Cellular Manufacturing through Composite Part Formation: A Genetic Algorithm Approach

Nirjhar Roy and V.R. Komma
Department of Mechanical Engineering
Motilal Nehru National Institute of Technology
Allahabad (U.P.), India

Abstract

Maintenance of cell efficiency in Cellular Manufacturing System is important. Once the cell is formed, new parts come naturally for processing. These parts may not be completely processed in a single manufacturing group and need different machines for processing. This paper presents a procedure for allocating new parts to machine groups using the concept of composite parts. Initially for each of the part families, the hypothetical composite parts are prepared from historical data and coded in terms of alphabetical string. The method starts with identifying the specific part family from the drawing of the new part. Primitive libraries for surface elements are prepared from historic data for identification of processes of the new components. Combinations of these surface elements make the features of the parts. The strings of all new coming components are compared with hypothetical composite parts that represents the part family. The optimization criterion for proposed method is taken as minimization of summation of Levenshtein distance between composite strings and the set of new part strings that reach for processing in the dedicated manufacturing groups. The model developed by the authors can be utilized to guide planners to operate the manufacturing cells efficiently.

Keywords
Group Technology, Cellular Manufacturing, Composite Part, Levenshtein Distance, Genetic Algorithm

1. Introduction
Group Technology is a manufacturing philosophy which attend to the concept that similar things should be kept together (Burbidge (1975)). Economically, the realization that many problems are similar and that, by grouping similar problems, a single solution can be found to a set of problems saves money, time and effort. The traditional approach to this type of manufacture is to make use of a functional layout in the factory, i.e. similar machines are grouped according to type. As a result of this form of machine layout, the workpiece itself must travel a considerable distance around the workshop before all the operations are performed on it. This usually leads to a long throughput time. Further, the processing sequence being random, there is bound to be a lot of backtracking in the shop floor. The types of parts being huge production planning function becomes complex. As a result, the concept of group technology is being utilized instead of implementing functional layout. Technically, the factory, previously a large job shop is partitioned into smaller job shops (cells or machine groups) in such a way that each cell is equipped with all the machines and equipment needed to completely process a set of dedicated parts that require similar processing type. It has been found that by switching to this type of cellular manufacturing system (CMS), many benefits of flow line production can be attained in a batch production system.

The basic concept behind Group Technology is attributed to the Russians, who carried out initial investigations during the 1930s. The early work stressed the importance of industrial classification and initial applications were limited to the medium and large batch productions. The work was extended during the war years by Mitrofanov (1966) to include workpieces produced in small batches. His major publication on Group Technology first appeared in 1959 and was translated into English in 1966. Mitrofanov proposed that it was possible to produce a theoretical composite part which incorporated all the major features of parts belonging to a family, and that a machine could be tooled up to produce the composite part, thus providing the set-ups required for each part in the family. Formation of composite part is one approach to implement the concept of CMS. When numerous parts need to be manufactured, it is very difficult to plan production as per processing requirements of individual parts. This approach could be used to iron out the problems by merging the primitives into a single hypothetical part. The manufacturing facility could be planned on the basis of composite part to facilitate economical production.

In the early 1960s, Opitz carried out an investigation into workpiece statistics, which showed that although firms manufacture a variety of products, the spectrum of them all was remarkably similar. Based on the findings of this investigation, he established a classification system, which enabled parts to be codified by means of their geometrical similarity. A number of methods for classification and coding were being investigated at approximately the same time. A significant growth in the interest and application of Group Technology followed the publication of Opitz's work.

