MODELING AND OPTIMIZATION OF CELL FORMATION PROBLEM USING PROGRESSIVE MODELING AND FACTDESIGN

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Science

in

Industrial Systems Engineering

University of Regina

By

Taras Dmytryshyn

Regina, Saskatchewan

December, 2017

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Taras Dmytryshyn, candidate for the degree of Master of Applied Science in Industrial Systems Engineering**,** has presented a thesis titled, ***Modeling and Optimization of Cell Formation Problem Using Progressive Modeling and FactDesign,*** in an oral examination held on September 5, 2017. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr. Mohamed El-Darieby, Software Systems Engineering
Supervisor:	Dr. Mohamed Ismail, Industrial Systems Engineering
Committee Member:	Dr. Amr Henni, Industrial Systems Engineering
Committee Member:	*Dr. Hussameldin Ibrahim, Industrial Systems Engineering
Chair of Defense:	Dr. Shelagh Campbell, Faculty of Business Administration

*Not present at defense

Abstract

Cellular manufacturing is the most efficient manufacturing system design option in the mid-volume/mid-varietymanufacturing environments.Designing a cellular manufacturing system is a complex multi-level process consisting of four interdependent stages: 1) cell formation (CF), 2) inter and intra-cell layout design, 3) work scheduling and 4) resource allocation. In this thesis, our focus will be on cell formation stage, widely known in the literature as the Cell Formation Problem (CFP). In this study, a new comprehensive, user-friendly framework is proposed and developed to address a variety of problem formulations. A CFP could be simply modeled as a simple incidence matrix or as complex as data tables that can capture all the relevant data that may arise in a real-world system design setup. Several similarity indices have also been implemented to the data input/output logic to assist in solving both small- and large-scaleversions of the problem. The selected indices have been validated using several problems form the literature. A novel progressive algorithm is proposed to solve the simple case of the CFP using the novel representations of factory graphs. The efficiency and efficacy of the proposed algorithm have been demonstrated using several CFP problems in the literature.

Dedication

To a great spirit, talented professional, bright mind and, foremost, mymentor and friend, Professor Aristid Vasylyk.

Acknowledgements

I would like to express my sincere thanks to my supervisor Dr. Mohamed Ismail for devoting considerable time and effort to bring this thesis to completion. I greatly appreciate his professional guidance, encouragement and inspiration.

I am also thankful to the external examiner Dr. Mohamed El-Darieby and the committee members Dr. Amr Henni and Dr. Hussameldin Ibrahim for their valuable help and suggestions.

Finally, I would like to thank tomy family and friends for not only the words of encouragement but also for their immediate and unconditioned help, should I have needed one.

Table of contents

List of tables

List of figures

Nomenclatures

ACO	Ant Colony Optimization
BB	Branch-and-Bound
CCFP	Capacitated Cell Formation Problem
CF	Cell Formation
CFP	Cell Formation Problem
CIM	Computer Integrated Manufacturing
CLINK	Complete-linkage clustering
CM	Cellular Manufacturing
CMS	Cellular Manufacturing System
DCMS	Dynamic Cellular Manufacturing System
DEFC	Differential Evolutionary Fuzzy Clustering
EA	Evolutionary Algorithm
EnGGA	Enhanced Grouping Genetic Algorithm
FMS	Flexible Manufacturing System
GA	Genetic Algorithm
GCFP	Generalized Cell Formation Problem
GE	Grouping Efficacy
GGA	Grouping Genetic Algorithm
GP-SLCA	Genetic Programming Single Linkage Cluster Analysis
GT	Group Technology
GUI	Graphical User Interface
HGBPSO	Hybrid Group-Based Particle Swarm Optimization
HGGA	Hybrid Grouping Genetic Algorithm
HSAM	Hybrid Simulated Annealing with Mutation
JIT	Just In Time

MA	Memetic Algorithm
MOSS	Multi-Objective Scatter Search
MOTS	Multi-Objective Tabu Search
MPIM	Machine-Part Incidence Matrix
MS	Manufacturing System
NPGA/NSGA-II	Niched Pareto Genetic Algorithm/II
NSGA	Non-dominated Sorting Genetic Algorithm
PF	Part Family
PM	Progressive Modeling
PSO	Particle Swarm Optimization
RPP	Resource Planning Problem
SA	Simulated Annealing
SACF	Simulated Annealing Cell Formation
SC	Similarity Coefficient
SLINK	Single-linkage clustering
SS	Scatter Search
TS	Tabu Search
TSCF	TabuSarch Cell Formation
TQM	Total Quality Management
VNS	Variable Neighborhood Search
WIP	Work-In-Progress
XGA	Variant of Multi-Objective Genetic Algorithm
VEGA	Vector Evaluated Genetic Algorithm

Chapter 1 : Introduction

1.1 Introduction

In order to keep up with competition on the market, manufacturing companies have always sought for ways to increase performance and efficiency of their Manufacturing Systems (MS). Nowadays, this is especially true for the companies involved in the production of medium volume, medium variety products. Conventional approaches do not suit the requirements, necessary for efficient production of these products. One solution to this problem is CM – an approach that brings the efficiency of a flow line into batch production. CM has been of researchers' and practitioners' interest for several decades now. Some significant progress has been made in this area within the last two decades. At the same time, the complexity of the problem and multitude of ways in which it can be approached still call for research and development of new and foremost practical techniques. A comprehensive generalized framework for the CFP and a novel solution algorithm for a simple binary problem with the objective of minimizing inter-cell moves is presented for the first time in this work.

This chapter provides an overview of Cellular Manufacturing (CM) concept and Cell Formation Problem (CFP) thereby laying the ground for subsequent chapters. First, the prerequisites for emergence of CM are outlined. Then, the advantages and the drawbacks of CM are consideredand and the main aspects of Cell Formation (CF) are discussed. Finally, the research objectives and the thesis outline conclude this chapter.

1.2 Cellular manufacturing

There are two conventional approaches in manufacturing: job shop and flow line. They both represent opposite sides of ways in which manufacturing can be organized. Job shop is a manufacturing system where machines performing similar operations are grouped into departments. Manufacturing layout obtained this way is called functional layout. In a low volume and high variety production job shop outperforms other MSs. Indeed, having specialized departments allows a great degree of flexibility. Flexibility is the ability of MS to accommodate to changes in product type and mix demands. In other words, it is the availability of machines for part processing [1]. Flexibility is a key feature of functional layout. In case of machine breakdown, parts can be rerouted to an alternative machine. These advantages come at the cost of having extensive material handling due to large travel distances between departments, high throughput times, large quantities of Work-In-Progress (WIP), long setup times and, as a result, a need to manufacture products in batches. According to [2] non-productive activities in job shop system take up to 95% of production time. Due to high flexibility, functional layout creates large obstacles for automation. Besides, job shop system has significant management problems–namely difficulties in scheduling and controlling the system. Due to the mentioned inefficiencies, such system suffers long lead times and high production cost which are crucial in today's competitive market.

Flow line, on the other hand, is effective for high volume and low variety production. It is a product-oriented system designed for a specific process route. Those systems are very efficient and easy to manage. However, they are extremely vulnerable to machine

breakdowns, suffer from the lack of flexibility, and require high initial investment cost. Almost any change to the system requires substantial time, labor and financial resources.

Both of these MSs lack efficiency when it comes to medium volume and medium variety production. Ideal MS would incorporate only advantages specific to both job shop and flow line with flexibility and efficiency being key ones. However, this is impossible since these factors are mutually exclusive. MS cannot be as flexible as a job shop and at the same time as efficient as a flow line. Nonetheless, approaching such goal would mean better performance in medium volume and medium variety production. Whenever there is a large product mix calling for the same combination of resources, CM can be successfully implemented. Groover [3] points out that it is quite common for a factory that is producing 10,000 parts to be able to group most of them into 20-30 part families. Kusiak [4] defined CM as a concept that bridges the gap between job shop and low line systems and aims to reduce the complexity of system's management. The interrelation between CM and conventional approaches is illustratedin Figure 1-1.

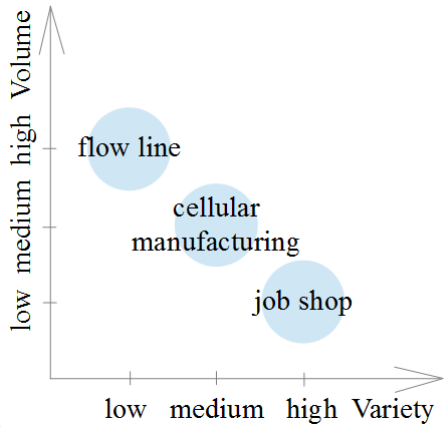


Figure 1-1: Interrelation between CM and conventional approaches

CM stems from a concept of group technology (GT) – a manufacturing philosophy that seeks to exploit similarities in products, their designs, and manufacturing processes. According to GT similar things should be done similarly. This is the main idea around which CM is built. Contrary to the information on the origins of GT found in many sources, the inception of GT dates back to 1925 when Flanders R. E. (listed in [5]) suggested organizingproduct-oriented departments in order to minimize transportation efforts during the manufacturing process. Mitrofanov's book “Scientific principles of group technology” was published in English in 1966 [6]. He generalized GT principles and specified design simplification and process standardization as a must for GT to be effective. Burbidge [7] adopted GT principles and developed a concept of Cellular Manufacturing System (CMS) – a system where parts requiring similar resources are grouped into Part Families (PF) and resources, in turn, are grouped into Manufacturing Cells (MC).

1.3 Advantages specific to CM

CM holds the following advantages over functional layout:

1. Simplified process planning and management. Planning and scheduling are complex when dealing with alarge number of parts and machines. MCs can be considered as quasi-independent units, and management problems can be addressed at the level of cells rather than individual machines. Reduced complexity and optimal location of machines, greatly simplify material flow. In turn, this makes manufacturing more transparent and easier to control.

2. Improved quality. It comes as a result of the elimination of large batches and employing highly qualified workers. Small batches ensure early the detection of defects and easy tracking of the problem. Workers trained to operate all the machines in a cell can provide a quick and accurate feedback and with their help solution to a problem can be found faster.
3. Faster response to the market. It is a direct consequence of reduced throughput (production) times. Small throughput time, in turn, comes from four interrelated factors:
   i. Reduced setup time. Since parts belonging to one family have very similar processing requirements, similar tools and fixtures are used and thus setup time is drastically reduced. Askin and Standridge [2] showed that cutting setup time in half reduces WIP by 50%.
   ii. Reduced material handling distance. It is one of the key features of CMS. Studies show that material handling makes up to 30-75% of the product cost [8]. Improved material handling is a result of efficient allocation of machines. It saves a lot of transportation-associated costs and decreases throughput time.
   iii. Reduced batch size. Short setup times and small traveling distances of parts allow producing parts in small batches. Small batches, in turn, significantly decrease the waiting time.
   iv. Less inventory. Close allocation of machines within cell and small batches result in smaller number of unfinished products in the shop. Besides, less

inventory means reduced WIP which essentially is smaller production-associated costs.

4. Reduced tooling requirements and costs. Tools that are being used in CM are specific to each cell and in most cases are universal for the whole part family associated with a particular cell, which means less expensive and handier tools and fixtures.
5. Less space is required.Since machines in a cell are brought closely together, less space is needed to install all the equipment.
6. Better work environment. Workers trained to operate the whole cell are more qualified and skillful. They can visualize their efforts in the form of a final product and make a valuable contribution to solving issues, should any arise. Working together in a cell is conducive to teamwork since cells represent a kind of sociological units. In a chapter titled "Personnel and group technology" Burbidge [7] explains how GT can improve job satisfaction and labor relations.

   Wemmerlov and Hyer [9] conducted a survey study among 32 U.S. manufacturing companies who were involved in CM. Average improvements after implementation of CM were reported as follows: 45.6% reduction in throughput time; 41.4% reduction in WIP inventory; 39.3% reduction in materials handling; 33.1% reduction in number of fixtures; 32.0% reduction in setup time; 29.6% improvement of part quality; 29.2% reduction in finished goods inventory. Similar figures were reported by Wemmerlov and Johnson [10] in a later survey on implementation experiences and performance improvements at 46 user plants.

It is important to note that the potential benefits of CM are valid only in case of low inter-cell movements.

1.4 Disadvantages specific to CM

1 Lower machine utilization and uneven machine load (in comparison with flow line). Sassani [11] and Kannanand and Ghosh [12] noted that in an attempt to create independent cells a problem of uneven workload and associated with ita problem of low machine utilization arises.

2 Lack of flexibility to product mix changes. Fair number of researchers agrees that CMS feature limited flexibility. This drawback, as well as machine load imbalance was considered by Djassemi [1]. While his simulation proves the point, he suggests employing flexible cross-trained workers to increase flexibility of CMS and reduce uneven machine load.

3 Problems related to machine breakdowns (in comparison with functional layout). Since MCs are designed to function as a whole unit, machine breakdown makes the entire cell nonfunctional. This problem can be solved through rerouting of parts to other cells. But because of the reduced flexibility, alternative machines are not always readily available.

4 Difficulty of managing out-of-cell operations. In real-world cases, it is almost impossible to create completely independent cells. Out of cell operations mean not only excessive material handling, but also interruption of manufacturing processes in a host cell. If the machine processing foreign part is in high demand, it will become a bottleneck machine and will impair the capacity of the cell.

5 High implementation cost. Changing existing machine layout is costly in terms of relocation of machines and long periods of machine downtime. On the top of that, the more emphasis was made on productivity during the design of CMS, the more duplicate machines have to be purchased. Companies involved in CM reported equipment relocation, cell design and purchase of new equipment as three major expenses categories associated with implementation of CM [9].

6 Absence of comprehensive yet simple tool for designing CMS. Presently there is no tool that could cheaply and rapidly identify machine groups and part families. Usually these tools must be developed individually for a specific case since there is a large variety of factors and each case has its own priorities and objectives.

1.5 Cell Formation Problem

Designing CMS is a complex multi-level process consisting of four problems [13]: (a) CF, (b) inter and intra-cell layout, (c) work scheduling and (d) resource allocation. First two are design problems. The focus of this work is on CFP. This is an important step in CMS design since it provides the basis for subsequent stages.

CFP is a combinatorial optimization problem that consists of identifying optimal sets of machines and part clusters. Considering the problem of grouping n parts and m machines into p clusters, the number of possible solutions is given by equation 1.1 [14]. CFP belongs to the class of combinatorial problems for which there is no polynomial time algorithm [15] and the most common way to solve them is by using heuristic-based algorithms [16].

$$\left(\frac{\sum_{i=1}^{p}(-1)^{p-i}\cdot i^{n}}{i!\cdot(p-i)!}\right)\cdot\left(\frac{\sum_{i=1}^{p}(-1)^{p-i}\cdot i^{m}}{i!\cdot(p-i)!}\right) \qquad (1.1)$$

In general, CFP can be solved using design-based or production-based methods. First category uses design features of parts to form part families, while the second one groups them into part families using their processing data. The vast majority of the existing approaches are production-based ones. CFP can be approached in 3 ways(a) first part families are formed; (b) first machine cells are formed; (c) part families and machine cells are formed simultaneously.

In its simplest form, CFP can be represented as Machine-Part Incidence Matrix (MPIM) (Figure 1-2(a)). In a given example, problem consists of grouping 6 parts (denoted as P1-P6) and 7 machines (denoted as M1-M7). Ones (1) indicate that part is being processed by a machine, whereas voids (sometimes also shown as zeros) show the absence of processing. By rearranging rows and columns, diagonal blocks can be formed as shown in Figure 1-2(b). Two obtained blocks represent two machine cells and two corresponding part families. Parts P1, P5 andP3 can be processed by cell#1 with machines M1, M3, M5 and M7. Similarly, parts P4, P2 and P6 can be processed by cell#2 with machines M2, M4 and M6.

	P1	P2	P3	P4	P5	P6
M1	1		1		1	
M2		1		1		1
M3	1		1		1	
M4		1		1		1
M5	1		1		1	
M6		1		1		1
M7	1				1	1

(a)

	P1	P5	P3	P4	P2	P6
M1	1	1	1			
M3	1	1	1			
M5	1	1	1			
M7	1	1	0			1
M2				1	1	1
M4				1	1	1
M6				1	1	1

(b)

Figure 1-2: Initial (a) and final (b) MPIM

Given solution has one void in cell #1 (M7; P3) and one element that does not belong to any cluster (M7; P6) and is referred to as exceptional element. Exceptional element means inter-cellular movement of parts. Here, part P6 that belongs to the second cell will have to visit machine M7 form the first cell. Also, in contrast to parts P1 and P5, part P3 doesnot require machine M7 thus creating a potential under-utilization problem of this machine in the cell. Exceptional elements and voids are undesirable since they reduce cell's performance by creating unnecessary transfer of parts between cells and idling of equipment respectively. In real-world cases it is almost impossible to form clusters completely free of voids and exceptional elements.

Being simple, MPIM disregards many important factors. Because of that, seemingly optimal clusters obtained this way may show poor performance when implemented in practice. For instance, if product volume (demand) is not taken into account, one exceptional element may seem to be better than two. In fact, if material handling caused

by one inter-cell move outweighs material handling caused by two inter-cell moves, second option would be preferable.

According to Tunnukij and Hicks [17] “Well designed manufacturing cells should maximize the machine utilization within each machine cell and minimize the inter-cell flow of parts.” (p.1990). One way of doing so is by taking alternative process plans/routes into account. In practice, parts can be produced in not only one, but several ways. In other words, several alternative process plans/routes already exist or can be generated for each part. According to Vin and Delchambre [18], “process plan is characterized by a sequence of machine types (lathe, etc.) required to transform raw material into some finished goods.” and “process route or routing represents the sequence of specific machines or work centers on which a part passes through to complete its processing, as defined by the process plan.” (p.1114). CFP where alternative plans/routes are considered is called Generalized Cell Formation Problem (GCFP). Kusiak [4] was one of the first who considered alternative routes in the context of CFP. He points out that they give more freedom for designing optimal CMS, allow better grouping due to flexibility, but also bring more complexity. A mathematical model for GCFP for the first time was developed by Choobineh [19].

Solving GCFP implies selecting the best process route for each part which is referred to as Resource Planning Problem (RPP) and secondly – solving CFP as such. RPP and CFP are mutually dependent (i. e. solution to GCFP depends on the process routes selected in the first stage). These problems can be solved sequentially, in iterations or

simultaneously. Simultaneous handling of two problems is difficult since both of them belong to the class of NP-hard problems [20]. Aljaber [21] gives an example where solving both problems concurrently with as little as 50 parts, each requiring 5 operations that can be performed on one of two alternative machines, results in more than $1.8 \cdot 10^{75}$ possible solutions. Despite these difficulties, in recent works done by [22], [23], [24] and [25], authors developed meta-heuristic and hybrid methods that realize concurrent solving of RPP and CFP.

Another way of minimizing inter-cellular traffic is through implementation of hybrid cells – cells where first portion of operations may be made in one cell, rather than the rest–in another one. This approach helps to reduce inter-cell movements to a single transition. In other words, allowing some inter-cell movement is compensated by more efficient cells with less cumulative inter-cell traffic.

As it was mentioned earlier, traditional CMS suffers from reduced flexibility and low machine utilization. This is aggravated especially in a highly dynamic environment with unpredictable demand, short product life cycles,and high variety. In case when previously optimal CMS design is no longer efficient in new conditions, a concept of Dynamic CMS (DCMS) is a viable alternative. DCMS concept was first introduced by Rheault et al. [26] and implies economic reconfiguration of MCs according to new circumstances in different time periods. Reconfiguration means machine relocation between and within cells. A comprehensive model for the design of DCMS was presented by Lokesh and Jain [27].

Another important factor to consider is the sequence of operations. “The operation sequence for a part is an order of the machines on which the part is sequentially processed.” [28] (p.86). Despite the abundant literature on CFP, few researchers address this factor in their work. Jayaswal and Adil [29] developed a model and solution methodology that incorporates alternative routes, operation sequence and some other factors to solve CFP. Sequence of operations not only affects the amount of inter-cell traffic, but also defines how parts will move inside a cell (intra-cell moves). Ideally, MC should operate as a small flow line. In reality, however, parts may go forward and backward depending on a process route. The higher the intra-cell traffic, the less efficient the cell is. Considering the sequence of operations is important because it can help to determine the optimal number of cells during the solution process instead of guessing upfront. According to Adil [30], the more cells in a system, the higher the cost if inter-cell and lower the cost of intra-cell transfer per part. Interrelationship of inter and intra-cell movement cost can be used to determine economic number of MC to be formed [31].

CFP can be solved based on single or multiple objectives. Multi-objective optimization is more realistic, but is more complex and oftentimesincorporates conflicting objectives. Also, CMS designer must assure that machine utilization levels are acceptable, production demands are satisfied, machine capacity levels are respectedand the total production cost is below the set limit. These and some other factors are referred to as “constraints”. Proper choice of constraints has great importance. Solving CFP without demand consideration leads to inefficient cells or cells that might not be able to provide

needed production volumes. CFP with explicit consideration of machine capacities is called Capacitated Cell Formation Problem (CCFP).

1.6 Research objectives

This work is the initial step in the endeavor of addressing a major challenge in the CFP research that was laid out by Askin [32] as follows: "Nonetheless, a comprehensive model that fully encapsulates factors likely to be encountered in practice does not exist. Moreover, there is a shortage of widely available, easy-to-use tools or decision support systems available to the practitioner." (p.6785). With this perspective in mind, three research objectives were defined:

1. Create a generalizeddata model for the cell formation problem.
2. Develop a generalized cell design framework that capturesreal-world data.
3. Develop a progressive algorithm for solving the CFP.

1.7 Thesis organization

This thesis is organizedin six chapters. The first chapter introduces the concept of cellular manufacturing in general and the cell formation problem in particular, sets the research objectives, and outlines the thesis. Chapter two reports and summarizes the related literature review which will be divided into three major sections: 1) Similarity indices 2) CFP solution optimization approaches with focus on meta-heuristics 3) Multi-objective models and algorithms of the CFP. Chapter three introduces a cellular design component-based framework for a cellular factory and utilizes it in implementing several CF

similarity/dissimilarity indices from the literature. Chapter 4 introduces similarity/dissimilarity indices and performance indicators of CMS. Chapter 5 introduces a novel Progressive Algorithm for the CFP. Chapter 6 summarizes the thesis and highlights direction for future research.

