

MODELISATION, SIMULATION ET OPTIMISATION D'UN ATELIER DE CONSTRUCTION NAVALE

F. BAIR, Ph.RIGO

ANAST

Université de Liège

4000 Liège, Belgique

F.Bair@ulg.ac.be, Ph.Rigo@ulg.ac.be

Y.LANGER

ROGP

Université de Liège

4000 Liège, Belgique

Yves.Langer@ulg.ac.be

RESUME : *Cette étude concerne la simulation d'un atelier de construction navale, ainsi que son optimisation dans le but d'améliorer sa productivité. Habituellement, la simulation est utilisée pour améliorer l'efficacité d'un atelier, mais l'optimisation est effectuée manuellement – particulièrement en construction navale. C'est en partie dû à la particularité des pièces à fabriquer qui sont continuellement différentes, ce qui rend difficile une automatisation complète, contrairement par exemple à l'industrie automobile. L'objectif est ici de modéliser de manière fine un atelier complexe, et d'évaluer les possibilités et la faisabilité d'une optimisation automatique. Le modèle virtuel développé sera couplé avec un algorithme génétique.*

MOTS-CLES : *Algorithme génétique, Construction Navale, Optimisation, Ordonnancement, Simulation*

1. INTRODUCTION

De nos jours, la simulation en construction navale devient de plus en plus importante. L'utilisation de la réalité virtuelle et de modèles de simulation permet une augmentation importante de l'efficacité, une amélioration de l'ordonnancement ainsi que des gains importants en productivité. De tels gains ne peuvent être obtenus facilement uniquement en utilisant des outils de simulation. Il est crucial de lier le modèle de simulation avec un outil d'optimisation.

Dans un premier temps, nous allons présenter brièvement l'atelier de construction navale étudié. Cette recherche est une recherche académique permise grâce à un programme de recherche de l'Université de Liège appelé « Crédit Impulsion ». Mais dans le but d'obtenir des résultats significatifs, plutôt que d'inventer un modèle virtuel qui n'aurait pas la garantie d'être réaliste, il a été choisi de modéliser un atelier réel. L'atelier choisi est un atelier des « Chantiers de l'Atlantique » (ALSTOM Marine, Saint-Nazaire, France). Ils ont acceptés de nous donner les informations confidentielles ainsi que tous les paramètres nécessaires à la modélisation d'un de leurs ateliers. Toutefois, le projet reste un projet académique, la calibration entre l'atelier réel et le modèle virtuel n'ayant pas encore été effectuée. Pour des raisons de confidentialité, le modèle présenté a été modifié (ainsi que les différentes figures et résultats) et les résultats auront donc des valeurs indicatives. L'atelier modélisé est un atelier de soudure : les données en entrée sont des pièces individuelles (ou des sous-assemblages) et les données de sortie sont des assemblages soudés. L'objectif est de réaliser un nombre maximum d'assemblage en un minimum de temps.

La deuxième partie du travail concerne l'optimisation du modèle. Certains chantiers navals (par exemple Flensburger Schiffbau-Gesellschaft) ou certains centres de recherche utilisent la simulation pour la production (Steinhauer 2003), mais l'optimisation du modèle est généralement effectuée manuellement, sans l'utilisation d'outil d'optimisation. Une des raisons et le temps de calcul qui est assez important pour des modèles complexes. C'est un problème sévère car les modèles sont fréquemment fortement stochastiques et il est dès lors nécessaire d'exécuter plusieurs simulations pour obtenir une moyenne du temps de production total. A cause de la complexité des systèmes, le meilleur outil d'optimisation pour résoudre notre problème semble être un algorithme génétique. Pour réduire notre temps de calcul, notre idée est de ne pas effectuer plusieurs simulations pour obtenir une moyenne mais de réaliser une seule simulation par individu (voir chapitre suivant pour la définition). Le principal danger de cette méthode est que l'algorithme va être tenté de garder des solutions qui ne sont peut-être pas les meilleures, mais qui sont bonnes uniquement par chance (pour rappel, le processus est fortement stochastique). En conséquence, l'algorithme génétique doit être configuré pour garder les meilleures solutions, mais pas à n'importe quel prix car celles-ci ne sont peut-être pas aussi bonnes que prévues. A la fin du processus d'optimisation, nous devons étudier de manière plus précise les meilleures solutions en effectuant plusieurs simulations et observant leur variance. Cette méthode ne va sans doute pas donner l'optimum global, mais le but est de voir si le gain obtenu grâce à cette méthode simplifiée est significatif. Même si l'optimum global permet une amélioration de la productivité de 30%, il est intéressant de trouver un optimum local qui fournit un gain de 10% !

