8-10	100	200	0
15	5	3-9-4	0	0	300
16	5	4-10	500	800	500
17	5	6-5	100	900	400
18	5	1-6-10	1000	1000	400
19	5	3-6-5	0	700	1000
20	5	11-9-4	800	300	500
21	5	8-7	500	400	1000
22	5	10-2-11	0	100	100
23	5	9-6-10	400	500	800
24	5	7-2	0	0	500
25	5	2-7-6	0	0	400

In addition to the system capacity and within cell capacity constraints, the following constraints were placed on the design of the cellular manufacturing system.

1. Three machine cells and part families are to be formed.
2. Each machine cell must contain a minimum of three machine types.
3. Each part family must have at least three members.

No changes in operation sequence, processing times, machine capacities, or cost data are anticipated over the three period planning horizon. The performance results of the system designs obtained by the three approaches are presented next.

TABLE 4-5
Resource Data for Design Problem 2

Machine Type	Acquisition Cost	Relocation Cost	Capacity	Number Available Period 1	Planned Acquisitions Period 2	Planned Acquisitions Period 3
1	$ 4,000	$ 2,000	15,000	1		
2	7,000	3,500	18,000	1		
3	5,000	2,500	18,000	1		
4	9,000	4,500	19,000	1		
5	5,000	2,500	15,000	1	1	
6	3,000	1,500	17,000	1	1	
7	9,000	4,500	17,000	1		
8	7,000	3,500	19,000	1		
9	5,000	2,500	18,000	1		
10	8,000	4,000	15,000	1	1	
11	3,000	1,500	19,000	1		

The system designs obtained using the multi-period and the single period approaches can be found in Appendix A.2. Table 4-6 summarizes the performance of these approaches with respect to the design objectives. The Optimal By Period approach performed the best with respect to the sum of total material handling cost and total machine acquisition cost. It is interesting to note, however, that this approach did not perform the best with respect to either individual objective. As was seen in Design Problem 1, the Optimal By Period approach performed the worst with respect to machine relocation cost.

The Fixed Cell / Fixed Family approach resulted in excessive material handling costs over the planning horizon. This is a good illustration of how inefficient a cellular manufacturing system can become when the conditions for which is was designed (period 1 data only) change over time. In Design Problem 1, the Fixed Cell / Fixed Family approach suffered due to the under-utilization of machines. In this example, the problem seems to lie with the assignment of new parts. Because the fixed cell design was obtained without regard to the production requirements of new products, the new products did not fit well into the cellular system (only one out of nine new products can be produced cell complete).

TABLE 4-6
Comparison of Results for Design Problem 2

		Multi-Period Approach	Fixed Cell / Fixed Family Approach	Optimum by Period Approach
Material Handling Cost	Period 1	$ 4,500	$ 5,000	$ 5,000
	Period 2	2,500	28,000	4,500
	Period 3	500	30,000	7,500
	Sub-Total	$ 7,500	$ 63,000	$ 17,000
Machine Acquisition Cost	Period 1	$ 29,000	$ 11,000	$ 11,000
	Period 2	18,000	13,000	13,000
	Period 3	0	0	3,000
	Sub-Total	$ 47,000	$ 24,000	$ 27,000
Machine Relocation Cost	Period 1	$ 0	$ 0	$ 0
	Period 2	0	0	14,000
	Period 3	0	0	4,500
	Sub-Total	0	0	$ 18,500
Total System Cost	Period 1	$ 33,500	$ 16,000	$ 16,000
	Period 2	20,000	41,000	31,500
	Period 3	500	30,000	15,000
	TOTAL	$ 54,000	$ 87,000	$ 62,500

The Multi-Period approach performed the best with respect to total system cost over the planning horizon. By looking at all of the periods simultaneously, this approach was able to justify the acquisition of duplicate machines in terms of the material handling savings over the entire planning horizon. The total system cost for the Fixed Cell / Fixed Family approach is 61.1% higher than the Multi-Period approach. The Optimal By Period approach results in a 15.7% increase in system cost over the Multi-Period approach. If the planned acquisition of resources is taken into account (totaling $16,000), these percentages become 86.8% and 22.4%, respectively.

4.3 Design Problem 3

The third design problem consists of 22 parts to be produced by seven unique machine types. The planning horizon for this problem is five periods. The machine-part incidence matrix, with processing times, is shown in Figure 4-3. The operation sequence and period demands for the 22 parts are listed in Table 4-7. Relevant resource data, including a schedule of planned acquisitions, are given in Table 4-8.

PARTS

MACHINES		1	2	3	4	5	6	7	8	9	10	11	12	13	14
	A		0.7			2.4			2.4	2.5					2.7
	B	0.5			0.6	0.9	2.1	1.4							
	C										2.5	3.0	0.7	1.6	
	D			3.1				1.4		1.0		0.6	1.0		
	E		1.2						4.4						3.8
	F	0.5					4.6		2.3						
	G	0.6			4.7	3.6	1.5			3.9	4.7				

FIGURE 4-3. Matrix of Processing Times for Design Problem 3

TABLE 4-7
Part Data for Design Problem 3

Part No.	M/H Cost per Unit	Operation Sequence	Demand Period 1	Period 2	Period 3	Period 4	Period 5
1	$ 1	2 - 6 - 7	1,040	347	0	0	0
2	1	1 - 5	1,040	1,388	1,388	833	278
3	1	4	1,040	1,040	624	208	0
4	1	2 - 7	1,040	1,388	1,388	833	278
5	1	1 - 2 - 7	1,040	1,010	624	208	0
6	1	2 - 6 - 7	1,040	347	0	0	0
7	1	2 - 4	1,040	1,560	2,080	2,080	1,248
8	1	1 - 5 - 6	1,040	347	0	0	0
9	1	1 - 4 - 7	1,040	1,388	1,388	833	278
10	1	3 - 7	1,040	1,388	1,388	833	278
11	1	3 - 4	1,040	1,560	2,080	2,080	1,248
12	1	3 - 4	1,040	347	0	0	0
13	1	3	1,040	624	208	0	0
14	1	1 - 5	1,040	347	0	0	0
15	1	3 - 1 - 4	0	0	879	1,319	1,759
16	1	4	0	0	983	1,475	1,966
17	1	2 - 6 - 5	0	0	1,023	1,535	2,046
18	1	1 - 4	0	0	913	1,370	1,826
19	1	3 - 1 - 4	0	0	963	1,446	1,926
20	1	2 - 5	0	0	0	958	1,437
21	1	3	0	0	0	0	1,035
22	1	4 - 5 - 1	0	0	0	0	873

TABLE 4-8
Resource Data for Design Problem 3

Machine Type	Acquisition Cost	Relocation Cost	Capacity	Number Available Period 1	Planned Acquisitions Period 2	Period 3	Period 4	Period 5
1	$ 1200	$ 600	8320	2				1
2	3000	1500	8320	1		1		
3	2500	1250	8320	1	1			1
4	800	400	8320	1		2		1
5	3000	1500	8320	2				
6	1000	500	8320	1				1
7	3000	1500	8320	3				

This problem is a multi-period extension of the 14 part, seven machine example used by Logendran (1990). The processing times and demand for the first 14 parts were taken from this example. Product lives for these parts were randomly generated. The processing times, operation sequence, and introductory period demand were randomly generated for the eight parts that are introduced to the system during the last three periods of the planning horizon.

In addition to constraints on system and within cell capacity, the following restrictions were placed on the design of the cellular manufacturing system.

1. Three machine cells and three part families are to be formed.
2. Each machine cell must contain at least two machine types.
3. Each part family must have at least one member.

The per unit material handling cost is $1 for all parts. No changes in the cost information, machine capacities, operation sequences, or processing times are anticipated during the planning horizon. Appendix A.3 contains the cellular designs obtained using the Multi-Period, Fixed Cell / Fixed Family, and Optimal By Period Approaches. Table 4-9 summarizes the performance of these approaches with respect to the design objectives.

As expected, the Optimal By Period approach performed the best with respect to material handling cost and machine acquisition cost. Compared to the other design problems, very little machine relocation is required to re-configure the system every period even though the planning horizon is longer. This result may be attributed in part to the size of the machine population (seven machines v. 19 and 11 in the other design problems).

Once again, the Fixed Cell / Fixed Family approach suffered due to the inability to relocate machines and/or reassign parts. This approach performed the worst with respect to both material handling costs and machine acquisition costs. With respect to total system cost over the five period planning horizon, the Multi-Period approach performed the best. However, the difference between the total system cost for the Multi-Period approach and the Optimal By Period approach is not as substantial as it was in the previous design problems (there is only a 2.3% increase in total cost).

TABLE 4-9
Comparison of Results for Design Problem 3

		Multi-Period Approach	Fixed Cell / Fixed Family Approach	Optimum by Period Approach
Material Handling Cost	Period 1	$ 0	$ 0	$ 0
	Period 2	0	0	0
	Period 3	0	1,842	0
	Period 4	0	2,765	0
	Period 5	556	$ 3,685	0
	Sub-Total	$ 556	$ 8,292	$ 0
Machine Acquisition Cost	Period 1	$ 2,200	$ 800	$ 800
	Period 2	5,000	2,500	2,500
	Period 3	4,600	5,800	3,800
	Period 4	800	0	800
	Period 5	800	8,500	4,700
	Sub-Total	$ 13,400	$ 16,800	$ 12,600
Machine Relocation Cost	Period 1	$ 0	$ 0	$ 0
	Period 2	0	0	0
	Period 3	1,100	0	2,350
	Period 4	600	0	1,500
	Period 5	3,250	0	2,900
	Sub-Total	$ 4,950	$ 0	$ 6,750
Total System Cost	Period 1	$ 2,200	$ 800	$ 800
	Period 2	5,000	2,500	2,500
	Period 3	5,700	7,642	6,150
	Period 4	1,400	2,765	2,300
	Period 5	$ 4,606	$ 12,185	$ 7,600
	TOTAL	$ 18,906	$ 25,092	$ 19,350

Generalizing the results over the three design problems, the superior overall performance of the multi-period approach is attributed to its ability to measure the impact of a design decision in one period over the planning horizon. For example, when deciding whether or not to acquire additional resources, the multi-period algorithm is able to trade-off the cost of the resource against the savings in material handling costs and/or machine relocation costs over the entire planning horizon. Single period approaches are limited to justifying extra resources based on the expected savings in only the current period. In addition, scheduled acquisitions of machines are an input to the multi-period algorithm. As a result, the algorithm is able to investigate the possibility of acquiring a machine in an earlier period to reduce other costs. Based on just three randomly generated test problems, these results are not conclusive. However, the results do encourage further study into using a multi-period approach to designing cellular manufacturing systems.

4.4 Summary

In this chapter, the potential benefits of using the multi-period PF/MC formation methodology were investigated. The results of the multi-period approach were compared to two single period design methodologies. Although the cost savings associated with the multi-period approach varied with each design problem, it has been demonstrated that in the presence of changing product mix and resource availability, a multi-period approach is beneficial.

Chapter 5 concludes the main body of this dissertation by highlighting the contributions of this research and identifying areas of future research.

5.0 Conclusions, Contributions, and Extensions

In this dissertation, a new methodology for the preliminary design of a cellular manufacturing system was developed. This methodology is unique in that it works with a forecast of product mix changes and resource availability. The problem considered was the formation of part families and machine cells. The objectives of the design methodology are the minimization of intercell transfers, the minimization of machine duplication, and the minimization of system re-configuration. In addition to the forecast of product mix and resource availability, the methodology includes many other relevant system parameters, such as the operation sequence of parts and the capacities of the resources. The advantage of using this methodology to form part families and machines cells is that a cellular system that performs well with respect to the design objectives over the entire planning horizon is obtained.

In Chapter 3, a mathematical model of the multi-period PF/MC formation problem was developed and a solution procedure based on a genetic algorithm was illustrated. The potential benefits of considering a changing product mix and resource population were identified in Chapter 4. This chapter highlights the contributions of this research and identifies areas for model improvement and future research.

5.1 Conclusions

The performance of the multi-period approach to PF/MC formation was evaluated in Chapter 4. In each of the three design problems considered, the multi-period approach outperformed the single period approaches. Although a sensitivity analysis was not performed, the following conclusions have been drawn from the results of the three design problems.

1. The multi-period approach tends to make planned acquisitions early and to purchase duplicate machines to reduce material handling costs. The planned acquisition of machines is derived from the inputs to the multi-period model. Thus the model is able to investigate purchasing the machines in an earlier period to reduce material handling costs. Machines are duplicated more often with the multi-period approach because the model is able to measure the potential

reduction in material handling cost and machine relocation cost across the entire planning horizon.

2. The Fixed Cell / Fixed Family approach tends to result in excessive material handling costs and machine acquisition costs. The increase in material handling cost results from the introduction of new parts. The composition of the machine cells is determined without considering parts produced beyond the first period. As a result, when new parts are introduced, there may not be a machine cell that is capable of producing the entire part. The increase in machine acquisition costs is a result of the fixed machine cells. By not considering the relocation of machines in underutilized cells, additional machines must be acquired to compensate for increases in demand.
3. The Optimal by Period approach performs well with respect to material handling costs and machine acquisition costs. However, this approach results in a significant amount of machine relocation and part reassignments between periods. How this approach compares to the multi-period approach depends on the cost of machine relocation.

The performance results with respect to the design problems indicate that the multi-period approach is capable of designing a system that performs well over the entire planning horizon. In addition, the multi-period approach can result in a better overall system design (with respect to total system cost) than a single-period approach.

5.2 Research Contributions

The majority of existing PF/MC formation procedures assume that the product mix and resource population are constant. The primary contribution of this research is the recognition of the variable nature of the production system in the design of part families and machine cells. In the presence of a changing product mix and resource population, a cellular system designed under these simplifying assumptions may not continue to provide the manufacturing efficiencies the system was originally designed to achieve. Even if periodic adjustments are made to the system as the part and machine populations change, the cost of some of these adjustments could have been avoided if the system had been designed with the variable nature of the production environment in mind. This was demonstrated in Chapter 4.

Few models of the PF/MC formation problem have attempted to address the changing nature of the production environment. Seifoddini (1990) addressed the uncertain nature of the product mix in his probabilistic PF/MC formation technique. Harhalakis *et al.* (1994) included random variations in product demand in their model. Unlike Seifoddini, who looked at multiple product mixes for a single period, Harhalakis *et al.* considered product demand changes over a series of periods in a planning horizon. However, neither of these models explicitly addressed the introduction of new parts to the production system or changes in the resource population.

Another research contribution derived from the multi-period PF/MC formation methodology is its ability to not only design a new cellular manufacturing system, but also its ability to investigate improvements to an existing system. When unanticipated changes in the product mix and/or resource availability occur after the cellular system has been designed and implemented, poor system performance may result. The multi-period model has the ability to include the current system configuration in the set of system parameters. As a result, the methodology is capable of investigating the re-configuration of the current system so that inherent efficiencies in the cellular system can be maintained.

Vakharia and Kaku (1993) developed a system redesign methodology that addresses long term demand changes. The redesign methodology consisted of reallocating parts to families to regain the benefits of the cellular system. Although machine relocation was mentioned as a potential solution to product mix changes, it was not included in their redesign strategy.

The use of a genetic algorithm to solve the multi-period PF/MC formation problem also has its advantages. Although the model developed in this dissertation focuses on three specific design objectives, the structure of the genetic algorithm procedure facilitates changing or adding to the list of design objectives. Changing the design objectives requires only that the evaluation function be modified. Entire re-programming of the procedure is not required.

A second benefit of using a genetic algorithm is that it inherently provides alternative system designs. A list of the best solutions found as the genetic algorithm moves from generation to generation can be maintained. These alternative designs can then be evaluated with respect to secondary design objectives and constraints.

5.3 *Improvements to the Basic Methodology*

As is the case with most (if not all) research, there is room for improvement in the multi-period PF/MC formation model and solution procedure described in this dissertation. The purpose of this section is to highlight some of the weaknesses of the multi-period model and to suggest ways to improve it. Some of these areas overlap with ideas for future research, which are to be discussed in the next section.

One weakness of the multi-period PF/MC formation model is the assumption that the required cost information is available; in particular, that the cost of relocating machines is known. While it may be possible to estimate the cost of special handling equipment and the labor involved in physically moving the machine to its new location, the other consequences of system re-configuration may not be as easily anticipated or measured. For example, it is likely that there will be some loss or backlog of production due to the unavailability of the machine as it is being moved. In addition, moving a machine into an existing cell may require some adjustments in the location of machines already in the cell due to space limitations and desired flow characteristics within the cell. Considerations such as these make the task of assigning a machine relocation cost difficult.

There are two primary opportunities for improvement in the multi-period PF/MC formation solution procedure as presented in this dissertation. The first opportunity for improvement involves the part assignment heuristic used within the genetic algorithm to assign parts to part families. The current heuristic seeks to assign parts to families such that intercell transfers are minimized. Each part is assigned to the cell that will result in the fewest intercell transfers. Although this assignment process is in line with the objective of minimizing material handling costs, it does not consider the impact of the part assignment on machine acquisition cost due to the within cell capacity constraint. For example, a part may be assigned to cell 1 because only one intercell transfer is required instead of being assigned to cell 2, where two intercell transfers would be required. As a result of this assignment, it may be necessary to acquire additional units of one or more machine types assigned to the cell so that the members of the part family requiring these machines can be fully processed within the cell. Since within cell capacity is not considered in the assignment of parts to cells, it is possible that the savings in material handling costs (due to one less intercell transfer) are outweighed by the increase in machine acquisition cost.

Ideally, the assignment of parts to families and machines to cells should occur simultaneously as part of the genetic algorithm. In this manner, the assignment of parts to families would reflect a trade-off among all design objectives, instead of just one objective. Modifying the genetic algorithm to include part assignment would require augmenting the current chromosomal structure with the part assignment decision variables. This addition would increase the computer requirements for running the genetic algorithm. It is likely that a workstation, instead of a PC, would be required to run the genetic algorithm.

The second opportunity for improvement involves the ability of the genetic algorithm to consistently find the best solution. Given the probabilistic nature of the genetic algorithm, it is common practice to run the algorithm several times and select the best solution found over those runs as the recommended system design. However, it has been observed that the genetic algorithm frequently converges to a sub-optimal solution. The likelihood of premature convergence appears to be related to the problem size. In the validation runs in Chapter 3, the genetic algorithm was able to consistently find the best solution. However, in the multi-period applications of Chapter 4, the reported solution typically occurred only once.

The solution to this problem requires further investigation into the selection of control parameters used in the genetic algorithm. Changing the reproduction operator and developing more problem specific crossover operators and mutation operators may improve the algorithms performance. In addition, the replacement strategy used to form the next generation of solution may need to be changed. A whole area of research could be devoted to finding the best control parameters for the PF/MC formation problem.

5.4 Areas for Future Research

In this section, a number of areas for future research are identified. Some of these involve the multi-period formulation of the PF/MC formation problem. Others apply to the PF/MC formation problem in general.

1. In the development of the multi-period PF/MC formation model, it was assumed that the processing sequence of parts was fixed and that alternate process plans did not exist. The general applicability of the model would be improved by

relaxing this assumption and making the necessary modifications to include process plan selection in situations where alternate process plans exist.

2. In this research, only three out of many possible design objectives were explicitly considered. The selection of design objectives is likely to be situation dependent. Therefore further research should be done to identify the most important design objectives, their interdependence, and measures of performance. Methods for combining these objectives into an evaluation function that can be used by the genetic algorithm also need to be investigated.

3. In Chapter 4, the cost advantage of using the multi-period PF/MC formation methodology varied across the design problems. In some problems, the cost advantage of the multi-period approach (versus the single period approaches) was significant; in other problems there was not much difference in the total system costs. This variation in results is due in part to the fact that the design problems used to evaluate the performance of the multi-period PF/MC formation methodology were randomly generated. An experiment is needed focusing on identifying what characteristics of the changing production system necessitate a multi-period approach.

4. An underlying assumption in the multi-period PF/MC formation procedure developed in this dissertation is that the forecast of part demand and resource availability is known with certainty. A valuable extension to this research would be to include uncertainty, especially considering the introduction of new parts and their production routings.

5. Research in the area of how to use information about an existing cellular manufacturing system to aid in the design of new parts would be beneficial. As a new part is being designed, the part specifications and the characteristics of the cellular system should be considered concurrently. Given that there are multiple processes by which a part feature can be realized, the development of process plans should consider machine capabilities, current machine cell utilization, and the existing system configuration. By considering these factors during the part design process, it should be possible to incorporate new parts into the system efficiently. An extension to the PF/MC formation procedure that would allow a designer to evaluate the impact of alternative process plans for new products before the design is finalized would be valuable.

6. The following areas of future research apply to the PF/MC formation problem in general.
    a) It may not be desirable to produce all parts within the cellular system. A methodology for identifying the subset of parts to be included in the PF/MC formation problem is needed.
    b) Guidelines for determining the number of cells to be formed and limits on the size of machine cells and part families need to be developed. Most procedures, including the procedure described in this dissertation, assume that these parameters are known.
    c) There are many system parameters that could be included in the PF/MC formation problem. Some existing models have considered parameters such as sequence dependent set up times and the layout of machines within each cell. Further research is needed to differentiate between the parameters appropriate for use in the preliminary design stage and those that should be included in the detailed design stage.

5.5 Summary

The multi-period PF/MC formation methodology developed in this dissertation is a significant step toward addressing the dynamic nature of the production environment in the design of a cellular manufacturing system. By including a forecast of product mix and resource availability as system parameters, it is possible to design a cellular manufacturing system that performs well with respect to the design objectives over the entire planning horizon.

The output of this methodology is a period by period description of the part families and machine cells. The need for the acquisition of additional machines and the relocation of the machines is known in advance and thus, can be adequately planned for. The multi-period PF/MC formation methodology is capable of designing a new system and improving an existing system. Lastly, the methodology provides a means for generating alternative system designs.