1.8 Summary

CFP is a complex combinatorial optimization problem which consists of forming machine groups and part families so that a desired peak performance is achieved. Formulation of CFP through simple MPIM has very limited application in the real-world cases. Alternative process plans/routes, machine capacities, product processing time and demand, consideration of multiple objectivesand other production factors help toobtainmuch more realistic solutions. Main contribution of this work is in pioneering the development of comprehensive CFP design framework and the development of a novel solution algorithm for a binary problem with the objective of maximizing grouping efficiency.

Chapter 2 : Literature review

2.1 Introduction

CFP has been studied for more than three decades now and a large number of solution techniques were developed. Early approaches were overly simplistic and were hardly applicable to real-world problems. As a research went on, more advanced models and methods emerged that could incorporate important production factors and considered multiple objectives. Despite the progress though, the majority of the existing CF techniques offer contribution only from a theoretical perspective thus making their application in industry still limited [33].

Literature on CFP itself is abundant. Comprehensive reviews on the subject of CFP can befound in [34-37]. This chapter offers a review of up-to-date advancements in the research of CFP. It is organized into three sections, each representing a certain area in CFP research. First section covers (dis)similarity indices that have been used when solving the CFP. Second and third sections cover single-objective and multi-objective meta-heuristic approaches respectively. Most information discussed in this chapter is concisely summarized in respective tables. Table 2.1 has production factors that were considered in the literature reviewed in this chapter.

Table 2-1: CFP-related parameters, constraintsand objectives

Machine-related parameters		8	Outsourcing cost	17	Min. machine breakdown rate/cost
1	Machine capacity	Constraints		18	Min. machine operation cost
2	Machine setup time/cost	1	Min/max/exact number of machine cells	19	Min. machine amortization cost
3	Machine duplication	2	Min/max cell size	20	Min. subcontracting cost
4	Machine procurement cost	Objectives		21	Min. machine maintenance cost
5	Machine breakdown rate/cost	1	Min. total number of inter and/or intra-cell moves	22	Min. tool consumption cost
6	Machine operation cost	2	Min. total distance/volume/cost of inter and/or intra-cell moves	23	Min. machine overhead cost
7	Machine relocation cost	3	Max. grouping efficiency	24	Min. inter-cell load variation
8	Machine amortization cost	4	Max. GE		
9	Machine maintenance cost	5	Max. generalized GE		
10	Machine overhead cost	6	Max. total bond energy		
11	Mean/exact time/rate between machine failures	7	Max. machine utilization		
12	Tool consumption cost	8	Max. total similarity between parts/machines in cells		
Part-related parameters		9	Min. total production time/costs		
1	Processing time/cost	10	Min. within-cell load variation		
2	Production volume/demand	11	Min. WIP		
3	Alternative process plans and/or routes	12	Min. machine idling time/cost		
4	Operation sequence	13	Min. number/cost of duplicate machines		
5	Part/batch handling cost	14	Min. setup time/costs		
6	Batch size/weight	15	Min. machine relocation costs		
7	Uncertain product demand/mix	16	Min. machine procurement costs		

17

2.2 Similarity indices

Similarity Coefficient (SC) methods rely on (dis)similarity measures in conjunction with clustering algorithms. From here onwards, under SC term we will implynot only similarity measures but also dissimilarity measures unless a particular method is described.SCs are easy to use, flexible and can incorporate more than one production parameter. SCs are calculated for pairs of given machines or parts and are presentedin a form of matrix of size MxM or PxP (M stands for the number of machines and P – for the number of parts in a given instance). The more/less pairs have in common, the higher/lower the value. For example, in a simple instance with MPIM, for a pair of machines that process the very same parts SC will have maximal value. SC matrices are inherently symmetrical with the diagonal consisting of "0s". Since the real-world problems usually have far lesser number of machines than parts, it is sensible to work with MxM rather than PxP sized matrix. After machine cells are identified, part families can be obtained. Similarity matrixis organized into clusters by a clustering algorithm.Yin [38] presented an extensive review of SC-based CF approaches and developed a taxonomy that systemizes them and clarifies the application of SCs. A large number of SCs has been proposed in the literature. This section provides a review of SCs that have beenused when solving CFP. First, general purpose SCsare covered followed by problem-oriented SCs. Concise summary on SCs covered in this section is given in Table 2.2.

General purpose SCs are used not only in CF but in other problems with similar structures. McAuley [39] was the first one who applied SC to CFP. He used Jaccard SC

[40] and single linkage cluster analysis to build a tree of solutions ranging from “all machines belong to a single group” to “all machines belong to separate groups.” Minimization of total inter-cell movement cost was proposed as criteria to select the best solution. Being simple, Jaccard SC also has limitations which are discussed by [41]. Other popular SCs in this group are Sorenson SC and Baroni-Urbani and Buser SC [40].

Problem-oriented SCs are somewhat different. Since consideration of different production factors and different objectives result in different cell configurations (for the same set of machines and parts), seemingly similar objects may not have high degree of “affinity” in the context of a particular problem. For example, if production volumes are taken into account with the goal of minimizing inter-cell material flow, those pairs that facilitate this goal may have higher similarity than they would in a case when only binary MPIM was considered. This peculiarity is also explained in [38].

Problem-oriented SCs can be divided into two groups: binary data-based and production data-based. Binary data-based group of SCs incorporates only part assignment information. The difference from general purpose SC is that specifics of CFP are taken into account when SC is developed. In [42] authors point out how different numbers of machines required by two parts cause bias when similarities are calculated and proposed simple and straightforward SC to overcome this issue. [43] developed SC that minimizes the number of inter-cell moves and maximizes within-cell machine utilization. It can take both positive and negative values and therefore represent similarity as well as dissimilarity of elements. Besides, the proposed approachis able tofindan optimal number

of cells without specifying it in advance (auto-clustering). In attempt to overcome the drawbacks of existing at that time SCs. [41] proposed simple dissimilarity coefficient that minimizes the average number of voids obtained when merging two machines or two groups of machines. Proposed approach was found to be reliable and efficient.

Production data-based group of SCs incorporates various production factors which can improve the quality of obtained solutions. DeWitte [44] was the first one who integrated production volume factor into SC.They are a weighted adaptation of Jaccard SC and according to author's findings demonstrate potential benefits in comparison with the previously presented measures. Steudel and Ballakur [45] were the pioneers in developing SC that took processing time into account. Also, they proposed heuristic algorithm that exploits similarities between machines and consists of two stages. First, SC is calculated and then the machines are organized into a chain so that the sum of bonds is maximized. Machine cells are formed in the second stage by partitioning a chain.

Selvam and Balasubramanian [46] were the first to incorporate operation sequences into SC and developed heuristic clustering algorithm for solving the CFP with the objective of minimizing material handling and machine idle time costs. Two-stage SC-based procedure to solve CFP with consideration of operation sequences was proposed by Choobineh [19]. New SC and special clustering algorithm form part families in the first stage and a linear integer programming model assigns machines to the cells in the second stage so that the production costs and machine-related costs are minimized. Another SC

with a sequence of operations was developed by [28] and [9]. Based on this measure they proposed an approach that explicitly considers the intra-cell machine sequence and machine loads. Parts are grouped based on the minimum number of non-common machine types in the group.

SC with consideration of alternative process routings was proposed by Gupta [47] for the first time. Besides, production volumes and processing times were taken into account and a heuristic algorithm for machine clustering and formation of part families was developed. Another SC-based approachwith multiple process routings and objective of minimizing thenumber of inter-cell moveswas developed by Won and Kim [48]. Unlike some previous authors, they consider machine pairs rather than pairs of parts. When calculating similarity, number of parts processed by a particular machine is counted as one even if alternative routes for that part also use that machine. To overcome machine chaining problem,variable clustering criteria was used. At each iteration, machines belonging to the current cells are removedand a new reduced similarity matrix is constructed. Based on Viswanathan [43] and previous work, two modified similarity coefficients for p-median model of CFP were proposed by Won [49]. With both coefficients incorporating alternative process plans, first reflects similarity between machines and second reflects similarity as well as dissimilarity between them.

Seifoddini and Tjahjana [50] introduced a batch size-based SC to solve CFP with high part variety. Given SC was applied to a case study with 39 machines and 22 part groups. A SC for machines that incorporates operation sequence, production volume and

processing time was proposed by Gupta and Seifoddini [51]. A SC assigns similarity values to machines in such a way that machines processing parts with high volumes will have higher affinity than machine pairs that are related to low-volume parts. Besides, higher similarityis given to machines that require more handling. Complete LINKage clustering (CLINK) was used to form machine cells. In addition to the factors considered, Yin and Yasuda added alternative process routes to their SC and used it in the first stage of their heuristic approach to obtain machine cells. Stage two improves solution and solves machine capacity problem. A combined dissimilarity coefficient with multiple parameters was developed by [52]. It employs operation sequence, production volume, processing time and machine capacity factors. Based on dissimilarity matrix, a minimum spanning tree is builtin order to obtain machine cells. At each iteration, part with the highest induced average cell load and the minimum number of inter-cell moves is assigned to a cell.

Table 2-2: Summary on covered SCs

Author	SC	Parameters						Type
		Machine-related	Part-related					
		1	1	2	3	4	Other	
1	Jaccard							Universal
2	Sorenson							Universal
3	Baroni-Urbani and Buser							Universal
4	$MAXCS_{il}$							Universal
5	s_{ij}							Machine-based
6	Average Void Value (Dissimilarity coefficient)							Machine-based
7	S_{ij}			+				Part-based
8a	Additive SC			+				Machine-based
8b	Multiplicative SC			+				Machine-based
9	Cell Bond Strength		+					Machine-based
10	S_{ij}			+		+	6	Part-based
11	$S_{ik}(L)$					+		Part-based

Table 2-2 (cont.): Summary on covered SCs

12	SO_{pq}					+		Part-based
13	S_{ij}		+	+	+			Machine-based
14	Generalized SC				+			Machine-based
15a	s^1_{hi}				+			Machine-based
15b	s^2_{hi}				+			Machine-based
16	Batch SC						9	Machine-based
17	S_{ij}		+	+		+		Machine-based
18	S_{ik}		+	+	+	+		Machine-based
19	Dissimilarity coefficient	+	+	+		+		Machine-based

2.3 CFP solution approaches with focus on meta-heuristics

This section covers up-to-date single-objective meta-heuristic approaches that were proposed to solve CFP. But first, concise information on meta-heuristic algorithms is provided. As it was mentioned already, CFP is NP-hard combinatorial optimization problem. These problems have large decision space and searching it entirely is impractical. Instead, domain-specific knowledge is used to help guiding a search towards a goal state. As a result, acceptable solution(s) can be found faster without exploring all the possibilities. Such algorithms are called heuristics. They significantly improve solution time, but cannot escape local optimum. Once a local optimum is found, the heuristic algorithm stops. Meta-heuristic algorithms are general purpose approximate algorithms that can go against heuristic function and thus escape local optimum. Essentially, the success of the method is defined by the balance between exploration and exploitation types of behavior. This is illustrated in Figure 2-1.

Figure 2-1: Concept of metaheuristics

Exploration side in its pure form is represented by a “random walk” algorithm which takes step in a random direction and evaluates chosen solution. Exploitation, on the other hand, is the process of refining promising solutions and corresponds to the “hill climbing” technique where algorithm chooses the best solution among neighbors which means following the steepest gradient. Since meta-heuristics do not fully explore the decision space, optimality cannot be guaranteed. Meta-heuristic algorithms are insensitive to the form of the objective function which gives the designer the flexibility of modifying the search depending on priorities. A major drawback for many meta-heuristics is the need for a pre-search (which is done by trial and error)to identify algorithm's optimal control parameters (like population size, mutation rate, initial temperature, aspiration level, etc.). Due to stochastic nature of meta-heuristics and depending on the algorithm, it might be a good idea to run algorithm several times in order toobtain a set of alternative solutions. Hybrid meta-heuristics have been widely used within the last decade. The idea is to combine meta-heuristics and any other search methodso that weaknesses of one are compensated by the strengths of another and algorithm’s performance is improved.

To this day, a large number of meta-heuristics have been developed. By studying different algorithms, researchers discovered that despite the existing diversity of names, virtually any meta-heuristic can be defined bya number of unified aspects. In other words, there can be two meta-heuristics that belong to different groups, but essentially will act the same, or there can be two meta-heuristics from the same group that are completely different. According to Talbi [53] and Liefooghe et al. [54], those aspects are

solution representation, solution initialization, solution evaluation, variation operators, fitness assignment, diversity preservation, selection and replacement strategies, elitism and continuation strategy. Subsequent literature reviewis done with this perspective in mind. Concise summary on the literature covered in this section is given in table 2.3.

Tabu Search (TS) was introduced by Glover [55] as a local search algorithm that can escape local optimum. During the search, self-updating tabu listis created where recent moves are stored. Moves in tabu tenure are not allowed for certain period of time even though they may be better than neighboring solutions. This gives algorithm exploration abilities and prevents it from cycling. The size of a tabu list is important. Small list impairs exploration abilities and may cause cycling whereas long list decreases algorithm's performance [56]. There is no general technique to determine the size of tabu list, but it is regarded to be proportional to the size of the problem [57].

Sun et al. [58] modeled CFP as a weighted graph partitioning problem where weights represent material flow between machines. Authors used TS-based approach with consideration of operation sequences, production volumeand minimization of inter-cell material flow as an objective. Like many TS-based approaches, a solutionwas represented as a string of integer numbers that represent cell number to which machine is assigned and position of an integer in a string represents machine number.Initial solution was generated randomly and single and double moves were implemented to generate neighborhood solutions. In order to improve algorithm’s efficiency, binary tree data structure and look-ahead strategy were implemented. Look-ahead strategy assesses not a

single next move, but a sequence of moves and chooses the one that gives the best objective. Authors reported superiority of proposed scheme over conventional greedy strategy. Lei and Wu [59] pointed out the significance of a high-quality initial solution for TS and adopted Generalized SC proposed by Won and Kim [48] and SLINK clustering procedure to generate initial solution. To find solutions with minimum inter-cell moves, TS with swap and insertion moves was proposed. Authors reported good performance of the algorithm when compared to other TS and SA algorithms.

Wu et al. [56] proposed TS-based method where production volumeis considered and part flow within cells is maximized. Authors compared two approaches for the generation of initial solutions: a random approach and a group-and-assign approach. In the second case, seed machine is assigned to each cell and then machines are added to cells so that the best objective value is achieved.Such approach was found to be superior and allowed finding optimum solutions for small and medium-sized problems in less than 0.005s. Besides, tabu list of dynamic, self-adjusting size with long-term memory mechanism was developed. Depending on the improvement of recent moves, measure of attractiveness of a move was adapted. If solution wasn't improved after a certain number of moves, diversification strategy was applied. If improvement was good, intensification strategy took place. Later, [60] proposed a three-stage algorithm with consideration of alternative process plans. During the first stage, process plans were selected randomly. TS procedure described in previous work coupled with Jaccard SC was used in the second stage to complete thepart assignment. In the third stage, machine cells were formed so that the number of exceptional elements was minimized. The whole procedure was repeated until

zero exceptional elements were found or the maximum number of iterations was reached. Algorithm was tested on problems of various sizes and was able to find optimum solutions for problems with unknown before optimum.

Simulated Annealing (SA) was introduced by [61] and popularized by Kirkpatrick [62] and is considered to be the oldest meta-heuristic search algorithm. SA was inspired by the annealing process in metallurgy which involves controlled cooling of a hot metal in order to obtain its equilibrium structure. Behavior of SA algorithm is governed by the probability function with parameter T which, similarly to the temperature of the system during annealing, decreases with time and thereby provides controlled randomization. Probability function determines the probability of going from one solution (node) to another. Algorithm starts at a high-temperature$T$. When T is high enough, the probability is equal to 100% regardless of the quality of solution. When T drops to 1.0, the probability of accepting best solution and rejecting worst is 100%,and the algorithm performs actually as “hill climbing.”

CFP with alternative process routings, production volume, machine capacity and the objective of minimizing inter-cell traffic was proposed by [22]. Branch-and-bound algorithmwas usedfor routing selection and SA to solve CFP simultaneously.[23] proposed SA-based auto-clustering algorithm forthe CFP with alternative process routes and objective of maximizing similarity between parts. New solution representation in a form oftwo-row matrix was adopted. First row of representation is a binary string that indicates whether a cell is selected for assignment and the second row is a commonly

used string of integers. Besides, the author proposed simple movement strategy that always generates feasible solutions. [32] proposed a three-stage algorithm to solve CFP with alternative process routings, operation sequenceand production volume. At first stage, RPP is solved by maximizing “0” elements in MPIM. Second stage involves SA and consists ofgrouping machines that machine chain similarity measureis maximized. Parts are grouped in the third stage so that the total inter-cell moves are minimized.

CFP with alternative routes was considered by [63]. The authors developed two-stage hybrid SA algorithm with auto-clustering and mutation operator (HSAM). The initial solution is generated in the first stage by grouping machines based on generalized SC by Won and Kim [48] and assigning part routes to the cells so that number of exceptional elements is minimized. Initial solution undergoes improvement stage followed by HSAM. In contrast to the vast majority of SA-based approaches where Boltzmann functionis commonly used [64] introduced Cauchy function to guide cooling schedule. According to authors’ findings, it allows to enhance algorithm’s exploration capabilities.Another point of this work is that solution representation consists of two segments. Essentially, it is a commonly used integer string that consists of two segments representing machines and parts respectively.

Genetic Algorithms (GAs) are non-deterministic meta-heuristic search and optimization techniques that were introduced by Holland [65]. GAs belong to evolutionary algorithms (EA) that have a distinctive feature of working with a population (set) of solution candidates rather than just only one candidate at a time. GAs were inspired by the

evolution of biological organisms – the “survival of the fittest” principle and incorporate 3 basic steps: selection, crossover and mutation.

Here is a description of the basic GA. GA works with a population of candidate solutions that are encoded into a string (or matrix) called “chromosome.” In turn, individual elements of the stringare called “genes”. GA search commences with the generation of initial population. Then, solutions are evaluated against fitness function and ranked depending on their fitness value. Next, selection takes place where the fittest candidates are chosen for crossover stage. Crossover operator shares fragments of solutions among pairs of chromosomes in order to obtain new population. In order to maintain population diversity, low-probability mutation operatoris applied. Finally, old population is replaced (partially or fully) with a new population and algorithm that repeats itself until stopping criteria are reached. As the size of the problem increases, so does the number of generations. An extensive overview of GAs can be found in [66].

Research work on CFP using GAs can be contingently divided into four domains. First one employs simple GAs. Second one uses problem-specific GAs called Grouping Genetic Algorithms (GGAs). Third one represents the multi-objective GAs that can produce sets of non-dominated solutions. Fourth group comprises hybrid GA-based algorithms.A thorough review onthe hybrid GAs was done by [67] where authors focus on possible ways of integrating various optimization techniques within GA framework. They explore the issues that accompany designing of hybrid GA and provide factors that influence algorithm's performance.

Simple GA considering batch size and processing time was introduced by Zolfaghari [68]. The algorithm utilizes commonly used crossover operator where random cut pointis chosenand second parts of chromosomes are exchanged. A GA with the objective function of maximizing Grouping Efficacy (GE) was introduced in [69]. The proposed chromosome structure guarantees solution feasibility and consists of three sections: a section representing the number of cells, a part section and a machine section. Values of genes range from 0 to 0.99 with each interval corresponding to a certain integer number. The position of a gene in the second and the third group corresponds to apart or a machine number. In order to preserve best solutions, they are always copied to the next generation. Roulette wheel, stochastic sampling, and tournament selection strategies were used for comparison. As for the crossover, a single-point, double-pointand uniform crossovers were tested. Authors report that algorithm was able to improve solutions for 40% of test problems. For finding solutions with the maximum GE, [70] proposed GA in conjunction with Variable Neighborhood Search (VNS). As the name suggests, VNS means using several neighborhood structures to generate neighbor solutions during the local search. Authors used solution representation in the form of a string of integers that contains part and machine sections and implemented as a simple one-point crossover and mutation. Another GA-based approach with the objective of maximizing GE and consideration of alternative routes was developed by [71]. GA is used to obtain the machine-cell assignments. The proposed algorithm uses the remainder stochastic sampling without replacement for reproduction, a double point crossoveranda mutation with low probability. Parts are assigned to cells where they have the maximum number of

operations and where the number of machines that donot require this part is minimum. The algorithm was able to improve 55.5% of solutions for 36 test problems.

Falkenauer [72] showed that when dealing with grouping problems, solution representation of the classical GA is highly redundant (which leads to unnecessarily large search space), standard crossover can cause overlapping of groups from different parents and standard mutation may cause too much disruption and consequently diversion from good solution. In order to overcome these problems, he developed GGA – GA specially adapted to grouping problems. Brown andSumichrast [73] were the first to apply GGA to CFP. Authors proposed solution representation consisting of part, machine and group sections. For selection, rank-based roulette-wheel selection with no duplications was developed. The proposed two-point crossover works only on a group section with subsequent modification of part and machine assignments. Such approach help savoiding inefficient manipulation of group structures. Infeasible solutions undergo a repair process during which parts are inserted into cells, where there are most machines for their processing.

[74] proposed a hybrid approach combining GGA with local search.Their design of GGA is very similar to the one described above. In order to improve offspring solutions, partial grouping efficacy-based local searchwas applied. Besides, for preservation of current best solutionan elitism strategy was implemented. [75] developed enhanced GGA with auto-clustering. The new algorithm uses GGA-like type of solution encoding and slightly modified crossover and mutation operators. Prior to crossover, chromosomes are

compared to avoid crossing of identical solutions. Authors implemented a new roulette-elitist strategy combining elitism and rank-based roulette wheelto preserve best solutions. In contrast to the approach proposed by Brown and Sumichrast [73], replacement heuristic was substituted by a greedy heuristic: the algorithm evaluates the fitness value of all possible chromosomes that can be obtained during the assigning of unassigned partsand the best outcome is chosen.

