Studies in Engineering and Technology
Vol. 8, No. 1; August 2021
ISSN 2330-2038 E-ISSN 2330-2046
Published by Redfame Publishing
URL: http://set.redfame.com

Mechanical System Topology Optimization for Better Maintenance

Ngnassi Djami Aslain Brisco[1], Nzié Wolfgang[2], Doka Yamigno Serge[3]

[1]PhD Candidate, National Advanced School of Agro-Industrial Sciences, University of Ngaoundere, Cameroon

[2]Department of Basic Sciences and Techniques for the Engineer, EGCIM, University of Ngaoundere, Cameroon

[3]Department of Physics, Faculty of Science, University of Ngaoundere, Cameroon

Correspondence: Ngnassi Djami Aslain Brisco, PhD Candidate, National Advanced School of Agro-Industrial Sciences, University of Ngaoundere, Cameroon.

Received: March 23, 2020 Accepted: April 24, 2020 Online Published: September 1, 2020

doi:10.11114/set.v8i1.4990 URL: https://doi.org/10.11114/set.v8i1.4990

Abstract

For a given mechanical equipment, knowing its modular topology has the advantage of facilitating its maintenance. Indeed, during a maintenance problem, we will not act on the whole product except on the failed module (product subsystem) and we would also gain time to detect, diagnose and compensate for the observed failure. On the other hand, the clustering algorithm, which has served as a reference for several works has several limits. It generates much more complex and more expensive modules in terms of coupling costs, which could require more resources, more intervention time and more maintenance work. This has worse consequences for product maintenance, because the more complex the product modules are, the more expensive the maintenance is. We therefore propose an improved clustering algorithm which has the advantage of reducing maintenance costs by reducing the coupling and decoupling costs (Disassembly and reassembly costs) of the modules, generated by the reference algorithm for good maintainability (dis-assemblability). The application is made on a soy roaster. The approach followed in the proposed algorithm consists first of all in defining a DSM (Design Structure Matrix) which will make it possible to define the correction coefficients of the coupling cost, then in formulating an objective function to reduce the coupling costs, and finally to take into account the integrating elements to reduce the size of the modules. The result achieved is the proposal for a modular topology (modular architecture) leading to a significant reduction in maintenance costs. The developed algorithm also allows an economy of scale in reducing the complexity of the modules, promoting good maintainability.

Keywords: design structure matrix, topology, module strength indicator, coupling cost, maintenance, modularity

1. Introduction

The algorithms proposed by Yu et al. (2003, 2007) have the advantage of a better convergence towards the optimum than the algorithm of Thebaud but a computation time with that can be longer (the authors do not provide these comparative data). Source programs are not available and difficult for a good design engineer to understand.

The algorithm used is based on the evaluation of the couplings between elements of the same domain to generate in a systematic way a grouping in modules of these same elements by optimizing an "objective function". It only manages the positive values of the couplings. Ngnassi et al. (2020) implemented the Idicula algorithm methodology where they opted for a more generic name for the "objective function" independent of the field studied, namely: "total cost of coupling". Idicula's algorithm allows modules to be stochastically produced from a DSM. However, we have found that this algorithm is cumbersome to use and depends on the control parameters which can be difficult to manipulate in order to find an optimal combination.

In this paper, we propose a new clustering algorithm which uses the matrix representation formalism to implement the identification of product domain topologies. The principle adopted is to use DSM (Design Structure Matrix) as input to offer optimal product topologies. Our contribution concerns the improvement of an existing clustering algorithm which has served as a reference for several previous works such as those of (Idicula, 1995); (Fernandez, 1998) and (Thebeau, 2001). We then propose several improvements to this algorithm, based on our bibliographic analysis on modular design and modularity indices. Finally, we compare the improved algorithm to the starting algorithm using as a criterion the reduction achieved in module coupling cost with the proposed algorithm.

2. Methodology

In the following, we will present the improved clustering algorithm and how it works.

It is first necessary to define the qualities expected from a good clustering algorithm.

In general, the identification of a topology is a multi-criteria decision problem. This involves taking into account the complexity of the design linked to this topology. The principle of a semi-decomposable system formulated by Simon (1997) and the axioms of the axiomatic concept goes in this direction and tend to favor the creation of subsystems which are as much as possible decoupled. It is a question of limiting the technical diversity by standardizing the components and making economies of scale in production. This is to ensure that the system will be easily maintainable. In summary, all stakeholders in the life cycle of the system will specify requirements on its topology. The MFD (Modular Function Deployment) modularization method tries to take into account several criteria to propose a module, but this method remains manual and subjective. Clustering algorithms generally seek to optimize an "objective function" which takes into account only the couplings of a domain of the system (for example, functional or organic) (Thebaud, 2001; Whitfield et al., 2002; Yu et al., 2003).

Regarding the rest of this work, we will define an optimal topology as the solution minimizing the "objective function" of a clustering algorithm.

Topology optimization is a phrase used to characterize design optimization formulations that allow for the prediction of the lay-out of a structural and mechanical system (Martin and Ole, 2004).

