Proceedings of ICAD2011
The Sixth International Conference on Axiomatic Design
Daejeon – March 30-31, 2011
ICAD-2011-03

EXTENDED ALGORITHM FOR DESIGN-MATRIX REORGANIZATION

Efrén Moreno Benavides
efren.moreno@upm.es
Universidad Politécnica de Madrid
Department of Aerospace Propulsion
Pza. Cardenal Cisneros, 3
28040 Madrid, Spain

Luis García Rodríguez
luisgr@alumnos.upm.es
Universidad Politécnica de Madrid
Department of Aerospace Propulsion
Pza. Cardenal Cisneros, 3
28040 Madrid, Spain

ABSTRACT

An algorithm, which is based on an extension of Nam P. Suh's algorithm, is proposed to reorganize the design matrix. The results are 1) the lowest-triangular and most diagonal design matrix and 2) a set of extra columns or rows depending on whether the design problem is redundant or not. The modification allows the reorganization of matrices with any distribution of Xs inside and works with rectangular matrices of any size. This paper describes the steps of the procedure and gives several examples of reorganization. In addition, the new algorithm is compared with the T. Lee, Acclaro, and N.P Suh algorithms.

Keywords: design-matrix, algorithm, matrix reformulation, off-diagonal term, decoupling strategy.

1 INTRODUCTION

Axiomatic Design (AD) developed by N. P. Suh [1990] provides a systematic approach to engineering design based on two axioms: the independence axiom and the information axiom. Axioms act on the mappings between design domains. Suh defined four domains in the design/manufacturing world: the customer domain, the functional domain, the physical domain and the process domain. Design matrices can express the relationships between functional requirements (FRs), which are defined in the functional domain, and design parameters (DPs), which are defined in the physical domain.

Axioms determine the best structure of these matrices. The larger the number of couplings between DPs and FRs, the worst the design is. As a result, designs can be classified according to the structure of the matrix. There are three main possibilities: coupled, decoupled and uncoupled designs. A diagonal design matrix characterizes an uncoupled design; a triangular design matrix characterizes a decoupled design; and a design matrix that cannot be converted to an uncoupled or decoupled characterizes a coupled design. Examples of these cases are shown in Fig. 1.

$$\begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix} \begin{bmatrix} X & 0 & 0 \\ X & X & 0 \\ 0 & X & X \end{bmatrix} \begin{bmatrix} X & 0 & X \\ X & X & 0 \\ X & X & X \end{bmatrix}$$

Uncoupled Decoupled Coupled

Figure 1. Uncoupled, decoupled and coupled matrices.

In general, functional requirements and design parameters can have any order, and hence the position of the non-zero elements in the design matrix is not fixed. For example, the decoupled matrix in Fig. 1 can be converted into a non-triangular matrix if the second and third rows and columns are permuted. Therefore, an algorithm, that transforms a general design matrix into the most diagonal or triangular matrix, could classify any design matrix as uncoupled, decoupled or coupled. The first axiom forces the designer to produce a set of DPs that maintains the independence of the FRs. This also implies that only two configurations of the design matrix are acceptable: decoupled or uncoupled. Additional off-diagonal terms cause coupling and should be eliminated. Then, the first step for assessing the goodness of a design is to rearrange the design matrix in order to find the off-diagonal terms. The importance of rearranging the design matrix lies in the fact that it allows the designer to discover the minimum number of matrix elements which have to be removed in order to decouple the matrix. Therefore, in an initial stage of the design process, it is very convenient to have an algorithm that gives the minimum set of off-diagonal terms that should be removed in order to obtain a decoupled or an uncoupled design.