Particle Swarm Optimization (PSO) is an evolutionary population-based optimization method which was introduced by Kennedy and Eberhart [76] and was inspired by the behavior of birds and fish groups. At each iteration, the behavior of a particle is a compromise among the three possible alternatives: (a) Following its current pattern of exploration; (b) Going back toward its best previous position; (c) Going back toward the best historic value of all particles. Position of a particle represents anencoded solution of the problem. [77] proposed a PSO-based auto-clustering method complemented with local searchin order to solve CFP. The objective of grouping efficacy was to be maximized. Proposed method uses group-based representation and operators instead of arithmetic-based ones and local search is applied to only 10% of particles. In order to produce permutations, two equations incorporating add operator (which acts as a typical GA crossover operator) as well as subtract and multiplication operators were developed. Authors obtained the same or improved results for 29 out of 31 problems when compared with alternative approaches. A hybrid approach combining PSO and local searchwas proposed by [78]. Solution representation is built upon a flow graph where edges represent part movement between machines. It is a binary string with a length equal to

thenumber of edges of flow graph. Each allele of representation denotes an edge of the graph and has value of “1” if there is an inter-cell move or “0” otherwise. In order to create diverse initial population, Scatter Search (SS) was utilized. A problem of slow convergence in the final stages of searchwas tackled by introducing the modified velocity equation. Local searchwas appliedin order to improve the algorithm’s convergence towards the objective of minimizing exceptional elements.

Ant Colony Optimization (ACO) is an evolutionary meta-heuristic that mimics the behavior of real ants. When searching for food, ants leave a pheromone – a chemical that marks the path to the food source and helps other ants finding it. With time pheromone tends to disappear, so long paths are given up whereas the shortest (optimal) one is being used the most. One of the most important aspects of the algorithm is the greedy stochastic policy used to construct the solutions. Additionally, pheromone evaporation incorporates a form of “forgetting” that keeps algorithm from premature convergence.

An idea of employing Boltzmann function with constant temperature from SA and mutation operator from GA was implemented by [79]. An initial solutionis formed by the rank order clustering and part-based Jaccard SC. Neighborhood solutions are generated by an insertion move where partis randomly moved from its current cell to another cell. A slightly modified generic mutation operator was applied to introduce diversification. In their work, Kao and Chen [80] argue that there is no clear boundary between machine cells. Based on the degree of belongingness, a machine or a part can be associated with several groups. Following this reasoning, a combination of differential evolutionary

algorithm and Fuzzy logic means clustering was proposed to solve CFP with maximizing the grouping efficacy and the automatic determination of the optimal number of cells. Solutionis represented in a membership-based matrix-like form with three rows. First row is a group row, second row is a fuzzy part relation matrix and third row is a fuzzy machine relation matrix. The offspring are generated using specially tailored crossover and mutation operators. After the recombination of the parent and the mutated solutions follow the local search proposed by[74].

Table 2-3: Summary of literature on single-objective meta-heuristics

Author	Parameters						Constraints		Objectives					Auto-clustering	Solution method
	Machine-related		Part-related												
	1	3	1	2	3	6	1	2	1	2	3	4	other		
1				+			+	+		+				-	TS
2					+		+	+	+					-	A
3				+		+		+		+				+	TSCF
4					+		+	+	+					-	TS+SC
5	+			+	+		+	+		+				-	SA+BB
6							+	+					8	+	SA
7				+	+		+	+	+					-	SA+SC
8					+		+	+	+		+			+	HSAM
9							+	+	+		+		6, 7	-	SA
10	+	+	+			+	+	+					5	-	GA
11							+	+				+		-	GA
12								+				+		+	GA+VNS
13					+			+				+		+	GA
14							+	+			+	+		-	GGA
15							+	+				+		-	HGGA
16							+	+				+		-	EnGGA
17							+	+				+		+	HGBPSO
18								+	+					+	PSO+SS
19							+	+				+		-	ACO-CF
20								+				+		+	B
21								+				+		+	DEFC

A – SC-based hierarchical clustering + TS
B –Hybrid algorithm employing Boltzmann function from SA and mutation from GA

2.4 Multi-objective models and algorithms of the CFP

This section covers up-to-date multi-objective meta-heuristic approaches that were proposed to solve CFP. Concise summary on the literature covered in this section is given in table 2.4. Before presenting a review,ashort introduction tomulti-objective optimization is provided. Multi-objective CFP solutions techniques proposed by most researchers usethe so-called scalar fitness assignment. A popular example is when multiple objectives are converted into a single objective through weighted aggregation [35, 81]. Such approach, however, is overly simplistic. Generally,there is no single decision vector for multi-criteria problem. Instead, there is a set of Pareto optimal (non-dominated) decision vectors. Pareto front for a bi-objective optimization problem is shown in Figure 2-2.

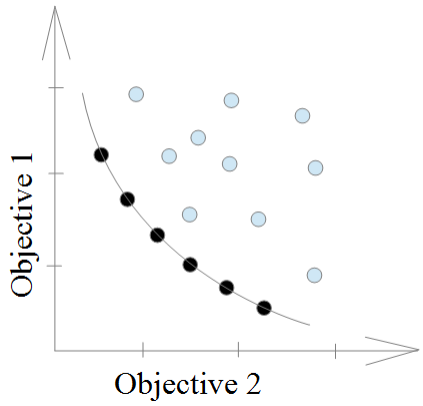


Figure 2-2: Pareto front in the objective space

Black dots represent Pareto front along which no solution dominates other solutions. Solution is non-dominated when it is not possible to improve one objective without deteriorating at least one of the other objectives. For complex optimization problems (like CFP), only a quasi-optimal solution usually can be found. In case of complex multi-

objective optimization problems, Pareto front is approximated rather than identified exactly. At the end of optimization process, a decision-maker is presented with a set of alternative solutions and makes a choice based on his priorities. [54] distinguish three aspects specific to multi-objective meta-heuristics: fitness assignment that will define how well a setof non-dominant solutions is converged in the objective space; diversity that will define how well solutions are diversified in the objective space (measures like crowding and sharing are usually used); elitism that will define how well best solutions will be preserved. For more information on multi-objective optimization, readers are referred to [82] and [53].

Shaller [83] developed two-step TS-based approach with the objective of minimizing costs of inter and intra-cell moves as well as machine amortization costs. Solutionis encoded as a part-based string of integers and the initial solution is created by assigning each part to an individual cell and then combining cells so that the objective function is minimized. Cost of intra-cell movements is assumed to be linear for a certain number of machines in a cell. Authors defined a move as random change of the assignment of a part to a cell. Variable size of tabu list was proposed. The idea is to reinforce the diversification or the intensification of the search depending on the improvement of objective function. Lei and Wu [59] were the first who developed TS for CFP capable of producing a set of Pareto optimal solutions.The proposed approach takes asequence of operations, processing time, production volume and machine capacities into account. Two objectives to minimize were the weighted sum of inter and intra-cell moves and within-cell load variation. New approach reducing computational time was proposed to

identify the non-dominated solutions. Besides, algorithm implements a policy of allowing any move if it produces a non-dominated solution. In addition to the factors considered by the previous authors, [84] incorporated alternative routes and mean time between machine failures into their model. The proposed algorithm provides an optimal number of cells automatically, but in contrast to the previous work, the multi-objective problem of minimizingthe sum of total inter-cell movement cost and machine breakdown costis simply reduced to a single-objective one. In order to increase algorithm's efficiency, initial solutionis constructed by SC-based method. A modified generalized SC by [48] was developed. Tabu list is a three-dimensional arrayand tabu move may be allowed if a move results in a solution that is better than the current best solution. The algorithm also keeps track of solution improvement dynamics. If there was no improvement within certain number of iterations, insertion and mutation moves are applied.

A simple two-stage approach was developed by [85]. First, the parts are assigned to cells by using Jaccard SC and greedy clustering procedure. Then the SA is used to group machines so that grouping efficacy is maximized. The proposed method can produce an optimum number of clusters without the previous specification. A comprehensive multi-period mathematical model with the weighted objective of minimizing the machine maintenance and the overhead costs, machine procurement costs, inter-cell material handling cost, machine operation and setup cost, tool consumption costand system reconfiguration cost was developed by [86]. In order to solve the model, SA algorithm with multiple Markov chains was proposed for the first time. Markov chain is the number of algorithm's iterations at each temperature level. Such approach allows carrying out

search in multiple directions thereby reducing the computational error. A solution representation consists of sections that correspond to part assignment for each period. It also encodes cell number for each operation, proportion of production volume among alternative routes and proportion of the total demand for subcontracted part. Six different perturbation operators were developed to generate neighborhood solutions. Five operators are applied in the first phase of the search and completely independent cells are formed. By eliminating duplicate machines, thesecond stage produces solutions with minimum inter-cell moves and best objective function value.

Another instance of multi-period dynamic CFP with the objective of minimizing inter-cell moves, machine constant and variable cost and reconfiguration cost was addressed by [87]. A solutionis representedin a form of two matrices that describe the assignment of operations to machines and the assignment of operations to cells respectively. Special mutation operators were developed to work with such representation. The proposed approachis also characterized by auto-clustering and the ability of simultaneous part and machine grouping. The same solution representation was adopted by [88]. Authors considered a comprehensive CFP with uncertain product mix and objective of minimizing total machine-associated constant cost, machine operating cost and inter and intra-cell material handling cost.

Venugopal and Narendran [89] were the first to develop GA-basedmethod to solve bi-criterion CFP. Authors proposed a mathematical model for minimizing the volume of total inter-cell material flow and total within-cell load variation. Developed GA features

single and double-point machine swap operator and machine swap mutation operator. The replacement policy is set up so that the best offspring solutions always have priority over parent population. A GA-based auto-clustering approach was developed by [90]. The algorithm is characterized by solution representation that consists of cell and machine rows and a normalized geometric selection scheme. During the selection, individuals are ranked according to their fitness value and then each individual is assigned a probability of being selected based on normalized geometric distribution.

In [91] authors developed multi-objective GA-based algorithm called XGA. Algorithm simultaneously minimizes inter-cell movements, cost of machine duplication and subcontracting cost, system underutilizationand cell load imbalances. Solution's fitness value is calculated using a non-dominated sorting method that was adopted from [92] and which works as follows: First, a set of non-dominated solutions in the population is identified. Next, a large dummy fitness value is assigned to each non-dominated candidate. To maintain diversity, that fitness value is then degraded by sharing it equally among candidates. By reducing the shared fitness value each time, several non-dominated fronts are obtained. Selection among obtained non-dominated sets is done by remainder stochastic sampling without replacementthat is combined with new elitism operator. Authors utilized simple crossover and mutation operators to produce offspring. Proposed algorithmwas tested against NSGA, NPGAand VEGA and around 20% diversity improvement over reference algorithms was reported.

In [93] authors considered CFP with various production and cost factors and proposed a method for maximizing thetotal similarity between parts, minimizing the overall processing time and costs and minimizing the total machine procurement cost. In order to ensure good diversity of solutions along the Pareto front in the objective space, uniform design technique is used to set weights of objective functions and selects a number of points (equal to the number of objectives) in objective space so that they are scattered uniformly. This allows searching objective space in multiple directions instead of a single direction. Solutionis represented by a chromosome where 2 genes are dedicated to each part and stand for cell numbers and selected process plan. A simple crossover and mutation were applied to generate offspring population.

Dimopoulos [81] proposed a multi-objective variant of GP-SLCA (a combination of GP-SLCA and NSGA-II [94]). GP-SLCA is a SC-based clustering technique where SC is not known and is obtained using genetic programming. Genetic programming evolves solutions in a form of computer programs of variable length. In short, genetic programming implies that every solution to a problem can be represented by a program that takes input in and produces a solution. One of the GP-SLCA's features is that it does not require any constraints. In given work authors tailored fitness assignment and selection of GP-SLCA. Several objectives are associated with generated SC and these values are subsequently used by NSGA-II ranking mechanism. NSGA-II-based technique was also developed by [95]. The proposed method uses discrete event simulation to obtain the objective values for candidate solutions. Asolutionis represented as a matrix of integers with rows representing a cell number and columns representing machine

numbers. Two different generation strategies were implemented to generate two halves of the initial population. The first half is generated randomly, whereas thesecond halfis generated by a technique that yields only feasible solutions. Authors used standard NSGA-II crowding tournament strategy for selection. To eliminate closely located solutions and make solution set to be more representative, the average linkage clustering is applied.

In [96] authors made a point that since the relative importance of the objectives is not known, a fuzzy programming can be applied to solve the multi-objective problem. In order to do so, authors proposed an idea of minimizing the maximum deviation of the objective function from its optimum level. A comprehensive multi-objective mathematical model with fuzzy goal programming approach and the objective of minimizing inter-cell travel distance, machine investment cost, machine idle time and keeping uniform workload on cellswas developed. A chromosome consisting of machine and part sections was proposed and a roulette wheel method was implemented for selection.

An example of combining global and local search was proposed by [97]. Authors introduced GA-based hybrid algorithm with TS. After mutation, TS-based local searchis applied to each offspringin order torefine the solutions. The proposed approach considers a variety of production factors and aims at minimizingaggregated sum of the total number of part moves and cell-load variation. [98] considered CFP with alternative process routes, machine duplication and four objectives. In the proposed approach, machine and

part groups are identified concurrently. A solutionis represented as three matrices where the first matrix encodes assignment of operations to machines; the second encodes the assignment of the operations to cells; and the third encodes the number of machines available in each cell. Inversion operators for rows, columns and diagonals are applied to improve initial solutions. In order to keep population diversity, an initial solutionis accepted only if its dissimilarity with existing solutions exceeds acertain threshold. In order to produce a trial set, machine level, cell level and conventional crossovers are used. Local searchis also applied to the newly generated trial set of solutions.

Scatter search (SS) is a population-based method that was introduced by [99]. It works with the two solution sets: a reference set and a trial set. Subsets of reference solutions are combined and a trial set is formed. After an improvement stage for trial solutions, a reference set is updated. As it can be seen SS resembles GA to a high degree. [100] developed SS method to solve dynamic CFP with the objective of minimizing machine relocation cost, machine underutilization and total number of inter-cell moves. The solution is represented as a string with segments each corresponding to a certain time period. Each section has two subsections: the firstone consists of cell numbers to which certain machine is assigned; the second one consists of machine numbers assigned to a certain cell. A diversification generator was adopted in order to increasethe quality of the initial population. After comparison against other popular techniques authors concluded that the proposed SS demonstrated very good results. Another SS-based approach (MOSS) to multi-objective dynamic CFP was proposed by [101]. The authors considered dynamic CFP with multiple factors and the objective of minimizing thetotal cell load

variation and the sum of various costs simultaneously. The proposed method allows for simultaneous machine and part grouping. The solution encoding consists of four matrices: the first matrix representsthe assignment of operations to machines in certain period, the second matrix represents the assignment of operations to cells in certain period, the third matrix represents the number of available machines in each cell in certain period and the fourth matrix represents the number of machines that were moved into a cell or excluded from it in certain period. Standard two-point crossover was utilized to produce the trial set of solutions. Similarity rate function was proposed to avoid the selection of similar solutions. A non-dominant sorting and crowding measure were adopted from NSGA-IIin order to find the Pareto set of solutions.

2.5 Literature summary

Literature on CFP is abundant.We have covered three large clusters, namely: similarity indices, meta-heuristics, and multi-objective approaches. Similarity indices are easy to implement and can incorporate many production factors and can be integrated into other solution techniques. Because of its complexity, CFP usually requires meta-heuristics such as TS, SA, GA, ACO, SS and many others. Meta-heuristic techniques are insensitive to the objective function and can provide quality solutions in a reasonable amount of time. In this chapter, we have tried to examine the different approaches from a perspective of unified parameters rather than their formal names. We extended our review to multi-objective models as well. Multi-objective optimization is more realistic than single-objective, but it is more complex at the same time.

Based on the review it can be concluded that although a large number of publications related to CFP exists,the majority of them, however, are suitable foremost for academic and not practical application. This manifests in anarrow focus of each particular approach, limited capabilities of modifying, intervening and influencing the search process as well as overall “rigidity” of proposed methods.

Table 2-4: Summary of the literature on the multi-objective meta-heuristics

Author	Parameters													Constraints		Objectives					Auto-clustering	Solution method	Pareto set	DCFP
	Machine-related						Part-related																	
	1	3	4	6	7	other	1	2	3	4	5	6	other	1	2	1	2	16	24	other				
1	+					8	+	+			+			+	+		+			19	-	TS	-	-
2	+						+	+		+				+	+	+				10	-	MOTS	+	+
3						5,11	+	+	+	+	+			+	+		+			17	-	TS	-	-
4														+	+	+					+	SACF	-	-
5	+	+	+	+	+	9,10,12	+	+			+	+	8	+	+		+	+		14,15,18, 21,22,23	-	SA	-	+
6	+	+	+	+	+		+	+	+		+	+		+	+		+			15,16,18	+	SA	-	+
7	+	+	+	+		8	+	+	+	+	+	+	7		+		+	+		18	-	SA	-	-
8	+						+	+						+	+		+			10	-	GA	-	-
9	+						+	+							+		+			10	+	GGA	-	-
10	+		+				+	+					10	+	+	+			+	7,13,20	-	XGA	+	-
11	+		+				+	+	+					+	+			+		8,9	-	GA	+	-
12																+				10	+	A	+	-
13	+	+	+				+	+				+		+	+	+		+		11	-	NSGA-II	+	-
14	+	+	+				+	+	+	+				+	+		+	+	+	12	-	B	-	-
15	+						+	+		+				+	+	+			+		-	MA	-	-
16	+	+	+	+		9	+	+	+	+	+	+		+	+		+	+	+	12,16,18, 21	-	SS	-	-
17	+				+		+	+				+		+	+	+				12,15	-	SS	-	+
18	+		+		+		+	+	+		+	+		+	+		+	+	+	15	-	MOSS	+	-

A – GP-SLCA+NSGA-II
B – GA+fuzzy goal programming

Chapter 3 : A generalized framework for the CFP

3.1 Introduction

In this chapter, we are going to present a novel framework for the cell formation problem. The framework is designed to serve several variants of the problem. The main focus of the framework presented is on the system inputs and outputs. The system has been implemented as an integrated module on FactDesign platform. FactDesign is an integrated manufacturing systems analysis and designbeing created and developed at the Systems Engineering Lab at the University of Regina. FactDesign is a propriety software of Prof. Mohamed Ismail. This chapter focuses only on the system's inputs and outputs of the proposed CMS design module implemented within the FactDesign framework.

3.2 System hierarchy

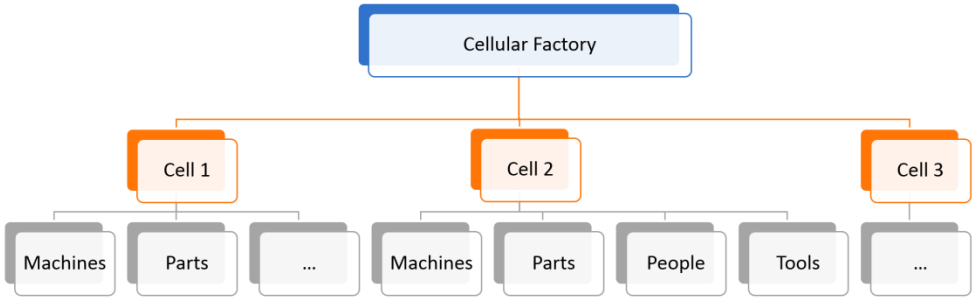


Figure 3-1: Hierarchy of a cellular factory

A cellular factory is composed of several minimally coupled cells or minimal inter-cells moves. Every cell should act almost as an autonomous unit. Every cell will have its machines, part family, workforce and a large set of cutting and processing tools. Figure

3-1 represents a system hierarchy or a tree where a cellular factory represents its root. Intermediate nodes represent a collection of system components or entities (parts/products) and end nodes represent very specific components or entities.

3.3 The new system data model

For many years, the machine/incidence matrix has been the main representation artifact for the CFP data. Several variants have been developed overtime to capture more information such as process plans/alternative process and production flow data. Figure 3-2 shows an example of incidence matrix built in FactDesign.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	1	1
2	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0
4	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	1	1	1
5	1	1	1	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
6	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0
7	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0
8	0	0	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	1	0	0	0
9	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0
10	1	1	1	0	1	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	0	0
11	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0

Figure 3-2: Incidence Matrix as depicted by FactDesign

In our framework, we prepared a detailed model that could capture all the possible data that we can collect about all the system component and entities .Depending on the analyst assumptions and theproblem model, some data could be optional. The first step of defining our problem is to determine the related data using the new CMS design options dialog box shown in Figure 3-3.

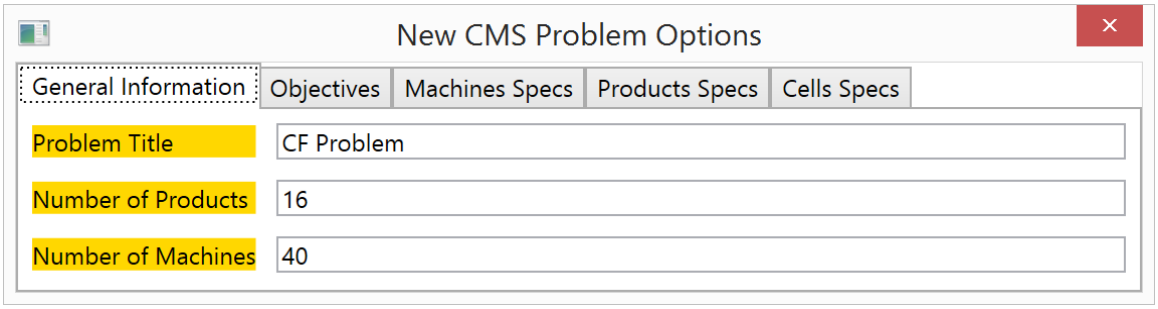


Figure 3-3: CMS problem options

3.3.1 General information tab

Problem Title:

Every problem has a title that distinguishes it from its counterparts. FactDesingsoftware can open several problems at the same time. Having a unique name is mandated by the FactDesign system specifications or guidelines.

Number of machines:

The number of machines (types) available in the system or available to the system designer. Machines here stand for different machine types rather than the overall number of machines. A certain type could have several copies.We are trying to abide by the same convention followed in the CFP literature.

Number of parts:

The number of parts in the system. Those parts are supposed to be clustered into part families and assigned to specific manufacturing cells.

3.3.2 The objectives tab

The literature of the CFP has reported many objectives that have been taken into consideration while working on the cell formation problem. Figure 3-4shows a list of objectives that hasbeen predefined in the FactDesign CMS Design Module.

