A Comparison of Bond Energy and Commonality Analysis Algorithms In Cellular Manufacture

Omer Ahmed Eltohami

Lecturer in Production Technology Dept. at Jeddah College Of Technology, Suadi Arabia

ABSTRACT

This study addresses the cell formation problem in group technology, through a comparison of the solutions provided by two algorithms. The first one is based on the ideas of Bond Energy formulation (Matching Algorithm (MA)), and the second is based on Commonality Scores analysis (Linear Cell Clustering Algorithm (LCCA)). The relative performance of MA and LCCA was investigated under three types of Datasets. The research applied Cell Generation, Percentage of Exceptions and Grouping Efficiency as performance measures. The results of the experiments indicate that, in general, MA is superior to LCCA in terms of Cell Generation and Percentage of Exceptions measures, while LCCA scores higher values of Grouping Efficiency.

1. Introduction

The cell formation problem in Group Technology (GT) has been addressed by several studies by many researchers. This problem which is embedded in a larger cell design process is very important indeed, since it influences the physical layout of the system, and thus influences the scheduling and control policies of the system. Numerous techniques for solving this problem are available in the literature, where a large number of cell formation algorithms, belonging to various approaches, have been developed over the last three decades to solve the machine - part grouping problem.

Manuscript received from Dr: Omer Ahmed El- Tohami at: 12/ 8/ 1997,
Accepted at: 26/ 8/ 1997, pp. 39-55,
Engineering research bulletin, Vol. 20, No. 3, 1997,
Menoufiya University, Faculty of Engineering,
Shebin El-Kom, Egypt, ISSN. 1110-1180.

Despite the existence of several procedures for solving the cell formation problem, still missing from the literature are studies directed to the problem of selecting among these numerous procedures. Hence, practitioners are still faced with the problem of selecting the appropriate cell formation algorithm that can be applied to their specific environments. Here, this study is an attempt to bridge this gap in cellular manufacturing (CM) systems studies by providing a comparison of two promising cell formation algorithms using cell generation, Percentage of Exceptions, and Grouping Efficiency as measures to assess the "goodness" of the solutions provided by each algorithm.

The problem of partitioning a manufacturing system into machine groups and the related problem of partitioning the set of manufactured parts into part families, according to the routing sheet information, is known as the cell formation problem. In its simplest form it can be formulated as a matrix diagonalization problem, where the incident matrix (a_{ij}) (explained below) is transformed into a block diagonal form such that mutually exclusive part families and their corresponding machine groups emerge. Actually, the cell formation problem involves three major interrelated issues: the identification of part families, the identification of machine groups, and the allocation of part families to machine groups. Block diagonalization of the incidence matrix is the first and most important step towards the application of **GT**.

2. Cell Formation Algorithms

Cell formation means the identification of a family of parts as being suitable for manufacture on a specific group of machines. There are several approaches that have been attempted to solve this problem, but only few researchers have attempted to develop frameworks for the cell formation process. Based on a comprehensive literature survey, the following two cell formation algorithms are selected to be tested and compared:

(1) Matching Algorithm (MA) (Bhat & Haupt, 1976);

(2) Linear Cell Clustering Algorithm (LCCA) (Wei & Kern, 1989).

It has to be mentioned that MA is always able to find a total division into groups, which complete all the parts they make, if such a configuration exists, i.e. it is able to form a completely cellular layout if one exists, which is not the case for **LCCA**. In addition, the two algorithms, for their operations, need nothing more than a machine - part route matrix.

On the other hand, it is noted that the two algorithms selected for testing here could be programmed on a computer, which is an essential feature since many real applications may be too large for manual approaches. Now, each of the tested algorithms will be considered in detail in the following sections.

