

Stratégie de désassemblage d'un modèle CAO basée sur le concept de sous-assemblage

Imen BELHADJ, Riadh BEN HADJ, Moez TRIGUI, Nizar AIFAOU, Abdelmajid BENAMARA

Laboratoire de Génie Mécanique, Ecole Nationale d'Ingénieurs de Monastir, Université de Monastir, Tunisie, email : imenne.belhadj@gmail.com

Résumé :

De nos jours, suite aux exigences croissantes des clients, les produits mécaniques deviennent de plus en plus complexes. Les outils de CAO doivent être capables de suivre cette évolution et offrir par conséquent les fonctionnalités requises pour assister les concepteurs dans les choix et la validation de ces produits. Les outils de calcul par la méthode des éléments finis, les outils de simulation des mouvements sont parmi les outils d'assistance au concepteur et totalement intégrés à la CAO. Dans cet article, une méthode de génération des séquences de désassemblage (SD) des produits mécaniques est proposée afin d'aider le concepteur durant le cycle de vie des produits. Cette méthode est basée sur le concept de sous-assemblage. Ainsi, une SD du produit en des sous-assemblages est proposée. Ensuite, à chaque sous-assemblage, une SD est générée. Pour mener à bien les propositions, une solution de co-simulation Solidworks-Matlab est détaillée. Un exemple concret est présenté afin de mettre en évidence la faisabilité et l'efficacité de l'approche proposée.

Abstract :

Nowadays, due to the increasing of the customer requirements, mechanical products become more and more complex. Computer Aided Design (CAD) / Computer Aided Engineering (CAE) tools should provide functionalities to assist the designers in the choice and validation of the proposed solutions of the product. This paper presents a new method for generating disassembly sequence plan (DS) of mechanical product. This method is based on the concept of subassembly. Thus, a DS of the product into subassemblies is proposed. Then, for each identified subassembly, a DS is generated. To carry the proposals, a Solidworks-Matlab co-simulation solution is detailed. A concrete example is presented in order to highlight the feasibility and effectiveness of the proposed approach.

Mots clefs : Mots clefs

Interopérabilité, Séquence de désassemblage, Sous-assemblage, Co-simulation, CAO, CAE.

1 Introduction

Dans un contexte de mondialisation, l'amélioration de la compétitivité reste un défi majeur pour les entreprises manufacturières. Ce défi sera atteint grâce à l'utilisation de processus innovants dans le cycle de vie des produits mécaniques qui garantissent un meilleur triplet (Coût/Délai/Qualité). Par ailleurs, les

produits mécaniques répondant aux exigences sévères des clients sont de plus en plus complexes. La planification des séquences d'assemblage et du désassemblage (SA/SD) devient une tâche importante dans la validation des solutions de conception des produits modernes ayant un nombre de pièces très élevé. Cependant, le temps d'exécution et l'échange avec l'utilisateur lors du processus de génération doit être raisonnable. Dans la littérature, l'intégration des SA/SD dans un environnement de conception assistée par ordinateur (CAO) a été largement discutée mais avec un faible niveau d'intégration. Ben hadj et al. [1] ont développé un outil de génération des SA basé sur les données issues de la CAO nommé « Computer Aided Assembly Sequence Plan » (C2ASP). L'outil proposé consiste en premier lieu à extraire les données CAO afin d'identifier tous les obstacles possibles lors du déplacement des composants. Un processus d'étude des interférences est également développé pour détecter les directions libres de montage. Cette méthode souffre du temps de calcul important dans le test d'interférence pour les mécanismes ayant un nombre de pièces élevé. Par ailleurs, la SA retenue n'est pas forcément la meilleure. Dans le but de contourner le problème combinatoire de la génération des SA/SD, Kheder et al. [2] ont proposé une approche de génération des SD basée sur les colonies de fourmis. La méthodologie développée permet de générer des SD faisables et optimales. Dans cette approche, plusieurs critères ont été introduits tels que le volume des pièces, le changement d'outil, les directions de démontage et la maintenabilité des pièces d'usure. Des travaux similaires sur l'optimisation des SA/SD par les techniques intelligentes (Algorithmes génétiques, réseaux de neurones, etc.) ont été développées [3, 4].