The fundamental idea of the material distribution technique for topology design is to associate to each pixel (or voxel) of a raster representation an independent design variable that defines the amount of material utilized in that subset of the domain (the raster representation is a discretization of a material distribution in a continuum formulation). Martin and Ole (2004) for simplicity, used the FEM mesh for analysis as the basis for the raster representation. For the topology design problem of determining which elements of the analysis domain should be part of the final structure, they formulated an optimization problem on the reference domain Ω in the following form:

$$\begin{aligned}&\min_{u,\rho}\left\{u_{out}=I^{T}u\right\}\\&s.t.:K\left(\rho\right)u=f\\&\sum_{e=1}^{N}v_{e}\rho_{e}\leq V\\&\rho_{e}\in\left\{0,1\right\},e=1,\dots,N\end{aligned} \tag{1}$$

ρ_e are design variables (one real variable per element in the FEM mesh) are discrete valued, $\rho_e \in \{0, 1\}$.

K is the stiffness matrix, *f* is the load vector and *u* the displacement vector. Moreover, **l** is a vector for which the inner product with **u** produces the relevant output displacement *u* out (l is interpreted as a (unit) load vector).

This is thus a FEM format of the maximum output compliant mechanism problem for a given input load and a prescribed volume V ($v_e$denotes the volume of element e).

The topology includes good modularity, but the distribution of these modules in an optimal volume (by reducing the size of complex modules). That is why Eric (2008) characterizes the expected qualities of an algorithm for a good module topology in the following way:

- S1: Converge repeatedly (reproducibility) and quickly (calculation time) towards the optimal solution, from the point of view of the "objective function" chosen;
- S2: Favor dense modules (density of modules);
- S3: Penalize large modules (number of elements in a module) because they can be complex to design and require many interfaces (difficult integration);
- S4: Be easy to program or modify;
- S5: Require few parameters to set;
- S6: To be understandable by its user. The principles of calculation must be easy to explain. The result must be easy to interpret.

The algorithm developed by Thebaud satisfies the specifications S1, S4 and S6. We will propose modifications to improve this algorithm on the other criteria.

In our approach to improving the reference algorithm, we integrated fundamental concepts of modular topology in order to guide the functioning of the algorithm.

The changes we have made relate to four specific points:

- The DSM;
- The MSI;
- The "objective function";
- Identification of integrating elements.

The flowchart of the new clustering algorithm that we propose is given in Figure 1.

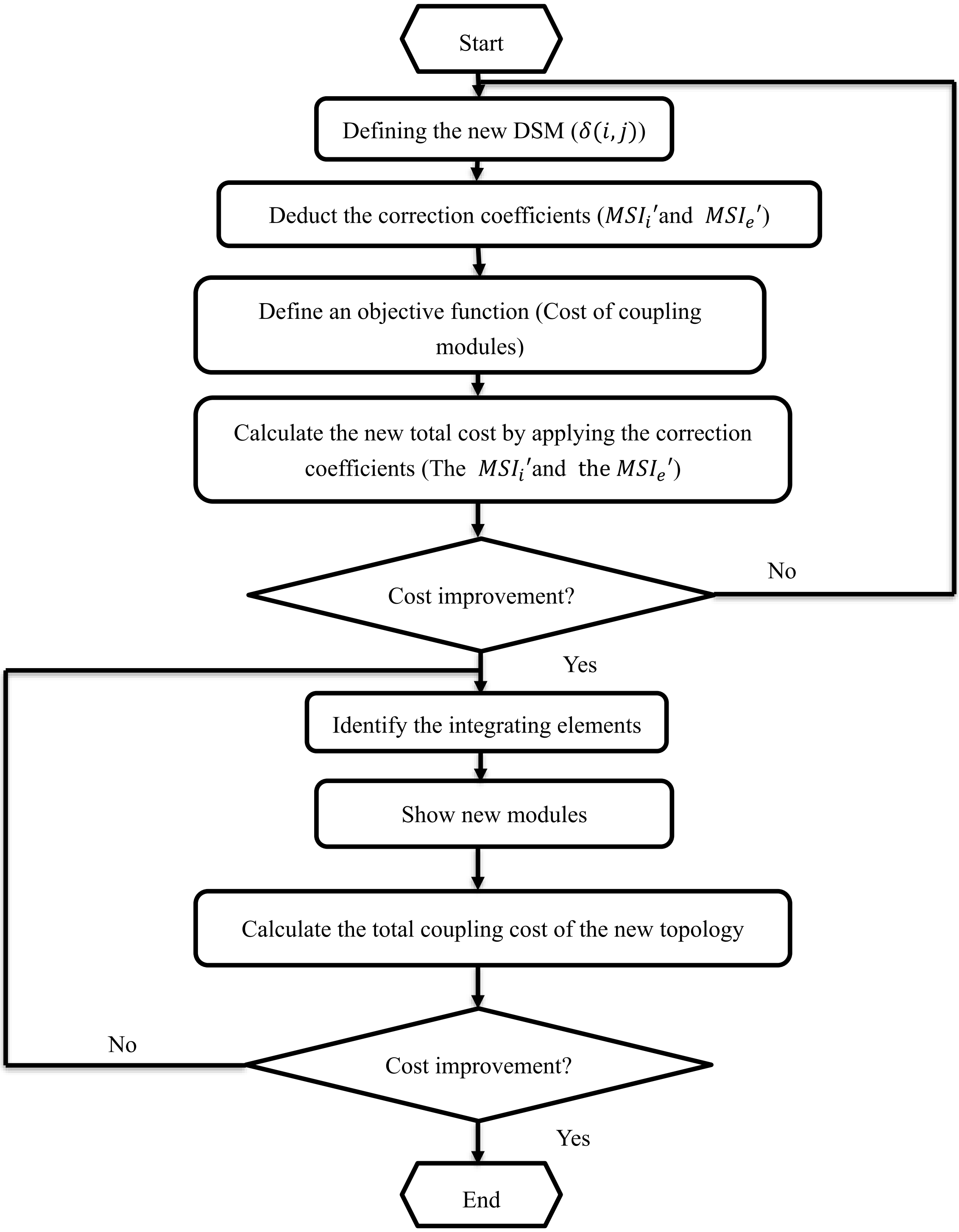
Start
Defining the new DSM ($\delta(i,j)$)
Deduct the correction coefficients ($MSI_i{}'$and $MSI_e{}'$)
Define an objective function (Cost of coupling modules)
Calculate the new total cost by applying the correction coefficients (The $MSI_i{}'$and the $MSI_e{}'$)
Cost improvement?
No
Yes
Identify the integrating elements
Show new modules
Calculate the total coupling cost of the new topology
No
Cost improvement?
Yes
End