2.1 Matching Algorithm (MA)

Bhat and Haupt (1976) proposed a clustering algorithm based on the ideas underlying the Bond Energy Algorithm (BEA) of McCormick et al (1972) and the shortest Spanning Path Algorithm of Slagle et al. (1975). According to the taxonomic frame suggested by Offodile et al.(1994), the model structure for this algorithm could be classified as having a matrix formulation of the Array-Based type, and its solution approach is also an Array-Based approach (Bond energy). It is noted that Array-Based methods group machines and parts without applying a similarity measure. Rows and columns of the incidence matrix are rearranged until a diagonal pattern of mutually separable clusters emerges. Machine - part groups are formed simultaneously. The strength of the proposed scheme is in its use of the special property of matrix $\mathbf{B = A*A^T}$. It reduces the required computations by a factor of **(n)** for row ordering and **(m)** for column ordering of an mxn matrix **(A)**. This algorithm , which is called the Matching Algorithm (**MA**), uses integer arithmetic similar to the methods of the **BEA** and the Shortest Spanning Algorithm. The **MA** is a general cluster analysis technique, and it is not specially developed to solve the part family / machine group formation problem. The **MA** is based on the fact that the formation of clusters from an unorganised data array can rest on a simple rule which reckons with the number of matchings of the zero and nonzero entries between two parallel lines (rows and columns) of the array. Depending on the specific problem that gives rise to a data array, one could further assign some weightings for the zero - nonzero structure in order to realise clusters that are germane to the problem. In fact, these ideas are implicit in the **BEA** and the shortest path algorithm, but the manner in which the **MA** has been conceptualised does make a significant difference to cluster analysis. The principal advantage as claimed by the authors is that the **MA** stems from the fact that when a row is rearranged, the change in the number of matchings need not be recomputed for each possible arrangement, instead only the change in ϕ - which is given by :

$$\mathbf{sum}\ \phi = \sum_{i=1}^{m-1} \mathbf{b i, i+1} \quad \text{, where; } b = \mathbf{A*A^T} \text{ - is evaluated.}$$

Furthermore, $\mathbf{A * A^T}$ is obtained by merely comparing and counting and not by matrix multiplication, so the computation of $\mathbf{A*A^T}$ is equal to the computation of matchings among the rows. The **MA** has a computational complexity of $\mathbf{O(m^2+n^2)}$.

2.2 Linear Cell Clustering Algorithm (LCCA)

Wei and Kern (1989) suggested an algorithm which is based on the calculation of a commonality score which indicates the similarity in the way two machines are used in the shop to manufacture the products. The algorithm generates consistent machine groupings regardless of the initial order of the input data. The clustering process allows sensitivity analysis of the solution, introducing various constraints on the characteristics of the machine clusters, without requiring repetition of major portions of the procedure. The algorithm has a linear computational complexity. The commonality score not only recognises the parts on which two machines do work, but also the parts on which the machines both do not work. The algorithm can always create the maximum number of cells dictated by the commonality scores. The similarity score used by the authors is an adaptation of the similarity score that Kusiak (1987) applied to the clustering problem. Kusiak's formulation produces a (P x P) matrix, while commonality scores produce a (M x M) matrix, and since the latter matrix is smaller, it therefore requires (in most cases) less processing time to cluster. Kusiak (1987) defined a similarity score for part pair i and j as:

$$S_{ij}=\Sigma\, \delta(a_{ik}, a_{jk})$$

Where $\delta(a_{ik}, a_{jk}) = \{1 \text{ if } a_{ik} = a_{jk}; O \text{ otherwise}\}$. Kusiak had one condition for adjusting his similarity score: if $a_{ik} = a_{jk}$ then add one point to the score. But, the commonality score includes two conditions for adjusting the similarity score:

(1) if $a_{ik} = a_{jk} = 1$, then add (p - 1) points to the commonality score C_{ij};

(2) if $a_{ik} = a_{jk} = 0$, then add one point to the commonality score C_{ij}.

In this way, only one case adds zero points to C_{ij}: if $a_{ik} = a_{jk}$, which means that one of the machines is used to manufacture part k, and the other is not.