Dans le même contexte et afin de réduire le temps d'exécution, Belhadj et al. [5] ont proposé une nouvelle méthode qui vise à décomposer un assemblage mécanique en des modules appelés « sous-assemblages » (SAS). Générer les SD du mécanisme revient à démonter les SAS identifiés. Ensuite, démonter les pièces associées à chaque SAS moyennant une approche générique. L'approche proposée est mise en œuvre à l'aide du logiciel Matlab. Cependant, le transfert de données de la CAO vers Matlab reste séquentiel et une conversion des paramètres CAO vers les variables Matlab consomme du temps excessif. Swaminathan et autres [6 - 14] ont utilisé le raisonnement à base de cas comme approche pour identifier les sous-ensembles dans un mécanisme. Cependant, cette technique est fortement liée à l'expertise et le savoir faire du mécanicien.

L'objectif de ce papier est de proposer une méthode permettant de puiser les données à partir de la CAO afin de générer les SD des mécanismes complexes. L'idée est de ramener le processus de simulation des interférences entre les pièces, qui consomme beaucoup de temps de calcul, vers un outil externe moyennant une approche de co-simulation Solidworks-Matlab. Cela permettra de remédier au faible niveau d'interopérabilité entre les logiciels CAO-CAE dans le domaine de la génération des SA/SD. La suite de ce papier est organisée comme suit: la méthode d'extraction des données CAO via sur le plug-in CADLAB sera détaillée. Ensuite, une procédure de transfert de ces données vers la plateforme de simulation Matlab est présentée afin de convertir automatiquement les données extraites de la CAO en variables Matlab. Un premier Output Matlab est une liste de sous-ensembles identifiés. Cette liste sera automatiquement renvoyée vers Solidworks afin de générer les matrices d'interférence des sous-ensembles le long des trois directions (X, Y et Z). Ensuite, une boucle d'interopérabilité est réalisée pour générer et simuler les SD du mécanisme en des sous-ensembles puis les SD des sous-ensembles eux mêmes. Un exemple illustratif est traité dans toutes les sections de cet article afin d'expliquer les différentes étapes de l'approche proposée et de mettre en évidence sa faisabilité et son efficacité.

2 Méthode proposée

La figure 1 montre l'approche globale d'interopérabilité proposée. Le traitement se fait en bouclant 3 étapes: l'extraction des attributs CAO de l'assemblage à l'aide du plug-in CADLab développé. Ces données sont ensuite transférées directement vers Matlab. Les résultats obtenus par Matlab seront transférés vers le plug-in CADLab et ainsi de suite le tout dans une approche de co-simulation. La condition d'arrêt est la génération d'une SD optimale et faisable du mécanisme.

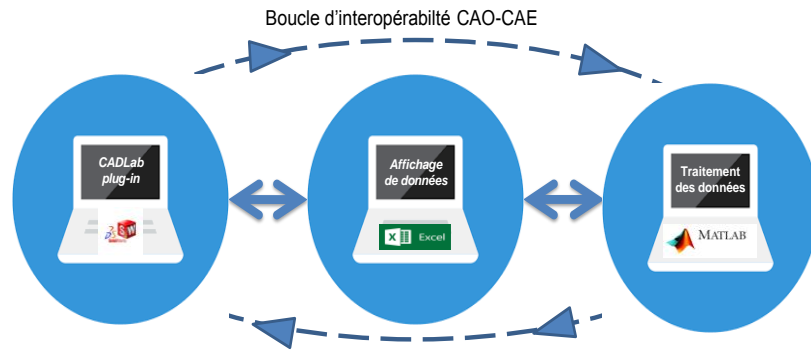


Figure 1. Approche globale d'interopérabilité CAO-CAE pour la génération des SD

Afin d'expliquer l'approche proposée, un exemple illustratif est introduit. L'exemple traité, est une boîte de vitesses d'un tracteur agricole composé de 63 pièces reliées par 24 vis et écrous (figure 2). Ce mécanisme est choisi pour montrer la faisabilité de l'approche et valider les résultats.