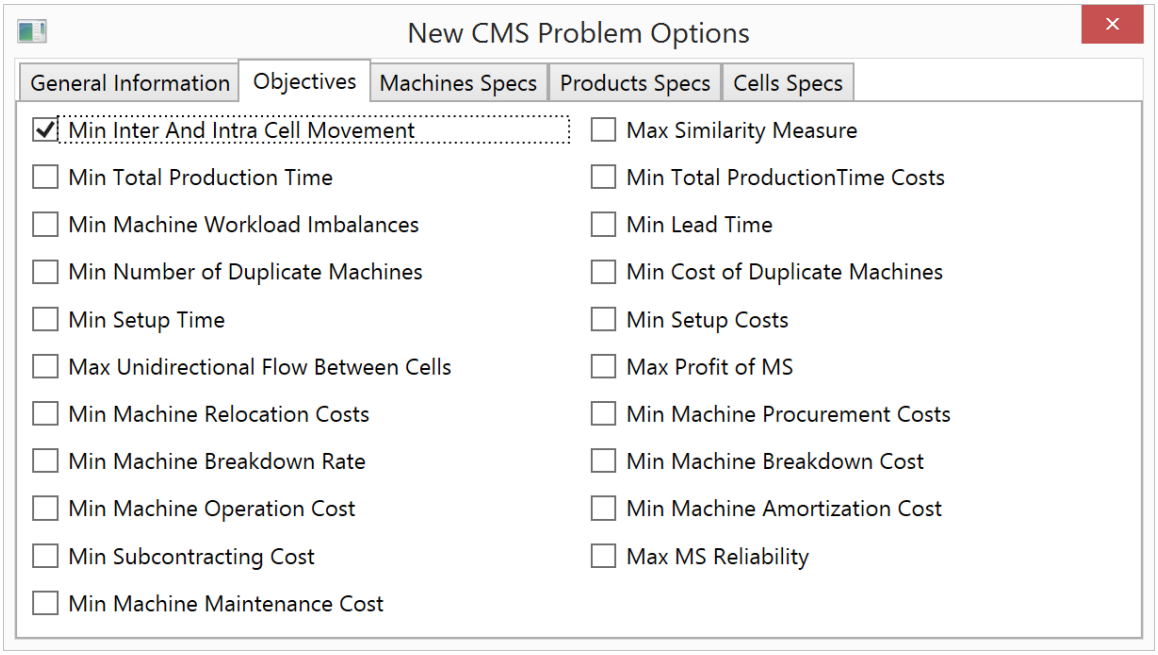


Figure 3-4: Objectives tab

Table 3.1 gives a brief description of every one of those objectives

Table 3-1: Description of available objectives

Objective	Description
Min inter and intra cell movement	Minimize the traffic among and inside different cells. In an ideal setup, there shouldn't be traffic at all.
Min total production time	Minimize total time it takes to produce a given set of parts/products
Min machine workload imbalances	Minimize differences between the loads of different machines in a cell

Table 3-1 (cont.): Description of available objectives

Min number of duplicate machines	Minimize the number of machines that are being introduced into MS as duplicates
Min setup time	Minimize total time it takes to set up machines before carrying operations needed for part production
Max unidirectional flow between cells	The maximum flow in the only forward flow
Min machine relocation costs	Minimize the costs needed for relocating of machines
Min machine breakdown rate	Minimize the rate at which machines break down thus minimizing overall machine downtime due to breakdowns
Min machine operation cost	Minimize the cost of operating of machines
Min subcontracting cost	Minimize the cost of producing parts outside of MS that's being designed
Min machine maintenance cost	Minimize the cost needed to maintain machines
Max similarity measure	Maximize a similarity measure that is being utilized to cluster parts/machines
Min total production time/costs	Minimize total time/cost it takes to produce a given set of parts/products
Min lead time	For make-to-stock products, minimizing the time between releasing the order for production and delivering parts/products into finished goods inventory
Min cost of duplicate machines	Minimize costs needed to acquire additional duplicate machines
Min setup costs	Minimize cost of setup operations
Max profit of MS	Maximize profit of MS in the context of factors that are being considered
Min machine procurement costs	Minimize the costs needed to equip MS with all necessary machines
Min machine breakdown cost	Minimize costs that will be incurred in case of machine breakdowns
Min machine amortization cost	Minimize the reduction of value (cost) of machines with time
Max MS reliability	Maximize the ability of MS to function without machine failures

Multiple objectives can be selected at once. In this case, the model will be solved using a multiple objective algorithm.

3.3.3 Machines

Machines tab (Figure 3-5) lists most of the data relevant to machines. Most models developed in the literature ignore such level of details. Making the data definition optional adds flexibility tothe models being defined.

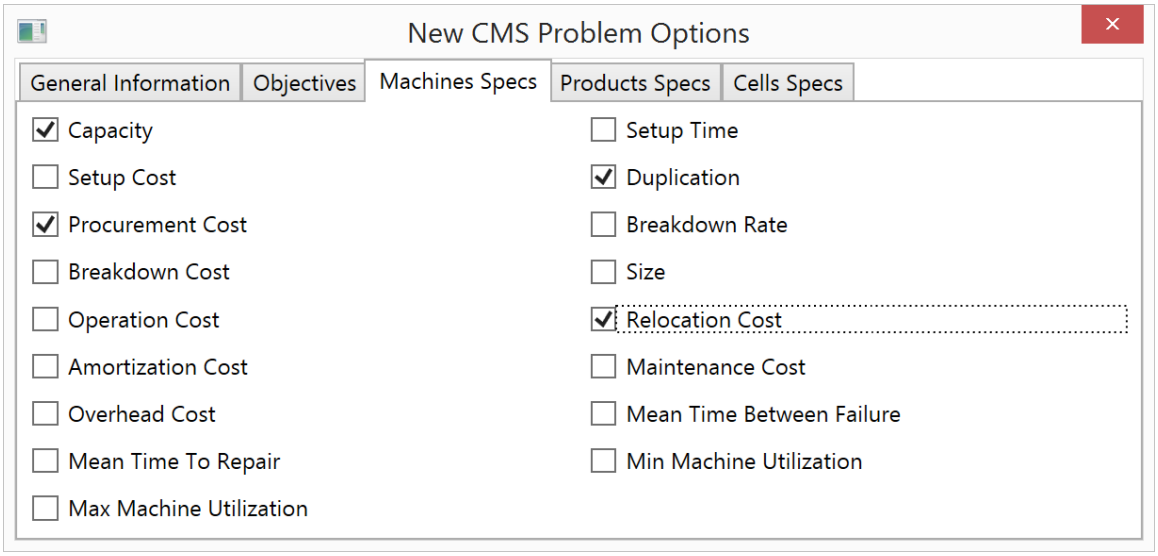


Figure 3-5: Machines attributes tab

Table 3.2 gives a description for every data piece on the list.

Table 3-2: Description of available machines' attributes

M/C Specification	Description
Capacity	Number ofproducts a machine can process during a set period of time
Setup cost	Amount of cost required to set up a machine for a particular operation
Procurement cost	Amount of funds needed to acquire a particular machine
Breakdown cost	Amount of funds that are being lost due to machine breakdown

Table 3-2 (cont.):Description of available machines' attributes

Operation cost	Amount of funds required to operate a machine
Amortization cost	Reduction of value (cost) of a machine with time
Overhead cost	Indirect production-related costs like: amortization cost, electricity consumed by a machine and maintenance personnel-related costs.
Mean time to repair	Mean time it takes to repair a machine
Max machine utilization	Maximum time that machine can be utilized during a certain period of time (work shift/day, etc.)
Setup time	The time it takes to set up a machine for a particular operation
Duplication	Option of adding to MS another same machine if needed
Breakdown rate	Rate at which a particular machine is expected to break down
Size	Physical dimensions of a particular machine
Relocation cost	Amount of funds needed to relocate a machine
Maintenance cost	Amount of funds needed to maintain a machine
Mean time between failure	Average time it takes for a machine to work without a failure
Min machine utilization	Min machine utilization allowed

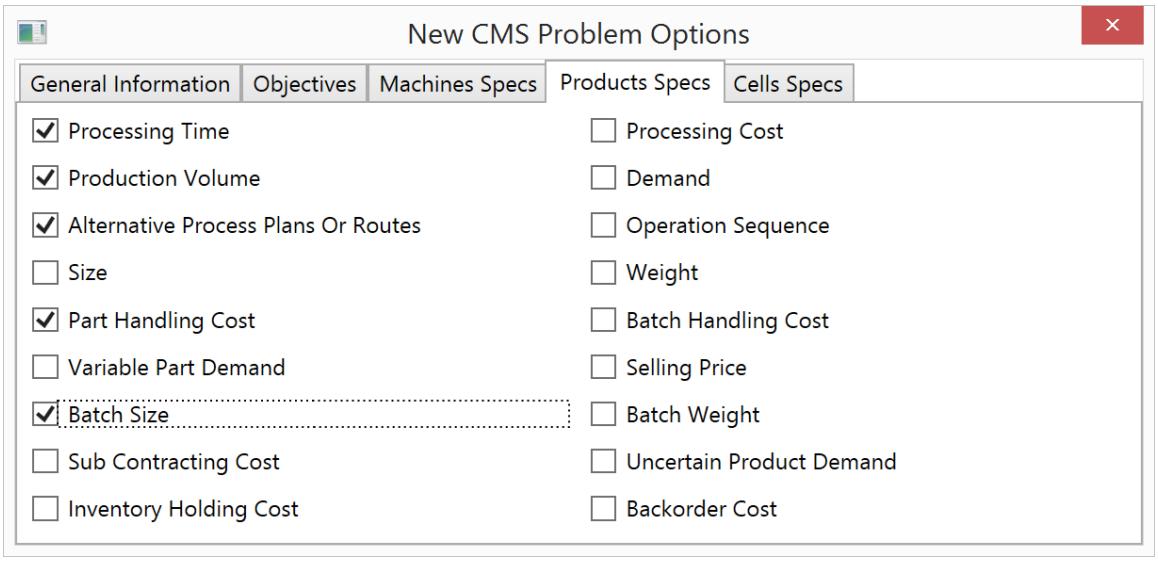


Figure 3-6: Products attributes tab

Figure 3-6 lists most of the available product specifications. Table 3-3 gives a detailed description of part/product specifications.

Table 3-3: Description of available part/product attributes

Part Specification	Description
Processing time	The time it takes to accomplish a particular operation on a particular machine
Production volume	The number of parts that is required during a particular planning horizon
Alternative process plans or routes	Option of having few interchangeable process plans or routes
Size	Physical dimensions of a particular part
Part handling cost	Cost of moving a particular part on a particular distance
Variable part demand	Option of having different part demand during different planning horizons
Sub contracting cost	Cost of outsourcing the production of a part
Inventory holding cost	Cost associated with storing unsold inventory

Table 3-3(cont.):Description of available part/product attributes

Processing cost	Cost associated with processing of a particular part on a particular machine
Demand	Number of parts that have to be produced
Operation sequence	Explicit consideration of operations' sequence
Weight	Weight of a part
Batch handling cost	Cost associated with handling a batch of parts on a particular distance
Selling price	Selling price of a part/product
Batch weight	Weight of a particular batch of parts
Uncertain product demand	Option of having product demand that is not known exactly
Backorder cost	Cost associated with inability to fulfill a particular order on time

3.3.4 Cells tab

Cells may have some constraints that could limit their design options or their size, see Figure 3-7. Cell attributes are listed in Table 3-4.

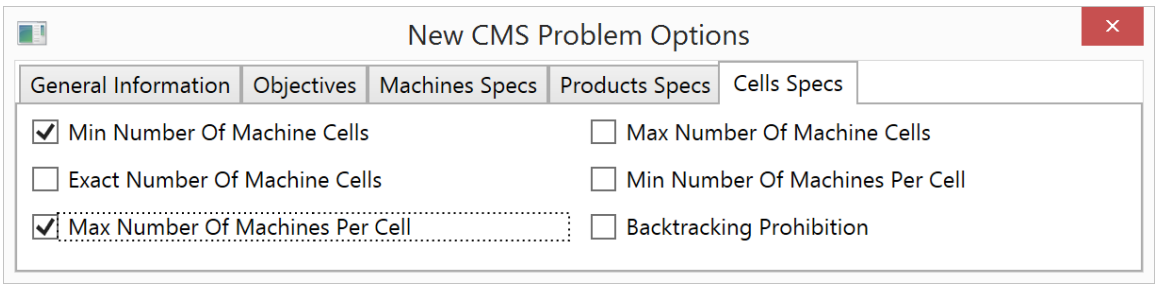


Figure 3-7: Cells specifications tab

Table 3-4: Description of available cell attributes

Cell Design Constraint	Description
Min number of machine cells	Smallest number of machine cells that can be formed
Exact number of machine cells	Number of cells in MS has to be equal to a set one
Max number of machines per cell	Largest number of machines that a cell can have
Max number of machine cells	Largest number of machine cells that can be formed
Min number of machines per cell	Smallest number of machines that a cell can have
Backtracking prohibition	This option allows to exclude part movement in direction that is different from the consecutive flow set by the sequence of machines

3.4 System components data

3.4.1 Machine data

Based on the design specifications definedinthe previous step, machine data forms are adapted to show only data listed in the design specifications step. Figure 3-8 shows sample machine data entry form.

Solimanpur 2 2004

Name	Procurement Cost
1	2500
2	2300
3	2000
4	2200
5	2000
6	2500
7	2500
8	2000
9	2000
10	2000

Figure 3-8: Machine procurement cost data [102]

3.4.2 Part data

Based on the part/product specifications determined inthe previous step, part/product data forms are adapted to show only the data listed in the design specifications step. Figure 3-9 shows sample parts data entry form.

Solimanpur 2 2004Parts Data

Name	Demand
1	80
2	80
3	80
4	80
5	80
6	80
7	80

Figure 3-9: Parts demand data [102]

3.4.3 Rout sheets and process plans

Route sheets show the details of the part manufacturing process. Figure 3-10 shows the details of this process.

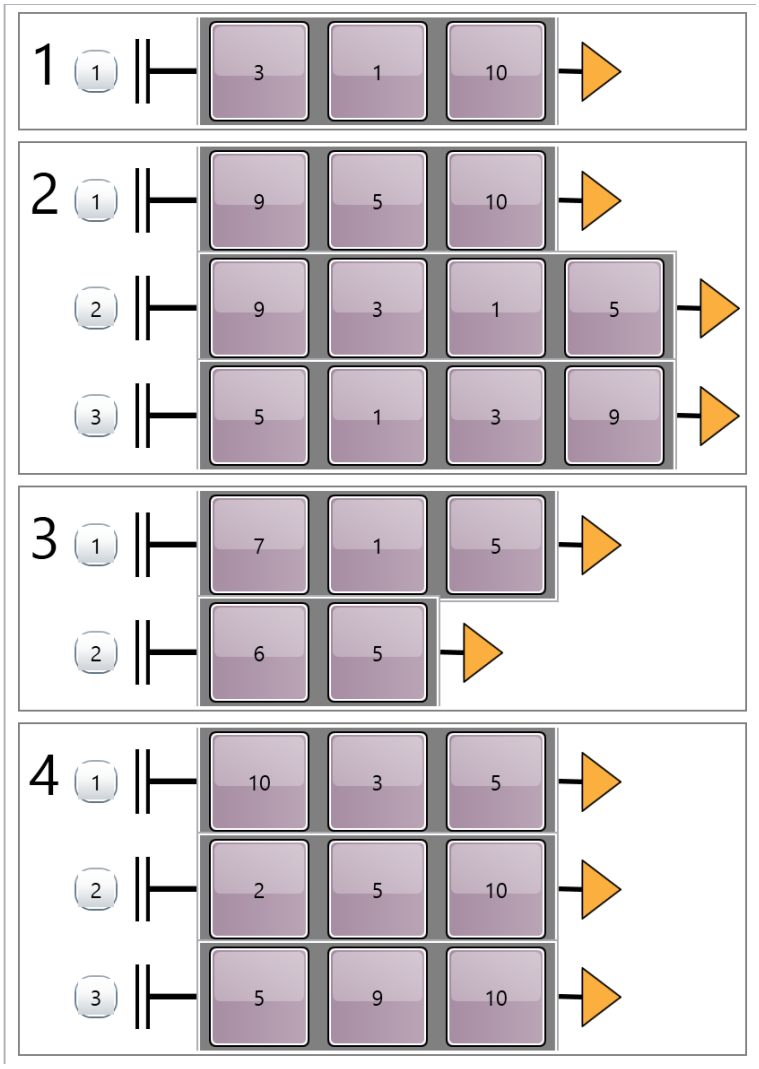


Figure 3-10: Route sheets for the first four parts with alternative process plans [102]

3.5 Matrix-based data input forms

Matrices have been used for decades in the CMS literature. In this thesis, we have also adopted this approach for benchmarking purposes. Incidence and sequence matrices constructed based on data from [102] are presented in Figure 3-11 and Figure 3-12

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

Figure 3-11: Incidence matrix [102]

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

Figure 3-12: Sequence matrix [102]

3.6 Output forms

The output forms are utilized to demonstrate solution results. There is one master form of several output portals are utilized to show the results. Factory graph portal (Figure 3-13) is a drag and drop diagram that enables the usersto redraw the auto-generated factory graph in order to create more appropriate factory network.

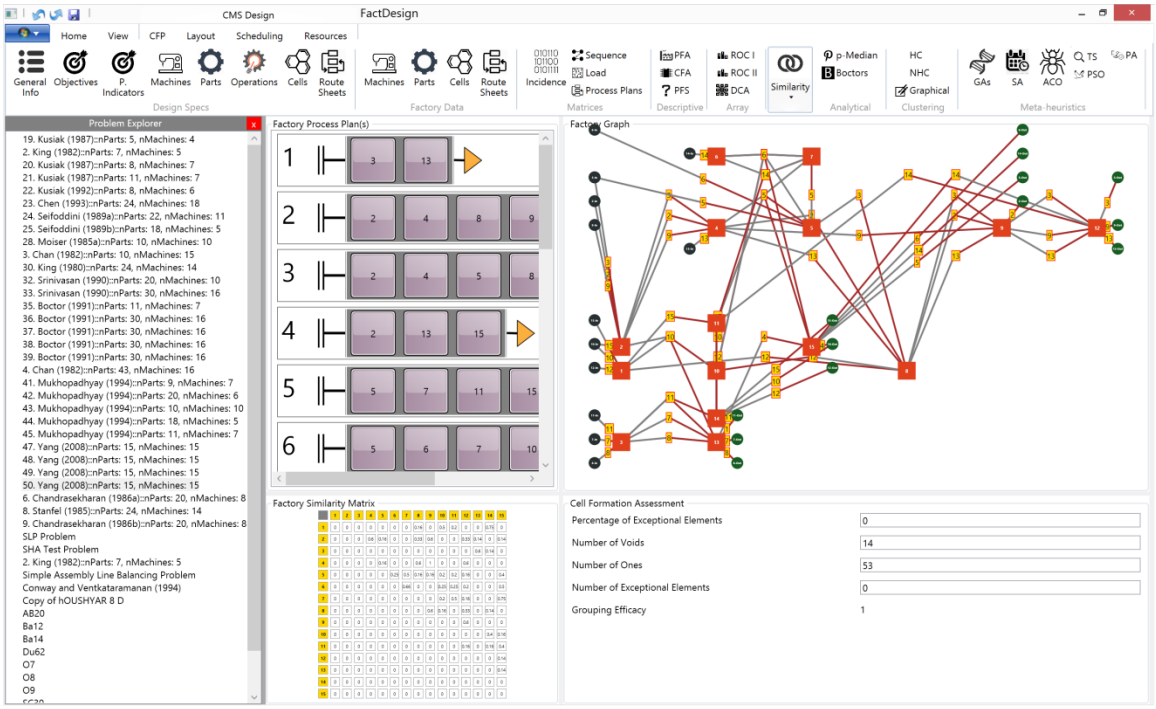


Figure 3-13: FactDesign CMS design output

3.6.1 Process plans viewer

The process plan viewer (Figure 3-14) is utilized to show the detailed process plan for every product. The viewer acts as a great help for tracking product workflow in a dense factory graph.

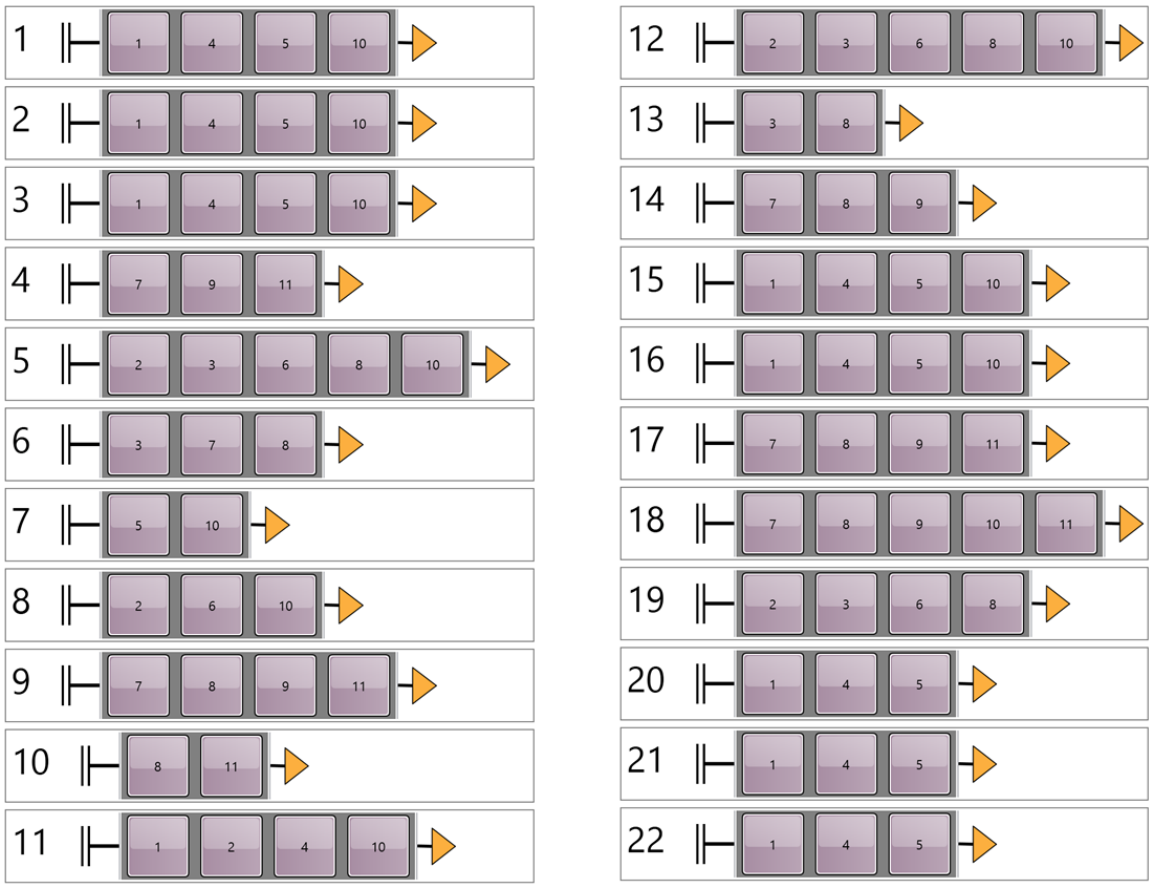


Figure 3-14: FactDesing process plan viewer

3.6.2 Factory graph

Factory graph portal is used to show all the workflow within the system.