Due to the importance of this step, previous authors have studied the problem: Suh [1990], Su [2003], Lee and Jeziorek [2006], Lee [2006], Cai [2009], and Acclaro Software [2010]. Principal contributions are described in Section 2. However, these algorithms have limitations. For example, the optimum algorithm proposed by T. Lee [2006] only works with square matrices that have non-zero elements in the main diagonal. This paper describes an algorithm that solves some of these deficiencies. However, it has other limitations (i.e., it is necessary to run the algorithm several times to achieve the optimum result). Thus, the purpose of this paper is to find an algorithm that rearranges the columns and rows of the design matrix in order to obtain a design matrix that is more diagonal or triangular than the original matrix. This algorithm must 1) be easy to implement, and 2) work with the most general design matrix (non-square matrices with zero elements in the main diagonal).

The main objective of this article is to explore the possibilities of an extended algorithm (EA), which is halfway between the N. P. Suh [1990] and T. Lee algorithms [2006]. The objectives of the EA are: 1) to obtain a design matrix that is more triangular than the original matrix, 2) to obtain a design matrix that is more diagonal than the original matrix, and 3) is valid for any matrix dimension, even for a rectangular matrix. Section 2 presents an overview of the methods used for matrix rearrangement. Section 3 describes the EA and gives some examples. Section 4 discusses and explains why EA is halfway between the Suh and Lee algorithms. Finally, conclusions are given in Section 5.

2 BACKGROUND

Some authors have dealt with the problem of rearranging the design matrix: Suh [1990], Lee and Jeziorek [2006] and Lee [2006]. Also, software programs like Acclaro DFSS [Axiomatic Design Solutions Inc., 2010] incorporate algorithms that rearrange the design matrix.

N. P. Suh [1990] describes a way to rearrange the design matrix. He uses an algorithm that moves rows and columns. The process is as follows:

(i) Find a row which has one non-zero element. Rearrange the order of FRs and DPs by putting the row and the column which contain the non-zero element first.

(ii) Excluding the first row and column, find the row which contains one non-zero element. Rearrange the components of FRs and DPs by putting the row and column which contains the non-zero element at the row and column second.

(iii) Repeat the procedure until there are no more sub-matrices to analyze.

This procedure works well when there are a few non-zero elements in each row. If the number of non-zero elements increases, the procedure has more possibilities of not finding the best rearrangement. To solve this problem Lee and Jeziorek [2006] use an optimal strategy based on graph theory. They aim to find the set of minimum elements that decouples the design matrix. An extended explanation of the complete process is given by T. Lee [2006]. To summarize, the process is as follows:

(i) Take the original design matrix DM(i,j) and construct the adjacency matrix A to determine the existence of coupling.
(ii) If coupled, construct incidence matrix B from A.
(iii) Identify a direct spanning tree.
(iv) Given B and the spanning tree, construct cycle matrix C.
(v) Search the combinations of the columns to find the first set of columns with non-zero entries when summed up.
(vi) Combination of columns found in the previous step indicates the minimum set of off-diagonal terms that decouples the design matrix DM.

This method is very powerful but it is more complex than the algorithm used by N. P. Suh, and needs a square design matrix with all the elements in the main diagonal different from zero (i.e., none of the elements in the main diagonal can be zero).

3 EXTENDED ALGORITHM

In this section the objective is to describe an extended algorithm (EA) which is easy to implement and useful for a large number of design matrices. In addition, it must rearrange the design matrix by finding 1) the most triangular matrix, and 2) the most diagonal matrix. In this paper the degree of "triangular" and "diagonal" is based on the number of off-diagonal elements. The larger the number of non-zero diagonal elements, the larger the degree of "diagonal". The smaller the number of non-zero elements above the diagonal, the larger the degree of "triangular". These measures are implicitly included in the algorithm.

To achieve the aforementioned objectives, the EA is divided in two phases. First, the EA rearranges the design matrix to obtain the most triangular one (Fig. 2). Second, for rectangular matrices, it selects the best DPs or FRs to be eliminated and obtains the most diagonal matrix (Fig.1).