A few weaknesses of the multi-period PF/MC formation methodology have been identified (e.g., the part assignment heuristic and repeatability of solutions). Future research is needed to correct these weaknesses and expand the applicability of the model. The need for research in this area continues.

References

Al-Qattan, Ibrahim, 1990, "Designing Flexible Manufacturing Cells Using a Branch and Bound Method," International Journal of Production Research, Vol. 28, No. 2, pp. 325-336.

Askin, Ronald G. and Kuohung Simon Chiu, 1990, "A Graph Partitioning Procedure for Machine Assignment and Cell Formation in Group Technology," International Journal of Production Research, Vol. 28, No. 8, pp. 1555-1572.

Askin, Ronald G., Steven H. Cresswell, Jeffrey B. Goldberg, and Asoo J. Vakharia, 1991, "A Hamiltonian Path Approach to Reordering the Part-Machine Matrix for Cellular Manufacturing," International Journal of Production Research, Vol, 29, No. 6, pp. 1081-1100.

Ballakur, Arvind and Harold J. Steudel, 1987, "A Within-Cell Utilization Based Heuristic for Designing Cellular Manufacturing Systems," International Journal of Production Research, Vol. 25, No 5, pp. 639-665.

Bhat, M.V. and A. Haupt, 1976, "An Efficient Clustering Algorithm," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 1, pp. 61- 64.

Biegel, John E. and James J. Davern, 1990, "Genetic Algorithms and Job Shop Scheduling, Computers and Industrial Engineering, Vol. 19, No. 1- 4, pp. 81-91.

Boctor, Fayez F., 1991, "A Linear Formulation of the Machine-Part Cell Formation Problem," International Journal of Production Research, Vol, 29, No. 2, pp. 343-356.

Boe, Warren J. and Chun Hung Cheng, 1991, "A Close Neighbour Algorithm for Designing Cellular Manufacturing Systems," International Journal of Production Research, Vol. 29, No. 10, pp. 2097-2116.

Burbridge, J.L., 1963, "Production Flow Analysis," Production Engineer, Vol. 42, p. 742.

Burbridge, J.L., 1973, "Production Flow Analysis on a Computer," Third Annual Conference of the Institute of Production Engineers.

Burbridge, J.L., 1975, The Introduction of Group Technology, John Wiley & Sons, New York.

Carrie, A.S., 1973, "Numerical Taxonomy Applied to Group Technology and Plant Layout," International Journal of Production Research, Vol. 11, No. 4, pp. 399-416.

Chan, H.M. and D.A. Milner, 1982, "Direct Clustering Algorithm for Group Formation in Cellular Manufacture," Journal of Manufacturing Systems, Vol. 1, No. 1, pp. 65-75.

Choobineh, F., 1988, "A Framework for the Design of Cellular Manufacturing Systems," International Journal of Production Research, Vol. 26, No. 7, pp. 1161-1172.

Dahel, N.-E. and S.B. Smith, 1993, "Designing Flexibility into Cellular Manufacturing Systems," International Journal of Production Research, Vol. 31, No. 4, pp. 933-945.

Davis, L., 1985, "Job-Shop Scheduling with Genetic Algorithms," Proceedings of the First International Conference on Genetic Algorithms and Their Applications.

Davis, L., 1991, Object-Oriented Genetic Algorithm (OOGA), (manual and diskette), TSP - The Software Partnership, Melrose, MA.

Davis, L. and S. Coombs, 1987, "Optimizing Network Link Sizes with Genetic Algorithms," Modelling and Simulation Methodology: Knowledge Systems Paradigms (Edited by M. Elzas, T. Orn and B.P. Zeigler), North Holland, Amsterdam.

De Jong, K.A., 1975, "Analysis of the Behavior of a Class of Genetic Adaptive Systems," (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, Vol. 36, 5140B, University Microfilms No. 76-9381.

Dutta, S.P., R.S. Lashkari, A. Padhye, G. Nadoli, and T. Ravi, 1985, "The Cell Formation Problem in Flexible Manufacturing Systems," Proceedings of the Fourth International Conference on Flexible Manufacturing Systems, Stockholm Sweden, pp. 199-207.

Dutta, S.P., R.S. Lashkari, G. Nadoli, and T. Ravi, 1986, "A Heuristic Procedure for Determining Manufacturing Families from Design-Based Grouping for Flexible Manufacturing Systems," Computers and Industrial Engineering, Vol. 10, No. 3, pp. 193-201.

Goldberg, David E., 1983, "Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning," (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, Vol. 44, No. 10, 3174B, University Microfilms No. 8402282.

Goldberg, David E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

Grefenstette, John J., 1986, "Optimization of Control Parameters for Genetic Algorithms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 16, No. 1, pp. 122-128.

Gefenstette, John J., 1987, A User's Guide to GENESIS, Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, D.C.

Grefenstette, J.J., R. Gopal, B.J. Rosmaita, and D. Van Gucht, 1985, "Genetic Algorithms for the Traveling Salesman Problem," Proceedings of the International Conference on Genetic Algorithms and Their Applications, pp. 160-168.

Gunasingh, K. Raja and R. S. Lashkari, 1989, "The Cell Formation Problem in Cellular Manufacturing Systems -- A Sequential Modelling Approach," Computers and Industrial Engineering, Vol. 16, No. 4, pp. 469-476.

Gunasingh, K. Raja and R. S. Lashkari, 1991, "Simultaneous Grouping of Parts and Machines in Cellular Manufacturing," Computers and Industrial Engineering, Vol. 20, No. 1, pp. 111-117.

Gupta, Tarun and Hamid Seifoddini, 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, No. 7, pp. 1247-1269.

Han, Chingping and Inyong Ham, 1986, "Multiobjective Cluster Analysis for Part Family Formations," Journal of Manufacturing System, Vol. 5, No. 4, pp. 223- 230.

Harhalakis, G., R. Nagi, and J.M. Proth, 1990, "An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications," International Journal of Production Research, Vol. 28, No. 1, pp. 185-198.

Harhalakis, G., G. Ioannou, I. Minis, and R. Naji, 1994, "Manufacturing Cell Formation Under Random Product Demand," International Journal of Production Research, Vol. 32, No. 1, pp. 47-64.

Holland, John H., 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan.

Jain, A.K., R.G. Kasilingam, and S.D. Bhole, 1991, "Joint Consideration of Cell Formation and Tool Provisioning Problems in Flexible Manufacturing Systems," Computers and Industrial Engineering, Vol. 20, No. 2, pp. 271-277.

Joines, Jeffrey A., C. Thomas Culbreth, and Russell E. King, 1995, "Manufacturing Cell Design Using an Integer-Based Genetic Algorithm," Proceedings of the Fifth International Conference on Flexible Automation & Intelligent Manufacturing, Stuttgart, Germany.

Joines, Jeffrey A., Russell E. King, and C. Thomas Culbreth, 1994, "A Comprehensive Review of Manufacturing Cell Design," North Carolina State University, Working Paper.

Kao, Y. and Y.B. Moon, 1991, "A Unified Group Technology Implementation Using the Backpropagation Learning Rule of Neural Networks," Computers and Industrial Engineering, Vol. 20, No. 4, pp. 425-437.

Khator, S.K. and S.A. Irani, 1987, "Cell Formation in Group Technology: A New Approach," Computers and Industrial Engineering, Vol. 12, No. 2, pp. 131-142.

King, J.R., 1980, "Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm," International Journal of Production Research, Vol. 18, No. 2, pp. 213- 232.

King, J.R. and V. Nakornchai, 1982, "Machine-Component Group Formation in Group Technology: Review and Extension," International Journal of Production Research, Vol. 20, No. 2, pp. 117-133.

Kumar, K. Ravi, Andrew Kusiak, and Anthony Vannelli, 1986, "Grouping of Parts and Components in Flexible Manufacturing Systems," European Journal of Operational Research, Vol. 24, No. 2, pp. 387-397.

Kusiak, Andrew, 1983, "Part Families Selection Model for Flexible Manufacturing Systems," Proceedings of the Annual Industrial Engineering Conference, Louisville, Kentucky, pp. 575-580.

Kusiak, Andrew and Wing S. Chow, 1987, "Efficient Solving of the Group Technology Problem," Journal of Manufacturing Systems, Vol. 6, No. 2, pp. 117-124.

Lashkari, R.S. and K.R. Gunasingh, 1990, "Machine-Part Grouping in the Presence of Alternate Process Plans in Flexible Manufacturing Systems," Proceedings of Manufacturing International '90, Atlanta, Georgia, pp. 101-105.

Leung, L.C., S.K. Maheshwari, and W.A. Miller, 1993, "Concurrent Part Assignment and Tool Allocation in FMS with Material Handling Considerations," International Journal of Production Research, Vol. 31, No. 1, pp. 117-138.

Logendran, Rasaratnam, 1990, "A Workload Based Model for Minimizing Total Intercell and Intracell Moves in Cellular Manufacturing," International Journal of Production Research, Vol. 28, No. 5, pp. 913- 925.

McAuley, John, 1972, "Machine Grouping for Efficient Production," The Production Engineer, Vol. 51, No. 2, pp. 53-57.

McCormick, William T. (Jr.), Paul J. Schweitzer, and Thomas W. White, 1972, "Problem Decomposition and Data Reorganization by a Clustering Technique," Operations Research, Vol. 20, No. 5, pp. 993-1009.

Nagi, Rakesh, George Harhalakis, and Jean-Marie Proth, 1990, "Multiple Routeings and Capacity Considerations in Group Technology Applications," International Journal of Production Research, Vol. 28, No. 12, pp. 2243-2257.

Nygard, K. and N. Kadaba, 1990, "Improving the Perfomance of Genetic Algorithms in Automated Discovery of Parameters," Proceedings of the International Conference on Machine Learning.

Rajagopalan, R. and J.L. Batra, 1975, "Design of Cellular Production Systems: A Graph-Theoretic Approach," International Journal of Production Research, Vol. 13, No. 6, pp. 567-579.

Rajamani, Divakar, N. Singh, and Y.P. Aneja, 1990, "Integrated Design of Cellular Manufacturing Systems in the Presence of Alternative Process Plans," International Journal of Production Research, Vol. 28, No. 8, pp. 1541-1554.

Sankaran, S., 1990, "Multiple Objective Decision Making Approach to Cell Formation: A Goal Programming Model," Mathematical Computer Modelling, Vol. 13, No. 9, pp. 71-82.

Seifoddini, Hamid, 1990, "A Probabilistic Model for Machine Cell Formation," Journal of Manufacturing Systems, Vol. 9, No. 1, pp. 69-75.

Shafer, Scott M. and David F. Rogers, 1991, "A Goal Programming Approach to the Cell Formation Problem," Journal of Operations Management, Vol. 10, No. 1, pp. 28-43.

Shtub, Avraham, 1989, "Modelling Group Technology Cell Formation as a Generalized Assignment Problem," International Journal of Production Research, Vol. 27, No. 5, pp. 775-782.

Slagle, J.R., C.L. Chang, and S.R. Heller, 1975, "A Clustering and Data-Reorganizing Algorithm," IEEE Transactions on Man, Systems, and Cybernetics, Vol. SMC-5, No. 1, pp. 125-128.

Sokal, R.R. and P.H.A. Sneath, 1968, Principles of Numerical Taxonomy, Freeman and Co.

Srinivasan, G., T.T. Narendran, and B. Mahadevan, 1990, "An Assignment Model for the Part-Families Problem in Group Technology," International Journal of Production Research, Vol. 28, No. 1, pp. 145- 152.

Steudel, Harold J. and Arvind Ballakur, 1987, "A Dynamic Programming Based Heuristic for Machine Grouping in Manufacturing Cell Formation," Computers and Industrial Engineering, Vol. 12, No. 3, pp. 215-222.

Vakharia, Asoo J. and Bharat K. Kaku, 1993, "Redesigning a Cellular Manufacturing System to Handle Long-Term Demand Changes: A Methodology and Investigation," Decision Sciences, Vol. 24, No. 5, pp. 909-930.

Vakharia, Asoo J. and Urban Wemmerlöv, 1990, "Designing a Cellular Manufacturing System: A Materials Flow Approach Based on Operation Sequences," IIE Transactions, Vol. 22, No. 1, pp. 84-97.

Venugopal, V. and T.T. Narendran, 1992, "A Genetic Algorithm Approach to the Machine-Component Grouping Problem with Multiple Objectives," Computers and Industrial Engineering, Vol. 22, No. 4, pp. 469-480.

Waghodekar, P.H. and S. Sahu, 1984, "Machine-Component Cell Formation in Group Technology: MACE," International Journal of Production Research, Vol. 22, No. 6, pp. 937-948.

Wei, J.C. and G.M. Kern, 1989, "Commonality Analysis: A Linear Cell Clustering Algorithm for Group Technology," International Journal of Production Research, Vol. 27, No. 12, pp. 2053-2062.

Wemmerlöv, Urban and Nancy L. Hyer, 1986, "Procedures for the Part Family / Machine Group Identification Problem in Cellular Manufacturing," Journal of Operations Management, Vol. 6, No. 2, pp. 125-147.

Wemmerlöv, Urban and Nancy L. Hyer, 1987, "Research Issues in Cellular Manufacturing," International Journal of Production Research, Vol. 25, No. 3, pp. 413-431.

Wu, H.L., R. Venugopal, and M.M. Barash, 1986, "Design of a Cellular Manufacturing System: A Syntactic Pattern Recognition Approach," Journal of Manufacturing Systems, Vol. 5, No. 2, pp. 81-88.

Xu, Haiping and Hsu-Pin (Ben) Wang, 1989, "Part Family Formation for GT Applications Based on Fuzzy Mathematics," International Journal of Production Research, Vol. 27, No. 9, pp. 1637-1651.

Bibliography

Al-Qattan, Ibrahim, 1990, "Designing Flexible Manufacturing Cells Using a Branch and Bound Method," International Journal of Production Research, Vol. 28, No. 2, pp. 325-336.

Askin, Ronald G. and Kuohung Simon Chiu, 1990, "A Graph Partitioning Procedure for Machine Assignment and Cell Formation in Group Technology," International Journal of Production Research, Vol. 28, No. 8, pp. 1555-1572.

Askin, Ronald G., Steven H. Cresswell, Jeffrey B. Goldberg, and Asoo J. Vakharia, 1991, "A Hamiltonian Path Approach to Reordering the Part-Machine Matrix for Cellular Manufacturing," International Journal of Production Research, Vol, 29, No. 6, pp. 1081-1100.

Askin, Ronald G. and Subramanian P. Subramanian, 1987, "Cost-Based Heuristic for Group Technology Configuration," International Journal of Production Research, Vol. 25, No. 1, pp. 101-113.

Balasubramanian, K.N. and R. Panneerselvam, 1993, "Covering Technique- Based Algorithm for Machine Grouping to Form Manufacturing Cells," International Journal of Production Research, Vol. 31, No. 6, pp. 1479-1504.

Ballakur, Arvind and Harold J. Steudel, 1987, "A Within-Cell Utilization Based Heuristic for Designing Cellular Manufacturing Systems," International Journal of Production Research, Vol. 25, No 5, pp. 639-665.

Bhaskaran, Kumar, 1990, "Process Plan Selection," International Journal of Production Research, Vol. 28, No. 8, pp. 1527-1539.

Bhat, M.V. and A. Haupt, 1976, "An Efficient Clustering Algorithm," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 1, pp. 61- 64.

Biegel, John E. and James J. Davern, 1990, "Genetic Algorithms and Job Shop Scheduling, Computers and Industrial Engineering, Vol. 19, No. 1- 4, pp. 81-91.

Boctor, Fayez F., 1991, "A Linear Formulation of the Machine-Part Cell Formation Problem," International Journal of Production Research, Vol, 29, No. 2, pp. 343-356.

Boe, Warren J. and Chun Hung Cheng, 1991, "A Close Neighbour Algorithm for Designing Cellular Manufacturing Systems," International Journal of Production Research, Vol. 29, No. 10, pp. 2097-2116.

Burbridge, J.L., 1963, "Production Flow Analysis," Production Engineer, Vol. 42, p. 742.

Burbridge, J.L., 1973, "Production Flow Analysis on a Computer," Third Annual Conference of the Institute of Production Engineers.

Burbridge, J.L., 1975, The Introduction of Group Technology, John Wiley & Sons, New York.

Burgess, A.G., I. Morgan, and T.E. Vollman, 1993, "Cellular Manufacturing: Its Impact on the Total Factory," International Journal of Production Research, Vol. 31, No. 9, pp. 2059-2077.

Carrie, A.S., 1973, "Numerical Taxonomy Applied to Group Technology and Plant Layout," International Journal of Production Research, Vol. 11, No. 4, pp. 399-416.

Chan, Ding-Yu and David D. Bedworth, 1990, "Design of a Scheduling System for Flexible Manufacturing Cells," International Journal of Production Research, Vol. 28, No. 11, pp. 2037-2049.

Chan, H.M. and D.A. Milner, 1982, "Direct Clustering Algorithm for Group Formation in Cellular Manufacture," Journal of Manufacturing Systems, Vol. 1, No. 1, pp. 65-75.

Chandra, Charu, Shahrukh A. Irani, and Sant R. Arora, 1993, "Clustering Effectiveness of Permutation Generation Heuristics for Machine-Part Matrix Clustering," Journal of Manufacturing Systems, Vol. 12, No. 5, pp. 388-408.

Chandrasekharan, M.P. and R. Rajagopalan, 1986a, "An Ideal Seed Non-Hierarchical Clustering Algorithm for Cellular Manufacturing," International Journal of Production Research, Vol. 24, No. 2, pp. 451- 464.

Chandrasekharan, M.P. and R. Rajagopalan, 1986b, "MODROC: An Extension of Rank Order Clustering for Group Technology," International Journal of Production Research, Vol. 24, No. 5, pp. 1221-1233.

Chandrasekharan, M.P. and R. Rajagopalan, 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.

Chandrasekharan, M.P. and R. Rajagopalan, 1989, "GROUPABILITY: An Analysis of the Properties of Binary Data Matrices for Group Technology," International Journal of Production Research, Vol, 27, No. 6, pp. 1035-1052.

Choobineh, F., 1988, "A Framework for the Design of Cellular Manufacturing Systems," International Journal of Production Research, Vol. 26, No. 7, pp. 1161-1172.

Chow, Wing S. and Ostap Hawaleshka, 1992, "An Efficient Algorithm for Solving the Machine Chaining Problem in Cellular Manufacturing," Computers and Industrial Engineering, Vol. 22, No. 1, pp. 95-100.

Chow, W.S. and O. Hawaleshka, 1993, "Minimizing Intercellular Part Movements in Manufacturing Cell Formation," International Journal of Production Research, Vol. 31, No. 9, pp. 2161-2170.

Chu, Chao-Hsien, 1993, "Manufacturing Cell Formation by Competitive Learning," International Journal of Production Research, Vol. 31, No. 4, pp. 829-843.

Chu, Chao-Hsien and Jack C. Hayya, 1991, "A Fuzzy Clustering Approach to Manufacturing Cell Formation," International Journal of Production Research, Vol. 29, No. 7, pp. 1475-1487.

Chu, Chao-Hsien and Mayshing Tsai, 1990, "A Comparison of Three Array-Based Clustering Techniques for Manufacturing Cell Formation," International Journal of Production Research, Vol. 28, No. 8, 1417- 1433.

Co, Henry C. and Abdelaziz Araar, 1988, "Configuring Cellular Manufacturing Systems," International Journal of Production Research, Vol. 26, No. 9, pp. 1511-1522.

Dahel, N.-E. and S.B. Smith, 1993, "Designing Flexibility into Cellular Manufacturing Systems," International Journal of Production Research, Vol. 31, No. 4, pp. 933-945.

Davis, Lawrence, 1985, "Job-Shop Scheduling with Genetic Algorithms," Proceedings of the First International Conference on Genetic Algorithms and Their Applications.

Davis, Lawrence, 1987, Genetic Algorithms and Simulated Annealing, Pitman Press, New York.

Davis, Lawrence, 1991a, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York.

Davis, Lawrence, 1991b, Object-Oriented Genetic Algorithm (OOGA), (manual and diskette), TSP - The Software Partnership, Melrose, MA.

Davis, L. and S. Coombs, 1987, "Optimizing Network Link Sizes with Genetic Algorithms," Modelling and Simulation Methodology: Knowledge Systems Paradigms (Edited by M. Elzas, T. Orn and B.P. Zeigler), North Holland, Amsterdam.

De Jong, K.A., 1975, "Analysis of the Behavior of a Class of Genetic Adaptive Systems," (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, Vol. 36, 5140B, University Microfilms No. 76-9381.

Dutta, S.P., R.S. Lashkari, A. Padhye, G. Nadoli, and T. Ravi, 1985, "The Cell Formation Problem in Flexible Manufacturing Systems," Proceedings of the Fourth International Conference on Flexible Manufacturing Systems, Stockholm Sweden, pp. 199-207.

Dutta, S.P., R.S. Lashkari, G. Nadoli, and T. Ravi, 1986, "A Heuristic Procedure for Determining Manufacturing Families from Design-Based Grouping for Flexible Manufacturing Systems," Computers and Industrial Engineering, Vol. 10, No. 3, pp. 193-201.

ElMaraghy, H.A. and P. Gu, 1989, "Feature Based Expert Parts Assignment in Cellular Manufacturing," Journal of Manufacturing Systems, Vol. 8, No. 2, pp. 139-152.

Frazier, Gregory V. and Norman Gaither, 1991, "Seed Selection Procedures for Cell Formation Heuristics," International Journal of Production Research, Vol. 29, No. 11, pp. 2227-2237.

Goldberg, David E., 1983, "Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning," (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, Vol. 44, No. 10, 3174B, University Microfilms No. 8402282.