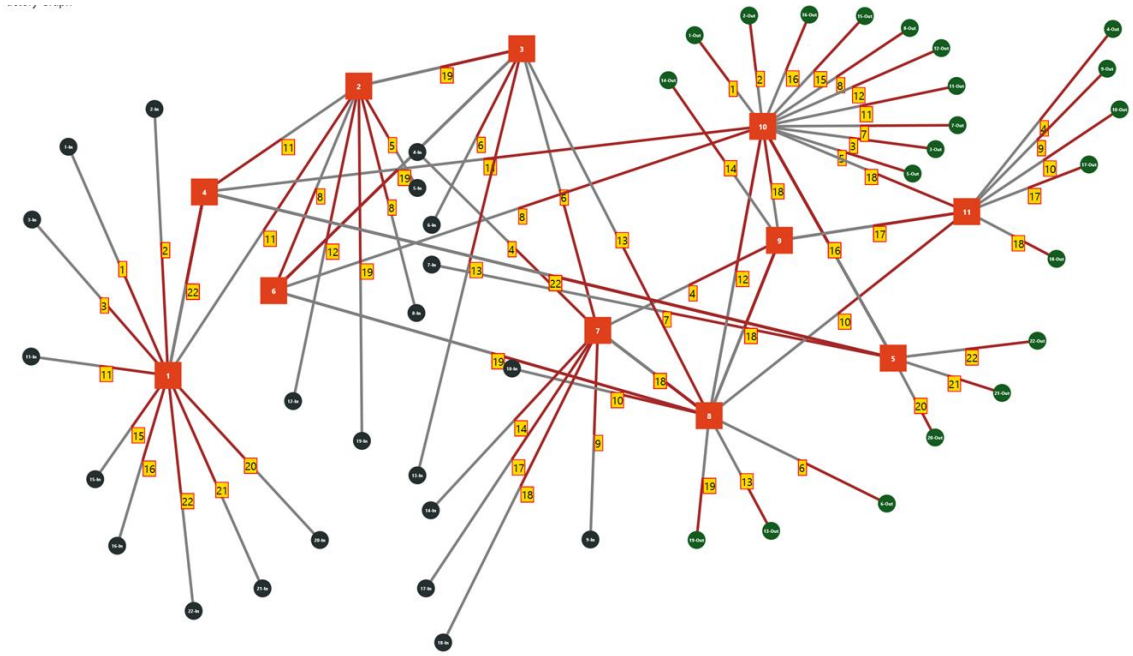


Figure 3-15: FactDesign factory graph: The graph is for [103]

3.6.3 Cell formation assessment portal

To evaluate the grouping efficiency and some other system design measures, this portal is utilized to show grouping efficiency.

3.6.4 Block diagonal matrix

Block diagonal matrix has always been utilized as a representation for CFP solutions. To bench mark our solutions we are using the same notation.

3.6.5 Objective(s) and performance

Objectives portal shows a list of objective(s) and their optimized value(s).

3.7 Summary

In this chapter, we introduced a generalized framework for the cell formation problem. We introduced a detailed data model that shows the system inputs and outputs. The CFP module has been integrated and embedded within the FactDesign software.

Chapter 4 : Similarity coefficients and performance measures

4.1 Introduction

Similarity and Dissimilarity indices of the CFP have been utilized for years in developing efficient heuristics for the cell formation problem. Similarity/dissimilarity coefficient-based methods rely on similarity measuresin conjunction with clustering algorithms as reported in chapter 2. Similarities of machines or parts provide the basis for machine/part grouping. Traditionally, similarity coefficients are presented in the form of machine-machine/part matrix. Machines/parts belonging to different groups will have high degree of dissimilarity. Machines/parts belonging to the same group may or may not have a high degree of similarity (consider parts being processed on two different sets of machines belonging to the same cell). Similarity measures are easy to use and can incorporate more than one production parameter. Since the number of machines is usually far less than the number of parts, machine-based similarity coefficients significantly reduce the size of the problem. Similarity matrices are inherently symmetrical. The novel algorithm that we are going to present in chapter 5 utilizes similarity coefficient in its algorithms during the search space exploration process. In this chapter, we will demonstrate our implementation of some of the most common similarity indices in the literature.

4.2 Similarity/dissimilarity indices

4.2.1 Jaccard similarity coefficient [104]

Jaccard SC (equation 4.1) is a general-purpose SC which can be used to calculate similarities among machines or part types. When two objects have identical values, Jaccard SC will have its maximum value of "1" and when two objects have no similarities, it will be "0". Jaccard SC is sensitive to coding [104] meaning objects will change their similarity when "0" and "1" are interchanged in initial data matrix.

$$C_{jk} = \frac{a}{a+b+c}, \qquad 0 \leq C_{jk} \leq 1 \quad (4.1)$$

where:

$j, k = 1, \ldots N$– index for objects (machine/part types)

N - number of objects (machine/part types)

a - number of part/machine types which visit both machine /part types j and k

b - number of part/machine types which visit only machine /part type j

c - number of part/machine types which visit only machine /part type k

The following example adopted from [105] illustrates the application of Jaccard SC to CFP. Consider MPIM given in Figure 4-1.

Machines \ Parts	1	2	3	4	5
1	1	1	0	1	1
2	1	1	0	0	1
3	1	0	1	1	0

Figure 4-1: Initial MPIM

Based on equation 4.1, Jaccard SC for machine pairs at hand will be:

$$S_{1,2} = \frac{3}{3+1} = 0.75;\ S_{2,3} = \frac{1}{1+4} = 0.20;\ S_{1,3} = \frac{2}{2+3} = 0.40$$

Figure 4-2 depicts the Jaccard similarity matrix for [106] dataset evaluated by the CMS design module of FactDesign.

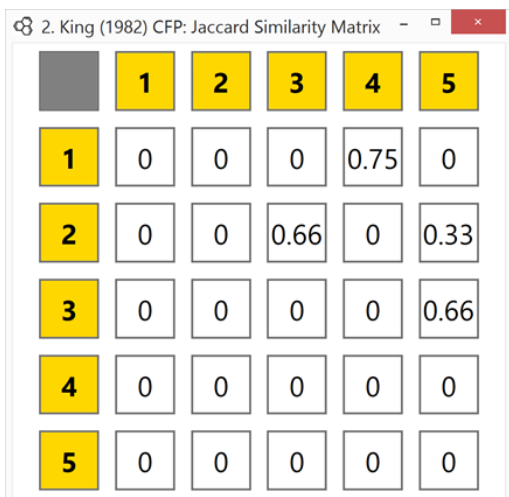


Figure 4-2: Jaccard similarity matrix for [106]—FactDesing

4.2.2 Sorenson similarity coefficient [104]

Sorenson SC (equation 4.2) is another general purpose SC that doesn't take 0-0 matches into account and doubles the weight of 1-1 matches at the same time.

$$C_{jk} = \frac{2 \cdot a}{2 \cdot a + b + c}, \qquad 0 \le C_{jk} \le 1 \quad (4.2)$$

where:

$j, k = 1, \ldots N$ – index for objects (machine/part types)

N – number of objects (machine/part types)

a – number of part/machine types which visit both machine /part types j and k

b – number of part/machine types which visit only machine /part type j

c – number of part/machine types which visit only machine /part type k

Figure 4-23 depicts the Jaccard similarity matrix for [106] dataset evaluated by the CMS design module of FactDesign.

2. King (1982) CFP: Sorenson Similarity M...

	1	2	3	4	5
1	0	0	0	0.85	0
2	0	0	0.8	0	0.5
3	0	0	0	0	0.8
4	0	0	0	0	0
5	0	0	0	0	0

Figure 4-3: Sorenson similarity matrix for [106]—FactDesign

4.2.3 Baroni-Urbani and Buser similarity coefficient [104]

Baroni-Urbani and Buser SC (eq.4.3) is a general purpose SC that was developed to improve data clustering of highly scattered matrices.

$$C_{ij} = \frac{a + \sqrt{a \cdot d}}{a + b + c + d + \sqrt{a \cdot d}}, \qquad 0 \leq C_{ij} \leq 1 \quad (4.3)$$

where:

$i, j = 1, \ldots M$ – index for machines

M – number of machines

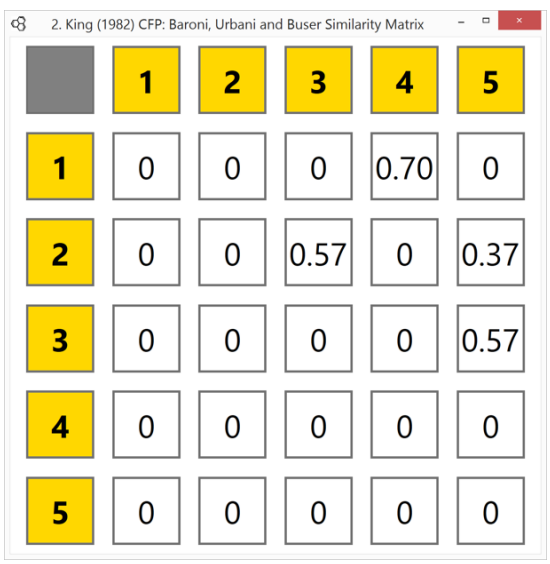
a – number of parts processed on both machines i and j

b – number of parts processed only on machine i

c – number of parts processed only on machine j

d – number of parts processed neither on machine i nor on machine j

Figure 4-24 depicts the Jaccard similarity matrix for [106] dataset evaluated by the CMS design module of FactDesign.



	1	2	3	4	5
1	0	0	0	0.70	0
2	0	0	0.57	0	0.37
3	0	0	0	0	0.57
4	0	0	0	0	0
5	0	0	0	0	0

Figure 4-4:Baroni-Urbani and Buser similarity matrix for [106]—FactDesing

4.2.4 Problem-oriented similarity coefficient [107]

Given SC (equation 4.4) was developed in order to overcome the bias when calculating similarities between two machines (parts) that process (require) different number of parts (machines).

$$MAXSC_{jk} = max\left[\frac{a}{a+b}, \frac{a}{a+c}\right], \qquad 0 \leq MAXSC_{jk} \leq 1 \quad (4.4)$$

where:

$j, k = 1, \ldots N$ – index for objects (machine/part types)

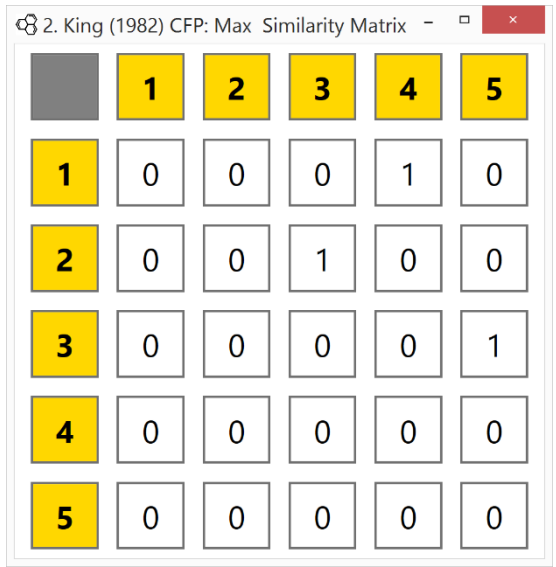
N – number of objects (machine/part types)

a – number of part/machine types which visit both machine /part types j and k

b – number of part/machine types which visit only machine /part type j

c – number of part/machine types which visit only machine /part type k

Figure 4-25 depicts the Shafer & Rodgers similarity matrix evaluated by the CMS design module of FactDesign.



2. King (1982) CFP: Max Similarity Matrix

	1	2	3	4	5
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	0	0
5	0	0	0	0	0

Figure 4-5: Shafer & Rodgers similarity coefficient—FactDesing

4.2.5 Production information based similarity coefficient [108]

Given SC (equation 4.5) takes into account similarities that parts have in terms of their operation sequences. The more similar the operation sequence, the higher the similarity.

$$S_{ik}(L) = \frac{1}{L} \cdot \left[S_{ik}(1) + \sum_{l=2}^{L} \frac{C_{ik}(l)}{N - l + 1} \right], \qquad 0 \leq S_{ik}(L) \leq 1, \qquad L \leq N \quad (4.5)$$

where:

L – length of common sequences between parts i and k

The higher L, the more discriminative power SC will have

$N = \min(N_i, N_k)$

$N_i, N_k -$ number of operations for part i or k

$i, k = 1, \ldots m -$ index for parts

$C_{ik}(l) -$ number of sequences of length l between parts i and $k, 2 \leq l \leq L$

$S_{ik}(1) -$ Jaccard SC

Following example [108] illustrates the application of given SC to CFP. Consider process plans for parts A, B and C (Figure 4-6).

Parts						
A	2	1	3	4	5	
B	1	3	2	1	4	5
C	1	3	2	4	5	

Figure 4-6: Process plans for given example

Since all five machines are being visited by all parts, Jaccard SC will be "1" for all pairs:

$$S_{AB}(1) = S_{AC}(1) = S_{BC}(1) = 1$$

Number of sequences of length $l = 2$ and $l = 3$ between parts A and B will be:

$C_{AB}(2) = 3 -$ for parts A and B there are 3 pairs with $l = 2$: $\{2; 1\}, \{1,3\}$ and $[[4,5]]$

$C_{AB}(3) = 0 -$ there are no common sequences for parts A and B with $l = 3$

Based on equation 4.5, SC of order $L = 3$ for parts A and B will be:

$$S_{AB}(3) = \frac{1}{3} \cdot \left[S_{AB}(1) + \frac{C_{AB}(2)}{5-2+1} + \frac{C_{AB}(3)}{5-3+1}\right] = \frac{1}{3} \cdot \left[1 + \frac{3}{4} + \frac{0}{3}\right] = \frac{7}{12}$$

Likewise, for parts pairs $A - C$ and $B - C$:

$C_{AC}(2) = 2 -$ for parts A and C there are 2 pairs with $l = 2$: $\{1; 3\}$and [[4,5]]

$C_{AC}(3) = 0 -$ there are no common sequences for parts A and C with $l = 3$

$$S_{AC}(3) = \frac{1}{3} \cdot \left[S_{AB}(1) + \frac{C_{AC}(2)}{5-2+1} + \frac{C_{AC}(3)}{5-3+1}\right] = \frac{1}{3} \cdot \left[1 + \frac{2}{4} + \frac{0}{3}\right] = \frac{1}{2}$$

$C_{BC}(2) = 3 -$ for parts B and C there are 3 pairs with $l = 2$: $\{1; 3\}$, [[3,2]] and [[4,5]]

$C_{BC}(3) = 1 -$ for parts B and C there is one pair with $l = 3$: $\{1; 3; 2\}$

$$S_{BC}(3) = \frac{1}{3} \cdot \left[S_{BC}(1) + \frac{C_{BC}(2)}{5-2+1} + \frac{C_{BC}(3)}{5-3+1}\right] = \frac{1}{3} \cdot \left[1 + \frac{3}{4} + \frac{1}{3}\right] = \frac{25}{36}$$

4.2.6 Production information based similarity coefficient [109]

Given SC (equation 4.6) incorporates such production parameters asoperation sequences, production volumes and operation times.

$$S_{ij} = \frac{\sum_{k=1}^{N}\left[X_k \cdot t_k^{ij} + \sum_{o=1}^{n_k} Z_{ko}\right] \cdot m_k}{\sum_{k=1}^{N}\left[X_k \cdot t_k^{ij} + \sum_{o=1}^{n_k} Z_{ko} + Y_k\right] \cdot m_k}, \qquad 0 \le S_{ij} \le 1 \quad (4.6)$$

where:

$i, j = 1, \dots M -$ index for machines

$M -$ number of machines

$N -$ number of part types

$m_k -$ planned production volume during a period for part type $k \; \forall k, k = 1, \dots N$

$n_k -$ number of times, part type k visits both machines in row

$$X_k = \begin{cases} 1, \text{ if part type } k \text{ visits both machines } i \text{ and } j \\ 0, \text{ otherwise} \end{cases}$$

$$Z_k = \begin{cases} 1, \text{ if part type } k \text{ visits both machines } i \text{ and } j \text{ in row} \\ 0, \text{ otherwise} \end{cases}$$

$t_k^{ij} -$ ratio of the total smaller unit operation time to the larger unit operation time

for machine pair ij, for part type k during visits to the machines i and j:

$$t_k^{ij} = \frac{min\left(\sum_{o=1}^{n_{ki}} t_{k,i}^o, \sum_{o=1}^{n_{kj}} t_{k,j}^o\right)}{max\left(\sum_{o=1}^{n_{ki}} t_{k,i}^o, \sum_{o=1}^{n_{kj}} t_{k,j}^o\right)} \qquad (4.6.1)$$

n_{ki} – number of trips part type k makes to machine i

n_{kj} – number of trips part type k makes to machine j

$t_{k,i}^o$ – unit operation time for part type k on machine i during oth visit

$t_{k,j}^o$ – unit operation time for part type k on machine j during oth visit

Following example [109] illustrates the application of given SC to CFP. Consider MPIM and production information (Figure 4-7).

Machines	Parts 1	Parts 2
A	1	0
B	1	1
C	0	1

Description	Parts 1	Parts 2
Production volume	10,000	10
Routing sequence	A-B	B-C-B
Unit operation time, *min*	50-45	01-10-05

Figure 4-7: Input data for a given example

Based on equations 4.6 and 4.6.1, SC between machines A and B is calculated as:

$$t_1^{AB} = \frac{min(\sum_{o=1}^{1} 50, \sum_{o=1}^{1} 45)}{max(\sum_{o=1}^{1} 50, \sum_{o=1}^{1} 45)} = \frac{45}{50}$$

$$t_2^{AB} = 0$$

$$S_{AB} = \frac{[X_1 \cdot t_1^{AB} + \sum_{o=1}^{n_1} Z_{1o}] \cdot m_1 + [X_2 \cdot t_2^{AB} + \sum_{o=1}^{n_2} Z_{2o}] \cdot m_2}{\left[X_1 \cdot t_1^{AB} + \sum_{o=1}^{n_1} Z_{1o} + Y_1\right] \cdot m_1 + \left[X_2 \cdot t_2^{AB} + \sum_{o=1}^{n_2} Z_{2o} + Y_2\right] \cdot m_2} =$$

$$= \frac{\left[1 \cdot \frac{45}{50} + 1\right] \cdot 10{,}000 + [0 \cdot 0 + 0] \cdot 10}{\left[1 \cdot \frac{45}{50} + 1 + 0\right] \cdot 10{,}000 + [0 \cdot 0 + 0 + 1] \cdot 10} = 0.999$$

Since machines A and C don't share any parts, $S_{AC} = 0$

Similarly, SC between machines B and C is calculated as:

$$t_1^{BC} = \frac{min(\sum_{o=1}^{1} 45, \sum_{o=1}^{0} 0)}{max(\sum_{o=1}^{1} 45, \sum_{o=1}^{0} 0)} = 1$$

$$t_2^{BC} = \frac{min(1+5, \sum_{o=1}^{1} 10)}{max(1+5, \sum_{o=1}^{1} 10)} = 1$$

$$S_{BC} = \frac{[X_1 \cdot t_1^{BC} + \sum_{o=1}^{n_1} Z_{1o}] \cdot m_1 + [X_2 \cdot t_2^{BC} + \sum_{o=1}^{n_2} Z_{2o}] \cdot m_2}{\left[X_1 \cdot t_1^{BC} + \sum_{o=1}^{n_1} Z_{1o} + Y_1\right] \cdot m_1 + \left[X_2 \cdot t_2^{BC} + \sum_{o=1}^{n_2} Z_{2o} + Y_2\right] \cdot m_2} =$$

$$= \frac{[0 \cdot 1 + 1] \cdot 10{,}000 + [1 \cdot 1 + 1 + 1] \cdot 10}{[0 \cdot 1 + 0 + 1] \cdot 10{,}000 + [1 \cdot 1 + 1 + 1 + 0] \cdot 10} = 0.003$$

4.2.7 Production information based similarity coefficient [110]

Given SC (equation 4.7) takes alternative process routes into consideration

$$S_{ij} = \frac{\sum_{k=1}^{N}\left[\sum_{r=1}^{r_k}(X_{kr} \cdot t_{kr} + n_{kr}) \cdot p_{kr}\right] \cdot m_k}{\sum_{k=1}^{N}\left[\sum_{r=1}^{r_k}(X_{kr} \cdot t_{kr} + n_{kr} + Y_{kr}) \cdot p_{kr}\right] \cdot m_k}, \qquad 0 \le S_{ij} \le 1 \quad (4.7)$$

where

$i, j = 1, \dots M -$ index for machines

$M -$ number of machines

$N -$ number of part types

$m_k -$ planned production volume during a period for part type $k \forall k, k = 1, \dots N$

$n_{kr} -$ number of trips, part type k makes between machines i and j for consecutive operations in the rth route

$r_k -$ number of alternative routes for part type k

$$X_{kr} = \begin{cases} 1, \text{if part type } k \text{ visits both machines } i \text{ and } j \\ 0, \text{otherwise} \end{cases}$$

$$Y_{kr} = \begin{cases} 1, \text{if part type } k \text{ visits either machine } i \text{ or } j \\ 0, \text{otherwise} \end{cases}$$

t_{kr} − ratio of the total smaller unit operation time to the larger unit operation time with machine pair i and j, for part type k in the rth route

$$t_{kr} = \frac{min\left(\sum_{o=1}^{n_{kri}} t_{kr,i}^{o}, \sum_{o=1}^{n_{krj}} t_{kr,j}^{o}\right)}{max\left(\sum_{o=1}^{n_{kri}} t_{kr,i}^{o}, \sum_{o=1}^{n_{krj}} t_{kr,j}^{o}\right)} \qquad (4.7.1)$$

n_{kri} − number of trips part type k makes to machine i in the rth route

n_{krj} − number of trips part type k makes to machine j in the rth route

$t_{kr,i}^{o}$ − unit operation time for part type k on machine i during oth visit