Figure 1. Flowchart of the improved clustering algorithm

2.1 Structural Matrix of Modified Design

The first modification of the basic algorithm concerns the structural design matrix. Using the modifications made by Ghassen (2007) and Eric (2008), the transformations of the elements of the DSM of reference that we propose are given in Table 1.

Table 1. Modifications to the structural design matrix.

Intensity of interactions	**Characteristics of interactions in the structural design matrix**
DSM(i, j)	Coupling value between elements $\boldsymbol{C_i}$ and $\boldsymbol{C_j}$. Note that where i = j, DSM (i, j) = 0.
$\boldsymbol{\delta(i,j)}$	Takes the value **1** when ***DSM (i, j) ≠ 0*** and the value **0** when ***DSM (i, j) = 0***.

The structural design matrix of the basic algorithm takes into account the intensity of the interactions between the elements (DSM (i, j)), the new structural design matrix that we propose takes into account only the existence or not of the interactions.

2.2 Corrective Coefficients of the Module Coupling Cost

The second modification concerns the Strength Indicator Module (MSI). Ngnassi et al. (2020) apply the basic algorithm developed by Idicula (1995), and observe the formation of large modules (including less dense inside), which are more restrictive for maintenance. To this end, we propose a method to avoid the formation of "too large" modules (forming denser modules inside). For this, we correct the coupling costs of the previous algorithm by two correction coefficients, which we call MSIi 'and MSIe'.

For the calculation of these coefficients, we focus on two characteristics of a modular topology:

- A module is the grouping of elements interacting strongly with each other;
- The elements belonging to a module interact weakly with the elements belonging to other modules.

In light of these two remarks, we were inspired by the modularity metric proposed by (Whitfield et al., 2002).

The MSIi 'part (for density of interactions inside) takes into account the internal interactions in a module and therefore gives a measure of the module's internal cohesion and Strength.

The MSIe 'part (for density of interactions outside) takes into account the external interactions which link the elements belonging to the module to the other elements external to the module, thus MSIe' measures the strength of the interactions external to the module.

It appears through the two definitions that MSIi 'and MSIe' are two conflicting indicators.

This is why the MSI' (Module Strength Indicator corrected for the module in question) is written as the difference between the two.

In this paper, we will not directly use the MSI indicator', but a form derived from the two indicators MSIi' and MSIe'. While these two original indicators take into account the intensity of interactions, the indicators we propose only take into account the existence or not of interactions. We then obtain the indicators MSIi 'and MSIe' presented in equations 2 and 3.

$$MSI_i^{'}\left(M_k\right)=\frac{\sum_{i=n_1}^{n_2}\sum_{j=n_1,j\neq i}^{n_2}\delta(i,j)}{\left(n_2-n_1+1\right)^2-\left(n_2-n_1+1\right)} \tag{2}$$

$$MSI_e^{'}\left(M_k\right)=\frac{\sum_{i=1}^{n_1-1}\sum_{j=n_1}^{n_2}\left(\delta(i,j)+\delta(j,i)\right)}{2\times\left(\left(n_1-1\right)\times\left(n_2-n_1+1\right)\right)}+\frac{\sum_{i=n_2+1}^{size(DSM)}\sum_{j=n_1}^{n_2}\left(\delta(i,j)+\delta(j,i)\right)}{2\times\left(\left(size(DSM)-n_2\right)\times\left(n_2-n_1+1\right)\right)} \tag{3}$$

With:

- Size (DSM): function which returns the size of a module or of the DSM, ie number of elements in a DSM sub-matrix;
- n_1: index of the first element of the module M_k ;

- n_2: index of the last element of the module $M_k(n_2 \geq n_1)$. If $n_2 = n_1$, the module actually contains only one element.

Note that the denominator of the MSIi’ corresponds to the total number of possible interactions of the module considered, outside the diagonal and that we only consider symmetrical DSMs.

The MSI indicator was developed for temporal DSMs and more specifically for parameter DSMs such as those developed in the basic clustering algorithm and which are inspired by design parameters. We then propose an indicator MSI 'developed for digital DSMs which take their values between 0 and 1. We then obtain indicators MSIi' and MSIe' having their values between 0 and 1. Because the indicators MSIi' and MSIe' take taking into account interactions worth 1 or 0, MSIi' and MSIe' have a value bounded by 0 and 1.

Before explaining the use of these indicators in the cost function, we will study the behavior of these two indicators.

The MSIi indicator is maximum and is worth 1, when the module is completely dense, that is to say, all the elements that compose it interact. The indicator is zero, when the elements that make up the module do not interact with each other at all (which calls into question the use of the term module to qualify this grouping of elements).