The algorithm is as follows:

- First phase (valid for rectangular and square matrices):

(i) Find the row with the minimum number of non-zero elements and, at least, with one non-zero element. We call this row $R(i)$, where i is the row index in the original matrix. See Fig. 2 (1).

(ii) Once the row $R(i)$ is found in step (i), proceed as follows.

  a. For each non-zero element present in $R(i)$, extract the associated column. We call these columns $C(i,j)$, where i is the row index of $R(i)$ and j is the column index. For each column $C(i,j)$ obtain the number of non-zero elements that are in the lower part of this column (i.e., if $C(i,j)_k$ is the kth element placed in the column, count up the number of non-zero elements with $k \geq i$). We call this number $L[C(i,j)]$.
  b. Among the columns selected in the previous step, take the one with the greatest value of $L[C(i,j)]$. We call its column index $C_0(i,j)$. See Fig. 2 (1).
  c. If there are more than one $C_0(i,j)$, because there are two or more columns with the same value for $L[C(i,j)]$, then it is necessary to find the column $C_0(i,j)$ with the least number of non-zero elements above the diagonal element. Therefore, we use the number $U[C(i,j)]$ obtained from counting up the number of non-zero elements $C(i,j)_k$ with $k \leq i$.
  d. Among the columns selected in the previous step, take the one with the lowest value of $U[C(i,j)]$. We call this index $C_0(i,j)$ See Fig. 2 (5).

(iii) Put the row $R(i)$ and the column $C_0(i,j)$ first in the matrix.

(iv) Excluding the first row and column, repeat the steps (i) to (iv) over the resultant sub-matrix.

(v) Repeat the process until there are no more sub-matrices to be analyzed.

-Second phase. (Valid only if the matrix is rectangular.)

For rectangular matrices, the number of DPs is not equal to the number of FRs. The procedure is first explained for the case where #DP > #FR. The explanation for the case where #DP < #FR can be found at the end of this section. Note that, if #DP > #FR, there are more columns than rows (i.e., there are #DP-#FR extra columns).

(i) Check if one of the extra columns can reduce the number L[*C*(*i,j*)] of any column that belongs to the square part of the matrix (i.e., the sub-matrix that accomplishes #DP = #FR) without increasing the number U[*C*(*i,j*)].

(ii) Permute columns that fulfill the anterior condition. See Fig. 3.

	1	2	3	4
1	X	X		
2			X	
3		X	X	X

$R(i)$=2,$C_0(i,j)$=3
(1)

	3	1	2	4
2	X			
1		X	X	
3	X		X	X

Rearrangement
(2)

	3	1	2	4
2	X			
1		X	X	
3	X		X	X

$R(i)$=1,$C_0(i,j)$=2
(3)

	3	2	1	4
2	X			
1		X	X	
3	X	X	X	X

Rearrangement
(4)

	3	2	1	4
2	X			
1		X	X	
3	X	X		X

$R(i)$=3,$C_0(i,j)$=4
(5)

	3	2	4	1
2	X			
1		X		X
3	X	X	X	

Final
Rearrangement
(6)

Figure 2. Phase 1 of EA.

	3	2	4	1
2	X			
1		X		X
3	X	X	X	

Change
column 2 by 1
(1)

	3	1	4	2
2	X			
1		X		X
3	X		X	X

Final
Rearrangement
(2)

Figure 3. Phase 2 of EA.

These two phases allow the EA to rearrange the design matrix in two ways. One way finds the most triangular matrix and the other way finds the most diagonal matrix. Whether or not to use the second phase of the procedure depends on the particular design problem. As an example of the process, consider the matrix present in Fig. 4.

	1	2	3	4	5
1	X				
2		X	X	X	X
3		X			
4		X		X	X
5	X			X	X

Figure 4. Example matrix.