Goldberg, David E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

Grefenstette, John J., 1986, "Optimization of Control Parameters for Genetic Algorithms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 16, No. 1, pp. 122-128.

Gefenstette, John J., 1987, A User's Guide to GENESIS, Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, D.C.

Grefenstette, J.J., R. Gopal, B.J. Rosmaita, and D. Van Gucht, 1985, "Genetic Algorithms for the Traveling Salesman Problem," Proceedings of the International Conference on Genetic Algorithms and Their Applications, pp. 160-168.

Gunasingh, K. Raja and R. S. Lashkari, 1989, "The Cell Formation Problem in Cellular Manufacturing Systems -- A Sequential Modelling Approach," Computers and Industrial Engineering, Vol. 16, No. 4, pp. 469-476.

Gunasingh, K. Raja and R. S. Lashkari, 1991, "Simultaneous Grouping of Parts and Machines in Cellular Manufacturing," Computers and Industrial Engineering, Vol. 20, No. 1, pp. 111-117.

Gupta, Tarun and Hamid Seifoddini, 1990a, "Clustering Algorithms for the Design of a Cellular Manufacturing System - An Analysis for Their Performance," Computers and Industrial Engineering, Vol. 19, No. 1-4, pp. 432-436.

Gupta, Tarun and Hamid Seifoddini, 1990b, "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, No. 7, pp. 1247-1269.

Gupta, Tarun, 1993, "Design of Manufacturing Cells for Flexible Environment Considering Alternative Routeing," International Journal of Production Research, Vol. 31, No. 6, pp. 1259-1273.

Han, Chingping and Inyong Ham, 1986, "Multiobjective Cluster Analysis for Part Family Formations," Journal of Manufacturing System, Vol. 5, No. 4, pp. 223- 230.

Harhalakis, G., R. Nagi, and J.M. Proth, 1990, "An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications," International Journal of Production Research, Vol. 28, No. 1, pp. 185-198.

Harhalakis, G., G. Ioannou, I. Minis, and R. Naji, 1994, "Manufacturing Cell Formation Under Random Product Demand," International Journal of Production Research, Vol. 32, No. 1, pp. 47-64.

Heragu, S.S. and Y.P. Gupta, 1994, "A Heuristic fo Designing Cellular Manufacturing Facilities," International Journal of Production Research, Vol. 32, No. 1, pp. 125-140.

Ho, Ying-Chin, Ching-En C. Lee, and Colin L. Moodie, 1993, "Two Sequence-Pattern, Matching-Based, Flow Analysis Methods for Multi- Flowlines Layout Design," International Journal of Production Research, Vol. 31, No. 7, pp. 1557-1578.

Holland, John H., 1975, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan.

Hyer, Nancy Lea and Urban Wemmerlöv, 1989, "Group Technology in the US Manufacturing Industry: A Survey of Current Practices," International Journal of Production Research, Vol. 27, No. 8, pp. 1287-1304.

Irani, S.A., T.M. Cavalier, and P.H. Cohen, 1993, "Virtual Manufacturing Cells: Exploiting Layout Design and Intercell Flows for the Machine Sharing Problem," International Journal of Production Research, Vol. 31, No. 4, pp. 791-810.

Jain, A.K., R.G. Kasilingam, and S.D. Bhole, 1991, "Joint Consideration of Cell Formation and Tool Provisioning Problems in Flexible Manufacturing Systems," Computers and Industrial Engineering, Vol. 20, No. 2, pp. 271-277.

Joines, Jeffrey A., C. Thomas Culbreth, and Russell E. King, 1995, "Manufacturing Cell Design Using an Integer-Based Genetic Algorithm," Proceedings of the Fifth International Conference on Flexible Automation & Intelligent Manufacturing, Stuttgart, Germany.

Joines, Jeffrey A., Russell E. King, and C. Thomas Culbreth, 1994, "A Comprehensive Review of Manufacturing Cell Design," North Carolina State University, Working Paper.

Kao, Y. and Y.B. Moon, 1991, "A Unified Group Technology Implementation Using the Backpropagation Learning Rule of Neural Networks," Computers and Industrial Engineering, Vol. 20, No. 4, pp. 425-437.

Kaparthi, Shashidhar and Nallan C. Suresh, 1992, "Machine-Component Cell Formation in Group Technology: A Neural Network Approach," International Journal of Production Research, Vol. 30, No. 6, pp. 1353-1367.

Kern, Gary M. and Jerry C. Wei, 1991, "The Cost of Eliminating Exceptional Elements in Group Technology Cell Formation," International Journal of Production Research, Vol. 29, No. 8, pp. 1535-1547.

Khator, S.K. and S.A. Irani, 1987, "Cell Formation in Group Technology: A New Approach," Computers and Industrial Engineering, Vol. 12, No. 2, pp. 131-142.

King, J.R., 1980, "Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm," International Journal of Production Research, Vol. 18, No. 2, pp. 213- 232.

King, J.R. and V. Nakornchai, 1982, "Machine-Component Group Formation in Group Technology: Review and Extension," International Journal of Production Research, Vol. 20, No. 2, pp. 117-133.

Kumar, K. Ravi, Andrew Kusiak, and Anthony Vannelli, 1986, "Grouping of Parts and Components in Flexible Manufacturing Systems," European Journal of Operational Research, Vol. 24, No. 2, pp. 387-397.

Kumar, C. Suresh and M.P. Chandrasekharan, 1990, "Grouping Efficacy: A Quantitative Criterion for Goodness of Block Diagonal Forms of Binary Matrices in Group Technology," International Journal of Production Research, Vol. 28, No. 2, pp. 233-243.

Kuo, Hsiu-Wei and Robert R. Inman, 1990, "A Practical Heuristic for the Group Technology Economic Lot Scheduling Problem," International Journal of Production Research, Vol. 28, No. 4, pp. 709-722.

Kusiak, Andrew, 1983, "Part Families Selection Model for Flexible Manufacturing Systems," Proceedings of the Annual Industrial Engineering Conference, Louisville, Kentucky, pp. 575-580.

Kusiak, A., 1985, "The Part Families Problem in Flexible Manufacturing Systems," Annals of Operations Research, Vol. 3, pp. 279-300.

Kusiak, A. and M. Cho, 1992, "Similarity Coefficient Algorithms for Solving the Group Technology Problem," International Journal of Production Research, Vol. 30, No. 11, pp. 2633-2646.

Kusiak, Andrew and Wing S. Chow, 1987, "Efficient Solving of the Group Technology Problem," Journal of Manufacturing Systems, Vol. 6, No. 2, pp. 117-124.

Lashkari, R.S. and K.R. Gunasingh, 1990, "Machine-Part Grouping in the Presence of Alternate Process Plans in Flexible Manufacturing Systems," Proceedings of Manufacturing International '90, Atlanta, Georgia, pp. 101-105.

Lee, H. and A. Garcia-Diaz, 1993, "A Network Flow Approach to Solve Clustering Problems in Group Technology," International Journal of Production Research, Vol. 31, No. 3, pp. 603-612.

Leung, L.C., S.K. Maheshwari, and W.A. Miller, 1993, "Concurrent Part Assignment and Tool Allocation in FMS with Material Handling Considerations," International Journal of Production Research, Vol. 31, No. 1, pp. 117-138.

Liao, T. Warren and Lian Jiang Chen, 1993, "An Evaluation of ART1 Neural Models for GT Part Family and Machine Cell Forming," Journal of Manufacturing Systems, Vol. 12, No. 4, pp. 282-290.

Logendran, Rasaratnam, 1990, "A Workload Based Model for Minimizing Total Intercell and Intracell Moves in Cellular Manufacturing," International Journal of Production Research, Vol. 28, No. 5, pp. 913- 925.

Logendran, Rasaratnam, 1991a, "Effect of the Identification of Key Machines in the Cell Formation Problem of Cellular Manufacturing Systems," Computers and Industrial Engineering, Vol. 20, No. 4, pp. 439-449.

Logendran, Rasaratnam, 1991b, "Impact of Sequence of Operations and Layout of Cells in Cellular Manufacturing," International Journal of Production Research, Vol. 29, No. 2, pp. 375-390.

Logendran, Rasaratnam, 1992a, "A Model for Duplicating Bottleneck Machines in the Presence of Budgetary Limitations in Cellular Manufacturing," International Journal of Production Research, Vol. 30, No. 3, pp. 683-694.

Logendran, Rasaratnam, 1992b, "Simultaneous Machine-Part Grouping Approach in Manufacturing Cells," Computers and Industrial Engineering, Vol. 23, No. 1-4, pp. 77-80.

Logendran, Rasaratnam, 1993, "A Binary Integer Programming Approach for Simultaneous Machine-Part Grouping in Cellular Manufacturing Systems," Computers and Industrial Engineering, Vol. 24, No. 3.

Logendran, Rasaratnam, P. Ramakrishna, and C. Sriskandarajah, 1994, "Tabu Search-Based Heuristics for Cellular Manufacturing Systems in the Presence of Alternative Process Plans," International Journal of Production Research, Vol. 32, No. 2, pp. 273-297.

Logendran, Rasaratnam and Thomas M. West, 1990, "A Machine-Part Based Grouping Algorithm in Cellular Manufacturing," Computers and Industrial Engineering, Vol. 19, No. 1-4, pp. 57-61.

Luong, L.H.S., 1993, "A Cellular Similarity Coefficient Algorithm for the Design of Manufacturing Cells," International Journal of Production Research, Vol. 31, No. 8, pp. 1757-1766.

McAuley, John, 1972, "Machine Grouping for Efficient Production," The Production Engineer, Vol. 51, No. 2, pp. 53-57.

McCormick, William T. (Jr.), Paul J. Schweitzer, and Thomas W. White, 1972, "Problem Decomposition and Data Reorganization by a Clustering Technique," Operations Research, Vol. 20, No. 5, pp. 993-1009.

Meenakshi, Sudaram, R. and Wen-Shiom Lian, 1990, "Approach for Designing Cellular Manufacturing Systems," Manufacturing Review, Vol. 3, No. 2, pp. 91-97.

Miltenburg, J. and A.R. Montazemi, 1993, "Revisiting the Cell Formation Problem: Assigning Parts to Production Systems," International Journal of Production Research, Vol. 31, No. 11, pp. 2727-2746.

Min, H. and D. Shin, 1993, "Simultaneous Formation of Machine and Human Cells in Group Technology: A Multi-Objective Approach," International Journal of Production Research, Vol. 31, No. 10, pp. 2307-2318.

Minis, Ioannis, George Harhalakis, and Satish Jajodia, 1990, "Manufacturing Cell Formation with Multiple, Functionally Identical Machines," Manufacturing Review, Vol. 3, No. 4, pp. 252-261.

Mosier, Charles T., 1989, "An Experiment Investigation the Application of Clustering Procedures and Similarity Coefficients to the GT Machine Cell Formation Problem," International Journal of Production Research, Vol. 27, No. 10, pp. 1811-1835.

Mukhopadhyay, S.K., P. Sarkar, and R.P. Panda, 1994, "Machine-Component Grouping in Cellular Manufacturing by Multi-dimensional Scaling," International Journal of Production Research, Vol. 32, No. 2, pp. 457-477.

Nagi, Rakesh, George Harhalakis, and Jean-Marie Proth, 1990, "Multiple Routeings and Capacity Considerations in Group Technology Applications," International Journal of Production Research, Vol. 28, No. 12, pp. 2243-2257.

Nygard, K. and N. Kadaba, 1990, "Improving the Perfomance of Genetic Algorithms in Automated Discovery of Parameters," Proceedings of the International Conference on Machine Learning.

Okogbaa, O. Geoffrey, Ming-Te Chen, Chaweng Changchit, and Richard L. Shell, 1992, "Manufacturing System Cell Formation and Evaluation Using a New Inter-cell Flow Reduction Heuristic," International Journal of Production Research, Vol. 30, No. 5, pp. 1101-1118.

Purcheck, Gunter, 1985, "Machine-Component Group Formation: An Heuristic Method for Flexible Production Cells and Flexible Manufacturing Systems," International Journal of Production Research, Vol. 23, No. 5, pp. 911-943.

Rajagopalan, R. and J.L. Batra, 1975, "Design of Cellular Production Systems: A Graph-Theoretic Approach," International Journal of Production Research, Vol. 13, No. 6, pp. 567-579.

Rajamani, Divakar, N. Singh, and Y.P. Aneja, 1990, "Integrated Design of Cellular Manufacturing Systems in the Presence of Alternative Process Plans," International Journal of Production Research, Vol. 28, No. 8, pp. 1541-1554.

Rajamani, Divakar, N. Singh, and Y.P. Aneja, 1992, "Selection of Parts and Machines for Cellularization: A Mathematical Programming Approach," European Journal of Operational Research, Vol. 62, No. 1, pp. 47-54.

Ribeiro, J.F. Ferreira and B. Pradin, 1993, "A Methodology for Cellular Manufacturing Design," International Journal of Production Research, Vol. 31, No. 1, pp. 235-250.

Sankaran, S., 1990, "Multiple Objective Decision Making Approach to Cell Formation: A Goal Programming Model," Mathematical Computer Modelling, Vol. 13, No. 9, pp. 71-82.

Seifoddini, Hamid, 1989, "A Note on the Similarity Coefficient Method and the Problem of Improper Machine Assignment in Group Technology Applications," International Journal of Production Research, Vol. 27, No. 7, pp. 1161-1165.

Seifoddini, Hamid, 1990, "A Probabilistic Model for Machine Cell Formation," Journal of Manufacturing Systems, Vol. 9, No. 1, pp. 69-75.

Seifoddini, Hamid, 1990, "Machine-Component Group Analysis Versus the Similarity Coefficient Method in Cellular Manufacturing Applications," Computers and Industrial Engineering, Vol. 18, No. 3, pp. 333-339.

Shafer, Scott M., Gary M. Kern, and Jerry C. Wei, 1992, "A Mathematical Programming Approach for Dealing With Exceptional Elements in Cellular Manufacturing," International Journal of Production Research, Vol. 30, No. 5, pp. 1029-1036.

Shafer, Scott M. and Jack R. Meredith, 1990, "A Comparison of Selected Manufacturing Cell Formation Techniques," International Journal of Production Research, Vol. 28, No. 4, pp. 661-673.

Shafer, Scott M. and David F. Rogers, 1991, "A Goal Programming Approach to the Cell Formation Problem," Journal of Operations Management, Vol. 10, No. 1, pp. 28-43.

Shafer, S.M. and D.F. Rogers, 1993, "Similarity and Distance Measures for Cellular Manufacturing. Part I: A Survey," International Journal of Production Research, Vol. 31, No. 5, pp. 1133-1142.

Shafer, S.M. and D.F. Rogers, 1993, "Similarity and Distance Measures for Cellular Manufacturing. Part II: An Extension and Comparison," International Journal of Production Research, Vol. 31, No. 6, pp. 1315-1326.

Shiko, Gjergji, 1992, "Process Planning-Orientated Approach to Part Family Formation Problem in Group Technology Applications," International Journal of Production Research, Vol. 30, No. 8, pp. 1739-1752.

Shtub, Avraham, 1989, "Modelling Group Technology Cell Formation as a Generalized Assignment Problem," International Journal of Production Research, Vol. 27, No. 5, pp. 775-782.

Slagle, J.R., C.L. Chang, and S.R. Heller, 1975, "A Clustering and Data-Reorganizing Algorithm," IEEE Transactions on Man, Systems, and Cybernetics, Vol. SMC-5, No. 1, pp. 125-128.

Sokal, R.R. and P.H.A. Sneath, 1968, Principles of Numerical Taxonomy, Freeman and Co.

Srinivasan, G., T.T. Narendran, and B. Mahadevan, 1990, "An Assignment Model for the Part-Families Problem in Group Technology," International Journal of Production Research, Vol. 28, No. 1, pp. 145- 152.

Srinivasan, G. and T.T. Narendran, 1991, "GRAFICS -- A Nonhierarchical Clustering Algorithm for Group Technology," International Journal of Production Research, Vol. 29, No. 3, pp. 463-478.

Stam, Atonie and Markku Kuula, 1991, "Selecting a Flexible Manufacturing System using Multiple Criteria Analysis," International Journal of Production Research, Vol. 29, No. 4, pp. 803-820.

Steudel, Harold J. and Arvind Ballakur, 1987, "A Dynamic Programming Based Heuristic for Machine Grouping in Manufacturing Cell Formation," Computers and Industrial Engineering, Vol. 12, No. 3, pp. 215-222.

Taboun, S.M., S. Sankaran, and S. Bhole, 1991, "Comparison and Evaluation of Similarity Measures in Group Technology," Computers and Industrial Engineering, Vol. 20, No. 3, pp. 343-353.

Vakharia, Asoo J. and Bharat K. Kaku, 1993, "Redesigning a Cellular Manufacturing System to Handle Long-Term Demand Changes: A Methodology and Investigation," Decision Sciences, Vol. 24, No. 5, pp. 909-930.

Vakharia, Asoo J. and Urban Wemmerlöv, 1990, "Designing a Cellular Manufacturing System: A Materials Flow Approach Based on Operation Sequences," IIE Transactions, Vol. 22, No. 1, pp. 84-97.

Vannelli, A. and R.G. Hall, 1993, "An Eigenvector Solution Methodology for Finding Part-Machine Families," International Journal of Production Research, Vol. 31, No. 2, pp. 325-349.

Venugopal, V. and T.T. Narendran, 1992, "A Genetic Algorithm Approach to the Machine-Component Grouping Problem with Multiple Objectives," Computers and Industrial Engineering, Vol. 22, No. 4, pp. 469-480.

Venugopal, V. and T.T. Narendran, 1992, "Cell Formation in Manufacturing Systems Through Simulated Annealing: An Experimental Evaluation," European Journal of Operational Research, Vol. 63, No. 3, pp. 409-422.

Waghodekar, P.H. and S. Sahu, 1984, "Machine-Component Cell Formation in Group Technology: MACE," International Journal of Production Research, Vol. 22, No. 6, pp. 937-948.

Wei, J.C. and G.M. Kern, 1989, "Commonality Analysis: A Linear Cell Clustering Algorithm for Group Technology," International Journal of Production Research, Vol. 27, No. 12, pp. 2053-2062.

Wemmerlöv, Urban and Nancy L. Hyer, 1986, "Procedures for the Part Family / Machine Group Identification Problem in Cellular Manufacturing," Journal of Operations Management, Vol. 6, No. 2, pp. 125-147.

Wemmerlöv, Urban and Nancy L. Hyer, 1987, "Research Issues in Cellular Manufacturing," International Journal of Production Research, Vol. 25, No. 3, pp. 413-431.

Wemmerlöv, Urban and Nancy L. Hyer, 1989, "Cellular Manufacturing in the US Industry: A Survey of Users," International Journal of Production Research, Vol. 27, No. 9, pp. 1511-1530.

Wu, H.L., R. Venugopal, and M.M. Barash, 1986, "Design of a Cellular Manufacturing System: A Syntactic Pattern Recognition Approach," Journal of Manufacturing Systems, Vol. 5, No. 2, pp. 81-88.

Wu, Naiqi and G. Salvendy, 1993, "A Modified Network Approach for the Design of Cellular Manufacturing Systems," International Journal of Production Research, Vol. 31, No. 6, pp. 1409-1421.

Xu, Haiping and Hsu-Pin (Ben) Wang, 1989, "Part Family Formation for GT Applications Based on Fuzzy Mathematics," International Journal of Production Research, Vol. 27, No. 9, pp. 1637-1651.

Appendix A. Design Problem Solutions

A.1 Solutions to Design Problem 1

The solutions to Design Problem 1 obtained via the Multi-Period, Fixed Cell / Fixed Family, and Optimal By Period approaches are presented. For each approach, the recommended cellular design for each period in the planning horizon is illustrated using a machine-part incidence matrix and a summary of performance with respect to the design objectives is given. A schedule of machine acquisitions and a list of changes in the composition of machine cells and part families are also provided.

A.1.1 Multi-Period Solution to Design Problem 1

The best combination of cellular designs found by the multi-period genetic algorithm for the two periods in the planning horizon are displayed in Figures A-1 and A-2. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance of this overall design with respect to the design objectives is shown in Table A-1. The total system cost for this design is $49,470.

TABLE A-1
Multi-Period Performance Results for Design Problem 1

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 7,300	$ 15,000	$ 0
Period 2	9,400	12,000	6,000
Total	$ 16,700	$ 27,000	$ 6,000
Total System Cost = $ 49,470			

Schedule of Machine Acquisitions

1. The planned acquisitions of an additional unit of machine types 9 and 15 take place during period 1 instead of period 2.
2. An additional unit of machine type 3 is acquired in period 1.

3. The planned acquisition of one unit of machine type 1 occurs during period 2 as scheduled.
4. An additional unit of machine type 12 is acquired in period 2.

Changes in Machine Cell Composition

1. Machine types 10 and 19 are moved from cell 1 in period 1 to cell 2 in period 2.

Changes in Part Family Composition

1. Parts 16 and 29 are reassigned from part family 1 in period 1 to part family 2 in period 2.
2. Parts 12 and 13 are reassigned from part family 1 in period 1 to part family 3 in period 3.
3. Parts 3, 5, 6, 7, 14, and 18 are introduced to the system in period 2. Parts 7, 14, and 18 are assigned to family 1; part 6 is assigned to family 2; parts 3 and 5 are assigned to family 3.