2. DESCRIPTION DU MODELE

2.1. Introduction

La principale caractéristique de l'atelier étudié (voir figure 1) est sa symétrie : le côté gauche est identique au côté droit. Chaque côté a ses propres outils et ses propres arrivées de pièces et sorties des assemblages terminés. Cependant, un outil est partagé en chaque côté : le robot de soudure automatique. C'est l'unique lien entre chacun des deux côtés.

L'atelier contient trois « lignes » de production, dont deux sont identiques et la troisième permet de traiter des assemblages plus grands. Chaque ligne (ou îlot) contient deux parties symétriques (ou demi îlot) qui produisent leurs propres assemblages. Chaque demi îlot possède une zone de stockage de containers (zone appelée STK) et une zone de travail sur laquelle est réalisée la séquence agrafage, soudage, parachèvement. Les pièces n'arrivent pas toutes dans l'atelier par le même chemin : les grandes tôles arrivent par des containers dans l'aire d'entrée, les plus petites pièces arrivent dans des petits containers juste en face de chaque zone de travail (zone STK).

Nous allons maintenant décrire les outils utilisés dans l'atelier avant de décrire plus précisément son fonctionnement.

2.2. Outils utilisés et leur modélisation

Deux ponts roulants, un de chaque côté, apportent des pièces de la zone d'entrée située dans le bas de l'atelier (voir Figure 1) à la zone de travail de l'îlot correspondant. Les tôles souhaitées ne sont pas toujours au sommet de la pile dans les containers, ce qui implique un dépilement de tôle dans une zone adjacente pour créer une nouvelle pile de pièces.

Le pont roulant est aussi utilisé pour évacuer les assemblages terminés et pour les évacuer sur un des

containers de sorties (deux containers – appelés PM – sont utilisés pour l'évacuation de chaque côté). Les assemblages réalisés ici sont par après soudés entre eux pour former des panneaux. Les assemblages peuvent être classés en différents groupes, en fonction du panneau auquel ils appartiennent. A la sortie, des assemblages appartenant à différents panneaux ne peuvent se trouver simultanément sur le même container de sortie. Nous avons deux containers de sortie différents, et nous ne pouvons dès lors avoir que deux différents groupes d'assemblages simultanément dans l'atelier. Malheureusement, ce n'est pas le cas et il faut évacuer les assemblages au sol si les deux assemblages de sortie sont occupés. Une fois un de ceux-ci libre, le grand pont doit alors prioritairement prendre ces assemblages excédentaires pour enfin les placer dans le container de sortie. Les panneaux doivent être réalisés dans un ordre bien précis (dépendance de cet atelier vis-à-vis du précédent et du suivant), nous ne possédons pas toute liberté pour l'ordre de fabrication des assemblages.

Pour chaque demi îlot, nous avons un outil de manutention (permettant également d'être utilisé pour souder) appelé préhenseur. Il est utilisé pour amener les pièces des containers de la zone STK jusqu'à la zone de travail et de les agraffer. Pour les très petites pièces, le préhenseur n'est pas utilisé mais la manutention et l'agrafage sont effectués manuellement par des ouvriers.

Le robot de soudage est le seul outil partagé entre les deux côtés de l'atelier. C'est un outil complètement automatisé et qui est utilisé pour souder les pièces qui viennent d'être agrafées manuellement ou par le préhenseur. En pratique, les temps totaux de soudage peuvent être calculés en amont via la programmation hors ligne qui se charge de paramétrer le robot. Ces données n'étant pas directement en notre possession, nous estimons ce temps de soudage par assemblage par une formule simplifiée dépendant des caractéristiques géométriques de ce dernier.