In this matrix, step (i) finds the row with maximum number of zeros, which is the first (*i*=1). There is only one non-zero element in *i*=1, so the first column (*j*=1) is the only one to be compared in the step (ii). For this column, *L*[*C*(1,1)]=1 and *U*[*C*(1,1)]=0. As a consequence, the first row and column do not move in the step (iii).

Step (iv) states that the previous steps must be repeated for the sub-matrix represented in Fig. 5. In this case, the selected row is *i*=3, which only has one non-zero element. Again, for the same reasons explained above, the chosen column is *j*=2, which has *L*[*C*(3,2)]=1 and *U*[*C*(3,2)]=1. Fig. 6 shows the new rearrangement of the matrix.

	1	2	3	4	5
1	X				
2		X	X	X	X
3		X			
4		X		X	X
5	X			X	X

Figure 5. Sub-matrix used in an intermediate step of EA.

	1	2	3	4	5
1	X				
3		X			
2		X	X	X	X
4		X		X	X
5	X			X	X

Figure 6. Result of an intermediate step of EA.

Once Again, step (v) gives the new matrix shown in Fig. 7. Finally, we can see that in order to obtain a decoupled design, only one element, which corresponds to (*i,j*)=(5,4) in the original matrix, must be eliminated.

	1	2	3	4	5
1	X				
2		X	X	X	X
3		X			
4		X		X	X
5	X			X	X

Original matrix

	1	2	4	5	3
1	X				
3		X			
4		X	X	X	
5	X		X	X	
2		X	X	X	X

Rearranged matrix

Figure 7. Original matrix and matrix rearranged by EA.

If we want to analyze a rectangular matrix, the algorithm is still valid. Suppose a rectangular matrix like the one in Fig. 8. After the rearrangement we can obtain the two matrices in Figs. 9 (matrix A) and 10 (matrix B). The first one is obtained after executing the first phase of the algorithm, and the second one, after executing the second phase.

	1	2	3	4	5	6
1	X		X	X		X
2		X			X	X
3			X		X	
4		X		X		X
5	X			X	X	X

Figure 8. Example matrix.

Matrix A is the most triangular matrix and matrix B is the most diagonal one. Matrix B indicates that, in the design problem, the designer should try to eliminate DP number 6, because it has the largest number of non-zero terms. However, for matrix A, it is more interesting to eliminate column 2 because this has fewer non-zero elements than column 6. Of course, in a real design process, the final decision will depend on the particular problem, but the proposed process will give information about the best design parameter to be eliminated. The same procedure can be executed for a matrix with more FRs than DPs. In this case, the algorithm will give us information about which one is the most coupled FR.

	5	2	4	1	3	6
3	X		X	X		X
2		X			X	X
4			X		X	
5		X		X		X
1	X			X	X	X

Original matrix

	5	6	4	1	3	2
3	X				X	
2	X	X				X
4		X	X			X
5	X	X	X	X		
1		X	X	X	X	

(A) Triangular rearrangement

Figure 9. Rearrangement obtained in the first phase of EA: most triangular matrix.

	5	2	4	1	3	6
3	X		X	X		X
2		X			X	X
4			X		X	
5		X		X		X
1	X			X	X	X

Original matrix

	5	2	4	1	3	6
3	X				X	
2	X	X				X
4		X	X			X
5	X		X	X		X
1			X	X	X	X

(B) Diagonal rearrangement

Figure 10 Rearrangement obtained in the second phase of EA: most diagonal matrix.

The EA works especially well for those matrices that are large and strongly coupled, like the design matrix shown in Fig. 11. The result of the algorithm for this matrix appears in Fig. 12 where it is possible to see that the design problem is coupled only due to two non-zero elements over the main diagonal.

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

Figure 11. Example matrix.

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

Figure 12. Rearranged matrix obtained by EA from design matrix in Fig. 11.

Note that phase two for the case where #FR > #DP is similar to the one previously described. However, the *L* and *U* numbers must be redefined as follows.