Askin and Zhou (1998) have described a methodology of cell formation in the situation where operations sequence is considered for cell formation. They have devised a similarity co-efficient, which is based on operations sequence and attempts to establish that a composite sequence for a part family can be obtained by iteratively adding one new part type. Pandey and Roy (2002) have described a procedure in which features of a part is identified from the two dimensional drawing using a list of primitive library.Roy(2005) has described a GA based cell formation technology in which sequence of processes are considered. Garbiey et al (2005) have considered the situation when the cell is tested for its efficiency when a new part is introduced in the already formed cell. They have computed various efficiency measures and considered a combined procedure as algorithm. Sarker and Xu (2000) described a clustering procedure based on this similarity coefficient which can be utilised to maximize the total in-sequence flow in a part family. An operation sequence-based similarity coefficient of part family is developed by them. Many research methodologies of cell formation have been developed and published. For an extensive review of literature please refer Papaioannou and Wilson (2010).

Most of the cell formation models make simple assumptions and ignore many manufacturing factors. These models consider machining operations of parts, and the manufacturing system is represented by a binary machine-part incidence matrix (MPIM) A, with the following convention:

A_{mn} = 1 if part m is processed on machine n
= 0 otherwise

The aim of the model is to manipulate the machine-part incidence matrix shown in Table 1 so that machine groups and the corresponding dedicated part families may be formed as shown in Table 2. Table 1 is the input matrix and Table 2 is the resultant matrix showing machine groups and dedicated part-families with one exceptional element as shown. Many of the algorithms in cell formation use machine-part incidence matrix.

Table 1: Machine-part incidence matrix

Machines / Parts	1	2	3	4
1	1	0	1	0
2	0	1	1	1
3	1	1	1	0
4	0	1	0	1

Table 2: Diagonalized matrix

Machines / Parts	1	3	2	4
1	1	1	0	0
3	1	1	1	0
2	0	1	1	1
4	0	0	1	1

In the machine-part incidence matrix '1' shows the incidence i.e. if the machine processes the part else it will be shown blank or '0'. For example, part 1 is processed by machines 1 and 3; part 4 is processed by machines 2 and 4. The matrix is now diagonalized as shown in Table 2. The part 2 cannot be fully processed in one single machine group and the machine 3 is called bottleneck machine. The objective of the cell formation algorithms is minimizing these bottleneck machines since they represent inter-cell movements.

The clustered matrix in Table 2 shows a particular machine-part relationship with a bottleneck machine. It is seen that although two distinct sub-matrices exist, 'nevertheless, part 2 hold relationship with both cells. It needs machines of both machine groups for its complete processing. However, if machine 3 is eliminated or duplicated in other cell, the condition for ideal cell formation is satisfied. Machine 3 is, therefore, a bottleneck machine. Existence of bottleneck machine creates disturbances in system as the machine loading becomes complex.

2. Composite Part

Mitrofanov (1959) and Edwards (1970) have proposed composite part approach to implement the concept of cellular manufacturing. A composite part is formed by merging the primitives of all the parts of a part family. Thus, the composite is a single hypothetical part that can be completely processed in a manufacturing cell/group. If a new part is loaded in a machine group, the degree of dissimilarity of the part from its related part family or the hypothetical composite should have minimum deviation and desired to be zero. The manufacturing facility could be planned on the basis of composite part to facilitate economical production. The primitives of three parts shown in Fig. 1 are merged into composite part as shown in Figure 2 by incorporating all the primitives of the three parts.