In Summary,

$$C_{ij} = \Sigma\, \Gamma\, (a_{ik}, a_{jk})$$

$$\text{Where: } \left(a_{ik}, a_{jk}\right) = \begin{cases} (p-1), \text{ if } a_{ik} = a_{jk} = 1 \\ 1, \quad\quad\; \text{if } a_{ik} = a_{jk} = 0 \\ 0, \quad\quad\; \text{if } a_{ik} \neq a_{jk} \end{cases}$$

Once the commonality scores have been calculated, the scores must be compared and the machines clustered accordingly. The worst-case complexity for the **LCCA** process is **$O(m\log m + m^2/2)$**.

3. General Comments

According to the framework suggested by Offodile et al (1994), the model structure for the LCCA could be classified as having a matrix formulation of the Similarity Coefficient - Based type, while its solution approach is of the heuristic type.. But, according to the classification suggested by Wemmerlov and Hyer (1986), which is more appropriate for performing our comparison, we find that:

MA is a techniques that identifies part families and machine groups simultaneously while the LCCA alone belongs to the techniques that identify machine groups only. Therefore, there is a need to augment the LCCA with an appropriate assignment procedure of another cell formation algorithm. By reviewing previous literature, the following was obtained:

(i) Kaparthi and Suresh (1994) augmented the LCCA by incorporating neural network logic for allocating parts, and they called that the Augmented Linear Clustering (ALC). Although, the authors showed that the ALC algorithm was superior for large datasets, they pointed out that the principal limitation with neural network methods, is the category proliferation problem. That is, the number of classes identified tends to increase rapidly when data size is large, and/or when vigilance threshold parameter is set at high level.

(ii) The possible algorithms for augmenting the LCCA were Rank Order Clustering algorithm (ROC2) (King and Nakornchai, 1982) and Direct Clustering algorithm (DCA) (Chan and Milner, 1982). Any of these two algorithms could have been used. However, the DCA was found to be more suitable for augmenting the LCCA, because it is simple and effective in clustering data directly from any given machine-component matrix, and in addition it was specifically designed for computer use and it can easily deal with large amounts of data obtainable in realistic situations. These two latter features of the DCA are essential since many real applications may be too large for manual approaches. The DCA is based on progressively restructuring the machine-part matrix by going through it sequentially, moving the rows with 'left-most' positive cells to the top and the columns with 'top-most' positive cells to the left of the matrix. Thus, if LCCA is augmented with the DCA, in relatively few trips the positive cells will be squashed toward the diagonal of the matrix and a clustered pattern will be formed, since by the original LCCA in the first stage of the augmented algorithm, the columns (machines) will already be in the right order. Therefore, in this study the LCCA is augmented with the part assignment procedure of the DCA, and hence the solutions obtained by this augmented form of the LCCA can be directly compared with the solutions generated by the other tested algorithms. In the rest of this study we refer to this augmented form of the LCCA briefly as LCCA.

4. Computer Programs

FORTRAN language (Version 5.1) was used to code all the tested algorithms, and they were executed on an ASI 486DX-33 personal computer (PAT 48 AV 486 VL-Bus/ISA System Board). For running any of these programs the user needs to input a part - machine incidence matrix, where a value of one in row i and column j means that the part i is processed on machine j, while a zero indicates that machine j is not involved in the processing of part i. Also, the user needs to specify the number of parts (rows) as well as the number of machines (columns).

5. Dataset Generation

The two algorithms were tested on one route sheet dataset from each of the following three data patterns:

(1) Block Diagonal (BD) pattern: where part/machine groups lie in separately defined blocks along the diagonal of the route sheet data matrix.

(2) Block Diagonal with Low Intercell Transfers (BD/L) pattern: in this dataset, approximately 10% of the parts require routing to more than one cell.

(3) Block Diagonal with Medium Intercell Transfers (BD/M) pattern: in this dataset, approximately 20% of the parts require routing to more than one cell.