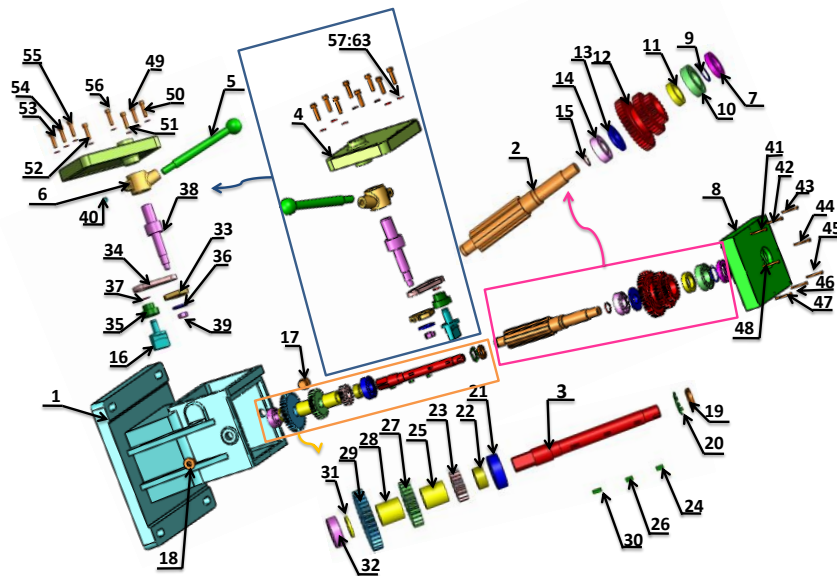


Figure 2. Modèle CAO de la boîte de vitesses

2.1 CADLab pour la collecte des données CAO

Un modèle CAO comprend généralement une grande quantité de données très utiles dans les applications CAE et plus particulièrement dans la génération des SD. L'extraction de ces données n'est possible que par l'exploitation des « application programming interfaces » (API) du système de CAO. CADLab est un plug-in intégré à Solidworks développé autour d'une bibliothèque d'APIs afin d'extraire et collecter les données d'un assemblage CAO. Ces données sont essentiellement structurées autour de deux types. D'une part, les données collectées qui comprennent à la fois les attributs de chaque pièce (propriétés de masse, données topologiques et géométriques) et les attributs des contraintes d'assemblage entre les pièces (types, entités, etc.). D'autre part, les données générées qui encapsulent une variété de matrices obtenues par plusieurs algorithmes intégrés dans CADLab (matrice d'adjacence, matrices de contact dans toutes les directions, matrices d'interférence dans toutes les directions, etc.). La figure 3 montre l'interface de l'outil CADLab permettant d'extraire et générer cette panoplie de données. La figure 4 présente la matrice d'adjacence de l'exemple traité ainsi que ses matrices de contact suivant les trois directions (x, y, z).

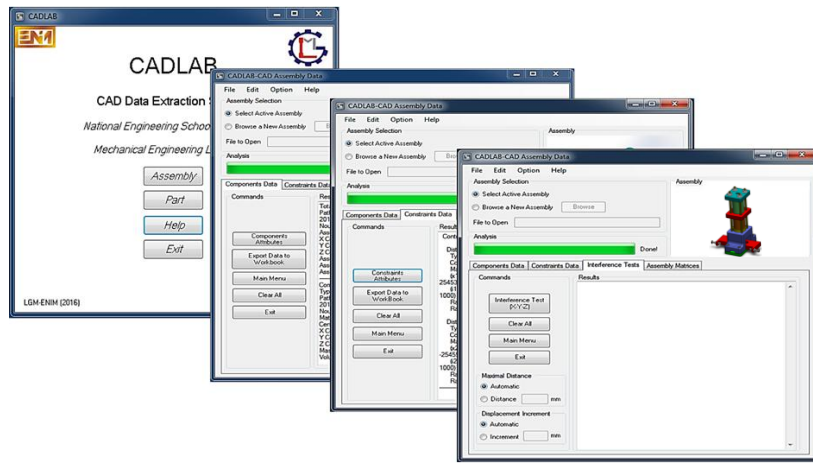


Figure 3 CADLab pour l'extraction et la génération des données CAO.