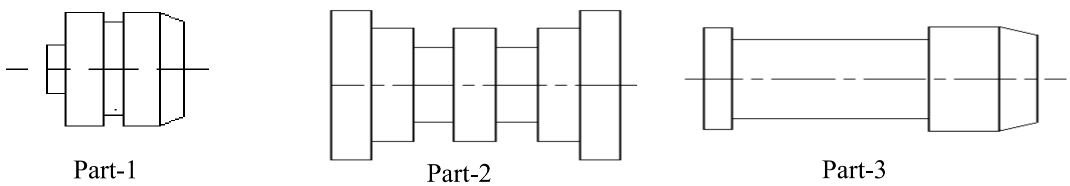

Figure 1: Individual parts

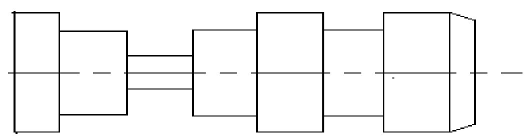

Figure 2: Composite part for the parts shown in Figure 1

Sometimes, it may not be judicious to merge all the primitives of parts due to various production considerations, as in that situation the shop will converge back to a large job shop and all the benefits of CMS will be lost. The size of the manufacturing group depends on initial capital investment capacity, machines available and outsourcing facilities. Therefore, individual parts features (in terms of primitives) could be merged in the composite part based on their repetitions in the parts. The primitives having more repetitions will be more eligible candidates for merging in the composite part. Various techniques could be used for selection of optimum primitives for merging in composite parts. The genetic algorithm is proved to be one of the effective techniques.

3. Primitive Library and Processing of New Parts

Finding the process requirements of any part is the pre-requisite of creating the part-families and subsequently the corresponding machine groups are formed. There are many researches in this direction. Liu (2004) has described a methodology for feature extraction and classification of rotational parts. In realistic situation, where 2D representation is utilized in the form of drawing, a methodology is need to be developed to identify the process required and the sequence of these processes.

To look into this aspect, a study was conducted in a tractor manufacturing shop. A sample of 110 rotational parts were collected. The data is analyzed and put up in a tabular form in Table 3 below. The frequency of various operations are shown in the Table 3 for simulating the occurrences of processes in a new parts which might come for processing in the shop and needed to be fitted in a part family.

A library can be constructed more elaborately by considering other alphabetical codes to various possible similar operations. For example, the turning operation can be further divided in other operations like step-up turning and step-down turning.

This library is user defined as any code could be used for any of the primitives. Primitives could be added and subtracted as per the requirement of any manufacturing facility. This provides flexibility to implement the proposed method from large to small-scale industry. One modification in this aspect was carried out in the study and a more flexible system of identifying the primitives were utilized as suggested by Pandey and Roy (2002) and explained below. The library could be used to provide alphabetical coding to any of the part feature by decomposing it into the number of primitives. This set of strings is used as input in the proposed method to generate optimal composite part string.

To cite an illustration, the reader may refer Fig. 3, in which the feature primitives of a part are shown. The primitives are in reference to machining surfaces. Thus, a feasible minimum set of primitives will generate a feature of the part. Figure 4 shows the surface elements or feature of a part. Thus, if a 2D object or drawing is considered as shown in Figure 5, the processing of the parts can be settled from its alternate process plans in view of processing the part in a single cell.

Table 3: Primitives and probability of occurrence of processes

S. No.	Name of Operation (Primitive)	Letter Code	Frequency (F_j)	% Probability of Occurrence, P_j
1	Reaming	a	6	1.049
2	Shaping	b	8	1.399
3	Spline Cutting	c	7	1.224
4	Marking	e	13	2.273
5	Counter-Sinking	f	12	2.098
6	Chamfering	k	23	4.021
7	Grooving	l	21	3.671
8	Grinding	n	23	4.021
9	Slotting	o	26	4.545
10	Boring	p	29	5.070
11	Taper Turning	q	39	6.818
12	Knurling	r	24	4.196
13	Threading	s	41	7.168
14	Drilling	u	82	14.336
15	Step Turning	v	76	13.287
16	Turning	x	142	24.825
		Total	572	100