The BD pattern dataset (see Appendix A) is based on the problem given by Morris (1988).

Then the BD/L and the BD/M patterns were obtained from this BD pattern by modifying the routings of some of the parts in order to obtain about 10% and 20% exceptional parts, respectively. These datasets were randomised, then the disguised datasets were clustered by each of the tested algorithms, and the original structures sought to be recovered.

6. Algorithms Comparison Measures

The following measures were considered to compare the alternative clustering algorithms by assessing the "goodness" of the solutions provided by each algorithm:

6.1 Cell Generation

By this measure it is meant to check the ability of each algorithm to produce BD solutions when they exist, and also their ability to correctly identify the cells that did not contain exceptions in the cases of BD/L and BD/M patterns.

6.2 Percentage of Exceptions

Is defined as the ratio of the parts requiring processing in more than one cell (exceptional parts) to the total number of parts, given as a percentage. It is noted that this measure is differently defined in other studies (Shafer and Rogers (Part II), 1993).

6.3 Efficiency Measures

The Grouping Efficiency (G.E.) as suggested by Chandrasekharan and Rajagopalan (1986b) will be used in this study. The concept of G.E. is based on two parameters: within - group utilisation and intercell movement. From the matrix point of view, the concentration of non-zero elements in the diagonal submatrices refers to utilisation, and the presence of such elements outside the diagonal submatrices represents intercell movements. G.E. is expressed as a weighted average of two efficiencies h_1 and h_2 as follows:

$$G.E. = q\,h_1 + (1 - q)\,h_2$$

Where:

$$0 \leq q \leq 1.0,$$

h_1 = number of non-zero elements in the diagonal blocks ÷ total number of elements in the diagonal blocks.

and h_2 = number of zeros in the off - diagonal blocks ÷ total number of elements in the off-diagonal blocks.

h_1 represents utilisation efficiency and h_2 represents intercell movements efficiency.

G.E. satisfies the basic requirements of non-dimensionality, non-negativity and zero - to - one range. The weighting factor (q) makes it possible to alter the emphasis between utilisation and intercell movement, depending on the specific requirements of the given problem.

The alternative algorithms in this study were compared for three values of q : 0.2, 0.5, and 0.8. Harhalkis et al. (1990) have suggested a similar evaluation criteria, where their efficiency concept is based on three efficiencies: Global Efficiency, Group Efficiency and Group Technology Efficiency. It is noted that these three efficiencies are not entirely independent, and it seems more appropriate to use the G.E. measure as defined above.

It is worth mentioning that, in this study the computational time measure is not applied due to the following:

(i) the execution times may not be of importance, since cell formation is a design exercise that may be required once, and it will not be executed on a day-to-day basis;

(ii) the four tested algorithms, as appeared during the pilot runs are very fast indeed, and this also was supported by results obtained in previous literature concerning the time requirements for these algorithms.

7. Discussion and Analysis of Clustering Solutions Obtained by the Algorithms

7.1 Experimental Results

The cell formation results, for each of the tested algorithms using 3 input datasets : Dataset of BD pattern, Dataset of BD/L pattern, and Dataset of BD/M pattern were obtained as part/machine clusters. These results were then used to compare the solutions generated by the tested algorithms in terms of their ability to generate cells, percentage of exceptions, and G.E. measure for three levels of q (0.2, 0.5, and 0.8). Table 1.1 shows the number of cells together with the Percentage of Exceptions given by the tested algorithms under the 3 Datasets. Table 1.2 presents the values of the within-cell utilisation parameter (h1) and intercell movements parameter (h2). Tables 1.3 through 1.5 summarise the results obtained for the G.E. measure for the two tested algorithms, under 3 Datasets, under three levels of the value of q. The next sections present the results of these tests together with their discussion and analysis, and a number of important conclusions were drawn.