$t_{kr,j}^{o}$ − unit operation time for part type k on machine j during oth visit

Following example [110] illustrates the application of given SC to CFP. Consider four machine (A,B,C,D) and three part example with production data given in table 4.1. Each of three parts (1-3) has two feasible routes and a two-step operation sequence performed on two different machines. Fourth column on table 4.1 consists of alternative routing sequences with unit operation time $t_{kr,i}^{o}$ and $t_{kr,j}^{o}$. Based on equation 4.7.1 we get:

$$t_{1,1} = \frac{\min(4,10)}{\max(4,10)} = 0.4$$

All other t_{kr} values are calculated in the same way (see column #5, Table 4.1)

Table 4-1: Input data for a given example

Part number	Production volume, m_k	Utilization factor	Routing sequence, $M(t_{kr,i}^{o})$-$M(t_{kr,j}^{o})$	Ratio, t_{kr}
1	50	0.80	A(4)-B(10)	0.40
		0.20	A(4)-C(6)	0.67
2	5	0.25	C(6)-D(3)	0.50
		0.75	B(5)-D(3)	0.60
3	20	0.90	B(8)-A(10)	0.80
		0.10	D(6)-A(10)	0.60

Based on equation 4.7, SC between machines A and B is calculated as:

$$S_{AB} = \frac{\sum_{k=1}^{3}[(X_{k1} \cdot t_{k1} + n_{k1}) \cdot p_{k1} + (X_{k2} \cdot t_{k2} + n_{k2}) \cdot p_{k2}] \cdot m_k}{\sum_{k=1}^{3}[(X_{k1} \cdot t_{k1} + n_{k1} + Y_{k1}) \cdot p_{k1} + (X_{k2} \cdot t_{k2} + n_{k2} + Y_{k2}) \cdot p_{k2}] \cdot m_k} =$$

$$= \frac{(0.4 + 1) \cdot 0.8 \cdot 50 + (0.8 + 1) \cdot 0.9 \cdot 20}{(0.4 + 1 + 0) \cdot 0.8 \cdot 50 + 1 \cdot 0.2 \cdot 50 + 1 \cdot 0.75 \cdot 5 + (0.8 + 1 + 0) \cdot 0.9 \cdot 20} = 0.86$$

SCs for other pairs of machines are calculated in the same way:

$S_{AC} = 0.33; S_{AD} = 0.046; S_{BC} = 0; S_{BD} = 0.099; S_{CD} = 0.040$

4.2.8 Problem-oriented dissimilarity coefficient [111]

Given dissimilarity measure (equation 4.8) calculates the average number of new voids that are produced as a result of merging machines or groups of machines.

$$AVV_{ij} = \frac{\sum_{m=1}^{M_i}(vc_{im}^{j} - vc_{im})}{M_i} + \frac{\sum_{m=1}^{M_j}(vc_{jm}^{j} - vc_{jm})}{M_j}, \qquad AVV \geq 0 \quad (4.8)$$

where:

$i, j -$ index for machine groups

$m, n -$ index for machine belonging to i or j group respectively

$c_i(c_j) -$ ith(jth) machine group in the problem

$c_{im}(c_{jn}) -$ mth(nth) machine in the ith (jth) machine group

$vc_{im}(vc_{jn}) -$ number of voids caused by machine m (n) in ith(jth) machine group

$c_i^j = c_j^i -$ new machine group formed by c_i and c_j

$c_{im}^j(c_{jn}^i) -$ the mth (nth) machine of ith (jth) machine group in c_i^j (c_j^i)

$vc_{im}^j(vc_{jn}^i) -$ number of voids produced by mth (nth) machine

of ith(jth) machine group in the c_i^j (c_j^i)

$M_i(M_j) -$ number of machines in ith (jth)machine group

Following example [111] illustrates the application of given SC to CFP.

Consider MPIM provided on Figure 4-8:

Parts

Machines	1	2	3	4	5	6	7	8	9	
1					1	1		1		c_i
2		1		1		1	1			c_j
3	1		1		1				1	
4		1		1			1			
5			1		1				1	
6		1		1			1	1		
7	1		1		1				1	

$c_i^j = c_j^i$

$vc_{im}^j = 3 \quad vc_{im} = 0 \quad M_i = 1$

$vc_{jn}^i = 2 \quad vc_{jn} = 0 \quad M_j = 1$

Figure 4-8: MPIM for given example

Machine group c_i consists of m machines and machine group c_j consists of n machines.

Based on equation 4.8, average voids value for machine pair 1-2 will be:

$$AVV_{12} = \frac{3-0}{1} + \frac{2-0}{1} = 5$$

AVV values for other machine groups are calculated in the same manner. Results are presented in AVV matrix (Figure 4-9).

Machines

Machines	1	2	3	4	5	6	7
1	-	5	5	6	4	5	5
2		-	8	1	7	2	8
3			-	7	1	8	0
4				-	6	1	7
5					-	7	1
6						-	8
7							-

Figure 4-9: AVV matrix

Next, consider grouping 2-4 and 1-6 machine groups:

Parts

Machines	1	2	3	4	5	6	7	8	9	
3	1		1		1				1	
2		1		1		1	1			m=1
4		1		1			1			m=2
1					1	1		1		n=1
6		1		1			1	1		n=2
5			1		1				1	
7	1		1		1				1	

c_i c_j $c_i^j = c_j^i$

$vc_{11}^2 = 5 \quad vc_{11} = 1$

$vc_{12}^2 = 3 \quad vc_{12} = 0$

$M_i = 2$

$vc_{21}^1 = 7 \quad vc_{21} = 2$

$vc_{22}^1 = 5 \quad vc_{22} = 3$

$M_j = 2$

Figure 4-10: Modified MPIM for given example

Process of finding AVV value for 2-4 and 1-6 machine groups is illustrated in Figure 4-10. and based on equation 4.8 is calculated as follows:

$$AVV_{2-4\ 1-6} = \frac{(5-1)+(3-0)}{2} + \frac{(7-2)+(5-3)}{2} = 7$$

By merging groups in such a manner, machine groups are formed. By using the same method, part families can be formed as well.

4.2.9 Cell Bond strength similarity coefficient [112]

Cell Bond Strength (CBS) SC (equation 4.9) was designed to take part routing and operation time data into account.

$$CBS_{mn} = \frac{\sum_{j=1}^{P}(a_{mj} \cdot X)}{\sum_{j=1}^{P} a_{mj}} + \frac{\sum_{j=1}^{P}(a_{nj} \cdot X)}{\sum_{j=1}^{P} a_{nj}}, \qquad \mathrm{CBS_{mn}} \geq 0 \quad (4.9)$$

where:

$m, n = 1, \ldots M -$ index for machine types

$M -$ number of machine types

$j -$ index for part types $j = 1, \ldots P$

a_{mj} − production time on machine type m for part type j

a_{nj} − production time on machine type n for part type j

$$X = \begin{cases} 1, \text{ if part type } j \text{ visits both machines } m \text{ and } n \\ 0, \text{ otherwise} \end{cases}$$

Following example [112] illustrates the application of CBS SC to CFP. Consider MPIM with parts' production time information (Figure 4-11):

Machines \ Parts	1	2	3	4	5	6	7
1	0.5	1.5	2.5	1.0			
2		3.0		4.0			
3	2.0		1.5				
4					3.5	1.5	2.5
5				1.0		2.5	3.5

Figure 4-11: Machine-part matrix with part production time information

Based on equation 4.9, CBS for machine pair is calculated as:

$$CBS_{12} = \frac{1.5 + 1}{0.5 + 1.5 + 2.5 + 1} + \frac{3.0 + 4.0}{3.0 + 4.0} = 1.45$$

$$CBS_{13} = \frac{0.5 + 2.5}{0.5 + 1.5 + 2.5 + 1} + \frac{2.0 + 1.5}{2.0 + 1.5} = 1.55$$

By calculating CBS for other machine pairs in the same manner, similarity matrix will be as shown in Figure 4-12.

Machinea \ Machines	1	2	3	4	5
1	-	1.45	1.55	0.00	0.32
2		-	0.00	0.00	0.71
3			-	0.00	0.00
4				-	1.39
5					-

Figure 4-12: BS matrix

4.2.10 Batch similarity coefficient [113]

Batch Similarity Coefficient (BSC) (equation 4.10) incorporates the ratio of production volume and batch size of parts.

$$BS_{ij} = \frac{\sum_{k=1}^{n} \frac{V_k}{b_k} \cdot X_{ijk}}{\sum_{k=1}^{n} \frac{V_k}{b_k} \cdot Y_{ijk}}, \qquad BS_{ij} \geq 0 \quad (4.10)$$

where:

$n -$ number of part types

$V_k -$ production volume for part type k

$b_k -$ batch size for part type k

$$X_{ijk} = \begin{cases} 1, \text{ if part type } k \text{ visits both machine } i \text{ and } j \\ 0, \text{ otherwise} \end{cases}$$

$$Y_{ijk} = \begin{cases} 1, \text{ if part type } k \text{ visits either machine } i \text{ or machine } j \\ 0, \text{ otherwise} \end{cases}$$

4.2.11 Production information based SC [114]

Given SC (equation 4.11) incorporates alternative process routes, operation sequences, operation times and, production volumes of parts.

$$S_{ik} = \frac{N_{ik}}{N_i + N_k - N_{ik}} \cdot SR_{ik} \cdot MLR_{ik}, \qquad 0 \leq S_{ik} \leq 1 \quad (4.11)$$

where:

$i, k = 1, \ldots M -$ index for machine

$M -$ number of machines

$N_{ik} -$ number of common parts with multiple process plans

processed by both machines i and k

N_i − number of parts with multiple process plans processed by machine i only

N_k − number of parts with mult iple process plans processed by machine k only

$$\text{Sequence ratio}: SR_{ik} = \frac{X_{ik}}{D_{ik}} \quad (4.11.1)$$

X_{ik} − number of actual movements of parts between machines i and k

D_{ik} − number of possible movements of parts between machines i and k

$$\text{Machine load ratio}: MLR_{ik} = \frac{Y_{ik}}{E_{ik}} \quad (4.11.2)$$

Y_{ik} − minimum production volume factor between machines i and k

E_{ik} − maximum production volume factor between machines i and k

4.2.12 Problem-oriented SC [115]

Given SC (equation 4.12) is a binary data based SC that can take both negative and positive values and therefore identifies both similarity and dissimilarity)

$$s_{ij} = \sum_{k=1}^{n} \delta(a_{ik}, a_{jk}) \quad (4.12)$$

where:

$$(a_{ik}, a_{jk}) = \begin{cases} +2, \text{ if element } a_{ik} = a_{jk} = 1 \\ -1, \text{ if } a_{ik} \neq a_{jk} \\ 0, \text{ otherwise (i.e. } a_{ik} = a_{jk} = 0) \end{cases}$$

$a_{ik} = a_{jk} = 1$ − part k is processed on both machines i and j

$a_{ik} \neq a_{jk}$ − part k is processed on either machine i or j

n − number of parts in the system

4.3 Performance indicators

4.3.1 Machine utilization [116]

Machine utilization index is given by equation 4.13.

$$MU = \frac{N}{\sum_{k=1}^{Q} m_k \cdot p_k}, \qquad 0 \leq MU \leq 1 \quad (4.13)$$

where:

$k = 1, \ldots C -$ index for machine cell

$C -$ number of machine cells

$N -$ total number of non-exceptional elements

$Q -$ number of cells

$m_k -$ number of machines in cell k

$p_k -$ number of parts belonging to cell k

4.3.2 Machine utilization in a cell [117]

Machine utilization in a cell index is given by equation 4.14.

$$U_{jc} = \frac{\sum_{k=1}^{N}(P_{kjc} \cdot D_k)}{C_{jc}}, \qquad U_{jc} \geq 0 \quad (4.14)$$

where:

$k\ (j)(c) = 1, \ldots P\ (M)\ (C) -$ index for part (machine)(cell)

$P\ (M)(C) -$ number of parts (machines)(cells)

$N -$ total number of parts being processed on machine j in cell c

$P_{kjc} -$ processing time of part k on machine j in cell c

$D_k -$ demand of part k that is being processed on machine j in cell c

$C_{jc} -$ capacity of machine j in cell c

4.3.3 Machine uptime [118]

Machine uptime index is given by equation 4.15.

$$UTR_n = \frac{MTBF_n}{MTBF_n + MTTR_n}, \qquad 0 \leq UTR_n \leq 1 \quad (4.15)$$

where:

$n = 1, \dots M$ − index for machine types

M − number of machine types

$MTBF_n$ − mean time between failures of a machine type n

$MTTR_n$ − mean time to repair a machine of type n

4.3.4 Cell utilization [117]

Cell utilization index is given by equation 4.16.

$$C_k = \frac{\sum_i U_i}{N_k}, \qquad C_k \geq 0 \quad (4.16)$$

where:

$k = 1, \dots C$ − index for machine cells

C − number of machine cells

U_i − machine utilization in cell k

N_k − number of machines in cell k

4.3.5 Cell utilization (with respect to part type *k* only) [119]

Cell utilization index with respect to part type k only is given by equation 4.17.

$$CU = \frac{1}{k_j} \cdot \sum_{l=1}^{c} \sum_{k=1}^{k_j} y_{jl} \cdot z_{jkl}, \qquad CU \geq 0 \quad (4.17)$$

where:

$k\ (j)(l) = 1, \ldots O\ (P)(C) -$ index for operation (part)(cell)

$O\ (P)(C) -$ number of operations (parts)(cells)

k_j – number of operations scheduled to be performed on part j

$$y_{jl} = \begin{cases} 1, \text{ if part } j \text{ is associated to cell } l \\ 0, \text{ otherwise} \end{cases}$$

$$z_{jkl} = \begin{cases} 1, \text{ if part} j's\ k\text{th operation is performed in cell } l \\ 0, \text{ otherwise} \end{cases}$$

4.3.6 Average cell utilization [117]

Average cell utilization index is given by equation 4.18.

$$ACU = \frac{\sum_k C_k}{N_c}, \qquad ACU \geq 0 \quad (4.18)$$

where:

$k = 1, \ldots C -$ index for machine cells

$C -$ number of machine cells

$C_k -$ cell utilization (see eq.4.16)

$N_c -$ number of cells

4.3.7 System utilization [120]

System utilization index is given by equation 4.19

$$U_k = \frac{P_k}{P_{TFk}}, \qquad U_k \geq 0 \ (4.19)$$

where:

$k = 1, \ldots K -$ index for time period

$P_k -$ maximum number of parts that can be processed by the considered CMS

during a time period k

$P_{TFk} -$ maximum number of parts that could be processed by the correspondent

totally flexible CMS during a time period k

4.3.8 Grouping efficiency [121]

Grouping efficiency performance measure is given by equation 4.20.

$$\eta = q \cdot \eta_1 + (1 - q) \cdot \eta_2, \qquad 0 \leq \eta \leq 1 \quad (4.20)$$

$$\eta_1 = \frac{e_d}{\sum_{r=1}^{k} M_r \cdot N_r} \quad (4.20.1)$$

$$\eta_2 = 1 - \left(\frac{e_0}{mn - \sum_{r=1}^{k} M_r \cdot N_r} \right) \quad (4.20.2)$$

$$q = \frac{\sum_{r=1}^{k} M_r \cdot N_r}{mn} \quad (4.20.3)$$

where:

$e_d -$ total number of 1s in the diagonal blocks

$e_0 -$ total number of 0s in the off-diagonal blocks

$k -$ limiting number of groups

$M_r -$ number of machines in cell r

$N_r -$ number of parts belonging to cell r

$mn -$ size of part-machine matrix

$q - weighting\ factor\ (value\ of\ 0.5\ is\ commonly\ used).$

[122] suggested that q can be selected based on the size and number of cells.

4.3.9 Weighted grouping efficiency [123]

Weighted grouping efficiency performance measure is given by equation 4.21.

$$\eta_q = \frac{q \cdot e_1 + (1-q) \cdot e_v}{e_1 + e_v} - \frac{(1-q) \cdot e_0}{e_1 + e_0} \quad (4.21)$$

where:

$q -$ weighting factor for exceptional and block-diagonal elements

$e_1 -$ number of 1s in the diagonal blocks

$e_v -$ number of voids in the diagonal blocks

$e_0 -$ number of exceptional (off-diagonal) elements

4.3.10 Grouping efficacy [124]

Grouping efficacy performance measure is given by equation 4.22.

$$\Gamma = \frac{1 - \psi}{1 + \phi} \quad (4.22)$$

where:

$$\psi = \frac{Number\ of\ exceptional\ elements}{Total\ number\ of\ operations}$$

$$\phi = \frac{Number\ of\ voids\ in\ the\ diagonal\ blocks}{Total\ number\ of\ operations}$$

4.3.11 Weighted grouping efficacy [125]

Weighted grouping efficacy performance measure is given by equation 4.23.

$$\gamma = \frac{q \cdot (e - e_0)}{q \cdot (e + e_v - e_0) + (1 - q) \cdot e_0} \quad (4.23)$$

where:

$q -$ weighting factor for exceptional and block-diagonal elements

$e -$ total number of 1s in the part-machine incidence matrix

$e_0 -$ number of exceptional (off-diagonal) elements

e_v − number of voids in the diagonal blocks

4.4 Summary

In this chapter, we have introduced a wide set of similarity coefficients and performance indicators. All the similarity indices reported in this chapter have been implemented in FactDesign-CFP module. Some examples from the literature have been utilized to validate the embedded algorithms. Results for large scale problems have been reported as well.

Chapter 5 : Solving the CFP using progressive modeling

5.1 Introduction

Progressive Modeling [126] is a new modeling approach developed at the outset to address the ever-growing complex system optimization problems. The new approach handles optimization problems from a systems perspective. The new approach has an iterative cycle. The problem at hand is decomposed into several integrating components connected using "a component model." Those components interact with each other using a well-defined set of interfaces that are supposed to be strictly defined all over the life cycle of a progressive model.

Despite the plethora of the CFP solution approaches, a practical tool for cellular factory is still lacking [32]. Most models try to simplify the problem or address sub-set of its variables in order to successfully solve the developed model. Lack of practicality of most models in theacademic literature was one of the major reasons that inspired the creation of Progressive Modeling at its early stages. The existing approaches offer very little room for the designer to interact and intervene during the design process or modify the solution to meet the real-world needs. In this chapter, we will demonstrate the first stage of an ongoing project for integrated cellular manufacturing design. The PM approach will be demonstrated in detail while working on the CFP.

5.2 Progressive Modeling process

Progressive modeling is a new modeling methodology for analyzing, formulating, modeling, and solving industrial problems form systems perspective, especially the complex ones. A complex problem is either a large scale or a large scope or both. PM integrates several principles from Operations Research, Optimization Metaheuristics, Software Engineering, and Object-oriented Systems Analysis and Design. PM process is an iterative process composed of a double-loop iterative process as shown in Figure 5-1.

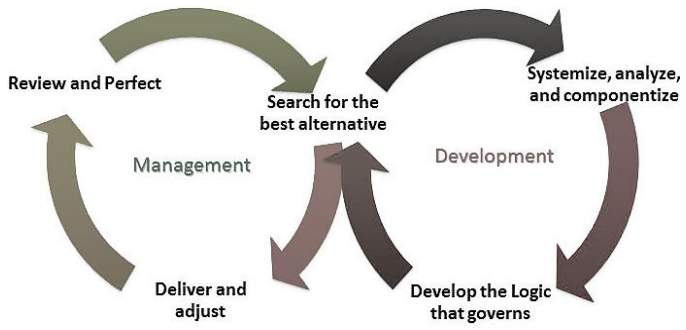


Figure 5-1: PM Process

Step 1: Systemize, componentize, and analyze: The Progressive Modeling process starts wtihanalyzing the problem at hand and decomposing it into several interacting components. Component models culminate this step. Component model demonstrates the separation of concerns of the interacting components in a black-box communication fashion.

Step 2: Develop the logic that governs: In order to be well-understood, controlled, or managed, systems behavior should be modeled. If this behavior can be described in a sophisticated way by a group of governing equations, a mathematical model can be

defined. Operations research defines decision variables, constraints, and objectives as the basic building blocks of math models. PM separates the decision space, from the objective space and defines a new term called model space. Decision variables or search neighborhood define(s) the search space.Solution objective or objectives define the objective space. A problem model that can progressively updateddefine the “model space”. With time, models could be further developed to reflect a better understanding of the underlying problem or underlying system changes or adaptations. Step 2, is problem specific and is concerned with the real world large-scope complex systems. CM design is considered by default a large-scope problem that could be a large scale one as well.

Step 3: Search for the best alternative: this step is concerned with solution algorithm development. A Progressive algorithm is a compilation of algorithms that aim to explore both the search space where different alternatives are found and the objective space where all these alternatives are evaluated according to a certain selection criteria (one or more objectives). Solutions are represented using object-oriented design principles. Problem data are easily accessed by soliton objects. Unlike traditional metaheuristics, e.g. GAs, SA, and others, the search process is an informed one. Several moves are createdand only the best one is selected. Encoding and decoding operations are either minimized or eliminated entirely. Objective updating might happened just locally. The last two strategies can drastically affect the solution time needed to get similar quality solutions obtained by other metaheuristics.

Step 4: Deliver and Adjust: All the complexities of the modeling process are hidden from the end solution users. FactDesign is a modular software platform that is being built to host and demonstrate progressive models and their solution solutions.

Step 5: Review and Adapt: Unlike traditional models, the results of PM are there to change, to adapt, to co-evolve with the underlying systems; the objective is not only to optimize the underlying system behavior by exploring all the possible alternatives under study but also our understanding or modeling of that system. Developing better analytics, better logic, and better algorithms is the constant goal of PM.

5.3 CFP Progressive Model

No matter how the system data is selected a CFPsolution is universal: part families and machine cells. Unlike the traditional block matrix, we introduce a new network diagram to represent a factory designsolution. Machines are represented as nodes while parts are represented as arc data. Hovering over any machine, part(s) or edge displays a data-rich graphical object that highlights the design requirements determined by problem data model.