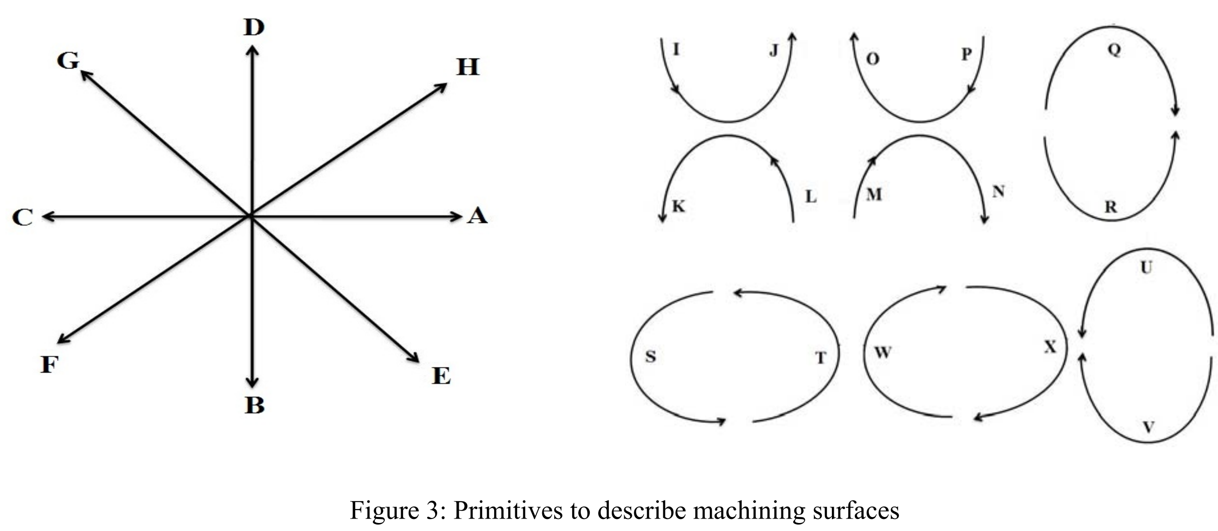

Figure 3: Primitives to describe machining surfaces

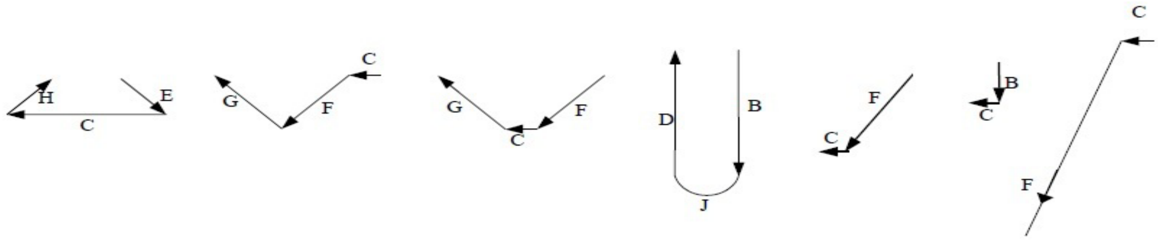

Figure 4: Surface features made out of primitives, on the basis of primitives as shown in Figure 3

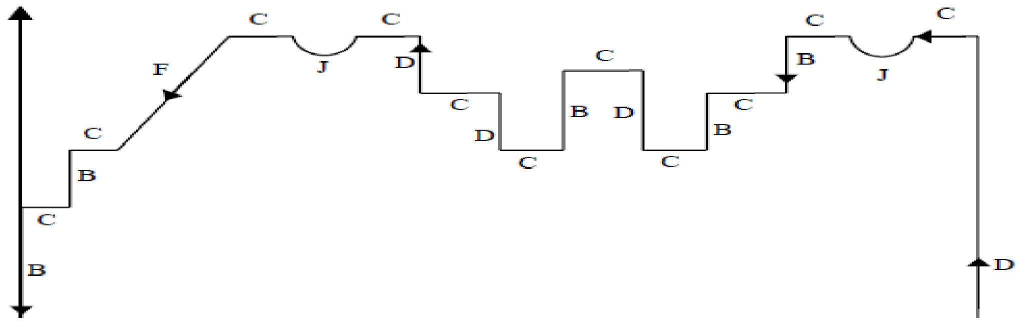

Figure 5: Drawing of the upper half of a rotational part surface

The representation of the diagram of Fig. 5 in terms of primitives is "DCJCBCBCDCBCDCDCJCFCBCB". The features and corresponding machining can be obtained by analysing the diagram (drawing of the part). Features can be derived using a suitable algorithm as given below. The algorithm is applied the rotational part having symmetric surfaces. With some modifications, the algorithm may be applied for non-rotational parts also.