PART FAMILIES

MACHINE CELLS	1	2	12	13	16	17	21	22	23	29	30	10	11	24	25	4	8	9	15	19	20	26	27	28
1 (1)									1	1	1													
3 (2)			1		1		1		1		1													
4 (1)						1	1	1						*										
5 (1)					1		1	1											*					
6 (1)			1		1	1			1										*					
7 (1)								1		1														
8 (1)	1			1			1	1			1			*										
9 (2)		1	1			1	1	1																
10 (1)			1	1	1												*							*
11 (1)		1									1													
12 (2)	1	1			1		1	1			1													
15 (1)			1	1						1														
17 (1)	1	1				1																		
19 (1)			1					1												*				
2 (1)				*								1		1										
3 (1)																								
5 (1)												1		1					*					
6 (1)													1	1	1				*					
7 (1)														1										
10 (1)												1					*							*
13 (1)															1									
14 (1)									*						1									
16 (1)													1	1										
2 (2)				*														1	1		1		1	1
3 (1)																		1						
4 (1)														*						1			1	
7 (1)																	1	1		1		1		
8 (1)														*		1			1		1			
9 (2)																1				1		1		
11 (1)																	1			1				
12 (1)																				1	1	1		
13 (1)																	1					1		
14 (2)									*								1	1	1	1			1	
15 (1)																		1						
16 (1)																1			1			1		
17 (2)																	1	1				1		1
18 (1)													*		*		1		1					1

FIGURE A-1. Design Problem 1: Multi-Period Design for Period 1

PART FAMILIES

MACHINE CELLS	1	2	7	14	17	18	21	22	23	30	6	10	11	16	24	25	29	3	4	5	8	9	12	13	15	19	20	26	27	28
1 (2)				1					1	1	*						*													
3 (2)				1			1		1	1																				
4 (1)					1		1	1							*															
5 (1)			1				1	1																	*					
6 (1)				1	1	1			1														*		*					
7 (1)								1																						
8 (1)	1		1				1	1		1					*															
9 (2)		1			1	1	1	1																						
11 (1)		1		1						1																				
12 (2)	1	1	1	1			1	1		1																				
15 (1)				1		1											*													
17 (1)	1	1			1						*																			
2 (1)											1	1			1															
3 (1)											1			1																
5 (1)											1	1		1	1										*					
6 (1)													1	1	1	1							*		*					
7 (1)															1		1													
10 (1)				*								1		1																
12 (1)														1																
13 (1)																1														
14 (1)									*							1														
16 (1)													1		1															
2 (1)																		1				1		1	1		1		1	1
3 (1)																						1	1							
4 (1)															*					1						1			1	
7 (1)																					1	1				1		1		
8 (1)															*				1					1	1		1			
9 (2)																			1				1			1		1		
10 (1)				*																	1		1	1						1
11 (1)																					1					1				
12 (1)																		1								1	1	1		
13 (1)																					1							1		
14 (2)									*												1	1			1	1			1	
15 (1)																	*	1				1	1	1						
16 (1)																			1	1					1			1		
17 (2)											*										1	1						1		1
18 (1)													*			*				1	1				1					1
19 (1)			*					*												1			1			1				

FIGURE A-2. Design Problem 1: Multi-Period Design for Period 2

A.1.2 Fixed Cell / Fixed Family Solution to Design Problem 1

The cellular designs obtained using the Fixed Cell / Fixed Family approach are given in Figures A-3 and A-4. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance results for this design are shown in Table A-2. The total system cost for this design is $64,200.

TABLE A-2
Fixed Cell / Fixed Family Performance Results for Design Problem 1

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 7,700	$ 0	$ 0
Period 2	8,500	48,000	0
Total	$ 16,200	$ 48,000	$ 0
Total System Cost = $ 64,200			

Schedule of Machine Acquisitions

1. No machines are acquired during period 1.
2. In addition to the planned acquisition of units of machine types 1, 9, 15, units of machine types 3, 4, 8, 12, 16, and 17 are also acquired during period 2 to satisfy system capacity and within cell capacity constraints.

Changes in Machine Cell Composition

As per the description of the Fixed Cell / Fixed Family approach, machines are not relocated between periods. The only changes made to the machine cells are the placement of new machines.

Changes in Part Family Composition

In the Fixed Cell / Fixed Family approach, parts are not reassigned to families between periods in the planning horizon. When new parts are introduced, they are assigned to the family that results in the fewest number of intercell transfers. The new parts introduced in this design problem are assigned to families as follows.

1. New parts 5, 7, 14, and 18 are assigned to family 1 in period 2.
2. New part 6 is assigned to cell 2 in period 2.
3. New part 3 is assigned to cell 3 in period 2.

PART FAMILIES

MACHINE CELLS

	1	2	4	11	12	13	16	17	19	20	21	22	26	27	30	9	15	23	24	29	8	10	25	28
3 (1)					1		1				1				1									
4 (1)								1	1		1	1		1										
5 (1)							1				1	1										*		
6 (1)				1	1		1	1															*	
7 (1)									1			1	1											
8 (1)	1		1			1				1	1	1			1									
9 (3)		1	1		1			1	1		1	1	1											
10 (1)					1	1	1																	
11 (1)		1							1						1									
12 (2)	1	1					1		1	1	1	1	1		1									
13 (1)													1											
15 (1)					1	1										*				*				
16 (1)			1	1									1											
17 (1)	1	1						1					1											
19 (1)					1				1			1												
1 (1)															*			1		1				
2 (2)						*				*				*		1	1		1					
3 (2)																1		1						
4 (1)																			1					
5 (1)																	1		1			*		
6 (1)																	1	1	1				*	
7 (1)																1			1	1				
8 (1)																	1		1					
14 (2)									*					*		1	1	1						
16 (1)																	1		1					
17 (1)																1								
2 (1)						*				*				*								1		1
7 (1)																					1			
10 (1)																					1	1		1
11 (1)																					1			
13 (1)																					1		1	
14 (1)									*					*							1		1	
17 (1)																					1			1
18 (1)				*													*				1		1	1

FIGURE A-3. Design Problem 1: Fixed Cell / Fixed Family Design for Period 1

PART FAMILIES

MACHINE CELLS	1	2	4	5	7	9	11	12	13	14	16	17	18	19	20	21	22	26	30	6	15	23	24	27	29	3	8	10	25	28
3 (2)						1		1		1	1					1			1											
4 (2)				1								1		1		1	1													
5 (1)					1						1					1	1											*		
6 (1)							1	1		1	1	1	1																*	
7 (1)						1								1			1	1												
8 (2)	1		1		1				1						1	1	1		1											
9 (4)		1	1					1				1	1	1		1	1	1												
10 (1)								1	1	1	1																			
11 (1)		1								1				1					1											
12 (3)	1	1			1					1	1			1	1	1	1	1	1							*				
13 (1)																		1												
15 (1)						1		1	1	1			1												*	*				
16 (2)			1	1			1											1												
17 (2)	1	1				1						1						1												
19 (1)				1	1			1						1			1													
1 (2)										*									*	1		1			1					
2 (2)						*			*						*					1	1		1	1						
3 (2)																				1		1								
4 (1)																							1	1						
5 (1)																				1	1		1					*		
6 (1)																					1	1	1						*	
7 (1)																							1		1					
8 (1)																					1		1							
14 (2)						*								*							1	1		1						
16 (1)																					1		1							
17 (1)																				1										
2 (1)						*			*						*											1		1		1
7 (1)																											1			
10 (1)																											1	1		1
11 (1)																											1			
13 (1)																											1		1	
14 (1)						*								*													1		1	
17 (1)																											1			1
18 (1)				*			*														*						1		1	1

FIGURE A-4. Design Problem 1: Fixed Cell / Fixed Family Design for Period 2

A.1.3 Optimal by Period Solution to Design Problem 1

The period by period cellular designs obtained by the Optimal By Period approach are shown in Figures A-5 and A-6. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance of this design combination with respect to the design objectives is given in Table A-3. The total system cost for this combination of period designs is $62,700.

TABLE A-3
Optimal by Period Performance Results for Design Problem 1

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 7,700	$ 0	$ 0
Period 2	5,500	18,000	31,500
Total	$ 13,200	$ 18,000	$ 31,500
Total System Cost = $ 62,700			

Schedule of Machine Acquisitions

1. No acquisition of resources takes place in period 1.
2. Units of machine type 1, 9, and 15 are acquired in period 2 as planned.

Changes in Machine Cell Composition

1. Machine types 3, 5, 8, and 1 unit of machine type 12 are moved from cell 1 in period 1 to cell 3 in period 2.
2. Machine type 4 is moved from cell 2 in period 1 to cell 3 in period 2.
3. Machine type 14 is moved from cell 2 in period 1 to cell 1 in period 2.
4. Machine types 10, 11, and 18 are moved from cell 3 in period 1 to cell 2 in period 2.
5. Machine type 17 is moved from cell 3 in period 1 to cell 1 in period 2.

Changes in Part Family Composition

1. Parts 1, 4, 11, 13, 16, 20, and 30 are moved from family 1 in period 1 to family 2 in period 2.
2. Parts 21 and 22 are moved from family 1 in period 1 to family 3 in period 2.
3. Part 27 is reassigned from family 2 in period 1 to family 3 in period 2.
4. Parts 8 and 25 are reassigned from family 3 in period 1 to family 1 in period 2.
5. Parts 10 and 28 are reassigned from family 3 in period 1 to family 2 in period 2.

PART FAMILIES

MACHINE CELLS	1	2	4	11	12	13	16	17	19	20	21	22	26	27	30	9	15	23	24	29	8	10	25	28
3 (1)					1		1				1				1									
4 (1)								1	1		1	1		1										
5 (1)							1				1	1										*		
6 (1)				1	1		1	1															*	
7 (1)									1			1	1											
8 (1)	1		1			1				1	1	1			1									
9 (3)		1	1		1			1	1		1	1	1											
10 (1)					1	1	1																	
11 (1)		1							1						1									
12 (2)	1	1					1		1	1	1	1	1		1									
13 (1)													1											
15 (1)					1	1										*				*				
16 (1)			1	1									1											
17 (1)	1	1						1					1											
19 (1)					1				1			1												
1 (1)															*			1		1				
2 (2)						*				*				*		1	1		1					
3 (2)																1		1						
4 (1)																			1					
5 (1)																	1		1			*		
6 (1)																	1	1	1				*	
7 (1)																1			1	1				
8 (1)																	1		1					
14 (2)									*					*		1	1	1						
16 (1)																	1		1					
17 (1)																1								
2 (1)						*				*				*								1		1
7 (1)																					1			
10 (1)																					1	1		1
11 (1)																					1			
13 (1)																					1		1	
14 (1)									*					*							1		1	
17 (1)																					1			1
18 (1)				*													*				1		1	1

FIGURE A-5. Design Problem 1: Optimal By Period Design for Period 1

PART FAMILIES

MACHINE CELLS	2	8	12	17	18	19	25	26	1	3	4	5	6	7	9	10	11	13	14	15	16	20	23	24	28	29	30	21	22	27
4 (1)				1		1						*												*						
6 (1)			1	1	1		1																							
7 (1)		1				1		1																						
9 (3)	1		1	1	1	1		1			*																			
10 (1)		1	1																											
11 (1)	1	1				1																								
12 (1)	1					1		1																						
13 (1)		1					1	1																						
14 (1)		1				1	1																							
15 (1)			1		1																									
16 (1)								1																						
17 (2)	1	1		1				1																						
19 (1)			1			1						*		*															*	
1 (2)													1						1				1			1	1			
2 (2)										1			1		1	1		1		1		1		1	1					
3 (2)			*										1		1				1		1		1				1			
5 (1)													1	1		1				1	1			1						
6 (1)																	1		1	1	1		1	1						
7 (1)															1									1		1				
8 (1)									1		1			1				1		1		1		1			1			
10 (1)																1		1	1		1				1					
11 (1)																			1								1			
12 (1)									1	1				1					1		1	1					1			
14 (1)															1					1			1							
15 (1)										1					1			1	1							1				
16 (1)											1	1					1			1				1						
17 (1)									1				1		1										1					
18 (1)		*					*					1					1			1					1					
2 (1)																														1
3 (1)			*																									1		
4 (1)												*												*				1	1	1
5 (1)																												1	1	
7 (1)																													1	
8 (1)																												1	1	
9 (1)											*																	1	1	
12 (1)																												1	1	
13 (1)																														
14 (1)																														1

FIGURE A-6. Design Problem 1: Optimal By Period Design for Period 2

A.2 *Solutions to Design Problem 2*

The solutions to Design Problem 2 obtained via the Multi-Period, Fixed Cell / Fixed Family, and Optimal By Period approaches are presented. For each approach, the recommended cellular design for each period in the planning horizon is illustrated using a machine-part incidence matrix and a summary of performance with respect to the design objectives is given. A schedule of machine acquisitions and a list of changes in the composition of machine cells and part families are also provided.

A.2.1 Multi-Period Solution to Design Problem 2

The composition of machine cells and part families obtained using the multi-period PF/MC formation algorithm for each period in the planning horizon are displayed in Figures A-7, A-8, and A-9. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance of the period by period designs with respect to the design objectives is given in Table A-4. The total system cost over the planning horizon for the multi-period solution is $54,000.

TABLE A-4
Multi-Period Performance Results for Design Problem 2

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 4,500	$ 29,000	$ 0
Period 2	2,500	18,000	0
Period 3	500	0	0
Total	$ 7,500	$ 47,000	$ 0
Total System Cost = $ 54,500			

Schedule of Machine Acquisitions

1. The planned acquisitions of machine types 6 and 10 take place in period 1 instead of period 2.
2. Additional units of machine types 1, 7, and 9 are acquired in period 1.

3. The acquisition of machine type 5 takes place as scheduled in period 2.
4. Additional units of machine types 2, 6, and 11 are acquired in period 2.

Changes in Machine Cell Composition

Other than the placement of newly acquired machines into the existing cells, no changes in the composition of the machine cells take place over the planning horizon.

Changes in Part Family Composition

1. Part 6 is reassigned from family 3 in period 1 to family 1 in period 2.
2. Parts 17 and 18 are moved from family 2 in period 1 to family 3 in period 2.
3. Parts 3, 4, 7, 19, and 22 are introduced to the system in period 2. Parts 4, 7, and 19 are assigned to family 1; parts 3 and 22 are assigned to family 2.
4. Part 22 is reassigned from family 2 in period 2 to family 1 in period 3.
5. Parts 11, 15, 24, and 25 are introduced to the system in period 3. Parts 11 and 15 are assigned to family 1; parts 24 and 25 are assigned to family 2.

PART FAMILIES

MACHINE CELLS		1	8	9	10	13	16	20	23	14	17	18	2	5	6	12	21
	1 (1)	1				1											
	3 (1)				1	1											
M	4 (1)		1				1	1									
A	6 (1)			1					1								
C	9 (1)	1	1					1	1								
H	10 (1)	1		1			1		1						*		
I	11 (1)		1	1	1			1									
E	1 (1)											1					
	6 (1)										1	1					
C	7 (1)									1							
E	10 (1)									1		1			*		
L	2 (1)													1			
S	5 (1)					*					*		1	1	1		
	7 (1)															1	1
	8 (1)									*			1		1		1
	9 (1)													1		1	

FIGURE A-7. Design Problem 2: Multi-Period Design for Period 1

PART FAMILIES

MACHINE CELLS	1	4	6	7	8	9	10	13	16	17	18	19	20	23	3	14	22	2	5	12	21
1 (1)	1							1			1										
3 (1)		1					1	1				1									
4 (1)					1				1				1								
5 (1)			1	1				1		1		1									
6 (2)		1		1		1				1	1	1		1							
9 (1)	1				1								1	1							
10 (1)	1	1	1	1		1			1		1			1							
11 (1)					1	1	1						1								
1 (1)															1						
2 (1)															1		1				
6 (1)																					
7 (1)																1					
10 (1)																1	1				
11 (1)															1		1				
2 (1)																			1		
5 (1)																		1	1		
7 (1)																				1	1
8 (1)			*													*		1			1
9 (1)																			1	1	

FIGURE A-8. Design Problem 2: Multi-Period Design for Period 2

PART FAMILIES

MACHINE CELLS	1	4	7	8	9	10	11	13	15	16	17	18	19	20	22	23	3	24	25	2	5	12	21
1 (1)	1						1	1				1											
3 (1)		1				1	1	1	1				1										
4 (1)				1			1		1	1				1									
5 (1)			1					1			1		1										
6 (2)		1	1		1						1	1	1			1							
9 (1)	1			1					1					1		1							
10 (1)	1	1	1		1					1		1			1	1							
11 (1)				1	1	1								1	1								
1 (1)																	1						
2 (1)															*		1	1	1				
6 (1)																			1				
7 (1)																		1	1				
11 (1)																	1						
2 (1)															*						1		
5 (1)																				1	1		
7 (1)																						1	1
8 (1)																				1			1
9 (1)																					1	1	

FIGURE A-9. Design Problem 2: Multi-Period Design for Period 3

A.2.2 Fixed Cell / Fixed Family Solution to Design Problem 2

The individual period arrangements of parts into families and machines into cells obtained by the Fixed Cell / Fixed Family solution approach are shown in Figures A-10, A-11, and A-12. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance results of these designs with respect to the three design objectives are displayed in Table A-5. The total system cost over the three period planning horizon for this design is $87,000.

TABLE A-5
Fixed Cell / Fixed Family Performance Results for Design Problem 2

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 5,000	$ 11,000	$ 0
Period 2	28,000	13,000	0
Period 3	30,000	0	0
Total	$ 63,000	$ 24.000	$ 0
Total System Cost = $ 87,000			

Schedule of Machine Acquisitions

1. The planned acquisition of machine type 6 takes place in period 1 instead of period 2.
2. The acquisitions of machine types 5 and 10 take place in period 2 as planned.

Changes in Machine Cell Composition

As per the description of the Fixed Cell / Fixed Family approach, machines are not relocated between periods. The only changes made to the machine cells are the placement of new machines.

Changes in Part Family Composition

In the Fixed Cell / Fixed Family approach, parts are not reassigned to families between periods in the planning horizon. When new parts are introduced, they are assigned to the family that results in the fewest number of intercell transfers. The new parts introduced in this design problem are assigned to families as follows.

1. Parts 3, 4, 7, 19, and 22 are introduced in period 2. Parts 3, 7, and 22 are assigned to family 1; parts 4 and 19 are assigned to family 3.
2. Parts 11, 15, 24, and 25 are introduced in period 3. Parts 11 and 15 are assigned to family 1; parts 24 and 25 are assigned to family 3.

PART FAMILIES

MACHINE CELLS	1	8	9	16	18	20	23	10	13	17	2	5	6	12	14	21
1 (1)	1				1				*							
4 (1)		1		1		1										
6 (1)			1		1		1									
9 (1)	1	1				1	1									
10 (1)	1		1	1	1		1						*		*	
11 (1)		1	1			1										
3 (1)								1	1							
6 (1)										1						
11 (1)								1								
2 (1)												1				
5 (1)									*	*	1	1	1			
7 (1)														1	1	1
8 (1)											1		1		1	1
9 (1)												1		1		

FIGURE A-10. Design Problem 2: Fixed Cell / Fixed Family Design for Period 1

PART FAMILIES

	Machine	1	3	7	8	9	16	18	20	22	23	4	10	13	17	19	2	5	6	12	14	21
	1 (1)	1	1					1						*								
	4 (1)				1		1		1													
M	6 (1)			1		1		1			1											
A	9 (1)	1			1				1		1											
C	10 (1)	1		1		1	1	1		1	1	*							*		*	
H	11 (1)		1		1	1			1	1												
I																						
N	3 (1)											1	1	1		1						
E	6 (1)											1			1	1						
	11 (1)												1									
C																						
E	2 (1)		*							*								1				
L	5 (1)			*										*	*	*	1	1	1			
L	7 (1)																			1	1	1
S	8 (1)																1		1		1	1
	9 (1)																	1		1		

FIGURE A-11. Design Problem 2: Fixed Cell / Fixed Family Design for Period 2

PART FAMILIES

	Machine	1	3	7	8	9	11	15	16	18	20	22	23	4	10	13	17	19	2	5	12	21	24	25
	1 (1)	1	1				1			1						*								
M	4 (1)				1		1	1	1		1													
A	6 (1)			1		1				1			1											*
C	9 (1)	1			1			1			1		1											
H	10 (1)	1		1		1			1	1		1	1	*										
I	11 (1)		1		1	1					1	1												
N																								
E	3 (1)						*	*						1	1	1		1						
	6 (1)													1			1	1						*
C	11 (1)														1									
E																								
L	2 (1)		*									*								1			1	1
L	5 (1)			*												*	*	*	1	1				
S	7 (1)																				1	1	1	1
	8 (1)																		1			1		
	9 (1)																			1	1			

FIGURE A-12. Design Problem 2: Fixed Cell / Fixed Family Design for Period 3

A.2.3 Optimal By Period Solution to Design Problem 2

The optimal designs with respect to material handling cost and machine acquisition cost for each period in the planning horizon are shown in Figures A-13, A-14, and A-15. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance of the individual designs are summarized in Table A-6. The total system cost over the planning horizon is $62,500.

TABLE A-6
Optimal by Period Performance Results for Design Problem 2

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 5,000	$ 11,000	$ 0
Period 2	4,500	13,000	14,000
Period 3	7,500	3,000	4,500
Total	$ 17,000	$ 27,000	$ 18,500
Total System Cost = $ 62,500			

Schedule of Machine Acquisitions

1. The planned acquisition of machine type 6 takes place in period 1 instead of period 2.
2. Additional units of machine types 9 and 11 are acquired in period 1.
3. The acquisitions of machine types 5 and 10 take place in period 2 as planned.
4. An additional unit of machine type 6 is acquired in period 3.

Changes in Machine Cell Composition

1. Machine type 1 is relocated from cell 1 in period 1 to cell 2 in period 2.
2. Machine type 7 is relocated from cell 3 in period 1 to cell 1 in period 2.
3. Machine types 2 and 9 are moved from cell 3 in period 1 to cell 2 in period 2.

4. Machine type 6 is relocated from cell 1 in period 1 to cell 3 in period 2.
5. Machine type 7 is relocated from cell 1 in period 2 to cell 2 in period 3.