7.2 Results of Cell Generation and Percentage of Exceptions

(1) Dataset 1 : (BD pattern)

(I) MA algorithm, produced a pure block solution, but the LCCA algorithm failed to produce a pure block diagonal solution, i.e. the solution obtained for this Dataset contained exceptional parts, where this solution contains a higher number of cells when compared to the solution offered by the MA algorithm.

It should be noted that the number of cells (or clusters) obtained by the LCCA are the maximum number of cells possible by this algorithm, without having any constraints on the number of cells or alternatively the cell size. If we assume that the numbers of cells obtained by the solution offered by the MA algorithm are the desired numbers of cells, the LCCA will produce a solution that have much lower percentages of Exceptions than those shown in Table 1.1. This shows the flexibility of the LCCA, which is obtained by the joining operations of the algorithm, based on selecting the highest commonality score that has not yet been considered in the clustering process.

(ii) The MA algorithm produced a pure block solution - but not block diagonal solution - where the clusters were interspersed above, below and along the diagonal. For the LCCA, although the clusters start from the north-west and end at the south-east diagonal, the clusters are still interspersed above, below and along the diagonal.

(iii) The MA algorithm obtained a solution with zero Percentage of Exceptions, while the LCCA obtained a solution with 52.5% Percentage of Exceptions.

(2) Dataset 2 : (BD/L Pattern)

The MA algorithm generally ranked better than LCCA algorithm with respect to the Percentage of Exceptions measure. The LCCA produced the highest percentages of Exceptions when considering its initial solution. But, as mentioned before, if fewer cells were desired, the LCCA would be expected to give a lower percentages of Exceptions.

(3) Dataset 3 : (BD/M Pattern)

Again, the high percentages of Exceptions obtained by the LCCA were due to the high number of cells created by this algorithm.

Finally, it should be noted that the MA algorithm, is sensitive to the initial arrangement of the dataset matrix, while the LCCA algorithm is not.

7.3 Results of Grouping Efficiency Measure

(i) It was apparent from Tables 1.3 through 1.5, that the LCCA scored the highest values of G.E. This was true with the highest value of q (0.8) i.e. when much emphasis is being placed on the within - cell utilisation factor, and it was also true when more weight was given to the frequency of intercell movements parameter i.e. at q = 0.2. Thus, whatever the value selected for q, the LCCA attained, generally and consistently, the best values of G.E. when compared to the MA algorithm.

(ii) The relative performance of the two algorithms tested was not affected by the value of q for all the datasets.

8. Conclusions and Recommendations for Future Reasearch

(1) Generally, the MA algorithm outperformed the other algorithm in the Percentage of Exceptions measure i.e. it has the lowest values with respect to this measure.

(2) Only the MA algorithm was able to identify the existence of a (BD) solution (Dataset 1), but the LCCA was not successful in doing so.

(3) Irrespective of the value chosen for q, the LCCA, generally and consistently, outperformed the other tested algorithm, under all the datasets, with respect to the G.E. measure.

(4) The form used here in this research for the LCCA is a modified one, since the original algorithm just gives the group of machines for each cell, without identifying the part families associated with each cell. Thus the algorithm is successfully augmented with the part assignment procedure of the DCA .

(5) The LCCA sometimes failed to identify the originally specified clusters, but this is not a problem, since this is generally associated with using the algorithm without forcing any constraints on the number of cells. But, generally when the number of cells is decided, the algorithm could easily identify the block diagonal solutions. Also, it should be known that these originally specified clusters are not necessarily the best solutions.

For example, with Dataset 1 the LCCA solution has 7 cells, while the initial data has a 4 cells (BD) solution. If we then apply the constraint that the number of cells is 4, immediately from the LCCA solution chart, it can bee seen that cell 2 [3,8,11] should be joined with cell 1 [17,22,20,25,2], and cell 3 [4,12,16,6] with cell 5 [5,19,14,24], in order to obtain a (BD) solution, which is exactly the same result when the commonality score values are used. Alternatively, without knowing the number of cells that are required to create a (BD) solution, the cell joining process could continue, according to the values of the commonality score, until no exceptional parts exist, if originally there exists a BD solution.