-For each row $R(i,j)$ obtain the number of non-zero elements that are in the left side of this row (i.e., if $R(i,j)_k$ is the *k*th element placed in the row, count up the number of non-zero elements with $k \leq i$). We call this number $L[R(i,j)]$.

- The number $U[R(i,j)]$ is obtained from counting up the number of non-zero elements $R(i,j)_k$ with $k \geq i$.

The above procedure must be changed as follows:

(i) Check if one of the extra rows can reduce the number $L[R(i,j)]$ of any row that belongs to the square part of the matrix (i.e., the sub-matrix that accomplishes #DP = #FR) without increasing the number $U[R(i,j)]$.

(ii) Permute rows that fulfill the anterior condition.

4 DISCUSSION

Various authors have already studied matrix rearrangement as we mentioned above. The present discussion is useful to find the advantages and disadvantages of the proposed algorithm. For this purpose, the N. P. Suh [1990], T. Lee [2006], and Acclaro [2010] algorithms are used in the following subsections.

4.1 T. LEE ALGORITHM

We discuss the T. Lee [2006] algorithm first. For this purpose we use the design matrix in Fig. 13, which is an example obtained from Lee and Jeziorek [2006].

	1	2	3	4	5
1	X			X	
2	X	X			X
3		X	X	X	
4		X		X	X
5	X		X		X

Figure 13. Original matrix.

Applying the optimal strategy for eliminating coupling terms, T. Lee obtains a triangular matrix and only two non-zero off-diagonal elements (1,4) and (5,3). Fig. 14 shows the resulting matrix.

	1	2	3	4	5
1	X			X	
2	X	X			X
3		X	X	X	
4		X		X	X
5	X		X		X

Original matrix

	1	5	2	4	3
1	X			X	
5	X	X			X
2	X	X	X		
4		X	X	X	
3			X	X	X

Rearranged matrix by T. Lee algorithm

Figure 14. Rearrangement obtained with the optimum algorithm developed by T. Lee [2006].

Our EA leads to the matrix shown in Fig. 15. According to the EA, it is necessary to eliminate three terms to achieve a triangular matrix instead of two as T. Lee algorithm indicates. This shows that our EA is not an optimum algorithm.

	1	2	3	4	5
1	X			X	
2	X	X			X
3		X	X	X	
4		X		X	X
5	X		X		X

Original matrix

	1	5	2	4	3
1	X				X
5	X	X		X	
2		X	X		X
4	X		X	X	
3		X		X	X

Rearrange matrix by EA

Figure 15. Rearrangement obtained with the EA.

The difference between the T. Lee and EA algorithms comes from an arbitrary decision adopted by the EA: when there are two or more rows with the same number of non-zero elements, the selected row is always the first one. Perhaps, the second one might be a better option, but the algorithm does not have any criteria to fix it. This means that the EA is not an optimum algorithm. To solve this problem, it is necessary to run the algorithm for all of the indecisions presented in the design matrix, by selecting a different row in each case. For example, the EA obtains the optimal result reported by T. Lee if the rows are shifted down one step (i.e., if the original matrix had the rows in the order 5,1,2,3,4). For this matrix, the EA gives the optimal result provided by the T. Lee method.

The T. Lee method can rearrange square matrices that have the main diagonal completely filled with non-zero elements. In this procedure, the Xs in the design matrix (DM) are replaced with ones. Matrix A is equal to the transposed design matrix minus the identity matrix of a size m x m ($A=DM^T-I$). If the DM is rectangular, the numbers of FRs and DPs are not equal. Consequently, matrix B cannot be constructed because the value of m is not well defined (m = vertices of graph = FR-DP pairs). This problem exists because there is at least one FR that does not have a free DP to form a pair (or vice versa). If the diagonal is not completely filled with non-zero elements, matrix A will have some negative element in the diagonal (i.e., if $DM(i,i)=0$ then $DM^T(i,i)-I(i,i)=-1$). In this case, the digraph will not have the corresponding vertex. Although EA is not an optimum algorithm, it can deal with rectangular matrices with blanks in the diagonal. In some cases, this can be an advantage that compensates for the fact that it is not an optimal algorithm.