5.3.1 Component model

The component model of the CFP is shown in Figure 5-2. Once the solution process is triggered using the GUI solve command, the modeler starts taking over the control role. The modeler acts as the broker among all other framework components. The molder access the problem data, initialize the search process, and explore the search space to find better candidates. The Modeler also creates a decoupled objective space where candidates are evaluated and utilized to identify the next move(s). The objective space deals with the problem solutionin a black-box fashion. The search space has its own controller that determines how solutions are represented and how search space is explored. The objective space has its own controller that determines the next move(s), keep track of previous solutions, or best solutions etc. Objective space controller and algorithms know nothing about how the problem is represented or what kind of problem is being solved. Such a kind of modular architecture and black boxing enable us to improve our algorithms locally and enable algorithm reuse; that is why our modeling approach is called Progressive Modeling.

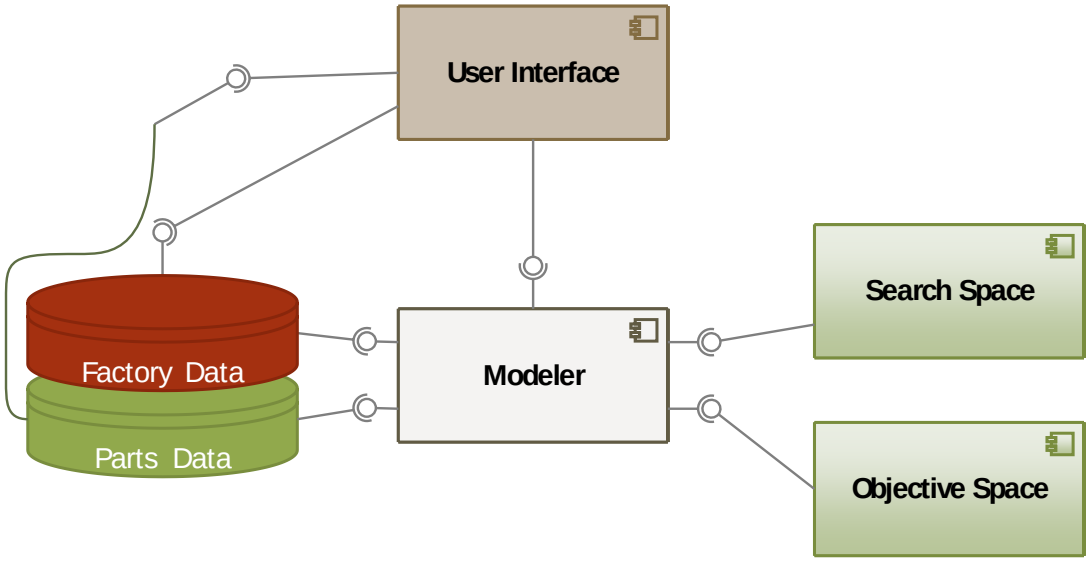


Figure 5-2: Progressive Modeling: A simplified block diagram

5.3.2 Solution representation (Factory Graphs)

A CFP could be represented as a factory graph which needs to be sorted out in a way that minimizes the intersection of its edges. *Figure 5-3* shows a solution for the 5-machine factory [106]. Machines are represented as nodes, and the traffic is represented with edges; multiple edges could connect several machines. Products are represented with dark-green circles when they are entering the system and with light-green ones when they are exiting. In a computer memory, a problem solution is a graph *G(V, E, C)* where *V* stands for vertices, *E* stands for edges, and *C* stands for Cells.

Our framework is still in its first phase of development,and thatis why we will discuss the simplest case of the CFP, the one with the simple incidence matrix only. We use the binary matrix to create the factory graph and the corresponding process plans. We assume the process plan happens in the same order they appear in the precedence matrix.

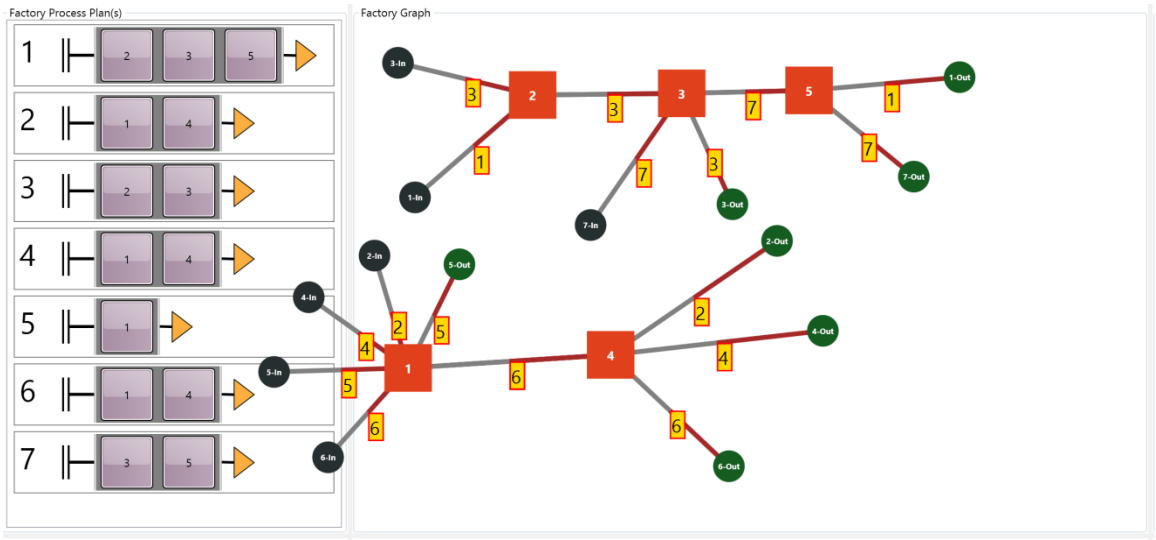


Figure 5-3: 5-machine 7-parts problem [106]

5.3.3 Solution Algorithm

5.3.3.1 Initialization

Since the cellular factory is represented as a graph, (see *Figure 5-3*) no encoding or decoding is required. We assume that there are upper and lower bounds on possible cell size. Using those bounds, we formulate the first cell using a seed machine. Once a seed machine is identified the next machine will attract the most similar one using any similarity measure, Jaccard'sfor example. Once the cell exceeds its bound, a new cell is created until all machines are assigned.

5.3.3.2 Exploring the search space

Alternative solutions could be generated as follows:

1. Any cell with at least the double number of the lower bond per cell could be split without constrain violations.

2. Any current cell could be chosen randomly andmost dissimilar machine is selected to migrate. The destination cell will be the most similar cell to this machine. Ties are broken arbitrarily. upper and lower bounds should be checked first guarantee solution feasibility

3. Any two cells could be merged as long the resulting cell size does not violated the cell size upper bound constraint

4. Operator number 2 could be done blindly (no similarity or dissimilarity are required).

5.3.3.3 Exploring the objective space

The objective space controller deals with the problem solution in a complete black-box fashion. We are borrowing some partial algorithms from well-known metaheuristics to do this job. In this study, we imported the roulette wheel random selection algorithm from GAs to select the candidate solutions to create the next neighborhood that should undergo further exploration at the search space.

5.3.3.4 The master solution algorithm

The master algorithm is a population-based algorithm that starts with an initial population and is terminated after a certain number of iterations. The algorithm is described in Listing 5-1. The input parameters for this algorithm are the input pop size, the max number of iterations, the factorydata,and the objectives. Any component could be updated now independently from other counter parts without violating the master algorithm. Ifwe want to use the partial swarm selection algorithm instead of GAs selection algorithm the update will be just limited to the objective space controller. Developing our own controller is left for future reach projects.

Listing 5-1: Master Progressive Algorithm

Step 1:	Use the GUI to intact with the user, or read the problem data from an Excel file (User interface)
Step 2:	Request fromthe modeler to initialize the solution process
Step 3:	The modeler Requests the search space controller to create the initial neighborhood.

Listing 5-1 (cont.): Master Progressive Algorithm

Step 4a:	For i=1 to max number of iterations
Step 4b:	-Request from the objective space controller to identify the best candidates to undergo the exploration process. -Keep a record of the best solution developed so far.
Step 4c:	-Request from the search space controller to explore the current neighborhood, identify the informed next moves, and update the current neighborhood.
Step 4d:	Next i
Step 5:	Report the results and the solution statistics

5.4 Experimental study and results

In order to test our CFP module, its proposed model, and solution algorithm, we used a set of well-defined problems in the CFP literature. Best well-know or optimum solutions for most of those problems have been found during the very early generations. For small and mid-size problems, the optimum solution was even hit during the initialization phase, thanks to the informed initialization and neighborhood search algorithms. See Figures 5-4 to 5-7 for sample solutions.

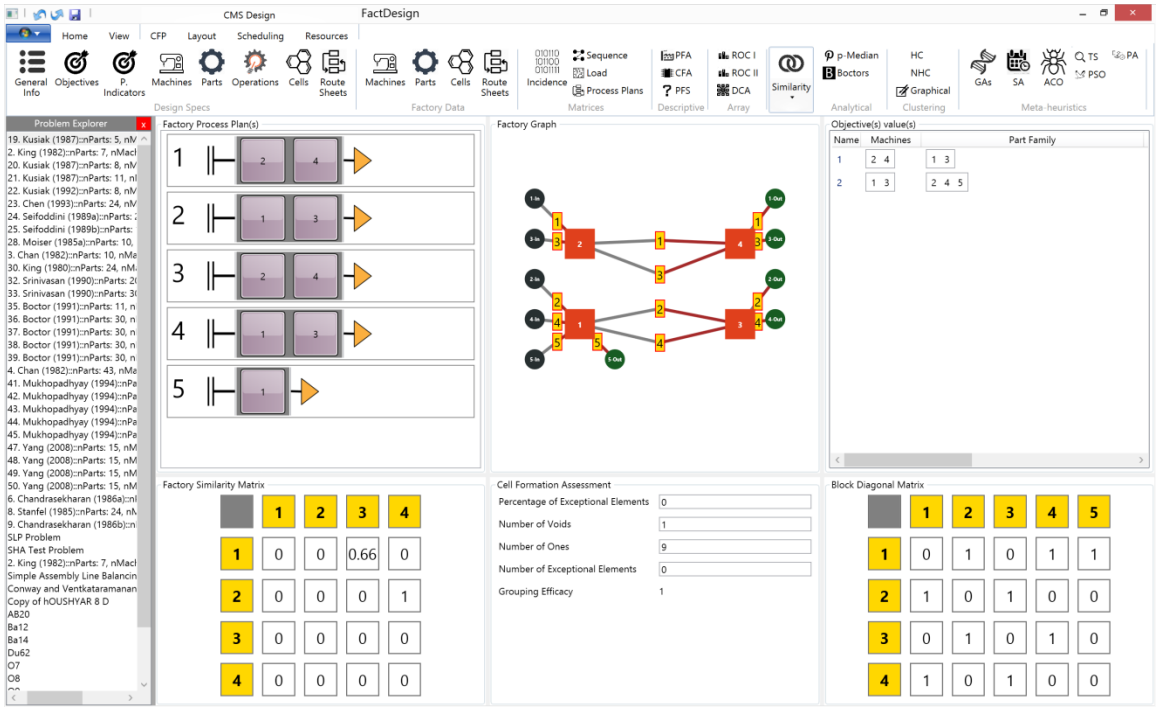


Figure 5-4: FactDesign solution for [127]

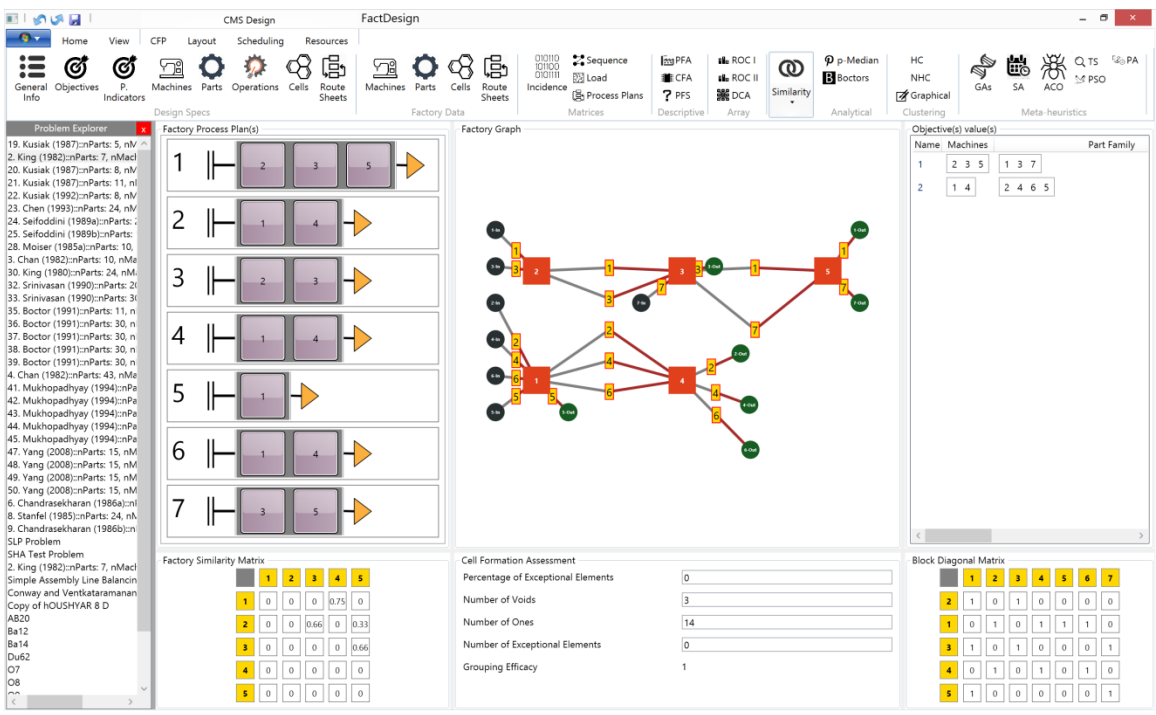


Figure 5-5: FactDesign solution for [106]

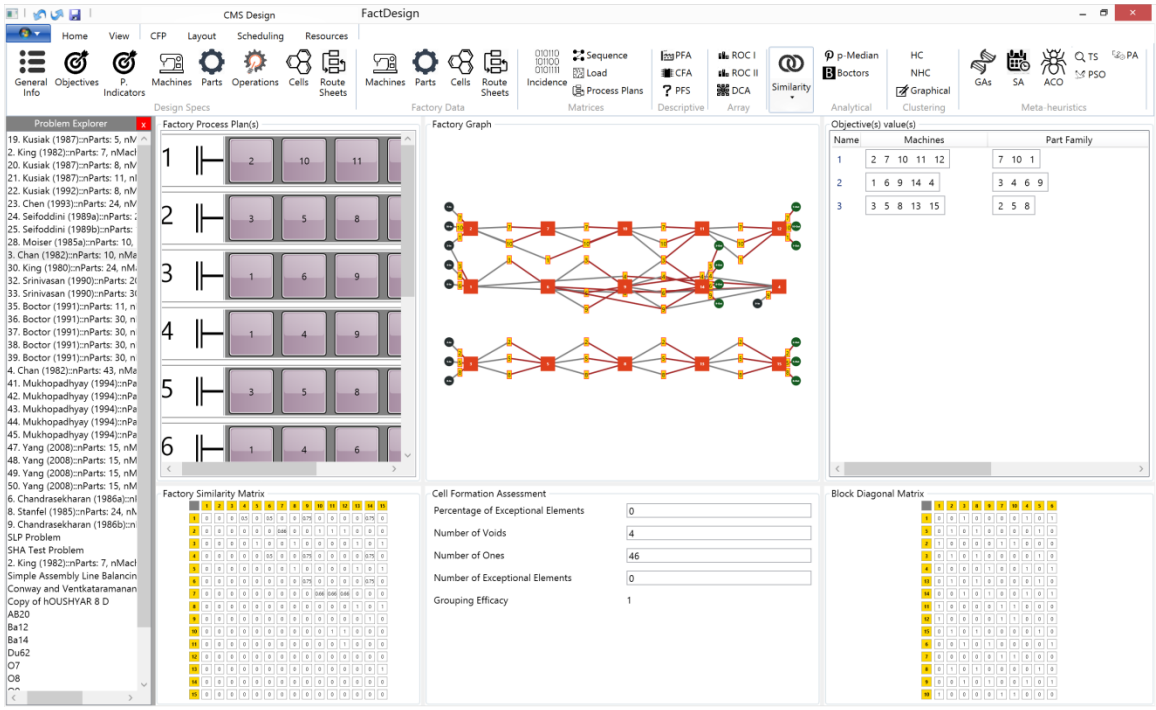


Figure 5-6: FactDesign solution for [128]

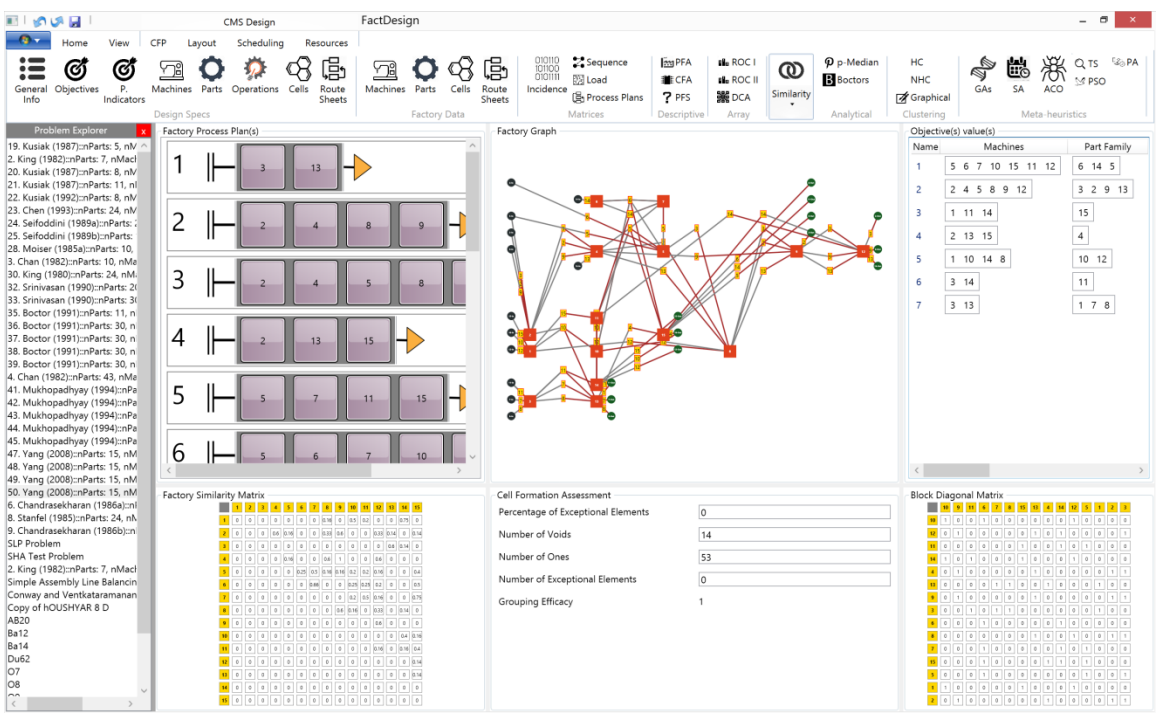


Figure 5-7: FactDesign solution for [129]

5.5 Summary

In this chapter, a new modeling approach called Progressive Modeling has been presented and applied to the cell formation problem. The problem has been analyzed from a systems-perspective,and a software framework has been developed to capture the data, implement the proposed solution algorithm, and display the results. A component model has been presented to demonstrate the problem definition, modeling and solution have been deployed over several interacting components. A novel solution representation has been introduced. The new presentation eliminates the encoding and decoding operations of other meta-heuristics and created an informed neighborhood search process. The developed model and framework can both be easily maintained and updated on an ongoing basis.

Chapter 6 : Summary and conclusions

Owing to the advantages that CMS brings to a medium-volume medium-variety manufacturing, there is a growing interest in such systems among researchers and practitioners. A major step in CMS design is CFP– a complex multi-aspect optimization problem, successful solution of which provides a solid ground for further design and development of CMS.

The presented literature review covers a multitude of approaches ranging from simple SC-based to sophisticated multi-criteria metaheuristics. Nonetheless, our findings show that the vast majority of proposed approaches present academic rather than practical interest. In this regard, a novel CMS design framework and a cell formation method using Progressive Modeling were proposed in this thesis.

CFP module of FactDesign software with focus on inputs and outputs of the module has been developed and described in detail. Developed module allows to capture numerous machines's, parts' and cells' attributes as well as variety of objectives. Route sheets and process plan viewer provide quick and convenient access to parts' routing data. Results can be presentedin form of a block diagonal matrix as well as factory graph that can be manipulated to obtain a desired factory network. CF assessment portal evaluates grouping based on grouping efficiency measure.

A number of Similarity Coefficients and performance indices have been presented and implemented in FactDesign CMS design module. Examples from the literature have been used to validate the implemented algorithms.

Principles of Progressive Modeling as well as solution algorithm were presented in chapter 5. In short, progressive modeling is an iterative modeling process that breaks down the system into several interacting components and thereby considers a problem at hand from the systems perspective. PM's modular design allows adapting to changes in problem formulations and design priorities. Solutions are represented using object-oriented design principles which eliminates encoding and decoding of solutions. Moreover, search space is explored through informed search processand the local objective update is performed to avoid computational redundancy. A set of well-defined problems from literature was used to test the proposed approach and best known or optimum solutions for most of those problems have been found during the early generations of the solution process.