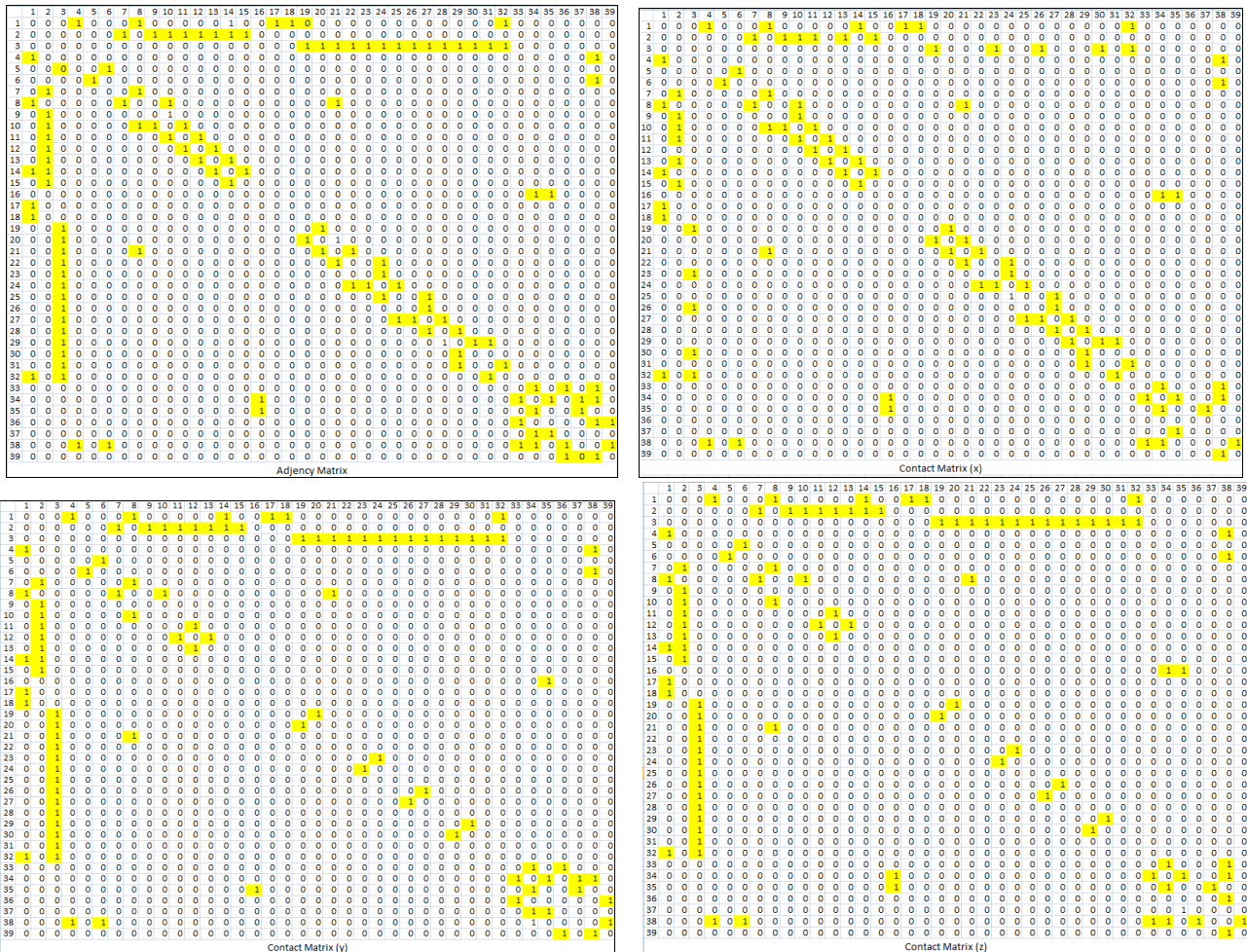


Figure 4. Matrices générées de l'exemple illustratif : Adjacence, contact suivant (x), contact suivant (y), et contact suivant (z)

2.1 Traitement des données sous Matlab

Une fois les données de CAO extraites et générées, le processus de traitement sous Matlab sera initié. Cela est déclenché à partir de CADLab. Ce processus s'articule essentiellement autour de deux étapes: la recherche des SAS et la génération des SD du mécanisme.

2.1.1 Recherche des SAS

Il s'agit d'une étape cruciale dans la démarche de génération des SD. L'organigramme de la figure suivante illustre cette étape.