4.2 Acclaro Algorithm

Acclaro Software [2010] also solves square matrices that have non-zero elements in the main diagonal. In other cases, Acclaro considers these cases doubtful. As a consequence, it is not possible to use Acclaro for rearranging rectangular matrices or square matrices with zeros in the main diagonal.

If we compare the EA with Acclaro we obtain different results depending on the matrix considered. Due to the reasons given above, differences are obtained in matrices with one or more diagonal elements equal to zero and in rectangular matrices. Some different results are obtained for strongly coupled matrices like the one in Fig. 16. Acclaro gives a rearranged matrix exposed in Fig. 17 where five elements are over the main diagonal. For this case, the EA provides the matrix in Fig. 18 that has four non-zero elements above the main diagonal.

	1	2	3	4	5	6
1	X		X	X		X
2	X	X			X	
3			X		X	X
4	X			X		
5		X	X	X	X	
6	X			X		X

Figure 16. Original matrix.

	1	2	3	4	5	6
1	X		X	X		X
2	X	X			X	
3			X		X	X
4	X			X		
5		X	X	X	X	
6	X			X		X

Original matrix

	1	4	6	2	3	5
1	X	X	X		X	
4	X	X				
6	X	X	X			
2	X			X		X
3			X		X	X
5		X		X	X	X

Matrix rearranged by Acclaro

Figure 17. Rearrangement obtained with Acclaro Software [2010].

	1	2	3	4	5	6
1	X		X	X		X
2	X	X			X	
3			X		X	X
4	X			X		
5		X	X	X	X	
6	X			X		X

Original matrix

	1	5	3	4	2	6
4	X			X		
2	X	X			X	
3		X	X			X
1	X		X	X		X
5		X	X	X	X	
6		X		X		X

Matrix rearranged by EA

Figure 18. Rearrangement obtained with the EA.

4.3 N. P. SUH ALGORITHM

The N. P. Suh algorithm [1990] accepts rectangular matrices and non-zero elements in the main diagonal. However, it presents some decision problems when all of the rows in the matrix have the same number of non-zero elements. To discuss the N. P. Suh algorithm we use the example given in [Suh, 1990, p. 116]. The design given is redundant (see Fig. 19), because there are more DPs than FRs.

	1	2	3	4	5	6	7
1	X		X			X	X
2	X	X		X			X
3	X	X			X	X	

Figure 19. Original matrix.

This matrix is very particular because it has the same number of non-zero elements in each row. This fact makes Suh's algorithm leaves the matrix unaltered. However, as Suh [1990] discussed, it is possible to select three DPs and "freeze" the unnecessary ones. Depending on which three parameters are chosen, there are different square matrices that solve the problem. They are collected in Fig. 18. One of them is uncoupled and the others are decoupled.

	3	4	5
1	X		
2		X	
3			X

	7	2	5
1	X		
2	X	X	
3		X	X

	1	2	5
1	X		
2	X	X	
3	X	X	X

Figure 20. Three possible rearrangements given by N. P. Suh [1990].

When the matrix in Fig. 19 is introduced in the EA, the result depends on whether we have decided to use the second part of the EA or not. When the objective is to find the most triangular matrix, the second phase of the EA is not executed and the result is the matrix in Fig. 21. When the second phase of the algorithm is executed the resulting matrix is the one in Fig. 22, which is the most diagonal one that the EA can achieve. Both matrices are the first and the third ones in Fig. 18. In the first case (the most diagonal matrix) it would be convenient to eliminate DPs 1, 6 and 7 for obtaining a total diagonal matrix. In the second case (the most triangular matrix), it would be convenient to eliminate DPs 3, 6 and 7 for obtaining a total triangular matrix.