Changes in Part Family Composition

1. Parts 1, 9, 18, and 23 are reassigned from family 1 in period 1 to family 2 in period 2.
2. Part 17 is reassigned from family 2 in period 1 to family 3 in period 2.
3. Part 5 is reassigned from family 3 in period 1 to family 2 in period 2.
4. Parts 12 and 14 are moved from family 3 in period 1 to family 1 in period 2.
5. Parts 3, 4, 7, 19, and 22 are introduced to the system in period 2. All of these parts are assigned to family 2.
6. Part 12 is reassigned from family 1 in period 2 to family 2 in period 3.
7. Parts 11, 15, 24, and 25 are introduced to the system in period 3. All of these parts are assigned to family 2.

PART FAMILIES

MACHINE CELLS	1	8	9	16	18	20	23	10	13	17	2	5	6	12	14	21
1 (1)	1				1				*							
4 (1)		1		1		1										
6 (1)			1		1		1									
9 (1)	1	1				1	1									
10 (1)	1		1	1	1		1						*		*	
11 (1)		1	1			1										
3 (1)								1	1							
6 (1)										1						
11 (1)								1								
2 (1)												1				
5 (1)									*	*	1	1	1			
7 (1)														1	1	1
8 (1)											1		1		1	1
9 (1)												1		1		

FIGURE A-13. Design Problem 2: Optimal By Period Design for Period 1

PART FAMILIES

MACHINE CELLS	8	12	14	16	20	1	3	4	5	7	9	10	13	18	19	22	23	2	6	17	21
4 (1)	1			1	1																
7 (1)		1	1																		*
9 (1)	1	1			1																
10 (1)			1	1															*		
11 (1)	1				1																
1 (1)						1	1						1	1							
2 (1)							1		1							1					
3 (1)								1				1	1		1						
5 (1)									1	1			1		1						
6 (1)								1		1	1			1	1		1				
9 (1)						1			1								1				
10 (1)						1		1		1	1			1		1	1		*		
11 (1)							1				1	1				1					
5 (1)																		1	1	1	
6 (1)																				1	
8 (1)			*															1	1		1

FIGURE A-14. Design Problem 2: Optimal By Period Design for Period 2

PART FAMILIES

MACHINE CELLS	8	16	20	1	3	4	5	7	9	10	11	12	13	15	18	19	22	23	24	25	2	17	21
4 (1)	1	1	1								*			*									
6 (1)																							
9 (1)	1		1																				
10 (1)		1																					
11 (1)	1		1																				
1 (1)				1	1						1		1		1								
2 (1)					1		1										1		1	1			
3 (1)						1				1	1		1	1		1							
5 (1)							1	1					1			1							
6 (1)						1		1	1						1	1		1		1			
7 (1)												1							1	1			*
9 (1)				1			1					1		1				1					
10 (1)				1		1		1	1						1		1	1					
11 (1)					1				1	1							1						
5 (1)																					1	1	
6 (1)																						1	
8 (1)																					1		1

FIGURE A-15. Design Problem 2: Optimal By Period Design for Period 3

A.3 Solutions to Design Problem 3

The solutions to Design Problem 3 obtained via the Multi-Period, Fixed Cell / Fixed Family, and Optimal By Period approaches are presented. For each approach, the recommended cellular design for each period in the planning horizon is illustrated using a machine-part incidence matrix and a summary of performance with respect to the design objectives is given. A schedule of machine acquisitions and a list of changes in the composition of machine cells and part families are also provided.

A.3.1 Multi-Period Solution to Design Problem 3

The composition of machine cells and part families obtained using the multi-period PF/MC formation algorithm for each period in the planning horizon are displayed in Figures A-16 through A-20. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance of the period by period designs with respect to the design objectives is given in Table A-7. The total system cost over the planning horizon for the multi-period solution is $18,506.

TABLE A-7
Multi-Period Performance Results for Design Problem 3

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 0	$ 2,200	$ 0
Period 2	0	5,000	0
Period 3	0	4,600	1,100
Period 4	0	800	600
Period 5	556	0	3,650
Total	$ 556	$ 12,600	$ 5,350
Total System Cost = $ 18,506			

Schedule of Machine Acquisitions

1. The planned acquisitions of machine types 1 and 6 scheduled for period 5 are made instead in period 1.
2. The planned acquisitions of machine type 3 in periods 3 and 5 are made instead in period 2.
3. The acquisitions of two units of machine type 4 and one unit of machine type 2 are made in period 3 as originally planned.
4. The unit of machine type 4 originally scheduled for acquisition in period 5 occurs in period 4.

Changes in Machine Cell Composition

1. Other than the placement of new machines, no changes are made to the machine cells between periods 1 and 2.
2. Machine type 6 is moved from cell 1 in period 2 to cell 2 in period 3.
3. Machine type 1 is moved from cell 2 in period 2 to cell 1 in period 3.
4. Machine type 1 is moved from cell 3 in period 3 to cell 2 in period 4.
5. Machine type 2 is moved from cell 1 in period 4 to cell 3 in period 5.
6. Machine type 4 is moved from cell 3 in period 4 to cell 2 in period 5.

Changes in Part Family Composition

1. Part 13 is reassigned from family 1 in period 1 to family 3 in period 2.
2. Parts 15, 16, 17, 18, and 19 are introduced to the system in period 3. Parts 15, 16, 18, and 19 are assigned to family 1; part 17 is assigned to family 2.
3. Part 2 is reassigned from family 2 in period 2 to family 3 in period 3.
4. Part 20 is introduced to the system in period 4 and is assigned to family 2.
5. Part 3 is moved from family 1 in period 3 to family 3 in period 4.
6. Part 2 is moved from family 3 in period 3 to family 2 in period 4.
7. Parts 21 and 22 are introduced to the system in period 5. Part 21 is assigned to family 1 and part 22 is assigned to family 2.
8. Parts 7 and 9 are reassigned from family 1 in period 4 to family2 in period 5.
9. Part 17 is reassigned from family 2 in period 4 to family 3 in period 5.

PART FAMILIES

MACHINE CELLS	Machine	1	3	4	5	6	7	9	10	11	12	13	2	14	8
Cell 1	1 (1)				1			1							
	2 (1)	1		1	1	1	1								
	3 (1)								1	1	1	1			
	4 (1)		1				1	1		1	1				
	6 (1)	1				1									
	7 (3)	1		1	1	1		1	1						
Cell 2	1 (1)												1	1	
	5 (1)												1	1	
Cell 3	1 (1)														1
	5 (1)														1
	6 (1)														1

FIGURE A-16. Design Problem 3: Multi-Period Design for Period 1

PART FAMILIES

MACHINE CELLS	Machine	1	3	4	5	6	7	9	10	11	12	2	14	8	13
Cell 1	1 (1)				1			1							
	2 (1)	1		1	1	1	1								
	3 (2)								1	1	1				
	4 (1)		1				1	1		1	1				
	6 (1)	1				1									
	7 (3)	1		1	1	1		1	1						
Cell 2	1 (1)											1	1		
	5 (1)											1	1		
Cell 3	1 (1)													1	
	3 (1)														1
	5 (1)													1	
	6 (1)													1	

FIGURE A-17. Design Problem 3: Multi-Period Design for Period 2

PART FAMILIES

MACHINE CELLS	Machine	3	4	5	7	9	10	11	13	15	16	18	19	17	2
Cell 1	1 (2)			1		1				1		1	1		
	2 (1)		1	1	1										
	3 (2)						1	1	1	1			1		
	4 (3)	1			1	1		1		1	1	1	1		
	7 (3)		1	1		1	1								
Cell 2	2 (1)													1	
	5 (1)													1	
	6 (1)													1	
Cell 3	1 (1)														1
	3 (1)														
	5 (1)														1
	6 (1)														

FIGURE A-18. Design Problem 3: Multi-Period Design for Period 3

PART FAMILIES

MACHINE CELLS	Machine	4	5	7	9	10	11	15	16	18	19	2	17	20	3
Cell 1	1 (2)		1		1			1		1	1				
	2 (1)	1	1	1											
	3 (2)					1	1	1			1				
	4 (3)			1	1		1	1	1	1	1				
	7 (2)	1	1		1	1									
Cell 2	1 (1)											1			
	2 (1)												1	1	
	5 (1)											1	1	1	
	6 (1)												1		
Cell 3	3 (1)														
	4 (1)														1
	5 (1)														
	6 (1)														

FIGURE A-19. Design Problem 3: Multi-Period Design for Period 4

```
                              PART FAMILIES

             |  4 10 11 15 16 18 19 21 |  2  7  9 20 22 | 17 |
      ----------------------------------------------------------
M     1 (2)  |           1        1  1 |                |    |
A     3 (3)  |     1  1  1           1  1 |             |    |
C     4 (3)  |        1  1  1  1  1    |                |    |
H     7 (1)  |  1  1                   |        *       |    |
I     ----------------------------------------------------------
N     1 (1)  |                         |  1     1     1 |    |
E     2 (1)  |  *                      |     1     1    |    |
      4 (1)  |                         |     1  1     1 |    |
C     5 (1)  |                         |  1        1  1 |    |
E     ----------------------------------------------------------
L     2 (1)  |  *                      |                |  1 |
L     5 (1)  |                         |                |  1 |
S     6 (2)  |                         |                |  1 |
      ----------------------------------------------------------
```

FIGURE A-20. Design Problem 3: Multi-Period Design for Period 5

A.3.2 Fixed Cell / Fixed Family Solution to Design Problem 3

The individual period arrangements of parts into families and machines into cells obtained by the Fixed Cell / Fixed Family solution approach are shown in Figures A-21 through A-25. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance results of these designs with respect to the three design objectives are displayed in Table A-8. The total system cost over the three period planning horizon for this design is $25,092.

TABLE A-8
Fixed Cell / Fixed Family Performance Results For Design Problem 3

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 0	$ 800	$ 0
Period 2	0	2,500	0
Period 3	1,842	5,800	0
Period 4	2,765	0	0
Period 5	3,865	8,500	0
Total	$ 8,292	$ 16,800	$ 0
Total System Cost = $ 25,092			

Schedule of Machine Acquisitions

1. An additional unit of machine type 4 is acquired in period 1.
2. The acquisition of machine type 3 is made in period 2 as planned.
3. An additional unit of machine type 1 is acquired in period 3.
4. Machine type 2 and two units of machine type 4 are acquired in period 3 as planned.
5. The acquisition of machine types 1, 3, 4, and 6 takes place in period 5 as scheduled.
6. An additional unit of machine type 5 is acquired in period 5.

Changes in Machine Cell Composition

As per the description of the Fixed Cell / Fixed Family approach, machines are not relocated between periods. The only changes made to the machine cells are the placement of new machines.

Changes in Part Family Composition

In the Fixed Cell / Fixed Family approach, parts are not reassigned to families between periods in the planning horizon. When new parts are introduced, they are assigned to the family that results in the fewest number of intercell transfers. The new parts introduced in this design problem are assigned to families as follows.

1. Parts 15, 16, 17, 18, and 19 are introduced to the system in period 3. All of these parts are assigned to family 2.
2. Part 20 is introduced in period 4 and is assigned to family 2.
3. Parts 21 and 22 are introduced to the system in period 5. Part 21 is assigned to family 1 and part 22 is assigned to family 2.

PART FAMILIES

MACHINE CELLS	10	11	12	13	1	3	4	5	6	7	8	9	2	14
3 (1)	1	1	1	1										
4 (1)		1	1											
7 (1)	1													
1 (1)								1			1	1		
2 (1)					1		1	1	1	1				
4 (1)						1				1		1		
5 (1)											1			
6 (1)					1				1		1			
7 (2)					1		1	1	1			1		
1 (1)													1	1
5 (1)													1	1

FIGURE A-21. Design Problem 3: Fixed Cell / Fixed Family Design for Period 1

PART FAMILIES

MACHINE CELLS	10	11	12	13	1	3	4	5	6	7	8	9	14	2
3 (2)	1	1	1	1										
4 (1)		1	1											
7 (1)	1													
1 (1)								1			1	1	1	
2 (1)					1		1	1	1	1				
4 (1)						1				1		1		
5 (1)											1		1	
6 (1)					1				1		1			
7 (2)					1		1	1	1			1		
1 (1)														1
5 (1)														1

FIGURE A-22. Design Problem 3: Fixed Cell / Fixed Family Design for Period 2

PART FAMILIES

	Machine	10	11	13	3	4	5	7	9	15	16	17	18	19	2
M	3 (2)	1	1	1						*				*	
A	4 (1)		1												
C	7 (1)	1													
H															
I	1 (2)						1		1	1			1	1	
N	2 (2)					1	1	1				1			
E	4 (3)				1			1	1	1	1		1	1	
	5 (1)											1			
C	6 (1)											1			
E	7 (2)					1	1		1						
L															
L	1 (1)														1
S	5 (1)														1

FIGURE A-23. Design Problem 3: Fixed Cell / Fixed Family Design for Period 3

PART FAMILIES

	Machine	10	11	3	4	5	7	9	15	16	17	18	19	20	2
M	3 (2)	1	1						*				*		
A	4 (1)		1												
C	7 (1)	1													
H															
I	1 (2)					1		1	1			1	1		
N	2 (2)				1	1	1				1			1	
E	4 (3)			1			1	1	1	1		1	1		
	5 (1)										1			1	
C	6 (1)										1				
E	7 (2)				1	1		1							
L															
L	1 (1)														1
S	5 (1)														1

FIGURE A-24. Design Problem 3: Fixed Cell / Fixed Family Design for Period 4

```
                           PART FAMILIES

           | 10 11 21 |  4  7  9 15 16 17 18 19 20 22 |  2 |
      --------------------------------------------------------
M   3 (3)  |  1  1  1 |           *           *        |    |
A   4 (1)  |     1    |                                |    |
C   7 (1)  |  1       |                                |    |
H     --------------------------------------------------------
I   1 (3)  |          |        1  1        1  1     1  |    |
N   2 (2)  |          |  1  1           1        1     |    |
E   4 (4)  |          |     1  1  1  1     1  1     1  |    |
    5 (2)  |          |                 1        1  1  |    |
C   6 (2)  |          |                 1              |    |
E   7 (2)  |          |  1     1                       |    |
L     --------------------------------------------------------
L   1 (1)  |          |                                |  1 |
S   5 (1)  |          |                                |  1 |
      --------------------------------------------------------
```

FIGURE A-25. Design Problem 3: Fixed Cell / Fixed Family Design for Period 5

A.3.3 Optimal By Period Solution to Design Problem 3

The optimal designs with respect to material handling cost and machine acquisition cost for each period in the planning horizon are shown in Figures A-26 through A-30. The number of each machine type in a cell is given in parentheses next to the machine number. In these figures, an '*' indicates an intercell transfer. The performance of the individual designs are summarized in Table A-9. The total system cost over the planning horizon is $19,350.

TABLE A-9
Optimal by Period Performance Results for Design Problem 3

	Material Handling Cost	Machine Acquisition Cost	Machine Relocation Cost
Period 1	$ 0	$ 800	$ 0
Period 2	0	2,500	0
Period 3	0	3,800	2,350
Period 4	0	800	1,500
Period 5	0	4,700	2,900
Total	$ 0	$ 12,600	$ 6,750
Total System Cost = $ 19,350			

Schedule of Machine Acquisitions

1. One of the units of machine type 4 scheduled for acquisition in period 3 is acquired in period 1.
2. Machine type 3 is acquired in period 2 as planned.
3. The acquisitions of machine type 2 and the second unit of machine type 4 take place in period 3 as planned.
4. The planned acquisition of machine type 4 in period 5 takes place instead in period 4.
5. Machine types 1, 3, and 6 are acquired as planned in period 5.

Changes in Machine Cell Composition

1. Other than the placement of the new unit of machine type 3, no changes are made in the machine cells between periods 1 and 2.
2. Machine type 6 is relocated from cell 2 in period 2 to cell 3 in period 3.
3. Machine type 1 is relocated from cell 3 in period 2 to cell 1 in period 3.
4. Machine type 3 is moved from cell 1 in period 2 to cell 2 in period 3.
5. Machine type 2 is relocated from cell 2 in period 3 to cell 1 in period 4.
6. Machine types 1 and 2 as well as two units of machine type 4 are relocated from cell 1 in period 4 to cell 2 in period 5.

Changes in Part Family Composition

1. Part 3 is reassigned from family 2 in period 1 to family 1 in period 2.
2. Parts 11, 12, and 13 are reassigned from family 1 in period 1 to family 2 in period 2.
3. Part 14 is moved from family 3 in period 1 to family 2 in period 2.
4. Parts 15, 16, 17, 18, and 19 are introduced to the system in period 3. Parts 16 and 18 are assigned to family 1; parts 15 and 19 are assigned to family 2; part 17 is assigned to family 3.
5. Part 9 is moved from family 2 in period 2 to family 1 in period 3.
6. Part 2 is reassigned from family 3 in period 2 to family 2 in period 3.

7. Part 10 is reassigned from family 1 in period 2 to family 2 in period 3.
8. Part 20 is introduced in period 4 and is assigned to family 3.
9. Parts 4, 5, and 7 are moved from family 2 in period 3 to family 1 in period 4.
10. Parts 21 and 22 are introduced to the system in period 5. Part 21 is assigned to family 3 and part 22 is assigned to family 2.
11. Parts 4, 7, 9, 16, and 18 are reassigned from family 1 in period 4 to family 2 in period 5.
12. Part 10 is moved from family 2 in period 4 to family 3 in period 5.
13. Part 20 is moved from family 3 in period 4 to family 2 in period 5.
14. Part 17 is reassigned from family 3 in period 4 to family 1 in period 5.

PART FAMILIES

MACHINE CELLS	10	11	12	13	1	3	4	5	6	7	8	9	2	14
3 (1)	1	1	1	1										
4 (1)		1	1											
7 (1)	1													
1 (1)								1			1	1		
2 (1)					1		1	1	1	1				
4 (1)						1				1		1		
5 (1)											1			
6 (1)					1				1		1			
7 (2)					1		1	1	1			1		
1 (1)													1	1
5 (1)													1	1

FIGURE A-26. Design Problem 3: Optimal By Period Design for Period 1

PART FAMILIES

	Machine	3	10	1	4	5	6	7	8	9	11	12	13	14	2
M	3 (1)		1												
A	4 (1)	1													
C	7 (1)		1												
H															
I	1 (1)					1			1	1				1	
N	2 (1)			1	1	1	1	1							
E	3 (1)										1	1	1		
	4 (1)							1		1	1	1			
C	5 (1)								1					1	
E	6 (1)			1			1		1						
L	7 (2)			1	1	1	1			1					
L															
S	1 (1)														1
	5 (1)														1

FIGURE A-27. Design Problem 3: Optimal By Period Design for Period 2

PART FAMILIES

	Machine	3	9	16	18	2	4	5	7	10	11	13	15	19	17
M	1 (1)		1		1										
A	4 (1)	1	1	1	1										
C	7 (1)		1												
H															
I	1 (1)					1		1					1	1	
N	2 (1)						1	1	1						
E	3 (2)									1	1	1	1	1	
	4 (2)								1		1		1	1	
C	5 (1)					1									
E	7 (2)						1	1		1					
L															
L	2 (1)														1
S	5 (1)														1
	6 (1)														1

FIGURE A-28. Design Problem 3: Optimal By Period Design for Period 3

PART FAMILIES

MACHINE CELLS	3	4	5	7	9	16	18	2	10	11	15	19	17	20
1 (1)			1		1		1							
2 (1)		1	1	1										
4 (2)	1			1	1	1	1							
7 (1)		1	1		1									
1 (1)								1			1	1		
3 (2)									1	1	1	1		
4 (2)										1	1	1		
5 (1)								1						
7 (1)									1					
2 (1)													1	1
5 (1)													1	1
6 (1)													1	

FIGURE A-29. Design Problem 3: Optimal By Period Design for Period 4

PART FAMILIES

MACHINE CELLS	10	21	2	4	7	9	11	15	16	18	19	20	22	17
3 (1)	1	1												
7 (1)	1													
1 (3)			1			1		1		1	1		1	
2 (1)				1	1							1		
3 (2)							1	1			1			
4 (4)					1	1	1	1	1	1	1		1	
5 (1)			1									1	1	
7 (1)				1		1								
2 (1)														1
5 (1)														1
6 (2)														1