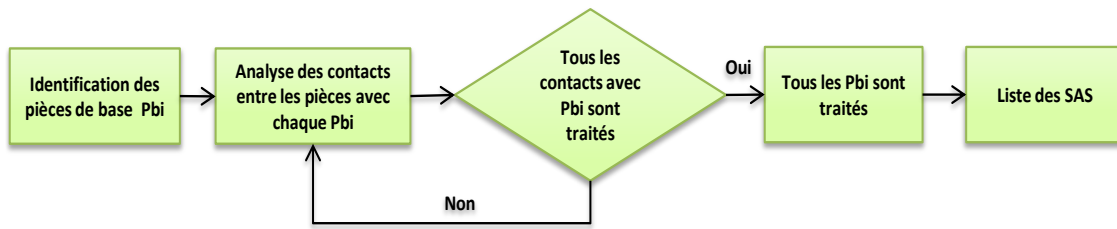


Figure 5. Organigramme de recherche des SAS

Il est à noter qu'une pièce de base Pb est définie à l'image d'un bâti sur lequel d'autres pièces seront montées. Afin de trouver la liste des pièces de base possibles dans un mécanisme, une fonction d'évaluation est développée. Elle combine, au moyen de coefficients de pondération (λ , σ et ρ), des paramètres tel que le nombre de contact (N_c) que peut avoir une pièce avec d'autres, le volume de la pièce (V) par rapport au volume total du mécanisme (V_t) et la surface de sa peau (S) par rapport à la surface totale du mécanisme (S_t). Une fois trouvé, les Pbi vont jouer le rôle d'initiateurs des SAS. La fonction d'évaluation (FE) est donnée par l'équation suivante :

$$FE = \lambda.N_c + \sigma.V/V_t + \rho.S/S_t \quad (1)$$

Pour l'exemple traité, la liste des Pb trouvée est [1, 2, 3, 8, 38]. Cela est basé sur la figure 7.

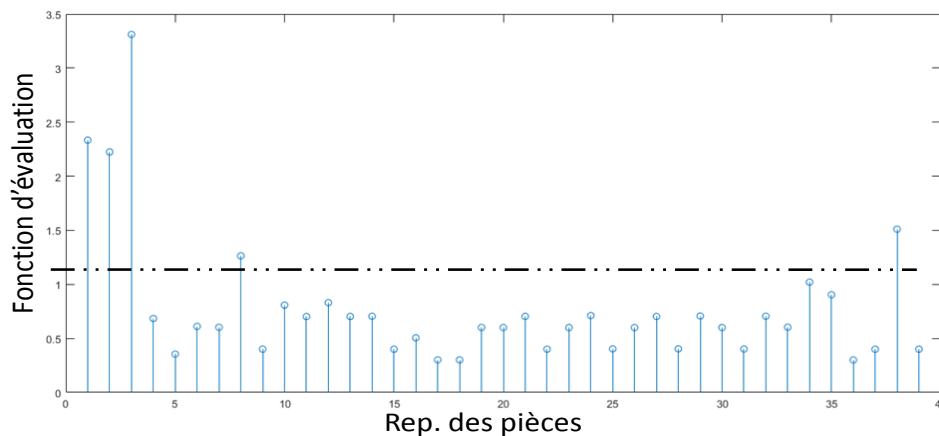


Figure 6. Evaluation des pièces par la fonction d'évaluation FE

L'algorithme de recherche des SAS commence par parcourir les matrices de contact suivant les trois directions (x , y , z) afin de supprimer tous les contacts mutuels entre les Bpi. Ensuite, et en se basant sur les mêmes matrices de contact, il recherche tous les contacts éventuels de chaque Bpi avec d'autres pièces en vue d'identifier les sous-ensembles. L'ampleur de la surface de contact et le type de l'ajustement permettront de trancher dans le cas d'une prise de décision de l'appartenance d'une pièce à deux sous-assemblages candidats. La sortie de l'algorithme est une liste de SAS. Pour l'exemple traité, la liste des SAS identifiés est représentée par la Figure 7.