(6) This research serves to bring into focus the utilisation of the MA as a promising cell formation algorithm, although it is a general clustering technique, and it was not specially developed to solve the cell formation problem. Its effectiveness has been demonstrated in the following aspects:

- It was capable of producing a pure block solution with the (BD) pattern of input Dataset 1;

- generally, it yielded the lowest percentage of exceptional parts, when compared to the other tested algorithm.

(7) The findings of this study open many more doors than they close in terms of future research directions. An obvious extension of this research is to compare the tested algorithms using larger input datasets than the ones used in this study, such that they would bear more resemblance to industry realities. This is useful in determining which algorithms are amenable for the much larger datasets encountered in practice.

Table 1.1. The number of cells created and the Percenta of Exceptions (%)

Dataset Number	Pattern	Algorithms			
		MA		LCCA	
		Cell No.	%	Cell NO.	%
5	BD	4	0.0	7	52.5
10	BD/L	4	7.5	7	47.5
15	BD/M	4	12.5	7	52.5

Table 1.2 Values of within - cell utilisation parameter (h_1)and intercell movements parameter (h_2)

Dataset Number	Pattern	Algorithms			
		MA		LCCA	
		h1	h2	h1	h2
5	BD	0.52	1.00	0.66	0.95
10	BD/L	0.52	1.00	0.68	0.96
15	BD/M	0.52	0.99	0.68	0.96

Table 1.3 Grouping Efficiency (G.E.) values at $q_1 = 0.2$

Dataset	Pattern	MA	LCCA
5	BD	0.90	0.89
10	BD/L	0.90	0.90
15	BD/M	0.90	0.90

Table 1.4 Grouping Efficiency (G.E.) values at q = 0.5

Data set	Pattern	MA	LCCA
5	BD	0.76	0.81
10	BD/L	0.76	0.82
15	BD/M	0.76	0.82

Table 1.5 Grouping Efficiency (G.E.) values at q = 0.8

Data set	Pattern	MA	LCCA
5	BD	0.62	0.72
10	BD/L	0.62	0.74
15	BD/M	0.61	0.74

References

Bhat, M.V. and Haupt, A.(1976). 'An efficient clustering algorithm'. IEEE Transactions on System, Man and Cybernetics, Vol. 6,61-64.

Chandrasekharan, M.P. and Rajagopalan, R. (1986b). 'An ideal seed non-hierarchical clustering algorithm for cellular manufacturing'. International Journal of Production Research, vol. 24, no. 2, 451-464.

Chan, H.M. and Milner, D.A. (1982). 'Direct clustering algorithm for group formation in cellular manufacture'. Journal of Manufacturing Systems, vol. 1, no. 1, 65-75.

Harhalakis, G., Nagi, R. and Proth, J.M. (1990). 'An efficient heuristic in manufacturing cell formation for group technology applications'. International Journal of Production Research, Vol. 28, no. 1, 185-198.

Kaparthi, S. and Suresh, N.C. (1994). 'Performance of selected part-machine grouping techniques for data sets of wide ranging sizes and imperfection'. Decision Sciences, vol. 25, no. 4, 515-539.

King, J.R. and Nakornchai, V. (1982). 'Machine - component group formation in group technology: review and extension'. International Journal of Production Research, vol. 20, no. 2, 117-133.

Kusiak, A. (1987). 'The generalized group technology concept'. International Journal of Production Research, Vol. 25, no. 4, 561-569.

McCormick, W.T., Schweitzer, P.J., and White, T. (1972). 'Problem decomposition and data reorganization by a clustering technique'. Operations Research, vol. 20, no. 5, 993-1009.