FIGURE A-30. Design Problem 3: Optimal By Period Design for Period 5

Appendix B. Genetic Algorithm Source Code Listing

```
'*********************************************************
'*  MAIN8.BAS     SIMULTANEOUS PERIOD SOLUTIONS
'*********************************************************
REM $INCLUDE: 'COMMON8.VAR'
CLEAR , , 15000
'*********************************************************
'PARAMETERS FOR GENETIC ALGORITHM
'PS1! = POPULATION SIZE
'G1!  = MAXIMUM NUMBER OF GENERATIONS
'CR1  = CROSSOVER RATE
'MR1  = MUTATION RATE
'*********************************************************
PS1! = 30
G1! = 300
CR1 = .6
MR1 = .01
'*********************************************************
'PROGRAM PARAMETERS
'P! = NUMBER OF PERIODS IN PLANNING HORIZON
'TOPRINT = PRINT FLAG FOR DETAILED GENERATION INFORMATION
'MULTI$ = IF TRUE (T), INCLUDE RELOCATION COST IN THE FITNESS FUNCTION
'MIN$ = IF TRUE (T), USE TRANSFORMATION PROCEDURE ON RAW FITNESS VALUES
'CAPBUFFER = CAPACITY BUFFER
'MAXRUNS = MAXIMUM NUMBER OF RUNS
'*********************************************************
P! = 2
TOPRINT = 1
MULTI$ = "T"
MIN$ = "T"
CAPBUFFER = 0
MAXRUNS = 100
REM $INCLUDE: 'DATAFILE\TEST_1.VAR'
FRESULTS$ = "RESULTS\TEST_1R.EX2"
FSTATS$ = "RESULTS\TEST_1S.EX2"
FBEST$ = "RESULTS\TEST_1B.EX2"
FRUNS$ = "RESULTS\TEST_1RN.EX2"
FDATA$ = "DATAFILE\TEST_1.DAT"
FVAR$ = "DATAFILE\TEST_1.VAR"
CFILE$ = "RESULTS\TEST_1C.EX2"
'*********************************************************
'INPUT DATA ARRAYS
'MP = PROCESSING TIMES
'PD = PART DEMANDS
'MC = MACHINE CAPACITIES
'MT = AVAILABLE RESOURCES
'MI = ACQUISITION COSTS
```

```
'MR = MINIMUM MACHINE REQUIREMENTS
'P$ = OPERATION SEQUENCES
'RC = RELOCATION COSTS
'**********************************************************
DIM MP(PARTS!, MACHINES!)
DIM PD(PARTS!, PERIODS!)
DIM MC(MACHINES!)
DIM MT(MACHINES!)
DIM MI(MACHINES!)
DIM MR(MACHINES!, P!)
DIM P$(PARTS!)
DIM RC(MACHINES!, P!)
'**********************************************************
'OUTPUT DATA ARRAYS
'MAO = BEST MACHINE ASSIGNMENT
'PAO = BEST PART ASSIGNMENT
'**********************************************************
DIM MAO!(CELLS!, MACHINES!, P!)
DIM PAO!(PARTS!, P!)
CALL READINPUTDATA(FBEST$)
IF MIN$ = "F" THEN BESTM = 0 ELSE BESTM = 1000000
OPEN FBEST$ FOR APPEND AS 1
PRINT #1, DATE$; :PRINT #1, "     ";
PRINT #1, FBEST$:PRINT #1, ""
PRINT #1, "CELLS! ="; CELLS!
PRINT #1, USING "PERIODS = #"; P!
PRINT #1, USING "MMC = ##"; MMC
PRINT #1, USING "MPC = ##"; MPC
PRINT #1, USING "MHCOST = ##"; MHCOST
PRINT #1, USING "CAPACITY BUFFER = #.#"; CAPBUFFER
PRINT #1, ""
PRINT #1, "AVAILABLE MACHINES: ";
FOR J = 1 TO MACHINES!
  PRINT #1, USING " ## "; MT(J);
NEXT J
PRINT #1, "":PRINT #1, ""
PRINT #1, "REQUIRED MACHINES:":PRINT #1, ""
FOR K = 1 TO P!
  PRINT #1, "PERIOD "; :  PRINT #1, USING "##:  "; K;
  FOR J = 1 TO MACHINES!
    PRINT #1, USING " ## "; MR(J, K);
  NEXT J
  PRINT #1, ""
NEXT K
PRINT #1, ""
CLOSE 1
FOR RUNNUM = 1 TO MAXRUNS
  IF TOPRINT = 1 THEN
    OPEN "CROSS.TXT" FOR APPEND AS 1
    PRINT #1, USING "CROSSOVER RESULTS - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
```

```
    OPEN "MUTATE.TXT" FOR APPEND AS 1
    PRINT #1, USING "MUTATION RESULTS - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
    OPEN "PARENTS.TXT" FOR APPEND AS 1
    PRINT #1, USING "PARENTS RESULTS - RUN ###";RUNNUM:PRINT #1, ""
    CLOSE 1
    OPEN "MATES.TXT" FOR APPEND AS 1
    PRINT #1, USING "MATES RESULTS - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
    OPEN "MPOP.TXT" FOR APPEND AS 1
    PRINT #1, USING "POPULATION RESULTS - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
    OPEN "COSTS.TXT" FOR APPEND AS 1
    PRINT #1, USING "FITNESS RESULTS - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
    OPEN "MATEPOOL.TXT" FOR APPEND AS 1
    PRINT #1, USING "MATING POOL - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
    OPEN "PARTPOP.TXT" FOR APPEND AS 1
    PRINT #1, USING "PART ASSIGNMENTS - RUN ###"; RUNNUM:PRINT #1, ""
    CLOSE 1
  END IF
  IF MIN$ = "F" THEN BESTM = 0 ELSE BESTM = 1000000
  STARTTIME$ = TIME$
  REDIM MA(CELLS!, MACHINES!, PS1!, P!), NMA!(CELLS!,MACHINES!,PS1!,P!)
  REDIM RF(PS1!), FR(PS1!), FS(PS1!), NF(PS1!), NFR(PS1!), NFS(PS1!)
  REDIM INVEST(PS1!, P!), MOVES(PS1!, P!), RL(PS1!, P!), EFF(PS1!, P!)
  REDIM UTIL(PS1!, P!), PURCHASED!(MACHINES!, PS1!)
  RANDOMIZE TIMER
  CHLENGTH1! = CELLS! * MACHINES! * P!
  CALL INITIALIZEMACHINES(MA!())
  REDIM PA!(PARTS!, PS1!, P!)
  CALL RESULTS(FRESULTS$,PS1!,RL(),1,PURCHASED!(), MAO!(),EFF(),UTIL())
  CALL MACHINEFEASIBILITY(MA!())
  FOR POP = 1 TO PS1!
    FOR K = 1 TO P!
      CALL PARTASSIGN(MA!(), POP, PA!(), K)
    NEXT K
  NEXT POP
  CALL MACHINEFITNESS(0,MA!(),PA!(),MOVES(),RF(),FR(),FS(),INVEST(),
                      RL(),PURCHASED!(),EFF(),UTIL())
  CALL STATISTICS(0, RF(), FR(), FS(), PS1!, FSTATS$)
  FOR GEN = 1 TO G1!
    REDIM PURCHASED!(MACHINES!, PS1!)
    MUTATED! = 0
    FOR POP = 1 TO PS1!
      FOR K = 1 TO P!
        FOR C = 1 TO CELLS!
          FOR J = 1 TO MACHINES!
            IF MA!(C, J, POP, K) > 0 THEN MA!(C, J, POP, K) = 1
          NEXT J
```

```
      NEXT C
    NEXT K
  NEXT POP
  CALL REPROMACHINES(GEN, FS(), MA!(), NMA!())
  CALL MACHINEFEASIBILITY(NMA!())
  REDIM NPA!(PARTS!, PS1!, P!)
  FOR POP = 1 TO PS1!
    FOR K = 1 TO P!
      CALL PARTASSIGN(NMA!(), POP, NPA!(), K)
    NEXT K
  NEXT POP
  CALL MACHINEFITNESS(GEN,NMA!(),NPA!(),MOVES(),NF(),NFR(),NFS(),
                      INVEST(),RL(),PURCHASED!(),EFF(),UTIL())
  CALL NEWMACHINEPOP(RF(), NF(), MA!(), NMA!(), PA!(), NPA!())
  IF MIN$ = "T" THEN
    CALL MACHINEFITNESS(GEN,MA!(),PA!(),MOVES(),RF(), FR(), FS(),
                        INVEST(),RL(),PURCHASED!(),EFF(),UTIL())
  ENDIF
  CALL STATISTICS(GEN, RF(), FR(), FS(), PS1!, FSTATS$)
NEXT GEN
GEN = BG
CALL MACHINEFITNESS(GEN,MA!(),PA!(),MOVES(),RF(),FR(),FS(),INVEST(),
                    RL(),PURCHASED!(),EFF(),UTIL())

REDIM SDATA1(PS1!, 2)
FOR POP = 1 TO PS1!
  SDATA1(POP, 1) = POP:SDATA1(POP, 2) = RF(POP)
NEXT POP
SIZE% = PS1!
CALL QSORT(SDATA1(), 1, SIZE%)
IF MIN$ = "T" THEN BEST = SDATA1(1, 1) ELSE BEST = SDATA1(PS1!, 1)
FOR K = 1 TO P!
  FOR I = 1 TO PARTS!
    PAO!(I, K) = PA!(I, BEST, K)
  NEXT I
  FOR C = 1 TO CELLS!
    FOR J = 1 TO MACHINES!
      MAO!(C, J, K) = MA!(C, J, BEST, K)
    NEXT J
  NEXT C
NEXT K
OPEN FBEST$ FOR APPEND AS 1
PRINT #1, "": PRINT #1, ""
PRINT #1, "RUN "; RUNNUM
PRINT #1, "NUMBER OF GENERATIONS"; BG
PRINT #1, "LAST IMPROVEMENT"; LASTIMPROVEMENT:PRINT #1, ""
BESTMOVES = 0:BESTINVEST = 0:BESTRL = 0
FOR K = 1 TO P!
  FOR J = 1 TO MACHINES!
    PURCHASED!(J, BEST) = 0:ASSIGNED = 0
    FOR C = 1 TO CELLS!
```

```
      ASSIGNED = ASSIGNED + MAO!(C, J, K)
    NEXT C
    NEW = ASSIGNED - MT(J) - PURCHASED!(J, BEST)
    IF NEW < 0 THEN NEW = 0
    PURCHASED!(J, BEST) = PURCHASED!(J, BEST) + NEW
  NEXT J
  PRINT #1, ""
  PRINT #1, "BEST SOLUTION FOR PERIOD"; K:PRINT #1, ""
  PRINT #1, "PART ASSIGNMENT":PRINT #1, ""
  FOR CELL = 1 TO CELLS!
    PRINT #1, USING "CELL ##:"; CELL;
    FOR I = 1 TO PARTS!
      IF PAO!(I, K) = CELL THEN PRINT #1, USING " ##"; I;
    NEXT I
    PRINT #1, ""
  NEXT CELL
  PRINT #1, "":PRINT #1, "MACHINE ASSIGNMENT":PRINT #1, ""
  FOR CELL = 1 TO CELLS!
    PRINT #1, USING "CELL ##:"; CELL;
    FOR J = 1 TO MACHINES!
      FOR M = 1 TO MAO!(CELL, J, K)
        PRINT #1, USING " ##"; J;
      NEXT M
    NEXT J
    PRINT #1, ""
  NEXT CELL
  PRINT #1, ""
  PRINT #1, "MOVES"; MOVES(BEST, K):
  BESTMOVES = BESTMOVES + MOVES(BEST, K)
  PRINT #1, "INVESTMENT"; INVEST(BEST, K)
  BESTINVEST = BESTINVEST + INVEST(BEST, K)
  PRINT #1, "RELOCATION"; RL(BEST, K)
  BESTRL = BESTRL + RL(BEST, K)
  PRINT #1, "EFFICACY"; EFF(BEST, K)
  PRINT #1, "CELL UTILIZATION"; UTIL(BEST, K):PRINT #1, ""
  PRINT #1, "FITNESS"; MOVES(BEST, K) + INVEST(BEST, K) + RL(BEST, K)
  PRINT #1, ""
  FOR J = 1 TO MACHINES!
    IF PURCHASED!(J, BEST) > 0 THEN
      PRINT #1, USING "PURCHASED ##"; J;
      PRINT #1, USING " ##,"; PURCHASED!(J, BEST)
    END IF
  NEXT J
  PRINT #1, "":PRINT #1, "":PRINT #1, "    |";
  FOR C = 1 TO CELLS!
    FOR I = 1 TO PARTS!
      IF PAO!(I, K) = C THEN PRINT #1, USING " ##"; I;
    NEXT I
    PRINT #1, " |";
  NEXT C
  PRINT #1, "":PRINT #1, STRING$(PARTS! * 3 + 4 + CELLS!, "-")
```

```
    FOR C = 1 TO CELLS!
      FOR J = 1 TO MACHINES!
        IF MAO!(C, J, K) > 0 THEN
          PRINT #1, USING "## |"; J;
          FOR C2 = 1 TO CELLS!
            FOR I = 1 TO PARTS!
              IF PAO!(I, K) = C2 THEN
                IF MP(I, J) > 0 THEN
                  IF C2 = C THEN
                    PRINT #1, "  1";
                  ELSEIF MAO!(C2, J, K) = 0 THEN
                    PRINT #1, "  *";
                  ELSE
                    PRINT #1, "   ";
                  END IF
                ELSE
                  PRINT #1, "   ";
                END IF
              END IF
            NEXT I
            PRINT #1, " |";
          NEXT C2
          PRINT #1, ""
        END IF
      NEXT J
      PRINT #1, STRING$(PARTS! * 3 + 4 + CELLS!, "-")
    NEXT C
  NEXT K
  PRINT #1, "":PRINT #1, ""
  PRINT #1, "OVERALL FITNESS SCORES"
  PRINT #1, "":PRINT #1, "MOVES        ";
  PRINT #1, USING "        ###,###"; BESTMOVES
  PRINT #1, "INVESTMENT   ";
  PRINT #1, USING "###,###,###,###"; BESTINVEST
  PRINT #1, "RELOCATION   ";
  PRINT #1, USING "###,###,###,###"; BESTRL
  PRINT #1, "":PRINT #1, "TOTAL FITNESS";
  PRINT #1, USING "###,###,###,###"; BESTMOVES + BESTINVEST + BESTRL
  PRINT #1, ""
  CLOSE 1
  OPEN FRUNS$ FOR APPEND AS #1
  PRINT #1, USING "RUN ###"; RUNNUM;
  PRINT #1, FOUND$
  PRINT #1, USING "BEST ######"; BESTM
  PRINT #1, USING "GENERATIONS ##"; BG
  PRINT #1, STARTTIME$, TIME$
  PRINT #1, ""
  CLOSE 1
NEXT RUNNUM
END
```

```
'**********************************************************
FUNCTION COUNTMACHINES! (MX!(), POP, MACHINE, K)
COUNT = 0
FOR CELL = 1 TO CELLS!
  COUNT = COUNT + MX!(CELL, MACHINE, POP, K)
NEXT CELL
COUNTMACHINES! = COUNT
END FUNCTION
'**********************************************************
SUB COUNTMOVES (K, PX!(), PS!, MOVES(), MX!(), MINMOVES)
FOR POP = 1 TO PS!
  MOVES(POP, K) = 0
  FOR I = 1 TO PARTS!
    IF PD(I, K) > 0 THEN
      NUMOPS = LEN(P$(I)) / 3
      C = PX!(I, POP, K)
      FIRSTOP = VAL(MID$(P$(I), 1, 3))
      IF MX!(C, FIRSTOP, POP, K) = 0 AND NUMOPS = 1 THEN
        MOVES(POP, K) = MOVES(POP, K) + PD(I, K)
      ENDIF
      IF OPSEQ$ = "T" THEN
        FOR O = 1 TO NUMOPS - 1
          CO = VAL(MID$(P$(I), O * 3 - 2, 3))
          NO = VAL(MID$(P$(I), O * 3 + 1, 3))
          IF MX!(C,CO,POP,K)=0 OR MX!(C,NO,POP,K) = 0 THEN
            MOVES(POP, K) = MOVES(POP, K) + PD(I, K)
          ENDIF
        NEXT O
      ELSE
        FOR O = 1 TO NUMOPS
          CO = VAL(MID$(P$(I), O * 3 - 2, 3))
          NO = VAL(MID$(P$(I), O * 3 + 1, 3))
          IF MX!(C, CO, POP, K) = 0 THEN
            MOVES(POP, K) = MOVES(POP, K) + PD(I, K)
          ENDIF
        NEXT O
      END IF
    END IF
  NEXT I
  MOVES(POP, K) = MOVES(POP, K) * MHCOST!
NEXT POP
END SUB
'**********************************************************
SUB INVESTMENT (K, MX!(), INVEST(), PS!, PURCHASED!())
FOR POP = 1 TO PS!:INVEST(POP, K) = 0:NEXT POP
FOR POP = 1 TO PS!
  FOR MACHINE = 1 TO MACHINES!
    ASSIGNED = COUNTMACHINES!(MX!(), POP, MACHINE, K)
    NEW = ASSIGNED - MT(MACHINE)-PURCHASED!(MACHINE, POP)
    IF NEW < 0 THEN NEW = 0
    PURCHASED!(MACHINE,POP)=PURCHASED!(MACHINE,POP) + NEW
```

```
      INVEST(POP, K) = INVEST(POP, K) + NEW * MI(MACHINE)
    NEXT MACHINE
  NEXT POP
  END SUB
  '*********************************************************
  SUB MACHINEFITNESS (GEN,MX!(),PX!(),MOVES(),FT(),FRT(),FST(),INVEST(),
                      RL(),PURCHASED!(),EFF(),UTIL())
  REDIM PURCHASED!(MACHINES!, PS1!)
  FOR K = 1 TO P!
    CALL PARTFEASIBILITY(PS1!, PX!(), MX!(), K)
    CALL CAPACITY(PX!(), MX!(), PS1!, K)
    FOR POP = 1 TO PS1!
      REDIM ADDM!(MACHINES!)
      CALL CHECKCAPACITY(POP, K, MX!(), ADDM!())
      CALL ADDMORECAPACITY(POP, K, MX!(), ADDM!(), PX!())
    NEXT POP
    CALL COUNTMOVES(K, PX!(), PS1!, MOVES(), MX!())
    IF K > 1 THEN CALL RELOCATION(PS1!, MX!(), RL(), K, PURCHASED!())
    CALL INVESTMENT(K, MX!(), INVEST(), PS1!, PURCHASED!())
    CALL UTILIZATION(K, MX!(), PX!(), UTIL())
  NEXT K
  IF TOPRINT = 1 THEN CALL PRINTPARTPOP(GEN, PX!())
  FOR POP = 1 TO PS1!
    SUM = 0
    FOR K = 1 TO P!
      SUM = SUM + MOVES(POP, K) + INVEST(POP, K)
      IF MULTI$ = "T" THEN SUM = SUM + RL(POP, K)
    NEXT K
    IF SUM > MAXCOST THEN MAXCOST = SUM
    IF SUM < MINCOST THEN MINCOST = SUM
  NEXT POP
  FOR POP = 1 TO PS1!
    FT(POP) = 0:SUM1 = 0:SUM2 = 0
    FOR K = 1 TO P!
      SUM1 = SUM1 + MOVES(POP,K) + INVEST(POP,K) + RL(POP,K)
      SUM2 = SUM2 + MOVES(POP,K) + INVEST(POP,K)
    NEXT K
    IF MIN$ = "F" THEN
      IF MULTI$ = "T" THEN
        FT(POP) = (MAXCOST - SUM1) / MAXCOST
      ELSE
        FT(POP) = (MAXCOST - SUM2) / MAXCOST
      END IF
    ELSE
      IF MULTI$ = "T" THEN FT(POP) = SUM1 ELSE FT(POP) = SUM2
    END IF
  NEXT POP
  MIN = 10000000000#:MAX = FT(1):SUM = 0
  FOR POP = 1 TO PS1!
    IF FT(POP) = 0 THEN FT(POP) = .1
    IF FT(POP) > MAX THEN MAX = FT(POP)
```

```
  IF FT(POP) < MIN THEN MIN = FT(POP)
NEXT POP
IF MIN$ = "F" THEN GOTO SCALE
IF GEN = 0 THEN XMAX = MAX
'*********************************************************
'* THE FOLLOWING ROUTINE CONVERTS THE MINIMIZATION PROBLEM
'* TO A MAXIMIZATION PROBLEM VIA REVERSE NORMALIZATION
'*********************************************************
FOR POP = 1 TO PS1!
  FRT(POP) = MIN / FT(POP):SUM = SUM + FRT(POP)
NEXT POP
SCALE:
CALL SCALEPOP(FRT(), FST(), PS1!)
IF TOPRINT = 1 THEN
  CALL PRINTMPOP("MPOP.TXT",GEN,MX!(), FT(), FRT(), FST())
  CALL PRINTCOSTS("COSTS.TXT",GEN,MOVES(), INVEST(), RL())
ENDIF
END SUB
'*********************************************************
SUB PARTASSIGN (MX!(), MPOP, PX!(), K)
REDIM CELLCOUNT(CELLS!)
FOR I = 1 TO PARTS!
  IF PD(I, K) > 0 THEN
    REDIM LOAD(CELLS!, 2)
    NUMOPS = LEN(P$(I)) / 3
    FOR C = 1 TO CELLS!
      LOAD(C, 1) = C
      FIRSTOP = VAL(MID$(P$(I), 1, 3))
      IF MX!(C, FIRSTOP, MPOP, K) = 0 AND NUMOPS = 1 THEN
        LOAD(C, 2) = LOAD(C, 2) + PD(I, K)
      ENDIF
      IF OPSEQ$ = "T" THEN
        FOR O = 1 TO NUMOPS - 1
          CO = VAL(MID$(P$(I), O * 3 - 2, 3))
          NO = VAL(MID$(P$(I), O * 3 + 1, 3))
          IF MX!(C,CO,MPOP,K)=0 OR MX!(C,NO,MPOP,K)=0 THEN
            LOAD(C, 2) = LOAD(C, 2) + PD(I, K)
          ENDIF
        NEXT O
      ELSE