4. Basic Assumptions

The following assumptions are made in this work:

1. The data solely represents the machine requirements on which operations are performed to produce any part.
2. The time needed for production is not considered.
3. The type of operation is not considered. For example, if a machine is utilised for performing many types of operations, no separate code is provided for each of the operations. However, the software can be modified for coding each operations separately.
4. Service departments like heat treatment, painting, quality control etc., are not considered for grouping, as they are always shared.
5. Workload for each type of machine has also not been considered initially for the obvious reason that in a dynamic condition the same cannot be correctly predicted. However the modification in the model has been made subsequently.
6. Alternate processing may be considered.

Algorithm for Feature Identification

Features are essentially generated by machining operations. Operations required for a part ultimately identify the part family. The algorithm below shows a working procedure for the same.

Step 1: (a) Create the profile view of the part. (It may be the initial drawing).
(b) Assign a unique identification number corresponding to each surface. The first surface being the left hand end of the part. Labeling will begin from the surface one and will continue in a anti-clockwise manner. Only upper half of the drawing is considered due to symmetry.

Step 2: Represent the part by creating a pattern string using the pattern primitives obtained from the scheme shown in the Fig. 3. Identify if any match is obtained with the features. Some examples of the features are cited in Fig 4.

Step 3: Create a new features if the same is obtainable from the pattern string of the part and is not included in the feature database.
Step 4: Retrieve the features in chronological order.

The algorithm is kept simple but may be applied in job-shops processing a huge number of parts. It is obvious that the number of rotational parts is very large in terms of percentages of the total parts processed. For assigning a part in the part family, the minimum distance of the part for all the composite parts is selected. A measure known as Lavenshtien distance is used in this work.

5. Levenshtein Distance

Levenshtein Distance (LD) measures the dissimilarity between any two parts represented in alphabetical strings. The distance between two strings, 1 and 2, is defined as the minimum numbers of transformations required to derive string 2 from string 1. Three types of transformations are accepted, viz., deletion, insertion, and substitution. For example,

String-1	String-2	Transformation	Levenshtein Distance
abcd	ab	2 deletions	2
abcd	abcdef	2 insertions	2
abcd	efgh	4 substitutions	4

Different weights can also be assigned separately to each type of the transformations for calculation of LD. However, in this work equal weight is assigned to the transformations. The minimization of LD is taken as optimization criterion in the proposed method. Modification of the computing methodologies are reported in Haldar and Mukhapadhyay (2011) for Optical Character Readers implementation in which different weights are assigned to characters. Hanov (2011) has described two efficient and simple algorithms based on the trie data structure that uses the digits in the keys to organize and search the dictionary. Tam (1990) has presented in his paper, a similarity coefficient based on the similarity of operation sequence. Further the use of similarity coefficient for part grouping has been discussed. They have also illustrated that such a coefficient, augmented with an advanced clustering algorithm, can improve production effectiveness by identifying part families that allow machine to interleave between identical operations of different parts.

Roy and Singh (1995) have proposed a heuristic methodology for machine part cell formation utilizing linguistic theory. In this paper, the processing requirements of a set of parts are expressed in terms of alphabetical strings. Each of the alphabets has been represented to denote machine/workcenter. Distance measures are calculated for each pair of strings to measure the dissimilarity between them; a part-part distance matrix is resulted where diagonal elements are all 0's and other elements indicated dissimilarity. Then a heuristic has been proposed to group parts into a set of parts families. The number of clusters is varied and for each set (a) sum of within group distances and (b) between group distances have been calculated. Tanaka (1986) has analyzed theoretical aspect of syntactic pattern recognition. In his paper, a pessimistic view of syntactic pattern recognition has been developed. This paper describes several aspects of syntactic pattern recognition from various points of view including the relation between the set of patterns and grammars, the semantics of a grammar, the expressive power of a grammar, grammatical inference, comparison between a syntactic method and a statistical method and a comparison of computing costs between a syntactic method and a prototype matching method.