Morris, J.S. (1988). 'Forming cellular layouts in group technology: a simulation evaluation of alternative approaches'. Unpublished Ph.D. Dissertation, University of Oklahoma, USA

Offodile, O.F., Mehrez, A. and Grznar, J. (1994). 'Cellular manufacturing: a taxonomic review framework'. Journal of Manufacturing Systems, vol. 13, no. 3, 196-220.

Shafer, S.M. and Rogers, D.F. (1993). 'Similarity and distance measures for cellular manufacturing. Part II. An extension and comparison'. International Journal of Production Research, vol. 31, no. 6, 1315 - 1326.

Slagle, J.R., Chang, C.L. and Heller, S.R. (1975). 'A clustering and data reorganization algorithm'. IEEE Transactions on System, Man and Cybernetics, Vol. SMC-5, 125-128.

Wei, J.C. and Kern, G.M. (1989). 'Commonality analysis: a linear cell clustering algorithm for group technology'. International Journal of Production Research, vol. 27, no. 12, 2053-2062.

Wemmerlov, U. and Hyer, N.L. (1986). 'Procedures for the part family/machine group identification problem in cellular manufacturing'. Journal of Operations Management, vol. 6, no. 2, 125-147.

Appendix A Datasets

A1. Dataset 1: Block Diagonal Pattern

(25 Machines & 40 Parts)

Parts/Machines

	00121022 38157202	011022110 138713509	11020101 49546246
1	[illegible]	000000000	00000000
2	[illegible]	000000000	00000000
4	[illegible]	000000000	00000000
5	[illegible]	000000000	00000000
6	[illegible]	000000000	00000000
7	[illegible]	000000000	00000000
13	[illegible]	000000000	00000000
14	[illegible]	000000000	00000000
17	[illegible]	000000000	00000000
18	[illegible]	000000000	00000000
12	[illegible]	000000000	00000000
16	[illegible]	000000000	00000000
9	[illegible]	000000000	00000000
10	[illegible]	000000000	00000000
15	[illegible]	000000000	00000000
11	[illegible]	000000000	00000000
3	[illegible]	000000000	00000000
8	[illegible]	000000000	00000000
19	00000000	[illegible]	00000000
21	00000000	[illegible]	00000000
22	00000000	[illegible]	00000000
27	00000000	[illegible]	00000000
29	00000000	[illegible]	00000000
25	00000000	[illegible]	00000000
26	00000000	[illegible]	00000000
20	00000000	[illegible]	00000000
23	00000000	[illegible]	00000000
24	00000000	[illegible]	00000000
28	00000000	[illegible]	00000000
30	00000000	000000000	[illegible]
33	00000000	000000000	[illegible]
32	00000000	000000000	[illegible]
31	00000000	000000000	[illegible]
34	00000000	000000000	[illegible]
35	00000000	000000000	[illegible]
36	00000000	000000000	[illegible]
37	00000000	000000000	[illegible]
38	00000000	000000000	[illegible]
39	00000000	000000000	[illegible]
40	00000000	000000000	[illegible]

A2. Dataset 2: Block Diagonal With Low Inter-cell Transfers

(25 Machines & 40 Parts)