The MSIe indicator is zero when the module concerned does not interact with any external element (it is completely decoupled). It is maximum when each of the elements that compose it interacts with all the elements external to this module. This last configuration is also contrary to the application of the term module to a grouping of elements.

Thus, a grouping of elements tends to become a perfect module when its MSIi tends towards 1 and its MSIe tends towards 0.

2.3 The Cost Function

The third modification concerns the "objective function" which is the determining element of the clustering algorithm. Note that its formulation corresponds to a modeling of what the expert who uses the algorithm can consider as an "optimal topology".

If we take up the "objective function" chosen in the work of Ngnassi et al. (2020), we note that we penalize external interactions more strongly than internal interactions. This configuration can lead, according to the example treated, to the creation of relatively large modules, even if the size of the module is penalized. Thus, to adjust and correct the "objective function" , we will draw inspiration from the work of (Ghassen, 2007) and (Eric, 2008). These works do not give constraints on the correction coefficients and do not integrate these constraints in the formulation of the objective function. We therefore formulate an objective function, defined by equation 4, which takes into account the constraints on the correction coefficients of the module coupling cost.

$$\begin{aligned} &minimize \quad && Total\ Coupling\ Cost = \sum_i \sum_j Total\ Coupling\ Cost(i,j) \\ &s.t: && MSI_i^{'} \prec MSI_i \\ & && MSI_e^{'} \prec MSI_e \end{aligned} \tag{4}$$

The corrections we make with this new objective function concern the internal coupling cost and the external coupling cost, for a module considered.

Thus, the corrections made are given by equations 5 and 6.

$$Corrected\ Internal\ Coupling\ Cost(M_k) = \ Internal\ Coupling\ Cost(M_k) / MSI_i^{'}(M_k) \tag{5}$$

$$Corrected\ External\ Coupling\ Cost(M_k) = \ External\ Coupling\ Cost(M_k) \times MSI_e^{'}(M_k) \tag{6}$$

These two corrections are interpreted as follows:

- by dividing the internal coupling cost by MSIi’, we favor the creation of rather dense than large modules;
- by multiplying the cost of external coupling by MSIe, we promote the minimization of interactions external to the modules.

Note the double logic of the new "objective function":

- the more MSIi 'increases, the denser the module, the lower the cost;
- the more MSIe 'increases, the more interactions there are outside the module, the more the cost increases.

The Total Corrected Cost is then written:

$$Total\ Corrected\ Coupling\ Cost = \sum_{M_k}^{Modules} \left(Corrected\ Internal\ Coupling\ Cost\left(M_k\right) + Corrected\ External\ Coupling\ Cost\left(M_k\right)\right) \tag{7}$$

In order to show the relevance of this last "objective function", we will compare in the results part of this paper our algorithm and the reference algorithm.

2.4 Identification of Integrating Elements

A final modification concerns the specific treatment that we reserve for integrating elements. Ghassen (2007) defines an integrating element as an element which does not belong to any module and which interacts with elements belonging to several modules. However, this definition poses a problem insofar as the basic topology does not include elements that do not belong to any module. Indeed, any module contains at least one component, so a single component defines its own module. To build small modules, we can consider an element already belonging to a module as an integrating element insofar as it interacts strongly with another module. We therefore define an integrating element as an element belonging to a module and which interacts with elements belonging to one or more modules.

The purpose of clustering algorithms is to create modules and therefore try to insert any element inside a module. The cost functions (that of Thebaud (2001) and or that which we propose) cannot guarantee that the integrating elements will be placed outside the other modules, which in the end, does not conform to the vision of the architect and disrupts the convergence of the algorithm.

This is why we propose to calculate a coupling index (IC) to identify "automatically" the integrating elements, defined as those which interact strongly in number with the other elements (Equation 8). We also offer the possibility for the user to manually classify an element that he considers to be integrative.

$$IC\left(element_j\right) = \frac{\sum_{i=1, i\neq j}^{size(DSM)} \delta\left(i, j\right)}{size\left(DSM\right) - 1} \tag{8}$$

The IC measures the coupling rate between an element and the rest of the system. To take into account the architects' judgment, Ghassen (2007) fixes the IC at 60%. Beyond this threshold, the elements are considered to be integrative. Eric (2008), on the other hand, offers the user the freedom to define the IC threshold.

3. Results

This part illustrates the application of the clustering algorithm proposed on the same equipment used in the work of Ngnassi et al. (2020): the soybean roaster. This choice was made with the aim of being able to easily justify our results. This therefore assumes that we still have a 19-component soy roaster and that we start from the modular topology materialized by the DSM parameter as indicated in the work of (Ngnassi et al., 2020).

3.1 Structural Design Matrix

By applying the modifications made to the reference DSM (DSM parameter) of Ngnassi et al. (2020), we obtain the digital DSM for our 19-component product given in Figure 2.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}
C_1	0			1											1				
C_2		0		1		1													
C_3			0	1												1			
C_4	1	1	1	0	1												1		
C_5				1	0	1													
C_6		1			1	0										1			
C_7							0			1							1		
C_8								0	1		1								
C_9								1	0	1									
C_{10}							1		1	0									
C_{11}								1			0								
C_{12}												0	1						1
C_{13}												1	0	1					
C_{14}													1	0	1	1			
C_{15}	1													1	0				
C_{16}			1			1								1		0			
C_{17}				1			1										0	1	
C_{18}																	1	0	1
C_{19}												1						1	0

Figure 2. Digital DSM of soybean roaster

The values of the interactions of this matrix will be very useful in the evaluation of the correction coefficients of the coupling costs of the modules of the soybean roaster.