        FOR O = 1 TO NUMOPS
          CO = VAL(MID$(P$(I), O * 3 - 2, 3))
          NO = VAL(MID$(P$(I), O * 3 + 1, 3))
          IF MX!(C, CO, MPOP, K) = 0 THEN
            LOAD(C, 2) = LOAD(C, 2) + PD(I, K)
          ENDIF
        NEXT O
      END IF
    NEXT C
    NUMCELLS% = CELLS!
    CALL QSORT(LOAD(), 1, NUMCELLS%)
```

```
    TIE = 0:LASTTIE = 0
    FOR C = 1 TO CELLS! - 1
      IF LOAD(C, 2) = LOAD(C + 1, 2) THEN
        TIE = 1:LASTTIE = C + 1
      ELSE
        C = CELLS!
      END IF
    NEXT C
    IF TIE = 1 THEN
      FOR C = 1 TO CELLS!
        IF C <= LASTTIE THEN
          TC = LOAD(C, 1)
          LOAD(C, 2) = 0
          FOR TI = 1 TO PARTS!
            IF PD(TI, K) > 0 AND PX!(TI, MPOP, K) = C THEN
              LOAD(C, 2) = LOAD(C, 2) + 1
            ENDIF
          NEXT TI
        ELSE
          LOAD(C, 2) = PARTS! + 1
        END IF
      NEXT C
      NUMCELLS% = CELLS!
      CALL QSORT(LOAD(), 1, NUMCELLS%)
      PLACE = 1:YESTIE = YESTIE + 1
    ELSE
      PLACE = 1:NOTIE = NOTIE + 1
    END IF
    PX!(I, MPOP, K) = LOAD(PLACE, 1)
  ELSE
    PX!(I, MPOP, K) = 0
  END IF
NEXT I
END SUB
'*****************************************************************
SUB RELOCATION (PS!, MX!(), RL(), K, PURCHASED!())
FOR POP = 1 TO PS!
  RL(POP, K) = 0
  FOR J = 1 TO MACHINES!
    MD! = 0:ASSIGNED = COUNTMACHINES!(MX!(), POP, J, K)
    NEW = ASSIGNED - MT(J) - PURCHASED!(J, POP)
    IF NEW < 0 THEN NEW = 0
    FOR CELL = 1 TO CELLS!
      M1! = MX!(CELL, J, POP, K - 1):M2! = MX!(CELL, J, POP, K)
      IF M2! > M1! THEN
        MD! = M2! - M1!
        IF NEW > 0 THEN
          IF NEW >= MD! THEN
            NEW = NEW - MD!:MD! = 0
          ELSE
            MD! = MD! - NEW:NEW = 0
```

```
          END IF
        END IF
        RL(POP, K) = RL(POP, K) + RC(J, K) * MD!
      END IF
    NEXT CELL
  NEXT J
NEXT POP
END SUB
'**********************************************************
SUB SCALEPOP (RX(), SX(), PS!)
UTOT = RX(1):UMAX = RX(1):UMIN = RX(1)
FOR POP = 2 TO PS!
  IF RX(POP) > UMAX THEN UMAX = RX(POP)
  IF RX(POP) < UMIN THEN UMIN = RX(POP)
  UTOT = UTOT + RX(POP)
NEXT POP
UAVG = UTOT / PS!:FMULT = 1.2
 IF UMIN > (FMULT * UAVG - UMAX) / (FMULT - 1) THEN
  DELTA = UMAX - UAVG
  IF DELTA > 0 THEN
    A = (FMULT - 1) * UAVG / DELTA
    B = UAVG * (UMAX - FMULT * UAVG) / DELTA:NOSCALE$ = "F"
  ELSE
    NOSCALE$ = "T"
  END IF
ELSE
  DELTA = UAVG - UMIN
  IF DELTA > 0 THEN
    A = UAVG / DELTA:B = -UMIN * UAVG / DELTA:NOSCALE$ = "F"
  ELSE
    NOSCALE$ = "T"
  END IF
END IF
IF NOSCALE$ = "F" THEN
  FOR POP = 1 TO PS!
    SX(POP) = A * RX(POP) + B
    IF SX(POP) < 0 THEN SX(POP) = 0
  NEXT POP
ELSE
  FOR POP = 1 TO PS!:SX(POP) = RX(POP):NEXT POP
END IF
END SUB
'**********************************************************
SUB ADDMORECAPACITY (POP, K, MX!(), ADDM!(), PX!())
FOR J = 1 TO MACHINES!
  IF ADDM!(J) > 0 THEN
    ZC = 0:REDIM ZEROCELL(CELLS!, 2)
    FOR C = 1 TO CELLS!
      IF MX!(C, J, POP, K) = 0 THEN
        ZM = MACHINESREQUIRED!(C, J, POP, K, MX!(), PX!())
        IF ZM > 0 THEN
```

```
      ZC = ZC + 1:ZEROCELL(ZC, 1) = C:ZEROCELL(ZC, 2) = ZM
     END IF
    END IF
   NEXT C
   ZF% = ZC
   CALL QSORT(ZEROCELL(), 1, ZF%)
   FOR C = ZC TO 1 STEP -1
    IF ZEROCELL(C, 2) <= ADDM!(J) THEN
     MX!(ZEROCELL(C, 1), J, POP, K) = ZEROCELL(C, 2)
     ADDM!(J) = ADDM!(J) - ZEROCELL(C, 2)
    END IF
   NEXT C
   IF ADDM!(J) > 0 THEN
    ZC = 0:REDIM ZEROCELL(CELLS!, 2)
    FOR C = 1 TO CELLS!
     IF MX!(C, J, POP, K) = 0 THEN
      ZM = MACHINESREQUIRED!(C,J,POP, K, MX!(), PX!())
      IF ZM > 0 THEN
       ZC = ZC + 1:ZEROCELL(ZC, 1) = C:ZEROCELL(ZC, 2) = ZM
      END IF
     END IF
    NEXT C
    ZF% = ZC
    CALL QSORT(ZEROCELL(), 1, ZF%)
    IF ZC > 0 THEN
     MX!(ZEROCELL(1, 1), J, POP, K) = ZEROCELL(1, 2)
    END IF
   END IF
  END IF
NEXT J
END SUB
'*********************************************************
SUB CAPACITY (PX!(), MX!(), PS!, K)
FOR POP = 1 TO PS!
 FOR CELL = 1 TO CELLS!
  FOR J = 1 TO MACHINES!
   IF MX!(CELL, J, POP, K) > 0 THEN
    NEEDED=MACHINESREQUIRED!(CELL,J,POP,K,MX!(),PX!())
    IF MX!(CELL, J, POP, K) < NEEDED THEN
     MX!(CELL, J, POP, K) = NEEDED
    ENDIF
   END IF
  NEXT J
 NEXT CELL
NEXT POP
END SUB
'*********************************************************
SUB CHECKCAPACITY (POP, K, MX!(), ADDM!())
FOR J = 1 TO MACHINES!
 AM = COUNTTYPE!(MX!(), POP, J, K)
 IF AM < MR(J, K) THEN
```

```
    ADDM!(J) = MR(J, K) - AM
  END IF
NEXT J
END SUB
'**********************************************************
FUNCTION COUNTTYPE! (MX!(), POP, MACHINE, K)
COUNT = 0
FOR CELL = 1 TO CELLS!
  COUNT = COUNT + MX!(CELL, MACHINE, POP, K)
NEXT CELL
COUNTTYPE! = COUNT
END FUNCTION
'**********************************************************
FUNCTION MACHINECOUNT! (MX!(), POP, CELL, K)
COUNT! = 0
FOR J = 1 TO MACHINES!
  COUNT! = COUNT! + MX!(CELL, J, POP, K)
NEXT J
MACHINECOUNT! = COUNT!
END FUNCTION
'**********************************************************
SUB MACHINEFEASIBILITY (MX!())
FOR POP = 1 TO PS1!
  FOR K = 1 TO P!
  'CHECK TO SEE IF EACH CELL HAS AT LEAST MMC MACHINES
    FOR CELL = 1 TO CELLS!
      NOMACHINES$ = "T"
      DO WHILE NOMACHINES$ = "T"
        NOMACHINES = 0:CM = 0
        CM = MACHINECOUNT!(MX!(), POP, CELL, K)
        IF CM < MMC THEN
          NOMACHINES = 1:AM = 1 + INT(RND * MACHINES!)
          MX!(CELL, AM, POP, K) = 1
        END IF
        IF NOMACHINES = 0 THEN NOMACHINES$ = "F"
      LOOP
    NEXT CELL
  NEXT K
NEXT POP
END SUB
'**********************************************************
FUNCTION MACHINESREQUIRED! (CELL, J, POP, K, MX!(), PX!())
LOAD = 0
FOR I = 1 TO PARTS!
  IF PX!(I, POP, K) = CELL THEN
    LOAD = LOAD + MP(I, J) * PD(I, K) * (1 + CAPBUFFER)
  END IF
NEXT I
IF LOAD > 0 THEN MR = INT(LOAD / MC(J)) + 1 ELSE MR = 0
MACHINESREQUIRED! = MR
END FUNCTION
```

```
'*************************************************************
SUB PARTFEASIBILITY (PS!, PX!(), MTX!(), K)
FOR POP = 1 TO PS!
  'CHECK TO SEE THAT EACH CELL HAS AT LEAST MPC PARTS ASSIGNED TO IT
  FOR CELL = 1 TO CELLS!
    NOPART$ = "T"
    DO WHILE NOPART$ = "T"
      NOPARTS = 0:CP = 0
      FOR I = 1 TO PARTS!
        IF PX!(I, POP, K) = CELL AND PD(I, K) > 0 THEN CP = CP + 1
      NEXT I
      IF CP < MPC THEN
        NOPARTS = 1:AP = 1 + INT(RND * PARTS!)
        IF PD(AP, K) > 0 THEN PX!(AP, POP, K) = CELL
      END IF
      IF NOPARTS = 0 THEN NOPART$ = "F"
    LOOP
  NEXT CELL
NEXT POP
END SUB
'*************************************************************
FUNCTION AVERAGEFIT (FX(), PS!)
SUM = 0
FOR POP = 1 TO PS!
  SUM = SUM + FX(POP)
NEXT POP
AVERAGEFIT = SUM / PS!
END FUNCTION
'*************************************************************
FUNCTION BESTFIT! (FX(), PS!)
MIN = FX(1):BEST = 1
FOR POP = 2 TO PS!
  IF FX(POP) < MIN THEN MIN = FX(POP):BEST = POP
NEXT POP
BESTFIT! = BEST
END FUNCTION
'*************************************************************
SUB CROSSMACHINES (CROSS$,GEN,P1,P2,CP1,CP2,TMA!(),NMA!())
PLACE = 0
FOR K = 1 TO P!
  FOR C = 1 TO CELLS!
    FOR J = 1 TO MACHINES!
      IF J + PLACE <= CP1 OR J + PLACE > CP2 THEN
        NMA!(C, J, P1, K) = TMA!(C, J, P1, K)
        NMA!(C, J, P2, K) = TMA!(C, J, P2, K)
      ELSE
        NMA!(C, J, P1, K) = TMA!(C, J, P2, K)
        NMA!(C, J, P2, K) = TMA!(C, J, P1, K)
      END IF
    NEXT J
    PLACE = PLACE + MACHINES!
```

```
  NEXT C
NEXT K
IF TOPRINT=1 THEN
  CALL PRINTCROSS(GEN,CROSS$,NMA!(),TMA!(),CP1,CP2,P1,P2)
ENDIF
END SUB
'*********************************************************
FUNCTION GETINTEGER! (RANGE)
X = RND
FOR R = 1 TO RANGE
  IF X <= R / RANGE THEN
    GETINTEGER! = R
    EXIT FUNCTION
  END IF
NEXT R
END FUNCTION
'*********************************************************
SUB NEWMACHINEPOP (RF(),NF(),MA!(), NMA!(), PA!(), NPA!())
REDIM SDATA1(PS1!, 2),SDATA2(PS1!, 2),TMA(CELLS!, MACHINES!, PS1!, P!)
REDIM TF(PS1!),TPA!(PARTS!, PS1!, P!)
FOR POP = 1 TO PS1!
  SDATA1(POP, 1) = POP:SDATA1(POP, 2) = RF(POP)
  SDATA2(POP, 1) = POP:SDATA2(POP, 2) = NF(POP)
NEXT POP
SIZE% = PS1!
CALL QSORT(SDATA1(), 1, SIZE%):CALL QSORT(SDATA2(), 1, SIZE%)
IF MIN$ = "T" THEN
  LOWPLACE = PS1!:COUNT = 0
ELSE
  LOWPLACE = 1:COUNT = 0
ENDIF
IF MIN$ = "F" THEN
  START = PS1!:EL = 1:SX = -1
ELSE
  START = 1:EL = PS1!:SX = 1
ENDIF
FOR POP = START TO EL STEP SX
  IF MIN$="T" AND SDATA2(POP,2) <= SDATA1(LOWPLACE,2) THEN
    LOWPLACE = LOWPLACE - 1:COUNT = COUNT + 1:PLACE = SDATA2(POP, 1)
    TF(COUNT) = NF(PLACE)
    FOR K = 1 TO P!
      FOR I = 1 TO PARTS!
        TPA!(I, COUNT, K) = NPA!(I, PLACE, K)
      NEXT I
      FOR C = 1 TO CELLS!
        FOR J = 1 TO MACHINES!
          TMA!(C, J, COUNT, K) = NMA!(C, J, PLACE, K)
        NEXT J
      NEXT C
    NEXT K
ELSEIF MIN$="F" AND SDATA2(POP,2)>=SDATA1(LOWPLACE,2) THEN
```

```
    LOWPLACE = LOWPLACE + 1:COUNT = COUNT + 1
    PLACE = SDATA2(POP, 1):TF(COUNT) = NF(PLACE)
    FOR K = 1 TO P!
      FOR I = 1 TO PARTS!
        TPA!(I, COUNT, K) = NPA!(I, PLACE, K)
      NEXT I
      FOR C = 1 TO CELLS!
        FOR J = 1 TO MACHINES!
          TMA!(C, J, COUNT, K) = NMA!(C, J, PLACE, K)
        NEXT J
      NEXT C
    NEXT K
  ELSE
    X = RND
    IF X <= 1 / PS1! THEN
      IF MIN$ = "T" THEN
        LOWPLACE = LOWPLACE - 1
      ELSE
        LOWPLACE = LOWPLACE + 1
      ENDIF
      COUNT = COUNT + 1:PLACE = SDATA2(POP, 1):TF(COUNT) = NF(PLACE)
      FOR K = 1 TO P!
        FOR I = 1 TO PARTS!
          TPA!(I, COUNT, K) = NPA!(I, PLACE, K)
        NEXT I
        FOR C = 1 TO CELLS!
          FOR J = 1 TO MACHINES!
            TMA!(C, J, COUNT, K) = NMA!(C, J, PLACE, K)
          NEXT J
        NEXT C
      NEXT K
    END IF
  END IF
NEXT POP
BOTTOM = PS1!
FOR POP = COUNT + 1 TO PS1!
  IF MIN$ = "T" THEN
    PLACET = POP - COUNT
  ELSE
    PLACET = BOTTOM:BOTTOM = BOTTOM - 1
  END IF
  PLACEO = SDATA1(PLACET, 1):TF(POP) = RF(PLACEO)
  FOR K = 1 TO P!
    FOR I = 1 TO PARTS!
      TPA!(I, POP, K) = PA!(I, PLACEO, K)
    NEXT I
    FOR C = 1 TO CELLS!
      FOR J = 1 TO MACHINES!
        TMA!(C, J, POP, K) = MA!(C, J, PLACEO, K)
      NEXT J
    NEXT C
```

```
  NEXT K
NEXT POP
FOR POP = 1 TO PS1!
  RF(POP) = TF(POP)
  FOR K = 1 TO P!
    FOR I = 1 TO PARTS!
      PA!(I, POP, K) = TPA!(I, POP, K)
    NEXT I
    FOR C = 1 TO CELLS!
      FOR J = 1 TO MACHINES!
        MA!(C, J, POP, K) = TMA!(C, J, POP, K)
      NEXT J
    NEXT C
  NEXT K
NEXT POP
BG = GEN
IF MIN$ = "F" THEN
  IF ABS(RF(B) - RF(W)) <= .000001 THEN
    GEN = G1!
  ELSEIF RF(W) > BESTM THEN
    BESTM = RF(W):LASTIMPROVEMENT = GEN
  ELSEIF RF(W) = BESTM THEN
    IF GEN - LASTIMPROVEMENT >= 15 THEN GEN = G1!
  END IF
ELSE
  IF ABS(RF(B) - RF(W)) <= .000001 THEN
    GEN = G1!
  ELSEIF RF(B) < BESTM THEN
    BESTM = RF(B):LASTIMPROVEMENT = GEN
  ELSEIF RF(B) = BESTM THEN
    IF GEN - LASTIMPROVEMENT >= 15 THEN GEN = G1!
  END IF
END IF
END SUB
'**********************************************************
SUB PRINTCOPIES (GEN, FX(), COPIES!(), PS!, CPROB())
OPEN "PARENTS.TXT" FOR APPEND AS 1
PRINT #1, USING "GENERATION ####"; GEN:PRINT #1, ""
SUM = SUMFITNESS(FX(), PS!)
FOR POP = 1 TO PS!
  PRINT #1, USING "### - "; POP;
  EXCOPIES = PS! * FX(POP) / SUM
  PRINT #1, USING "   SCALED FITNESS = ###.#####";FX(POP);
  PRINT #1, USING "   EXPECTED = ###.#####"; EXCOPIES;
  PRINT #1, USING "   CPROB = ###.#####"; CPROB(POP);
  PRINT #1, USING "   ACTUAL    = ###"; COPIES!(POP)
NEXT POP
PRINT #1, "":PRINT #1, ""
CLOSE 1
END SUB
'**********************************************************
```

```
SUB PRINTCROSS (GEN,CROSS$,NMA!(),TMA!(),CP1,CP2,P1!,P2!)
OPEN "CROSS.TXT" FOR APPEND AS 1
PRINT #1, "CROSSOVER OPERATOR = ";:PRINT #1, CROSS$;
IF CROSS$ <> "NONE" THEN
  PRINT #1, USING "   CP1 = ####"; CP1;
  PRINT #1, USING "   CP2 = ####"; CP2
ELSE
  PRINT #1, ""
END IF
PRINT #1, "":PRINT #1, "BEFORE: ";
FOR K = 1 TO P!
  FOR C = 1 TO CELLS!
    FOR J = 1 TO MACHINES!
      PRINT #1, USING " #"; TMA!(C, J, P1!, K);
    NEXT J
  NEXT C
NEXT K
PRINT #1, "":PRINT #1, "        ";
FOR K = 1 TO P!
  FOR C = 1 TO CELLS!
    FOR J = 1 TO MACHINES!
      PRINT #1, USING " #"; TMA!(C, J, P2!, K);
    NEXT J
  NEXT C
NEXT K
PRINT #1, "":PRINT #1, "":PRINT #1, "AFTER : ";
FOR K = 1 TO P!
  FOR C = 1 TO CELLS!
    FOR J = 1 TO MACHINES!
      PRINT #1, USING " #"; NMA!(C, J, P1!, K);
    NEXT J
  NEXT C
NEXT K
PRINT #1, "":PRINT #1, "        ";
FOR K = 1 TO P!
  FOR C = 1 TO CELLS!
    FOR J = 1 TO MACHINES!
      PRINT #1, USING " #"; NMA!(C, J, P2!, K);
    NEXT J
  NEXT C
NEXT K
PRINT #1, "":PRINT #1, ""
CLOSE 1
END SUB
'**********************************************************
SUB PRINTMATES (GEN, MATE!(), PS!)
OPEN "MATES.TXT" FOR APPEND AS 1
PRINT #1, USING "GENERATION ####"; GEN:PRINT #1, ""
FOR POP = 1 TO PS!
  PRINT #1, USING "MATE OF ### = "; POP;
  PRINT #1, USING "###"; MATE!(POP)
```

```
NEXT POP
PRINT #1, ""
CLOSE 1
END SUB
'**********************************************************
SUB PRINTPOOL (GEN, NMA!())
OPEN "MATEPOOL.TXT" FOR APPEND AS 1
PRINT #1, USING "GENERATION ####"; GEN
PRINT #1, ""
FOR POP = 1 TO PS1!
  PRINT #1, USING "MEMBER ### : "; POP;
  FOR K = 1 TO P!
    FOR C = 1 TO CELLS!
      FOR J = 1 TO MACHINES!
        PRINT #1, USING " #"; NMA!(C, J, POP, K);
      NEXT J
    NEXT C
  NEXT K
  PRINT #1, ""
NEXT POP
PRINT #1, "":PRINT #1, ""
CLOSE 1
END SUB
'**********************************************************
SUB QSORT (SDATA(), START%, FINISH%)
LOW% = START%:HIGH% = FINISH%:MIDDLE% = (LOW% + HIGH%) / 2
CHECK = SDATA(MIDDLE%, 2)
DO
  DO WHILE SDATA(LOW%, 2) < CHECK
    LOW% = LOW% + 1
  LOOP
  DO WHILE SDATA(HIGH%, 2) > CHECK
    HIGH% = HIGH% - 1
  LOOP
  IF LOW% <= HIGH% THEN
    SWAP SDATA(LOW%, 1), SDATA(HIGH%, 1)
    SWAP SDATA(LOW%, 2), SDATA(HIGH%, 2)
    LOW% = LOW% + 1:HIGH% = HIGH% - 1
  END IF
LOOP UNTIL LOW% > HIGH%
IF START% < HIGH% THEN
  CALL QSORT(SDATA(), START%, HIGH%)
END IF
IF LOW% < FINISH% THEN
  CALL QSORT(SDATA(), LOW%, FINISH%)
END IF
END SUB
'**********************************************************
SUB REPROMACHINES (GEN, FS(), MA!(), NMA!())
IF TOPRINT = 1 THEN
  OPEN "CROSS.TXT" FOR APPEND AS #1
```

```
  PRINT #1, "":PRINT #1, "":PRINT #1, USING "GENERATION ####"; GEN
  PRINT #1, ""
  CLOSE 1
  OPEN "MUTATE.TXT" FOR APPEND AS #1
  PRINT #1, "":PRINT #1, "":PRINT #1, USING "GENERATION ####"; GEN
  PRINT #1, ""
  CLOSE 1
END IF
REDIM MATE!(PS1!),COPIES!(PS1!),TMA!(CELLS!, MACHINES!, PS1!, P!)
CALL SELECTPARENTS(GEN, FS(), PS1!, COPIES!())
CALL SELECTMATES(GEN, MATE!(), PS1!)
X = 0
FOR POP = 1 TO PS1!
  FOR T = 1 TO COPIES!(POP)
    X = X + 1