References:

1. Djassemi, M., *A simulation analysis of factors influencing the flexibility of cellular manufacturing.* Int. J. Prod. Res., 2005. **43**(10): p. 2101-2111.
2. Askin, R.G., & Standridge, C. R., *Modeling and analysis of manufacturing systems*. 1993, New York, NY: Wiley & Sons.
3. Groover, M.P., *Fundamentals of modern manufacturing: materials, processes and systems, 4th edition*. 2010, Hoboken, NJ: Wiley & Sons.
4. Kusiak, A., *The generalized group technology concept.* Int. J. Prod. Res., 1987. **25**(4): p. 561-569.
5. Zhang, Y., *Group Technology: Manufacturing Cell Formation Incorporating Routing Flexibilities with Alternative Machines to Minimize the Material Handling Cost*. 2010: University of Houston.
6. Mitrofanov, S.P., *Scientific Principles of Group Technology:(Nauchnye Osnovy Gruppovoĭ Tekhnologii)*. 1966: National Lending Library for Science and Technology.
7. Burbidge, J.L., *The introduction of group technology*. 1975, New York, NY: Wiley & Sons.
8. Heragu, S.S., *Facilities design*. 2008, Boca Raton, FL: CRC Press.
9. Wemmerlov, U., & Hyer, N. M. L., *Cellular manufacturing in the U.S. industry: a survey of users.* Int. J. Prod. Res., 1989. **27**(9): p. 1511-1530.
10. Wemmerlov, U., & Johnson, D. J., *Cellular manufacturing at 46 user plants: Implementation experiences and performance improvements.* Int. J. Prod. Res., 1997. **35**(1): p. 29-49.
11. Sassani, F., *A simulation study on performance improvement of group technology cells.* Int. J. Prod. Res., 1990. **28**(2): p. 293-300.
12. Kannan, V.R., & Ghosh, S., *A virtual cellular manufacturing approach to batch production.* Decision Sciences, 1996. **27**(3): p. 519-539.
13. Wemmerlov, U., & Hyer, N. M. L., *Procedures for the part family/machine group identification problem in cellular manufacturing.* Journal of Operations Management, 1986. **6**(2): p. 125-147.
14. Wang, J., *A linear assignment clustering algorithm based on the least similar cluster representatives.* IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 1999. **29**(1): p. 100-104.
15. Garey, M.R., & Johnson, D. S., *Computers and intractability: A guide to the theory of NP-completeness.* 1979, San Francisco, CA: W. H. Freeman.
16. Hodiya, A.L., *A mathematical model and a simulated annealing algorithm for an integrated facility layout and cell formation*, 2015, University of Guelph: Guelph, ON, Canada.
17. Tunnukij, T.H., C., *An enhanced grouping genetic algorithm for solving the cell formation problem.* Int. J. Prod. Res., 2009. **47**(7): p. 1989-2007.

18. Vin, E., & Delchambre, A., *Generalized cell formation: iterative versus simultaneous resolution with grouping genetic algorithm.* Int. J. Prod. Res., 2014. **25**: p. 1113-1124.
19. Choobineh, F., *A framework for the design of cellular manufacturing systems.* Int. J. Prod. Res., 1988. **26**(7): p. 1161-1172.
20. Logendran, R., Ramakrishna, P., & Sriskandarajah, C., *Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plans.* Int. J. Prod. Res., 1994. **32**(2): p. 273-297.
21. Aljaber, N.J., *New approaches to cell formation problems in cellular manufacturing systems*, 1999, Mississippi State university: Mississippi State, MS, USA.
22. Caux, C., Bruniaux, R., & Pierreval, H., *Cell formation with alternative process plans and machine capacity constrains: A new combined approach.* Int. J. Prod. Res., 2000. **64**: p. 279-284.
23. Baykasoglu, A., Nabil, N. Z. G., & Richard, C. C., *Capability based formulation and solution of multiple objective cell formation problems using simulated annealing* Integrated Manufacturing Systems, 2001. **12**(4): p. 258-274.
24. Adenso-Diaz, B., Lozano, S., Racero, J., & Fernando, G., *Machine cell formation in generalized group technology.* Computers & Industrial Engineering, 2001. **41**: p. 227-240.
25. Vin, E., Delchambre, A., & Francq, P. *A generalized cell formation problem solved by an adapted GGA.* in *8th International Conference of Modeling and Simulation*. 2010. Hammamet, Tunisia.
26. Rheault, M., Drolet, J. R., & Abdulnour, G., *Physically reconfigurable virtual cells: A dynamic model for a highly dynamic environment.* Computers & Industrial Engineering, 1995. **29**(1-4): p. 221-225.
27. Lokesh, K.S., & Jain, P. K., *Dynamic cellular manufacturing systems design - a comprehensive model.* International Journal of Advanced Manufacturing Technology, 2011. **53**(11): p. 11-34.
28. Vakharia, A.J., & Wemmerlov, U., *Designing a cellular manufacturing system: A materials flow approach based on operation sequences.* IEEE Transactions, 1990. **22**(1): p. 84-97.
29. Jayswal, S., & Adil, G. K., *Efficient algorithm for cell formation with sequence data, machine replications and alternative process routings.* Int. J. Prod. Res., 2004. **42**(12): p. 2419-2433.
30. Adil, G.K., & Rajamani, D., *The trade-off between intracell and intercell moves in group technology cell formation.* Journal of Manufacturing Systems, 2000. **19**(5): p. 305-317.
31. Hu, L., & Yasuda, K., *Minimising material handling cost in cell formation with alternative processing routes by grouping genetic algorithm.* Int. J. Prod. Res., 2006. **44**(11): p. 2133-2167.
32. Askin, R. G., *Contributions to the design and analysis of cellular manufacturing systems*. International Journal of Production Research, 2013. **51**(23-24): p.6778-6787.

33. Vitanov, V., Tjahjono, B., & Marghalany, I., *Heuristic rules-based logic cell formation algorithm.* Int. J. Prod. Res., 2008. **46**(2): p. 321-344.
34. Joines, J.A., King, R. E., & Culbreth, C. T., *A comprehensive review of production-oriented manufacturing cell formation techniques*, 1996.
35. Mansouri, S.A., Moattar Husseini, S. M., & Newman, S. T. , *A review of the modern approaches to multi-criteria cell design.* Int. J. Prod. Res., 2000. **38**(5): p. 1201-1218.
36. Papaioannou, G., & Wilson J. M., *The evolution of cell formation problem methodologies based on recent studies (1997-2008): Review and directions for future research.* European Journal of Operational Research, 2010. **206**: p. 509-521.
37. Patel, J.N., & Patel, S. V., *Approaches to solve cell formation, machine layout and cell layout problem: A review.* Transactions on Machine Learning and Artificial Intelligence, 2014. **2**(5): p. 80-96.
38. Yin, Y., & Yasuda, K., *Similarity coefficient methods applied to the cell formation problem: A taxonomy and review.* International Journal of Production Economics, 2006. **101**: p. 329–352.
39. McAuley, J., *Machine grouping for efficient production.* Production engineer, 1972. **51**(2): p. 53-57.
40. Romesburg, C., *Cluster analysis for researchers*. 2004, Morrisville, NC: Lulu Press.
41. Yasuda, K., & Yin, Y., *A dissimilarity measure for solving the cell formation problem in cellular manufacturing.* Computers & Industrial Engineering, 2001. **39**: p. 1-17.
42. Shafer, S.M., & Rogers, D. F., *Similarity and distance measures for cellular manufacturing. Part II. An extension and comparison.* Int. J. Prod. Res., 1993. **31**(6): p. 1315-1326.
43. Viswanathan, S., *A new approach for solving P-median problem in group technology.* Int. J. Prod. Res., 1996. **34**(10): p. 2691-2700.
44. DeWitte, J., *The use of similarity coefficients in production flow analysis.* Int. J. Prod. Res., 1980. **18**: p. 503-514.
45. Steudel, H.J., & Ballakur, A., *A dynamic programming based heuristic for machine grouping in manufacturing cell formation.* Computers & Industrial Engineering, 1987. **12**(3): p. 215-222.
46. Selvam, R.P., & Balasubramanian, K. N., *Algorithmic grouping of operation sequences.* Engineering Costs and Production Economics, 1985. **9**: p. 125-134.
47. Gupta, T., *Design of manufacturing cells for flexible environment considering alternative routeing.* Int. J. Prod. Res., 1993. **31**(6): p. 1259-1273.
48. Won, Y., & Kim, S., *Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings.* Computers & Industrial Engineering, 1997. **32**(1): p. 207-220.
49. Won, Y., *New p-median approach to the cell formation problem with alternative process plans.* Int. J. Prod. Res., 2000. **38**(1): p. 229-240.

50. Seifoddini, H., & Tjahjana, B., *Part-family formation for cellular manufacturing: a case study at Harnischfeger*. Int. J. Prod. Res., 1999. **37**(14): p. 3263-3273.
51. Gupta, T., & Seifoddini, H., *Production data based similarity coefficient for machine-component grouping decisions in the design of cellular manufacturing system.* Int. J. Prod. Res., 1990. **28**(7): p. 1247-1269.
52. Prabhakaran, G., Janakiraman, T. N., & Sachithanandam, M., *Manufacturing data-based combined dissimilarity coefficient for machine cell formation.* International Journal of Advanced Manufacturing Technology, 2002. **19**: p. 889-897.
53. Talbi, E., *Metaheuristics: From design to implementation*. 2009, Hoboken, NJ: Wiley & Sons.
54. Liefoghe, A., Jourdan, L., & Talbi, E., *A software framework based on a conceptual unified model for evolutionary multiobjective optimization: ParadisEO-MOEO*. International Journal of Operational Research, 2011. **209**: p. 104-112.
55. Glover, F., *Future paths for integer programming and links to artificial intelligence.* Computers & Operations Research, 1986. **34**(5): p. 533-549.
56. Wu, T.H., Low, C., & Wu, W. T., *A tabu search approach to the cell formation problem.* International Journal of Advanced Manufacturing Technology 2004. **23**: p. 916-924.
57. Cao, D., & Chen, M., *Using penalty function and tabu search to solve cell formation problems with fixed cell cost.* Computers & Operations Research, 2004. **31**: p. 21-37.
58. Sun, D., Li, L., & Batta, R., *Cell formation using tabu search.* Computers & Industrial Engineering, 1995. **28**(3): p. 485-494.
59. Lei, D., & Wu, Z., *Tabu search for multiple-criteria manufacturing cell design.* International Journal of Advanced Manufacturing Technology, 2006. **28**: p. 950-956.
60. Wu, T.H., Chen, J. F., & Yeh, J. Y., *A decomposition approach to the cell formation problem with alternative process plans.* International Journal of Advanced Manufacturing Technology, 2004. **24**: p. 834-840.
61. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller A. H., & Teller E., *Equation of state calculation by fast computing machines.* The Journal of Chemical Physics, 1953. **21**.
62. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P., *Optimization by simulated annealing.* Science, 1983. **220**(4598): p. 671-680.
63. Wu, T.H., Chung, S. H., & Chang, C. C., *Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings.* Expert Systems with Applications, 2009. **36**: p. 3652–3661.
64. Lin, S.W., Ying, K. C., & Lee, Z. J., *Part-machine cell formation in group technology using a simulated annealing-based meta-heuristic.* Int. J. Prod. Res., 2010. **48**(12): p. 3579-3591.
65. Holland, J.H., *Adaptation in natural and artificial systems*. 1975, Ann Arbor, MI: University of Michigan Press.

66. Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning* 1989: Addison-Wesley Professional;.
67. El-Mihoub, T.A., Hopgood, A. A., Nolle, L., & Battersby, A., *Hybrid genetic algorithms.* Engineering Letters, 2006. **13**(2): p. 124-137.
68. Zolfaghari, S., & Liang , M., *Comprehensive machine cell/part family formation using genetic algorithms.* Journal of manufacturing Technology Management, 2004. **15**(6): p. 433-444.
69. Kumar C. S., C., M. P., *Grouping efficacy: a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology.* Int. J. Prod. Res., 1990. **28**(2): p. 233-243.
70. Paydar, M.M., & Saidi-Mehrabad, M., *A hybrid genetic-variable neighborhood search algorithm for the cell formation problem based on grouping efficacy.* Computers & Operations Research, 2013. **40**: p. 980-990.
71. Shiyas, C.R., & Pillai, V. M., *Cellular manufacturing system design using grouping efficacy-based genetic algorithm.* Int. J. Prod. Res., 2014. **52**(12): p. 3504-3517.
72. Falkenauer, E., *Genetic algorithms for grouping problems* 1998, New York, NY: Wiley & Sons.
73. Brown, E.C., & Sumichrast, R. T., *CF-GGA: a grouping genetic algorithm for the cell formation problem.* Int. J. Prod. Res., 2001. **39**(16): p. 3651-3669.
74. James, T.L., Brown, E. C., & Keeling, K. B., *A hybrid grouping genetic algorithm for the cell formation problem.* Computers & Operations Research, 2007. **34**: p. 2059-2079.
75. Tunnukij, T., & Hicks, C., *An enhanced grouping genetic algorithm for solving the cell formation problem.* Int. J. Prod. Res., 2009. **47**(7): p. 1989-2007.
76. Kennedy, J., & Eberhart, R. C. *Particle swarm optimization*. in *IEEE international conference on neural networks*. 1995. Perth, Australia.
77. Kashan, A.H., Karimi, B., & Noktehdan, A., *A novel discrete particle swarm optimization algorithm for the manufacturing cell formation problem.* International Journal of Advanced Manufacturing Technology, 2014. **73**: p. 1543-1556.
78. Anvari, M., Mehrabad, M. S., & Barzinpour F., *Machine-part cell formation using a hybrid particle swarm optimization.* International Journal of Advanced manufacturing Technology, 2010. **47**: p. 745-754.
79. Wu, T.H., Chang, C. C., & Yeh, J. Y., *A hybrid heuristic algorithm adopting both Boltzmann function and mutation operator for manufacturing cell formation problems.* International Journal of Production Economics, 2009. **120**: p. 669–688.
80. Kao, Y., & Chen, C., *A differential evolution fuzzy clustering approach to machine cell formation.* International Journal of Advanced Manufacturing Technology, 2013. **65**: p. 1247-1259.
81. Dimopoulos, C., *Explicit consideration of multiple objectives in cellular manufacturing.* Engineering Optimization, 2007. **39**(5): p. 551-556.
82. Collette, Y., & Siarry, P., *Multi-objective optimization: principles and case studies*. 2003, New York, NY: Springer.

83. Shaller, J., *Tabu search procedures for the cell formation problem with intra-cell transfer costs as a function of cell size.* Computers & Industrial Engineering, 2005. **49**: p. 449–462.
84. Chung, S.H., Wu, T. H., & Chang, C. C., *An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations.* Computers and Industrial Engineering, 2011. **60**: p. 7-15.
85. Wu, T.H., Chang, C. C., & Chung, S. H., *A simulated annealing algorithm for manufacturing cell formation problems.* Expert Systems with Applications, 2008. **34**: p. 1609–1617.
86. Defersha, F.M., & Chen, M., *A parallel multiple Markov chain simulated annealing for multi-period manufacturing cell formation problems.* International Journal of Advanced Manufacturing Technology, 2008. **37**: p. 140-156.
87. Tavakkoli-Moghaddam, R., Safaei, N., & Sassani, F., *A new solution for a dynamic cell formation problem with alternative routing and machine costs using simulated annealing.* Journal of the Operational Research Society, 2008. **59**(4): p. 443-454.
88. Jayakumar, V., & Raju, R., *A simulated annealing algorithm for machine cell formation under uncertain production requirements.* Arabian Journal for Science and Engineering, 2014. **39**: p. 7345-7354.
89. Venugopal, V., & Narendran, T. T., *A genetic algorithm approach to the machine-component grouping problem with multiple objectives.* Computers & Industrial Engineering, 1992. **22**(4): p. 469-480.
90. Yasuda, K., Hu, L., & Yin, Y., *A grouping genetic algorithm for the multi-objective cell formation problem*. Int. J. Prod. Res., 2005. **43**(4): p. 829-853.
91. Mansouri, S.A., Moattar-Husseini, S. M., & Zegordi, S. H., *A genetic algorithm for multiple objective dealing with exceptional elements in cellular manufacturing.* Production Planning & Control, 2003. **14**(5): p. 437-446.
92. Srinivas, N., & Deb, K., *Multiobjective optimization using nondominated sorting in genetic algorithms.* Evolutionary Computation, 1994. **2**(3): p. 221–248.
93. Solimanpur, M., Vrat, P., & Shankar, R., *A multi-objective genetic algorithm approach to the design of cellular manufacturing systems.* Int. J. Prod. Res., 2004. **42**(7): p. 1419-1441.
94. Dimopoulos, C.M., N., *A hierarchical clustering methodology based on genetic programming for the solution of simple cell-formation problems.* Int. J. Prod. Res., 2001. **39**(1): p. 1-19.
95. Neto, A.R.P., & Filho, E. V. G., *A simulation-based evolutionary multiobjective approach to manufacturing cell formation.* Computers & Industrial Engineering, 2010. **59**: p. 64-74.
96. Saeidi, S., Solimanpur, M., & Mahdavi, I., *A multi-objective genetic algorithm for solving cell formation problem using a fuzzy goal programming approach.* International Journal of Advanced Manufacturing Technology, 2014. **70**: p. 1635-1652.

97. Muruganandam, A., Prabhaharan, G., Asokan, P., & Baskaran, V., *A memetic algorithm approach to the cell formation problem.* International Journal of Advanced Manufacturing Technology, 2005. **25**: p. 988–997.
98. Tavakkoli-Moghaddam, R., Ranjbar-Bourani, M., Amin, G. R., & Siadat, A., *A cell formation problem considering machine utilization and alternative process routes by scatter search.* Journal of Intelligent Manufacturing, 2012. **23**: p. 1127-1139.
99. Glover, F., *Heuristics for integer programming using surrogate constraints.* Decision Sciences, 1977. **8**(1): p. 156-166.
100. Wang, X., Tang, J., & Yung, K., *Optimization of the multi-objective dynamic cell formation problem using a scatter search approach.* International Journal of Advanced Manufacturing Technology, 2009. **44**: p. 318-329.
101. Bajestani, M.A., Rabbani, M., Rahimi-Vahed, A. R., & Khoshkhou, G. B., *A multi-objective scatter search for a dynamic cell formation problem.* Computers & Operations Research, 2009. **36**: p. 777-794.
102. Solimanpur, M., Vrat, P., & Shankar, R. *A multi-objective genetic algorithm approach to the design of cellular manufacturing systems*. International Journal of Production Research, 2004. **42**(7): p. 1419-1441.
103. Seifoddini, H. *A note on similarity coefficient method and the problem of improper machine assignment in group technology applications.* International Journal of Production Research, 1989. **27**(7): p. 1161-1165.
104. Romesburg C. H. *Cluster analysis for researchers.* Lifetime learning publications, 2004. Lulu Press, Releigh, North Carolina, USA.
105. McAuley, J. *Machine grouping for efficient production*. The Production Engineer, 1972. **51**(2): p. 53-57.
106. Hadley, S. W. *Finding part-machine families using graph-partitioning techniques.* International Journal of Production Research, 1996. **34**(7): p. 1821-1839.
107. Shafer, S. M., & Rogers, D. F. *Similarity and distance measures for cellular manufacturing. Part II. An extension and comparison.* International Journal of Production Research, 1993. **31**(6): p. 1315-1326.
108. Choobineh, F. *A framework for the design of cellular manufacturing systems*. International Journal of Production Research, 1988. **26**(7): p. 1161-1172.
109. Gupta, T., & Seifoddini, H. *Production data based similarity coefficient for machine-component grouping decisions in the design of cellular manufacturing systems.* International Journal of Production Research, 1990. **28**(7): p. 1247-1269.
110. Gupta, T. *Design of manufacturing cells for flexible environment considering alternative routeing.* International Journal of Production Research, 1993. **31**(6): p. 1259-1273.
111. Yasuda, K., & Yin, Y. *A dissimilarity measure for solving the cell formation problem in cellular manufacturing*. Computers & Industrial Engineering, 2001. **39**: p. 1-17.
112. Steudel, H. J., & Ballakur, A. *A dynamic programming based heuristic for machine grouping in manufacturing cell formation.* Computers & Industrial Engineering, 1987. **12**(3): p. 215-222.

113. Seifoddini, H., & Tjahjana, B. *Part-family formation for cellular manufacturing: a case study at Harnishfeger*. International Journal of Production Research, 1999. **37**(14): p. 3263-3273.
114. Yin, Y., & Yasuda, K. *Manufacturing cells' design in consideration of various production factors*. International Journal of Production Research, 2002. **40**(4): p. 885-906.
115. Viswanathan, S. *A new approach for solving the P-median problem in group technology*. International Journal of Production Research, 1996. **34**(10): p. 2691-2700.
116. Adil, J. K., & Rajamani, D. *The trade-off between intracell and intercell moves in group technology cell formation*. Journal of manufacturing Systems, 2000. **19**(5): p. 305-317.
117. Wei, J. C., & Gaither, N. *A capacity constrained multiobjective cell formation method*. Journal of manufacturing Systems, 1990. **9**(3): p. 222-232.
118. Beaulieu, A., Gharbi, A., & Ait-Kadi. *An algorithm for the cell formation and the machine selection problems in the design of a cellular manufacturing system*. International Journal of Production Research, 1997. **35**(7): p. 1857-1874.
119. Logendran, R. *A binary integer programming approach for simultaneous machine-part grouping in cellular manufacturing systems*. Computers & Industrial Engineering, 1993. **24**(3): p. 329-336.
120. Albino, V., & Caravelli, A. C. *Limited flexibility in cellular manufacturing systems: A simulation study*. International Journal of Production Economics, 1999. **60**(61): p. 447-455.
121. Chandrasekharan, M. P., & Rajagopalan, M. *MODROC: an extension of rank order clustering for group technology*. International Journal of Production Research, 1986. **24**(5): 1221-1233.
122. Kumar K. R., Kusiak, A., & Vanelli, A. *Grouping of parts and components in flexible manufacturing systems*. European Journal of Operational Research, 1986. **24**: p. 387-397.
123. SarkerB. R., & Khan, M. *A comparison of existing grouping efficiency measures and a new weighted grouping efficincy measure*. IIE Transactions, 2001. **33**: p. 11-27.
124. Kumar, C. S., & Chandrasekharan, M. P. *Grouping efficacy: a quantitative criterion fro goodness of block diagonal forms of binary matrices in group technology*. International Journal of Production Research, 1990. **28**(2): 233-243.
125. Ng, S. M. *Worst-case analysis of an algorithm for cellular manufacturing*. European Journal of Operational Research, 1993. **69**: 384-398.
126. Ismail, M. *Progressive Modeling: the Process, the Principles, and the Applications*. Submitted to Conference on Systems Engineering Research (CSER'13). March 20th, 2013. Georgia Institute of Technology, Atlanta, GA: Elsevier
127. Kusiak, A., & Chow, W. S. *Efficient solving of group technology problem*. Journal of Manufacturing Systems, 1987. **6**(2): p. 117-124.

128. Chan, H. M., & Milner, D. A. *Direct clustering algorithm for group formation in cellular manufacture*. Journal of Manufacturing Systems, 1982. **1**(1): p. 65-75.
129. Yang, M., & Yang, J. *Machine-part cell formation in group technology using a modified ART1 method*. European Journal of Operational Research, 2008. **188**: p. 140-152.