Computation of Distance between New and Composite Parts: The Algorithm

This methodology utilizes a heuristic based on linguistic method. The minimization of a dissimilarity measure called Levenshtein distance is used as an optimization criterion. By using the proposed algorithm, the authors were able to find out optimal composite part in limited number of iterations.

Step-1

Set n to be the length of s.
Set m to be the length of t.
If n = 0, return m and exit.
If m = 0, return n and exit.
Construct a matrix containing 0..m rows and 0..n columns.

Step-2

Initialize the first row to 0
Initialize the first column to 0

Step-3

Examine each character of s (i from 1 to n).

Step-4

Examine each character of t (j from 1 to m)

Step-5

If s[i] equals t[j], the cost is 0.
If s[i] does not equal t[j], the cost is 1.

Step-6

Set cell d[i, j] of the matrix equal to the minimum of:
a. The cell immediately above plus 1: d[i-1, j] + 1
b. The cell immediately to the left plus 1: d[i, j-1] + 1.
c. The cell diagonally above and to the left plus the cost: d[i-1, j-1] + cost.

Step-7

After the iteration steps 3 to 6 are complete, the distance is found in cell d[n, m].

After finding out the maximum LD between all the strings in this manner, the LD is summed up. The fitness function taken is:

Fitness Function, FF = 1 / (1 + LD)

The aim of taking this objective function is to enable one to select the parts, which can be taken up for processing by the imaginary job-shop. A threshold value of LD is fixed, and all the parts having LD beyond this value are rejected, and only those, which lie within the threshold value, are selected. The underlying logic is that the job-shop cannot process all the jobs that come to it and it has to take a decision what kind of jobs can it take up. This technique helps in making that decision.

The authors have proposed Genetic Algorithm (GA) Method, which is based on theory of natural selection and "survival of the fittest" chromosome. The design of a GA depends on six key concepts: representation, initialization, evaluation function, reproduction, crossover and mutation.

Genetic Algorithm Procedure:

Choose initial population
Evaluate each individual's fitness
Do
    Select best-ranking individuals to reproduce
    Mate pairs at random
    Apply crossover operator
    Apply mutation operator
    Evaluate each individual's fitness
While terminating condition reaches.

6. Model Development

The model is divided into two parts, and described hereunder:

1) In the first part of the model, the processing requirements of each part are indicated in terms of an alphabetical string. This coding is based on the predefined primitive library. However, this coding could be customized as per the individual requirements of any manufacturing facility.
2) The second part of the model attempts to generate the code for composite part. The set of strings for parts is used for creating sub optimal set of target strings. The target strings are created by combination of strings for the parts. The maximum string length of target string is equal to string length of composite part. The string length for composite part is user dependent based on the desired in-house production cost. The initial population is generated by random selection of target strings. If string length of any string in initial population is not equal to that of given string length, random characters from target strings are introduced in the strings of

initial population. GA is applied on this initial population to get a set of composite strings. Among this set of composite strings, best-fit string is selected. The optimization criterion for implementation of GA is taken as minimization of summation of LD between an individual string of initial population and the set of target strings.

Algorithm for Creating Composite Part

The algorithm can be described in six steps as follows:

Step-1: Represent processing of each part in terms of alphabetical strings. Here, each alphabet represents a primitive as in primitive library. These strings are used as input strings. The model is capable of considering sequential processing along with random processing. (If a part in processed in the order of ‘d-c-g-h-a’ where 'd', 'c', 'g', 'h', and 'a' represent the processing element. Then, the corresponding sequential processing plan will be ‘dcgha’, while random processing plan will be ‘acdgh’.)