3.2 Correction Coefficients

The correction coefficients of the coupling cost are given by the relations 2 and 3. For reasons of symmetry of the DSM, the relation 3 becomes:

$$MSI_e^{'}\left(M_k\right)=\frac{\sum_{i=1}^{n_1-1}\sum_{j=n_1}^{n_2}2\delta\left(i,j\right)}{2\times\left(\left(n_1-1\right)\times\left(n_2-n_1+1\right)\right)}+\frac{\sum_{i=n_2+1}^{size(DSM)}\sum_{j=n_1}^{n_2}2\delta\left(i,j\right)}{2\times\left(\left(size(DSM)-n_2\right)\times\left(n_2-n_1+1\right)\right)} \tag{9}$$

By using equations 2 and 9, we obtain Figure3 which makes a comparative study of the corrected MSI and that obtained with the reference algorithm. Remember that the Module Strength Indicator (MSI) is obtained by subtracting from the MSIi, the MSIe.

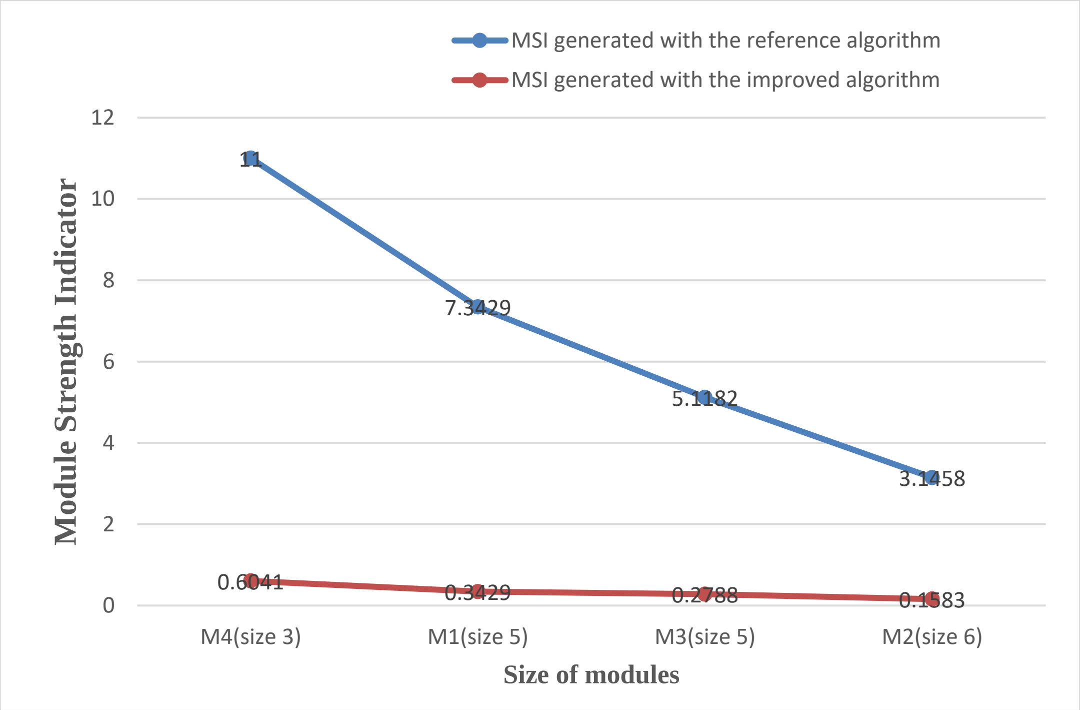

Figure 3. Comparative study of the corrected MSI and that obtained with the reference algorithm

In view of these results, a similar interpretation is made on the degree of modularity as that given by the application of the reference algorithm on the soy roaster in the work of (Ngnassi et al., 2020). Module 4 is the one that penalizes the most external interactions, so it is the easiest to maintain. On the other hand, module 2 is the one which has the weakest cohesive force (weak MSI), therefore it is more difficult to maintain. This interpretation is therefore quite logical because the larger the module, the more difficult it is to maintain. Overall, this step of our algorithm favored the creation of dense modules, and more specifically of module 4 which is the densest with a correction coefficient closer to 1.

3.3 Coupling Cost

Taking into account the correction coefficients (MSIi 'and MSIe') and knowing the coupling costs generated by the basic algorithm, we will evaluate the corrected coupling cost of each module constituting the soy roaster and deduce the new cost overall coupling of soybean roaster modules.

We evaluate the corrective costs defined by relations 5 and 6 where we give the rounded values.

The corrected total coupling cost of the modules of the soybean roaster is given by the relation 10.

$$Total\ Corrected\ Coupling\ Cost = \sum_{i=1}^{4} Corrected\ Coupling\ Cost\ \left(M_i\right) \tag{10}$$

Table 2 summarizes the coupling costs obtained with the dereference algorithm and the proposed new clustering algorithm and shows the reduction obtained in cost.

Table 2. Summary of costs generated by the reference algorithm and the new algorithm

Roaster modules	**Coupling cost** **Reference algorithm**	**Coupling cost** **New algorithm**	**Reduction achieved**
Module 1	140060	18380	80, 87 %
Module 2	74224	29020	60, 9 %
Module 3	29894	12306	58, 83 %
Module 4	666	999	

The Table 2 shows that with the proposed new clustering algorithm, there is a reduction in consequent coupling costs (by more than 50%) with each of modules 1, 2 and 3. On the other hand, module 4 which is the module less restrictive for the design and consequently for the maintenance underwent a cost increase, that is to say by 50%. This results in the simple fact that this last module of the reference architecture has no external coupling cost.