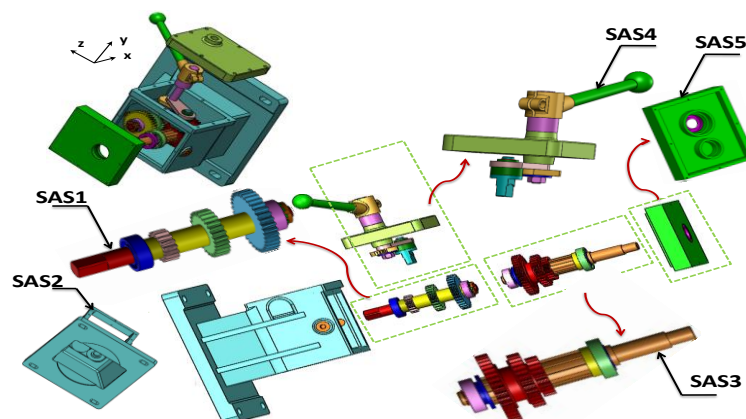


Figure 7. Identification des SAS de l'exemple traité

2.1.2 Génération des SD

L'algorithme de génération des SD est basé sur les matrices d'interférences générées par CADLab. Une Matrice d'interférence [Min] est de taille (s x s) où s représente le nombre de SAS. L'élément Min(i, j) correspondant à l'axe k est interprété comme suit : Si le SAS i n'interfère pas avec un autre SAS j dans la direction k, le SAS i peut être démonté librement à partir du SAS j dans la direction k; Dans ce cas, la valeur de Min(i, j) = 0. Si le déplacement du SAS i dans la direction k comporte une interférence avec un autre SAS j, le SAS j, s'il se déplace dans la direction k aura à son tour une interférence avec le SAS i; Dans ce cas, la valeur de Min(i, j) = 1. La figure suivante présente les nouvelles matrices d'interférence des SAS suivant les trois directions.

	SAS1	SAS2	SAS3	SAS4	SAS5
SAS1	0	1	1	0	0
SAS2	0	0	0	1	0
SAS3	0	1	0	0	0
SAS4	0	1	1	0	0
SAS5	1	1	1	1	0

Interference Matrix along (+x) direction

	SAS1	SAS2	SAS3	SAS4	SAS5
SAS1	0	1	0	1	1
SAS2	1	0	1	1	1
SAS3	0	1	0	1	1
SAS4	0	0	0	0	0
SAS5	1	1	1	1	0

Interference Matrix along (+y) direction

	SAS1	SAS2	SAS3	SAS4	SAS5
SAS1	0	1	0	0	1
SAS2	1	0	1	1	0
SAS3	1	1	0	0	1
SAS4	0	1	0	0	0
SAS5	0	0	0	0	0

Interference Matrix along (+z) direction

Figure 8. Matrice d'interférences des SAS suivant (x, y, z)

La figure 9 montre, la méthode itérative de génération des SD du mécanisme traité en des SAS à partir de la matrice [Min]. En effet, si tous les éléments d'une ligne ou d'une colonne sont nuls, le SAS associé peut être démonté librement dans la direction k. Le démontage sera possible selon (+ k) si toute la ligne est égale à 0 ou selon (-k) si toute la colonne est égale à 0. Dans le cas où plus qu'une ligne et/ou colonne sont nuls, la prise de décision sur quel SAS sera démonté est effectuée en comparant les scores de la fonction d'évaluation du Bp associé. Celui qui a le plus faible score sera démonté en premier lieu. Cette idée provient des pratiques industrielles, qui consistent généralement à démonter les pièces ou les ensembles ayant les plus faibles volumes avec le minimum de contact. Une fois qu'un SAS est démonté, sa ligne et sa colonne correspondantes dans la matrice [Min] seront supprimés. Une nouvelle matrice carrée [Min] est ensuite générée. Cette procédure est répétée jusqu'à ce que tous les SAS soient démontés. La même procédure est appliquée pour le démontage des SAS eux-mêmes une fois que les matrices [Min] associées sont générées par CADLab.

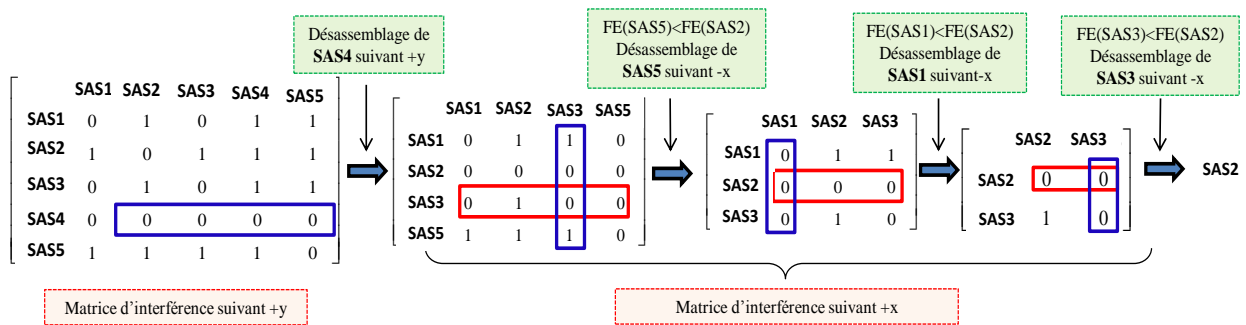


Figure 9. Génération des SD à partir des matrices [Min]