Step-2: Generate target strings by combination of input strings. All of the strings are not necessarily of equal string length. The maximum string length of target string is equal to that of the string length of composite part. The string length for composite part is user defined as per the desired in house production cost.

Step-3: Create initial population of strings by copying target strings randomly. If any chromosome of initial population is not equal to that of given string length, random characters from the set of target strings are copied.

Step-4: Calculate summation of Levenshtein Distance (∑LD) between each chromosome and the set of target strings. Calculate fitness value as 1/ (1+∑LD).

Step-5: Set iteration = 0.

Step-6: Reproduce by selecting and copying above average strings from initial population to form mating pool.

Step-7: Apply one point crossover to increase the average fitness of the mating pool.

Step-8: Mutate chromosome in mating pool. In this step first random number is generated to know the location of mutation. All the characters are assumed to be distributed equidistant in a range of random numbers. Another random number is generated to know the character to be substituted at the location of mutation.

Step-9: Iteration = Iteration + 1

Step-10: Stop if more than half of the population in the mating pool is of same functional value, otherwise go to step 6.

Example:

Step-1: Code the components based on the exemplary primitives’ library as follows:

Component 1: BJC
Component 2: CJG
Component 3: BCG

Step-2: Suboptimal strings are created as follows: BJCJG
BCG

Step-3: Initial population is generated as follows:

BJCJGC
BCGBBC
BCGCCC
BJCJGC
BJCJGJ
BCGBBJ
BCGJGG
BJCJGB
BJCJGB
BCGJBB

Step-4: Fitness value is evaluated for initial population through program developed in C++.

Step-5: Set iteration = 0.

Step-6: Reproduce the strings from the mating pool of initial population:

Step-7: Perform crossover on the string of mating pool of reproduced strings

Step-8: Perform mutation on the strings of mating pool of crossovered strings.

Step-9: Increase iteration value by 1.

Step-10: Set of composite strings are generated. Check the termination criteria. If not fulfilled go to step 6, otherwise pick the string of minimum ∑LD.

From the final mating pool, the strings of minimum LD are picked up to represent the composite part. The final composite string is BJCJGB. This method ensures the generation of composite part string in limited number of iterations. The authors have tested the proposed method on various examples and found the method consistent to generate the results. One of the examples is shown in the Table 4, where for the given case composite string of minimum ∑LD is shown:

Table 4: Composite string of minimum sum of LD

Input Strings	Composite String Size	Composite String	∑LD
BJC	3	BJE	3
CJE	4	BJEB	6
BCE	5	BJCJE	3
	8	BJCJEBCE	0

The above example of three strings of three characters each is given to show the effectiveness of the above method. However, the proposed method could be used on number of strings of more characters.

6. Conclusions

In this paper, a procedural step for assignment of new parts in cellular manufacturing system has been proposed using the concept of composite part. The described method can take care both the cases of random processing and sequential processing problems. Conventional clustering methods mostly solve the MPIM by matrix manipulation and cannot solve the cases where sequence of operations are maintained. Further, for the cases when a particular operation is performed more than once on a part, the representation of part processing in MPIM method is not possible as any element in the matrix will show only the incidence and not the number of incidence. For both the above cases where incidences are more than once or where sequential processing are considered, the method of applying dissimilarity coefficient similar to Lavenshtien distance is more applicable. The accuracy of the output of the present work is verified from the fact that the characters appeared in composite part string have the maximum frequency of occurrence at the corresponding places of the input strings. For example, if the strings of components are; BJLCEH, CJDAELF, DAEOCK, BCDAEP, BCEJOI and composite part is restricted to have a string length of 6, the output through the algorithm is found to be BCDAEP. It can be noted that the second place of the input strings ‘C’ and ‘J’ are appearing two times, but insertion of ‘C’ at second place in the composite part string will minimize ∑LD. Thus, the existence of bottleneck machines which complicate the process of diagonalisation in conventional clustering methods, will not add any restriction in this method.