	1	2	3	4	5	6	7
1	X		X			X	X
2	X	X		X			X
3	X	X			X	X	

Original matrix

	1	2	5	4	3	6	7
1	X				X	X	X
2	X	X		X			X
3	X	X	X			X	

Triangular rearrangement

Figure 21. Rearrangement with EA: most triangular (phase 1).

	1	2	3	4	5	6	7
1	X		X			X	X
2	X	X		X			X
3	X	X			X	X	

Original matrix

	3	4	5	2	1	6	7
1	X				X	X	X
2		X		X	X		X
3			X	X	X	X	

Diagonal rearrangement

Figure 22. Rearrangement with EA: most diagonal (phase 2).

5 CONCLUSION

This article proposes an algorithm, based on an extension of N.P. Suh's algorithm, which allows the design matrix to be rearranged in two different ways: the most triangular matrix and the most diagonal matrix. When the design matrix is rectangular, it also indicates the best DPs or FRs to be removed. The procedure is based on a comparison of three numbers: the number of non-zero elements in each row, the number of non-zero elements in each column below the diagonal element, and the number of non-zero elements in each column above the diagonal element. Due to the simplicity of the procedure, the extended algorithm is valid for a large number of design matrices, and it is especially useful for strongly coupled matrices, i.e., for matrices with a large number of DPs and FRs, and a large number of non-zero elements.

More complex algorithms, like the optimum procedure described by T. Lee, solve the problem more accurately; however, they can only deal with square matrices whose elements in the main diagonal are all non-zero. Simpler algorithms, like the one described by N. P. Suh, are easier to implement and hence they can deal with more general matrices, such as rectangular matrices with empty elements in the main diagonal. However, it cannot achieve a final result if the number of elements in each row is the same. This paper shows that it is possible to find a trade-off between both characteristics.

Simplicity is maintained because the procedure is based on changing the relative position of rows and columns with a decision criterion based on the number of non-zero elements in different positions of the rows and columns. Thus, this decision criterion is direct, and hence the new decision structure allows matrix elements to be moved directly by reordering rows and columns. This can be an advantage if the design matrix is large or if it is very populated.

Therefore, this paper shows that the proposed algorithm is halfway between the optimal and simplest algorithms. We conclude that it can be used to recognize when a matrix is coupled, decoupled or uncoupled. It can deal with large and very populated rectangular matrices without elements in the main diagonal. However, this paper shows that the proposed algorithm is not optimal, and hence optimal algorithms, like the one described by T. Lee, are needed.

6 ACKNOWLEDGMENTS

This work has been carried out under the contract P100130354 between ALTRAN Technologies and Universidad Politécnica de Madrid. We are grateful to ALTRAN Technologies for the financial support.

7 REFERENCES

[1] Acclaro DFSSS, Software by Axiomatic Design Solutions Inc., www.axiomaticdesign.com, 2010.

[2] Cai, C.-L., *et al.,* "The method for analysing and disposing of functional interaction in axiomatic design", *Mechanical Engineering Science,* 224, Part C, 2009.

[3] Lee, T., "Optimal strategy for eliminating coupling terms from a design matrix"*, Integrated Design and Process Technology, June,* 10(2), 45-55, 2006.

[4] Lee, T. and Jeziorek, P., "Understanding the value of eliminating an off-diagonal term in a desing matrix", *4th International Conference on Axiomatic Design*, Firenze, 2006.

[5] Su, J. C.-Y. *et al.*, "A structured approach to measuring functional dependency and sequencing of coupled tasks in engineering design". *Computer and industrial engineering*, 45, 195-214, 2003.

[6] Suh, N. P., *Axiomatic Design: Advances and Applications,* New York: Oxford University Press, 2001.

[7] Suh, N. P., *The Principles of Design*, New York: Oxford University Press, 1990. ISBN 0-19-504345-6.