This is justified by the example of calculating the coupling cost proposed on page 77 of the manuscript produced by (Eric, 2008). We can see that the terminal module (last module of the architecture) has an external coupling cost zero, but a non-zero internal coupling cost. However, the application of this calculation principle developed by Eric to our soy roaster with the new proposed alogorithm, allows us to realize that module 4 generates a coupling cost of 999, against a value of 666 obtained with the reference algorithm (Ngnassi et al., 2020). The conclusion which emerges is that the improved algorithm did not favor a reduction in coupling cost of module 4.

Overall, the proposed algorithm has reduced the overall cost of coupling soybean roaster modules by 75.2% (from 244,844 to 60,705). So the proposed topology, materialized by digital DSM is better from the maintenance point of view.

Figure 4 shows the evolution of the coupling cost generated by the new algorithm as a function of the size of the modules of the soybean roaster.

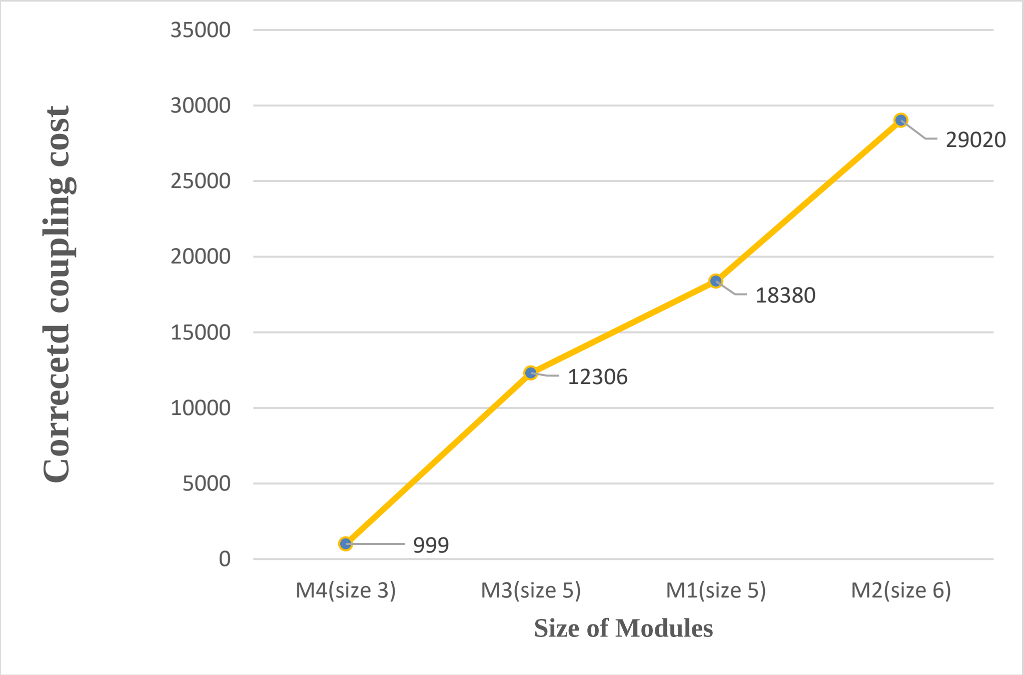

Figure 4. Evolution of the coupling cost corrected according to the size of the modules

With the reference algorithm (Ngnassi et al., 2020), we found that module 1 has the highest coupling cost and clearly higher than that of module 2. However, from a design point of view, module 2 is the most restrictive for designers due to its complexity with a larger size (ie 6 components), which therefore seems to be a problem. The algorithm that we proposed shows in Figure 4 that the cost of coupling is an increasing function of the size of the modules, which is also normal because the design complexity increases with the size of the product to be designed. On the other hand, molds 1 and 3 are the same size (i.e. 5 components), so it is difficult to perceive the problem of maintenance of its modules which will take the most intervention time where that will generate the most costs allocated for maintenance. Suddenly with the algorithm that we propose, it is strong from this observation that of the two modules, module 3 is the least restrictive and requiring less resources for maintenance.

3.4 Identification of the Integrating Elements of the Soybean Roaster

Referring to the reference algorithm deployed in the work of Ngnassi et al. (2020), we see that modules 1 and 2 require more work from the designers, because they have a higher coupling cost. This obviously translates into the fact that they are larger in size.

In order to facilitate the maintenance of these modules, we will take into account the integrating elements which will make it possible to reduce the size.

Considering the digital DSM of the soybean roaster described in Figure 2 we note that of modules 1 and 2, there are two integrating elements: component $C_4$and component C_6.

The first integrating element (C_4) has 5 interactions, including 4 with the first module and 1 with the last module. The second integrating element (C_6) meanwhile has 3 interactions, including 2 with module 1 and 1 with module 3. We therefore calculate the coupling index of the two components (defined in relation 7) to identify them "automatically "as integrating elements. Thus: with component C_4, we have an IC of 71% and with component C_6, we have an IC of 33%. Component C_4 is automatically identified as being integrator, but with regard to the threshold set by Ghassen (2007), component C_6 cannot be considered to be integrative. Since component C_6 does not interact with the components of the module to which it belongs (module 2), it therefore becomes completely normal for it to be integrated into a module where it interacts strongly with the components. We therefore place ourselves in the same posture of Eric (2008) where we leave the choice to the user to set an IC and we therefore classify the component C_6manually as being an integrating element.