The conventional cellular manufacturing system considers the forming of part families and the corresponding machine groups to form manufacturing cells. The large job-shop is partitioned to smaller job-shops and the advantages of flow shop is achieved. However, the conventional clustering methods do not consider many of the practical problems like operations sequence, cell layout, alternate process plans, cell efficiency or occurrences of more than one incidence in the MPIM. The methodology however generate inconsistent result if the bottleneck machines increases to the extent that one single jobshop becomes more efficient.

References

Askin, R. G., and Zhou, M., Formation of independent flow-line cells based on operation requirements and machine capabilities, *IIE Transactions*, vol. 30, pp. 319-329, 1998.

Burbidge, J. L., *The Introduction to Group Technology*, Heinemann, London, 1975.

Edwards, G. A. B., Principle of Composite Parts, Machine Shop Metal- working Econ., *Letter to Prod. Engr.*, vol. 31, no. 2, pp. 278, 1972.

Garbie, I. H., Parsaei H. R., and Leep, H. R., Introducing new parts into existing cellular manufacturing systems based on a novel similarity coefficient, *International Journal of Production Research*, vol. 43, no. 5, pp. 1007–1037, 2005.

Haldar, R., and Mukhopadhyay, D., Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach, working paper: *Cornell University library*, arXiv:1101.1232 [cs.IT], 2011

Hanov, S., Fast and Easy Levenshtein distance using a Trie, Available at: http://stevehanov.ca/blog/index.php?id=114, July 15,2013.

Liu, S. C., Feature extraction and classification for rotational parts taking 3D data files as input, *Journal of the Chinese Institute of Industrial Engineers*, vol. 21, no. 5, pp. 432-443, 2004.
Mitrofanov, S. P., *The Scientific Principles of Group Technology*, National Lending Library for Science and Technology, Boston, 1966.
Pandey, V., and Roy N., Extraction of features from a rotational part: a syntactic approach, *Journal of Institution of Engineers (India) - Production Engineering*, vol. 82, pp. 64-68, 2002.
Papaioannou, G., and 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*, vol. 206, pp. 509-521, 2010.
Roy, N., and Singh, M. D., Development of production cells in batch manufacturing: a linguistic model, *Institute of Engineers (India) Journal*, vol. 76, pp. 74-78, 1995.
Roy Nirjhar, A new genetic algorithm for manufacturing cell formation, Int. Journal of Manufacturing Technology and Research, Vol. 2, pp 83-90, 2006
Sarker, B. R., and Xu, Y., Designing multi-product lines: job routing in cellular manufacturing, systems. *IIE Trans.*, vol. 32, pp. 219–235, 2000.
Tam, K. Y., An operation sequence based similarity coefficient for part families information, *Journal of Manufacturing Systems*, vol. 9, no. 1, pp. 55-68, 1990.
Tanaka, E., Ikeda, M., and Ezure, K., Direct parsing, *Pattern Recognition*, vol. 19, no. 4, pp. 315–323, 1986.

Biography

Nirjhar Roy is a Professor of Mechanical Engineering Department, at Motilal Nehru National Institute of Technology Allahabad. He has received his Bachelors of engineering degree from Jadavpur University, Master's from IIT, Kharagpur and Doctoral degree from University of Allahabad. He has worked in industries and academic organisations for more than 35 years. He had given consultancy to Ministry of Industrial Development, Government of India and also for Coal India Ltd. for a number of projects. He has developed applications software for various industries in Northern India. He has written books for technical education and published many research papers.

V. Rao Komma is Assistant Professor in Department of Mechanical Engineering, Motilal Nehru National Institute of Technology Allahabad. He received his Master's and Doctoral degree in Mechanical Engineering from Indian Institute of Technology, Roorkee, in 2003 and 2009 respectively. He has published 17 research articles in various national and international journals and conference proceedings. His research interests include modeling & simulation of manufacturing systems, agent-based manufacturing systems, integration of CAD and CAM.