Parts/Machines

	00121022 38157202	011022110 138713509	11020101 49546246
1	[illegible]	000000000	00000000
2	[illegible]	000000000	00000000
4	[illegible]	000000000	00000000
5	[illegible]	000100000	00000000
6	[illegible]	000000000	00000000
7	[illegible]	000000000	00000000
13	[illegible]	000000000	00000000
14	[illegible]	000000000	00000000
17	[illegible]	000000000	00000000
18	[illegible]	000000000	00000000
12	[illegible]	000000000	00000000
16	[illegible]	000000000	00000000
9	[illegible]	000000000	00000000
10	[illegible]	000000000	00000000
15	[illegible]	000000000	00000000
11	[illegible]	000000000	00000000
3	[illegible]	000000000	00000000
8	[illegible]	000000000	00000000
19	00000000	[illegible]	00000000
21	00000000	[illegible]	00000000
22	00000000	[illegible]	00000000
27	00000000	[illegible]	00000000
29	00000000	[illegible]	00000000
25	00001000	[illegible]	00000000
26	00000000	[illegible]	00000000
20	00000000	[illegible]	00000000
23	00000000	[illegible]	00000000
24	00000000	[illegible]	00000000
28	00000000	[illegible]	00000000
30	00000000	000000000	[illegible]
33	00000000	000000000	[illegible]
32	00000000	000000010	[illegible]
31	00000000	000000000	[illegible]
34	00000000	000000000	[illegible]
35	00000000	000000000	[illegible]
36	00000000	000000000	[illegible]
37	00000000	000000000	[illegible]
38	00000000	000000000	[illegible]
39	00000000	000000000	[illegible]
40	00000000	000000000	[illegible]

A3. Dataset 3: Block Diagonal With Medium Inter-cell Transfers

(25 Machines & 40 Parts)

Parts/Machines

	00121022	011022110	11020101
	38157202	138713509	49546246
1	[illegible]	000000000	00000000
2	[illegible]	000000000	00000000
4	[illegible]	000000000	00000000
5	[illegible]	000100000	00000000
6	[illegible]	000000000	00000000
7	[illegible]	000000000	00000000
13	[illegible]	000010000	00000000
14	[illegible]	000000000	00000000
17	[illegible]	000000000	00000000
18	[illegible]	000000000	00000000
12	[illegible]	000000000	00000000
16	[illegible]	000000000	00000000
9	[illegible]	000000000	00000000
10	[illegible]	000000000	00000000
15	[illegible]	000000000	00000000
11	[illegible]	000000000	00000000
3	[illegible]	000000000	00000000
8	[illegible]	000000000	00000000
19	00000000	[illegible]	00000000
21	10000000	[illegible]	00000000
22	00000000	[illegible]	00000000
27	00000000	[illegible]	01000000
29	00000000	[illegible]	00000000
25	00001000	[illegible]	00000000
26	00000000	[illegible]	00000000
20	00000000	[illegible]	00000000
23	00000000	[illegible]	00000000
24	00000000	[illegible]	10000000
28	00000000	[illegible]	00000000
30	00000000	000000000	[illegible]
33	00000000	000000000	[illegible]
32	00000000	000000010	[illegible]
31	00000000	000000000	[illegible]
34	00000000	000000000	[illegible]
35	00000000	000000000	[illegible]
36	00000000	000000000	[illegible]
37	00000000	000000000	[illegible]
38	00000000	000000000	[illegible]
39	00000000	000000000	[illegible]
40	00000000	000000000	[illegible]

(الملخص العربى)

مقارنة بين استخدام خواريزم صيغة طاقة الترابط مع خواريزم تحليل الخدوش المشتركة فى خلايا المجموعات الصناعية .

تهتم هذه الدراسة بمشكلة تشكيل الخلية في المجموعات الصناعية من خلال مقارنة الحلول وذلك باستخدام خواريزمين . الخواريزم الأول يعتمد على أفكار صيغة الترابط (خواريزم التطابق) ، والخواريزم الثانى يعتمد على تحليل الخدوش المشتركة (خواريزم تراكم الخلية الخطية) .

تم دراسة وبحث الآداء النسبى لكلا الخواريزمين على ثلاثة أنواع من مجموعات المعطيات ، كذلك تم تطبيق انتاج الخلية ونسب الاستثناءات وكفاءة المجموعات للقياسات الآداء .

أوضحت نتائج التجارب على وجه العموم أن استخدام خواريزم التطابق أفضل من استخدام خواريزم تراكم الخلية الخطية فى حالتى انتاج الخلية وقياسات نسب الاستثناءات ، بينما خواريزم تراكم الخلية الخطية أعطت قيم أعلى فى حالة كفاءة المجموعات .