By applying the clustering algorithm with the procedures specific to integrating elements, we obtain the topology presented in Figure 5.

	C_1	C_2	C_3	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_4
C_1	0													1					1
C_2		0			1														1
C_3			0												1				1
C_5				0	1														1
C_6		1		1	0										1				
C_7						0			1							1			
C_8							0	1		1									
C_9							1	0	1										
C_{10}						1		1	0										
C_{11}							1			0									
C_{12}											0	1						1	
C_{13}											1	0	1						
C_{14}												1	0	1	1				
C_{15}	1												1	0					
C_{16}			1		1								1		0				
C_{17}						1										0	1		1
C_{18}																1	0	1	
C_{19}											1						1	0	
C_4	1	1	1	1												1			0

Figure 5. Topology of the soybean roaster with the integrating elements

This topology shows that the algorithm integrated the element C_4 in the last module and integrated the element C_6 in the first module with which it interacted with priority. We will now compare the topology obtained with the integrating elements and that obtained by applying the clustering algorithm without the integrating elements. By applying the relations 2 and 3 which give the corrective coefficients of the modules of the soybean roaster, we obtain Table 3.

Table 3. Correcting coefficients of the soy roaster taking into account the integrating elements

Roaster modules	MSI_i'	MSI_e'	*MSI'*
Module 1	0,2	0,1	0,1
Module 2	0,4	0,0222	0,3778
Module 3	0,4	0,1166	0, 2834
Module 4	0,5	0,1	0,4

Table 3 shows that module 4 always remains the module that penalizes external interactions the most. On the other hand, with the integration of integrators, it is module 1 which this time becomes the most difficult to maintain.

Figure 6 shows the evolution of the MSI with and without taking into account the integrators.

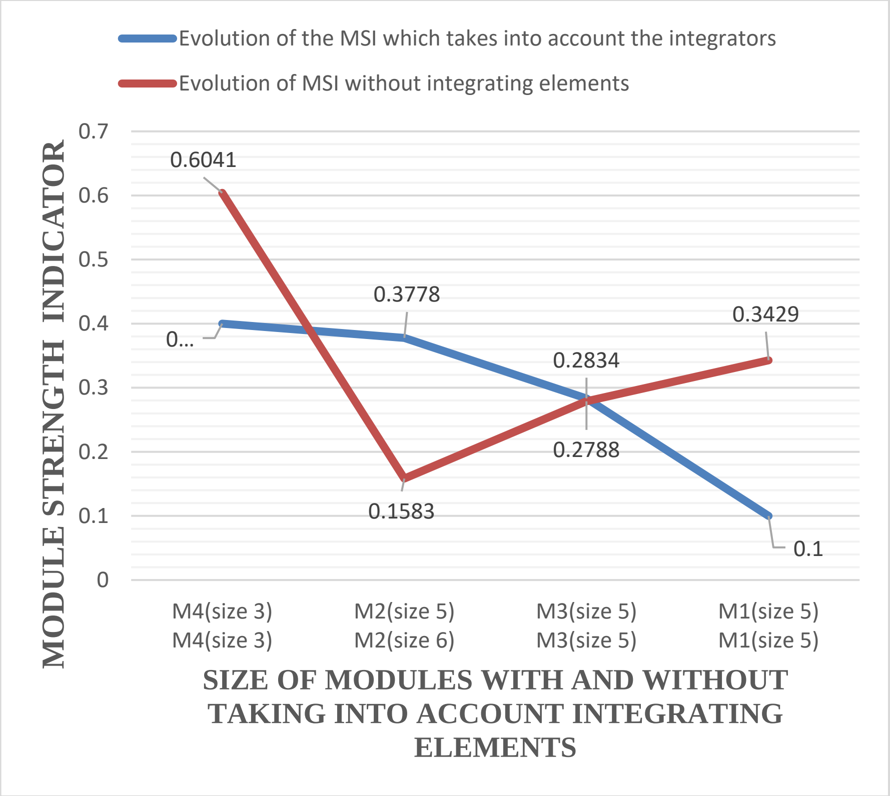

Figure 6. Correction of MSI with and without integrating elements

Figure 6 above shows that the addition of a component in module 4 following the integrating elements leads to a reduction in MSI of 33.7%, which favors the creation of a less dense module. On the other hand, the economy of scale observed at the level of module 2 (either with a reduction of 1 component) leads to a considerable increase in the MSI (that is to say 138.66%), which rather favors the creation of a more dense. The MSI of module 3 fluctuated slightly, this is reflected in the fact that the integrating elements did not affect the size of this module. Module 1, although having kept its size, has undergone a considerable reduction in its MSI. This is explained by the fact that this module by giving up an integrating element and having received another has considerably increased its external interactions.

By applying relations 5, 6 and 7, the results obtained in terms of coupling costs of the modules of the roaster are given in Table 4.

Table 4. Corrected coupling costs of the soybean roaster with integration elements taken into account

Roaster modules	**Internal coupling cost**	**External coupling cost**	**Corrected coupling cost**
Module 1	11250	3581	14831
Module 2	14400	1520	15920
Module 3	9125	3060	12185
Module 4	1332	0	1332

Table 4 shows that module 4 remains the least restrictive module for the design and module 2, the module which generates the most work and resources of the designers. integrating elements.