    FOR K = 1 TO P!
      FOR J = 1 TO MACHINES!
        FOR CELL = 1 TO CELLS!
          NMA!(CELL, J, X, K) = MA!(CELL, J, POP, K)
          TMA!(CELL, J, X, K) = MA!(CELL, J, POP, K)
        NEXT CELL
      NEXT J
    NEXT K
  NEXT T
NEXT POP
IF TOPRINT = 1 THEN CALL PRINTPOOL(GEN, NMA!())
IF P! > 1 THEN
  PR1 = .25:PR2 = .5:PR3 = .75
ELSE
  PR1 = 1 / 3:PR2 = 2 / 3:PR3 = 1
END IF
FOR POP = 1 TO PS1!
  IF MATE!(POP) > POP THEN
    X = RND
    IF X <= CR1 THEN
      X = RND
      IF X <= PR1 THEN
        CP1 = GETINTEGER!(CHLENGTH1! - 1):CP2 = CHLENGTH1! * 2
        CROSS$ = "SINGLE"
      ELSEIF X <= PR2 THEN
        P1!=GETINTEGER!(CHLENGTH1! - 1):P2!=GETINTEGER!(CHLENGTH1! - 1)
        IF P1! > P2! THEN CP2 = P1!:CP1 = P2! ELSE CP1 = P1!:CP2 = P2!
        CROSS$ = "DOUBLE"
      ELSEIF X <= PR3 THEN
        SPERIOD = GETINTEGER!(P!):SCELL = GETINTEGER!(CELLS!)
        CP1 = (SPERIOD-1)*CELLS!:CP1=CPR1*MACHINES!+(SCELL-1)*MACHINES!
        CP2 = CP1 + MACHINES!:CROSS$ = "CELLSWAP"
      ELSE
        SPERIOD = GETINTEGER!(P!):CP1=(SPERIOD-1) * CELLS! * MACHINES!
        CP2 = CP1 + MACHINES! * CELLS!:CROSS$ = "PERIODSWAP"
      END IF
```

```
    ELSE
      CP1 = CHLENGTH1!:CP2 = CHLENGTH1! * 2:CROSS$ = "NONE"
    END IF
    CALL CROSSMACHINES(CROSS$, GEN, POP, MATE!(POP), CP1,
                             CP2, TMA!(), NMA!())
  END IF
NEXT POP
IF TOPRINT = 1 THEN OPEN "MUTATE.TXT" FOR APPEND AS 1
FOR POP = 1 TO PS1!
  IF TOPRINT = 1 THEN
    PRINT #1, USING "###  "; POP;:PRINT #1, "BEFORE: ";
    FOR K = 1 TO P!
      FOR C = 1 TO CELLS!
        FOR J = 1 TO MACHINES!
          PRINT #1, USING " #"; NMA!(C, J, POP, K);
        NEXT J
      NEXT C
    NEXT K
    PRINT #1, ""
  END IF
  THISPOP = 0:PLACE$ = ""
  FOR K = 1 TO P!
    FOR CELL = 1 TO CELLS!
      FOR J = 1 TO MACHINES!
        X = RND:PLACE = (K-1)*CELLS!
        PLACE = PLACE*MACHINES!+(CELL - 1)*MACHINES!+J
        IF X <= MR1 THEN
          MUTATED! = MUTATED! + 1:THISPOP = THISPOP + 1
          PLACE$ = PLACE$ + STR$(PLACE)
          IF NMA!(CELL, J, POP, K) > 0 THEN
            NMA!(CELL, J, POP, K) = 0
          ELSE
            NMA!(CELL, J, POP, K) = 1
          ENDIF
        END IF
      NEXT J
    NEXT CELL
  NEXT K
  IF TOPRINT = 1 THEN
    PRINT #1, "     AFTER : ";
    FOR K = 1 TO P!
      FOR C = 1 TO CELLS!
        FOR J = 1 TO MACHINES!
          PRINT #1, USING " #"; NMA!(C, J, POP, K);
        NEXT J
      NEXT C
    NEXT K
    PRINT #1, "":PRINT #1, USING "     MUTATIONS = ###"; THISPOP;
    IF THISPOP > 0 THEN
      PRINT #1, "    PLACE:" + PLACE$
    ELSE
```

```
      PRINT #1, ""
    ENDIF
    PRINT #1, ""
  END IF
NEXT POP
IF TOPRINT = 1 THEN CLOSE 1
ERASE TMA
END SUB
'**********************************************************
SUB SELECTMATES (GEN, MATE!(), PS!)
DONEMATING$ = "F"
DO WHILE DONEMATING$ = "F"
  MATE = INT(RND * PS!) + 1
  FOR POP = 1 TO PS!
    IF MATE!(POP)=0 AND MATE<>POP AND MATE!(MATE) = 0 THEN
      MATE!(POP) = MATE:MATE!(MATE) = POP:POP = PS!
    ENDIF
  NEXT POP
  DONEMATING$ = "T"
  FOR POP = 1 TO PS!
    IF MATE!(POP) = 0 THEN
      DONEMATING$ = "F"
    ENDIF
    POP = PS!
  NEXT POP
LOOP
IF TOPRINT = 1 THEN CALL PRINTMATES(GEN, MATE!(), PS!)
END SUB
'**********************************************************
SUB SELECTPARENTS (GEN, FX(), PS!, COPIES!())
'STOCHASTIC REMAINDER WITHOUT REPLACEMENT SELECTION STRATEGY
REDIM CPROB(PS!):SUM = SUMFITNESS(FX(), PS!):TOT = 0
FOR POP = 1 TO PS!
  EXCOPIES = PS! * FX(POP) / SUM: COPIES!(POP) = INT(EXCOPIES)
  CPROB(POP) = EXCOPIES - COPIES!(POP):TOT = TOT + COPIES!(POP)
NEXT POP
SUM = SUMFITNESS(CPROB(), PS!)
FOR POP = 1 TO PS!
  CPROB(POP) = CPROB(POP) / SUM
NEXT POP
FOR POP = 2 TO PS!
  CPROB(POP) = CPROB(POP) + CPROB(POP - 1)
NEXT POP
DO WHILE TOT < PS!
  S = RND
  FOR PARENT = 1 TO PS!
    IF S < CPROB(PARENT) THEN
      COPIES!(PARENT) = COPIES!(PARENT) + 1
      IF PARENT < PS! THEN
        DELTA = CPROB(PARENT + 1) - CPROB(PARENT)
        FOR T = PARENT + 1 TO PS!
```

```
          CPROB(T) = CPROB(T) - DELTA
        NEXT T
      END IF
      CPROB(PARENT) = 0:PARENT = PS!:TOT = TOT + 1
    END IF
  NEXT PARENT
LOOP
IF TOPRINT = 1 THEN CALL PRINTCOPIES(GEN,FX(),COPIES!(),PS!,CPROB())
END SUB
'**********************************************************
FUNCTION SUMFITNESS (FX(), PS!)
SUM = 0
FOR POP = 1 TO PS!
  SUM = SUM + FX(POP)
NEXT POP
SUMFITNESS = SUM
END FUNCTION
'**********************************************************
FUNCTION WORSTFIT! (FX(), PS!)
MAX = FX(1):WORST = 1
FOR POP = 2 TO PS!
  IF FX(POP) > MAX THEN MAX = FX(POP) WORST = POP
NEXT POP
WORSTFIT! = WORST
END FUNCTION
'**********************************************************
REM $INCLUDE: 'DATAFILE\TEST_1.DAT'
'**********************************************************
FUNCTION CELLASSIGN!
X = RND
FOR CELL = 1 TO CELLS!
  IF X <= CELL / CELLS! THEN
    CELLASSIGN! = CELL
    EXIT FUNCTION
  END IF
NEXT CELL
END FUNCTION
'**********************************************************
SUB INITIALIZEMACHINES (MA!())
'RANDOMLY GENERATE PS1! MACHINE ASSIGNMENT SOLUTIONS
FOR POP = 1 TO PS1!
  FOR K = 1 TO P!
    FOR J = 1 TO MACHINES!
      FOR N = 1 TO MT(J)
        CELL = CELLASSIGN!:MA!(CELL, J, POP, K) = 1
      NEXT N
    NEXT J
  NEXT K
NEXT POP
END SUB
'**********************************************************
```

```
SUB READINPUTDATA (FBEST$)
OPEN FBEST$ FOR APPEND AS 1
PRINT #1, FBEST$, DATE$
'READ MACHINE-PART INCIDENCE MATRIX
PRINT #1, "":PRINT #1, "MACHINE-PART INCIDENCE MATRIX":PRINT #1, ""
FOR I = 1 TO PARTS!
  PRINT #1, USING "PART ##:"; I;
  FOR J = 1 TO MACHINES!
    READ MP(I, J):PRINT #1, USING "  ##.#"; MP(I, J);
  NEXT J
  PRINT #1, ""
NEXT I
'READ PART DEMAND
PRINT #1, "":PRINT #1, "PART DEMAND":PRINT #1, ""
FOR K = 1 TO PERIODS!
  FOR I = 1 TO PARTS!
    READ PD(I, K)
  NEXT I
NEXT K
FOR I = 1 TO PARTS!
  PRINT #1, USING "PART ##:"; I;
  FOR K = 1 TO PERIODS!
    PRINT #1, USING "  ##,###"; PD(I, K);
  NEXT K
  PRINT #1, ""
NEXT I
'READ MACHINE CAPACITY
PRINT #1, "":PRINT #1, "MACHINE CAPACITIES:";
FOR J = 1 TO MACHINES!
  READ MC(J):PRINT #1, USING "  ##,###"; MC(J);
NEXT J
PRINT #1, ""
'READ NUMBER OF EACH MACHINE TYPE AVAILABLE
PRINT #1, "":PRINT #1, "AVAILABLE MACHINES:";
FOR J = 1 TO MACHINES!
  READ MT(J):PRINT #1, USING "  ##"; MT(J);
NEXT J
PRINT #1, ""
'READ PART OPERATION SEQUENCE
PRINT #1, "":PRINT #1, "PART OPERATION SEQUENCES":PRINT #1, ""
FOR I = 1 TO PARTS!
  PRINT #1, USING "PART ##:  "; I;:READ P$(I):PRINT #1, P$(I)
NEXT I
'READ MACHINE ACQUISITION COST
PRINT #1, "":PRINT #1, "ACQUISITION COSTS":PRINT #1, ""
FOR J = 1 TO MACHINES!
  PRINT #1, USING "MACHINE ##:"; J;
  READ MI(J):PRINT #1, USING "  ##,###"; MI(J)
NEXT J
'READ MACHINE RELOCATION COST
IF P! > 1 THEN
```

```
  FOR K = 1 TO P!
    FOR J = 1 TO MACHINES!
      READ RC(J, K)
    NEXT J
  NEXT K
  PRINT #1, "":PRINT #1, "MACHINE RELOCATION COST":PRINT #1, ""
  FOR J = 1 TO MACHINES!
    PRINT #1, USING "MACHINE ##:";J;:PRINT #1,USING "   ##,###";RC(J, 1)
  NEXT J
END IF
'COMPUTE MINIMUM NUMBER OF MACHINES REQUIRED
FOR K = 1 TO P!
  FOR J = 1 TO MACHINES!
    LOAD = 0
    FOR I = 1 TO PARTS!
      LOAD = LOAD + MP(I, J) * PD(I, K) * (1 + CAPBUFFER)
    NEXT I
    RM = LOAD / MC(J)
    IF RM - INT(RM) = 0 THEN
      MR(J, K) = RM
    ELSE
      MR(J, K) = INT(RM) + 1
    ENDIF
  NEXT J
NEXT K
PRINT #1, "":PRINT #1, "MINIMUM NUMBER OF MACHINES":PRINT #1, ""
FOR J = 1 TO MACHINES!
  PRINT #1, USING "MACHINE ##:"; J;
  FOR K = 1 TO P!
    PRINT #1, USING "  ##"; MR(J, K);
  NEXT K
  PRINT #1, ""
NEXT J
PRINT #1, "":PRINT #1, "":PRINT #1, ""
CLOSE 1
END SUB
'**********************************************************
FUNCTION AVGFITNESS (TX(), PS!)
SUM = 0
FOR POP = 1 TO PS!
  SUM = SUM + TX(POP)
NEXT POP
AVGFITNESS = SUM / PS!
END FUNCTION
'**********************************************************
FUNCTION MAXFITNESS (TX(), PS!)
MAX = -1
FOR POP = 1 TO PS!
  IF TX(POP) > MAX THEN MAX = TX(POP)
NEXT POP
MAXFITNESS = MAX
```

```
END FUNCTION
'************************************************************
FUNCTION MINFITNESS (TX(), PS!)

MIN = TX(1)
FOR POP = 2 TO PS!
  IF TX(POP) < MIN THEN MIN = TX(POP)
NEXT POP
MINFITNESS = MIN
END FUNCTION
'************************************************************
SUB PRINTCOSTS (FILE$, GEN, MOVES(), INVEST(), RL())
OPEN FILE$ FOR APPEND AS 1
PRINT #1, USING "GENERATION ####"; GEN:PRINT #1, ""
FOR POP = 1 TO PS1!
  PRINT #1, USING "### - "; POP;:SUM = 0
  FOR K = 1 TO P!
    PRINT #1, USING "  M:###,###"; MOVES(POP, K);
    PRINT #1, USING "  I:###,###,###"; INVEST(POP, K);
    IF K > 1 THEN
      PRINT #1, USING "  R:###,###"; RL(POP, K);
    ENDIF
    SUM = SUM + MOVES(POP,K) + INVEST(POP,K) + RL(POP,K)
  NEXT K
  PRINT #1, USING "  T:###,###,###,###"; SUM
NEXT POP
PRINT #1, ""
CLOSE 1
END SUB
'************************************************************
SUB PRINTMPOP (GENFILE$, GEN, MX!(), FX(), FRX(), FSX())
OPEN GENFILE$ FOR APPEND AS 1
PRINT #1, USING "GENERATION ####"; GEN
PRINT #1, "":PRINT #1, "MACHINE ASSIGNMENTS":PRINT #1, ""
FOR POP = 1 TO PS1!
  PRINT #1, USING "MEMBER ### : "; POP;
  FOR K = 1 TO P!
    FOR C = 1 TO CELLS!
      PRINT #1, "[";
      FOR J = 1 TO MACHINES!
        PRINT #1, USING " #"; MX!(C, J, POP, K);
      NEXT J
      PRINT #1, " ]";
    NEXT C
    PRINT #1, " | ";
  NEXT K
  PRINT #1, USING "  RAW ###,###,###"; FX(POP);
  PRINT #1, USING "  REV #.#####"; FRX(POP);
  PRINT #1, USING "  SCL ##.#####"; FSX(POP)
NEXT POP
PRINT #1, ""
```

```
MINRAW = MINFITNESS(FX(), PS1!):MAXRAW = MAXFITNESS(FX(), PS1!)
AVGRAW = AVGFITNESS(FX(), PS1!):MINREV = MINFITNESS(FRX(), PS1!)
MAXREV = MAXFITNESS(FRX(), PS1!):AVGREV = AVGFITNESS(FRX(), PS1!)
MINSCL = MINFITNESS(FSX(), PS1!):MAXSCL = MAXFITNESS(FSX(), PS1!)
AVGSCL = AVGFITNESS(FSX(), PS1!)
PRINT #1, USING "MINRAW = ###,###,###"; MINRAW;
PRINT #1, USING "        MINREV = #.#####"; MINREV;
PRINT #1, USING "        MINSCL = #.#####"; MINSCL
PRINT #1, USING "MAXRAW = ###,###,###"; MAXRAW;
PRINT #1, USING "        MAXREV = #.#####"; MAXREV;
PRINT #1, USING "        MAXSCL = #.#####"; MAXSCL
PRINT #1, USING "AVGRAW = ###,###,###"; AVGRAW;
PRINT #1, USING "        AVGREV = #.#####"; AVGREV;
PRINT #1, USING "        AVGSCL = #.#####"; AVGSCL
PRINT #1, "":PRINT #1, ""
CLOSE 1
END SUB
'***********************************************************
SUB PRINTPARTPOP (GEN, PX!())
OPEN "PARTPOP.TXT" FOR APPEND AS 1
PRINT #1, USING "GENERATION ####"; GEN:PRINT #1, ""
FOR POP = 1 TO PS1!
  PRINT #1, USING "### : "; POP;
  FOR K = 1 TO P!
    FOR I = 1 TO PARTS
      PRINT #1, USING "## "; PX!(I, POP, K);
    NEXT I
    PRINT #1, "| ";
  NEXT K
  PRINT #1, ""
NEXT POP
PRINT #1, ""
CLOSE 1
END SUB
'***********************************************************
SUB RESULTS (GX,MOVES(),INVEST(),XT(),PX!(),MX!(),FILENAME$,PS!,RL(),K,
             PURCHASED!(),MAO!(), EFF(), UTIL())
OPEN FILENAME$ FOR APPEND AS 1
PRINT #1, "GENERATION"; GX
FOR POP = 1 TO PS!
  PRINT #1, "":  PRINT #1, "SOLUTION"; POP
  IF P! > 1 THEN PRINT #1, "":    PRINT #1, "PERIOD"; K
  PRINT #1, "":  PRINT #1, "PART ASSIGNMENT":  PRINT #1, ""
  FOR CELL = 1 TO CELLS!
    PRINT #1, USING "CELL ##:"; CELL;
    FOR I = 1 TO PARTS!
      IF PX!(I, POP, K) = CELL THEN PRINT #1, USING " ##"; I;
    NEXT I
    PRINT #1, ""
  NEXT CELL
  PRINT #1, "":PRINT #1, "MACHINE ASSIGNMENT":PRINT #1, ""
```

```
  FOR CELL = 1 TO CELLS!
    PRINT #1, USING "CELL ##:"; CELL;
    FOR J = 1 TO MACHINES!
      FOR M = 1 TO MX!(CELL, J, POP, K)
        PRINT #1, USING " ##"; J;
      NEXT M
    NEXT J
    PRINT #1, ""
  NEXT CELL
  PRINT #1, "":PRINT #1, "PERIOD FITNESS":PRINT #1, ""
  PRINT #1, USING "MOVES        ###,###,###"; MOVES(POP, K)
  PRINT #1, USING "INVESTMENT   ###,###,###"; INVEST(POP,K)
  PRINT #1, USING "RELOCATION   ###,###,###"; RL(POP, K)
  PRINT #1, USING "EFFICACY         #.#####"; EFF(POP, K)
  PRINT #1, USING "CELL UTILIZATION  #.#####"; UTIL(POP,K)
  PRINT #1, USING "TOTAL    ###,###,###,###"; XT(POP)
NEXT POP
PRINT #1, "":PRINT #1, "SUMMARY FITNESS":PRINT #1, ""
PRINT #1, "           MOVES";:PRINT #1, "      INVESTMENT";
PRINT #1, "      RELOCATION";:PRINT #1, "            TOTAL"
FOR POP = 1 TO PS!
  PRINT #1, USING "### "; POP;
  PRINT #1, USING "###,###,###,### "; MOVES(POP, K);
  PRINT #1, USING "###,###,###,### "; INVEST(POP, K);
  PRINT #1, USING "###,###,###,### "; RL(POP, K);
  PRINT #1, USING "###,###,###,###,###"; XT(POP)
NEXT POP
CLOSE 1
END SUB
'**********************************************************
SUB STATISTICS (GENERATION,X(),RX(), SX(), PS!, FILENAME$)
OPEN FILENAME$ FOR APPEND AS 1
PRINT #1, "GENERATION"; GENERATION
MAXRAW = MAXFITNESS(X(), PS!):MAXREV = MAXFITNESS(RX(), PS!)
MAXSCL = MAXFITNESS(SX(), PS!):MINRAW = MINFITNESS(X(), PS!)
MINREV = MINFITNESS(RX(), PS!):MINSCL = MINFITNESS(SX(), PS!)
AVGRAW = AVGFITNESS(X(), PS!):AVGREV = AVGFITNESS(RX(), PS!)
AVGSCL = AVGFITNESS(SX(), PS!)
PRINT #1, USING "MAXRAW ###,###,###"; MAXRAW;
PRINT #1, USING "   MAXREV   #.####"; MAXREV;
PRINT #1, USING "   MAXSCL   ##.####"; MAXSCL
PRINT #1, USING "MINRAW ###,###,###"; MINRAW;
PRINT #1, USING "   MINREV   #.####"; MINREV;
PRINT #1, USING "   MINSCL   ##.####"; MINSCL
PRINT #1, USING "AVGRAW ###,###,###"; AVGRAW;
PRINT #1, USING "   AVGREV   #.####"; AVGREV;
PRINT #1, USING "   AVGSCL   ##.####"; AVGSCL
PRINT #1, "NUMBER MUTATED"; MUTATED!:PRINT #1, ""
CLOSE 1
END SUB
```

Vita

Elin Laurene MacStravic Wicks was born on February 23, 1966. She grew up in Willingboro, New Jersey and graduated from Holy Cross High School (Delran, New Jersey) in 1984. Elin received a Bachelor of Science degree in Industrial Engineering from Rutgers, The State University of New Jersey, in May 1989. Immediately following graduation, Elin entered the graduate program at Rutgers University. She received a Master of Science in Industrial Engineering in August, 1991. Her thesis, A Multicriteria Decision Model for the Economic Justification of Advanced Manufacturing Technology, was sponsored by the Defense Logistics Agency. Elin entered the Manufacturing Systems option of the Ph.D. program in Industrial and Systems Engineering at Virginia Polytechnic Institute and State University in August, 1991 and received her Ph.D. in June, 1995.

Elin is a member of Alpha Pi Mu and the Institute of Industrial Engineers. She has been selected as an Outstanding Young Woman of America. Elin has accepted an Assistant Professorship position at the University of Missouri - Columbia for Fall 1995. She is married to Matthew L. Wicks.

Elin M. Wicks