3 Conclusion

Dans ce papier, une approche d'ingénierie simultanée pour générer automatiquement les SD d'un produit complexe est détaillée. L'approche proposée commence par identifier tous les SAS en fonction des données CAO générés à partir d'un outil CADLab totalement intégré à SolidWorks. Des matrices d'interférence [Min] des pièces et des SAS sont itérativement générées et simplifiées afin d'aboutir aux SD. Une boucle d'interopérabilité SolidWorks- Matlab est jouée en permanence pour mener à bien la démarche proposée. Un exemple illustratif, choisi de la littérature, a été utilisé afin de mettre en évidence l'efficacité des propositions dans la génération des SD faisables dans un temps de calcul raisonnable. Dans les travaux futurs, les directions combinées de démontage impliquant des mouvements de rotation seront considérés. Par ailleurs, actuellement, l'outil proposé génère une séquence séquentielle de démontage, il serait utile de penser au

paralléliste du démontage des pièces et des SAS surtout pour les mécanismes complexes (tracteur routier, navire, avion, etc.)

Références

- [1] R. Ben Hadj, M. Trigui, N. Aifaoui, Toward an integrated CAD assembly sequence planning solution, Proc IMechE Part C: J Mechanical Engineering Science 229(16): 2987–3001, 2015.
- [2] M. Kheder, M. Trigui, N. Aifaoui, Optimization of disassembly sequence planning for preventive maintenance, Int J Adv Manuf Technol DOI 10.1007/s00170-016-9434-2, 2017.
- [3] A. Aghaie, H. Mokhtari, Ant colony optimization algorithm for stochastic project crashing problem in PERT networks using MC simulation. International Journal of Advance Manufactur-ing Technology, Vol 45, pp. 1051–1067, 2009.
- [4] R. J. Mullen, D. Monekosso, S. Barman, P. Remagnino, A review of ant algorithms. Expert Systems with Application, Vol 36, pp 9608-9617, 2009.
- [5] I. Belhadj, M. Trigui, A. Benamara, Subassembly generation algorithm from a CAD model, Int J Adv Manuf Technol 87: 2829–2840, 2016.
- [6] A. Swaminathan, K.S. Barber, An experience-based assembly sequence planner for mechanical assemblies. IEEE Trans Robot Autom 12(2): 252–266, 1996.
- [7] R. Viganò, G. O. Gómez, Automatic assembly sequence exploration without precedence definition, Int J Interact Des Manuf 7, 79 - 89, 2013.
- [8] H. Wang, Y. Rong, D. Xiang, Mechanical assembly planning using ant colony optimization, Computer-Aided Design 47, 59 - 71, 2014.
- [9] Mohd Fadzil Faisae Rashid, Windo Hutabarat, Ashutosh Tiwari (2011). A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. Int J Adv Manuf Technol 59, 335 – 349
- [10] JXinhua Liu, Youhui Liu, Bihong Xu (2013). A converse method-based approach for assembly sequence planning with assembly tool. Int J Adv Manuf Technol 69, 1359 – 1371
- [11] Jiapeng Yu & Chengen Wang (2013). A max–min ant colony system for assembly sequence planning, Int J Adv Manuf Technol 67. 2819 – 2835
- [12] Meiping Wu, Yi Zhao, Chenxin Wang (2013). Knowledge-Based Approach to Assembly Sequence Planning for Wind-Driven Generator. Mathematical Problems in Engineering, Volume 2013
- [13] Mohamed Kashkoush & Hoda ElMaraghy (2015). Knowledge-based model for constructing master assembly sequence. Journal of Manufacturing Systems 34, 43 – 52
- [14] Ibrahim, I., Ibrahim, Z., Ahmad, H. et al. (2015). An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm. Int J Adv Manuf Technol 79: 1363.