Table 5 summarizes the coupling costs obtained with the clustering algorithm proposed with and without integrating elements and shows the reduction obtained in cost.

Table 5. Summary of costs generated by the proposed clustering algorithm with and without integrating elements

Roaster modules	**Coupling cost without integrating elements**	**Coupling cost with integrating elements**	**Reduction achieved**
Module 1	18380	14831	19, 87 %
Module 2	29020	15920	45, 14 %
Module 3	12306	12185	0,98 %
Module 4	999	1332	

According to the table above, considering the integrating elements, there is a consequent reduction in the cost of coupling modules 1 and 2 and a slight reduction in costs with module 3, which results in the fact that the elements integrators did not affect the size of the module. On the other hand, module 4 has suffered a cost increase because it has no external coupling cost.

The explanation given for table 2, according to which the reduction rate of module 4 does not appear, is also valid for table 5. It is important to note that in the evaluation of the overall cost, the cost of coupling the module 4 appears because it has a not-zero coupling cost.

Objectively, the topology obtained is better than the previous one from the point of view of "measuring modularity" with a cost of 44,268, ie achieving an overall reduction of 27.07%. Indeed, thanks to this reduction in coupling costs and with the inclusion of integrating elements, we opt for a system that is less restrictive for designers, therefore less complex and easier to maintain (easy to dismantle, to reassemble and to recycle).

4. Conclusion and Perspectives

This paper was devoted to the proposal of a new clustering algorithm with the aim of reducing the coupling costs generated by the reference clustering algorithm. Several corrections to the reference algorithm have been made such as the definition of a digital DSM allowing the correction coefficients of the coupling cost to be brought out, the formulation of an objective function (the coupling cost) making it possible to evaluate the cost. coupling of the new proposed topology (materialized by the digital DSM) and the definition of the integrating elements in order to make an economy of scale of the large modules which cause more constraints and resources of the designers. It turns out that the algorithm proposed without taking into account the integrating elements gave a topology which generated an overall reduction in costs of 75.2% on the reference algorithm. By taking into account the integrating elements, we had defined another topology of the soy roaster which is even better than that given by the new algorithm which does not take into account the integrating elements. The new topology has promoted an economy of scale for Module 2, reducing it by one component, making it less complex and easier to maintain. This module therefore underwent a cost reduction of 45.14% and overall achieved a reduction of 27.07% in the costs of the modules of the soybean roaster.

As perspectives for this research work, we could set out more restrictive rules regarding the construction of DSMs, study the robustness of the simulated architectures and develop a software platform based on the algorithms and methods developed.

Acknowledgements

This work was funded by the authors of the paper and was carried out in the Mechanical Engineering laboratory at the National Advanced School of Agro-Industrial Sciences at the University of Ngaoundere.

References

Eric, B. (2008). Contributions to the Instrumentation of the Profession of System Architecture: from Modular Product to the Organization of the Design System. *HDR*, University of Franche-Comte, UFR Sciences and Techniques, FEMTO Institute / AS2M Department, 1-206. https://tel.archives-ouvertes.fr/tel-00348034

Fernandez, C. (1998). Integration Analysis of Product Architecture to Support Effective Team Co-location. *Master's Thesis*, Massachusetts Institute of Technology MIT, USA. http://hdl.handle.net/1721.1/9600

Ghassen, H. (2007). Towards a joint design of product architectures and project organization in the context of system engineering. *Doctoral Thesis*, University of Franche-Comte, UFR Sciences and Techniques, Doctoral school of Physical Sciences for engineer and Microtechnology, 1-249. https://tel.archives-ouvertes.fr/tel-00206133

Idicula, J. (1998). Planning for Concurrent Engineering. *Thesis draft*, Nanyang Technology University.

Martin, P., Ole, S. (2004). Topology Optimization. Theory, Methods and Applications, Second Edition, Corrected Printing. https://doi.org/10.1007/978-3-662-05086-6

Ngnassi Djami, A. B., Nzié, W., & Doka Yamigno, S. (2020). Maintenance Modularity Optimization using Clustering Algorithm: Application. *International Journal of Industrial Engineering (IJIE)*, *7*(1), 12-24. https://doi.org/10.14445/23499362/IJIE-V7I1P102

Simon, H. A. (1997). Systems Sciences, Artificial Sciences, version translated into French by J.L. Le Moigne, Dunod.

Thebeau, R. (2001). Knowledge Management of System Interfaces and Interactions for Product Development Processes. *Master's Thesis*, Massachusetts Institute of Technology. https://www.semanticscholar.org/paper/Knowledge-management-of-system-interfaces-and-from-Thebeau/61b80812aa66cbd0588b6d76b8c733a0590dcaf1

Whitfield, R. I., Smith, J. S., & Duffy, A. H. B. (2002). Identifying Component Modules. *In Proceedings of Seventh International Conference on Artificial Intelligence in Design (AID'02)*, Cambridge, United Kingdom, 571-592. https://doi.org/10.1007/978-94-017-0795-4_27

Yu, T. A., & Yassine, D. (2003). Goldberg, Genetic Algorithm for Developing Modular Product Architectures. *Proc. ASME 15th Int. Conf. Design Theory Methodology*, Chicago. https://doi.org/10.1115/DETC2003/DTM-48647

Yu, T., & Yassine. A. (2007). Goldberg, An Information Theoretic Method for Developing Modular Architectures Using Genetic Algorithms. *Research in Engineering Design*, *18*(2), 91-109. https://doi.org/10.1007/s00163-007-0030-1