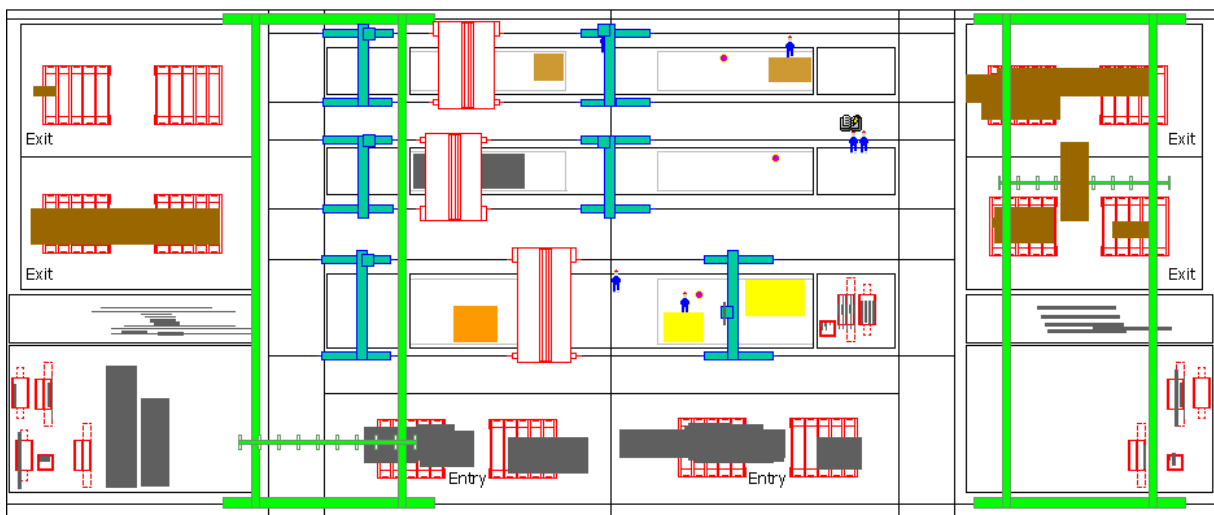


Figure 1. Modèle de l'atelier

2.3. Fonctionnement général de l'atelier

En entrée, nous avons une liste d'assemblages à réaliser pour chaque côté de l'atelier. Ces assemblages sont groupés pour constituer des « kits ». Un kit peut ne contenir qu'un seul assemblage, mais peut parfois être constitué de 4 ou 5 petits assemblages. Remarquons que cette notion de kit n'est utilisée qu'au sein de cet atelier et ne correspond à aucune restriction technologique antérieure ou ultérieure. Tous les assemblages d'un kit sont réalisés en même temps sur le même demi îlot. Notre objectif est de trouver la meilleure séquence de tous ces kits afin de minimiser le temps total de production. Ces kits ont été définis par l'utilisateur et ne peuvent pas être modifiés par la simulation. Une étape suivante et intéressante du travail serait d'essayer d'obtenir les meilleurs kits possibles grâce à un outil d'optimisation.

Les deux séquences de kits (une par côté) sont les données d'entrée que le processus d'optimisation devra optimiser afin de donner – en moyenne – le temps de production global le plus court. Il y a cependant deux manières différentes d'utiliser la simulation et de définir cette séquence : la séquence peut être soit une séquence « par demi atelier », soit une séquence « par demi îlot ».

Dans une séquence par demi atelier, nous fixons au départ deux séquences (listes) de kits qui seront les ordres d'exécution de ceux-ci. Dès qu'un demi îlot est libre, nous allouons le kit suivant de la liste à cet îlot. Cela signifie qu'au début de la simulation, nous ne savons pas où vont être réalisés les assemblages. Les grandes tôles sont stockées sur les containers d'entrée selon un ordre identique à celui des kits. Dès lors, l'avantage de cette méthode est que les tôles nécessaires seront toujours au sommet de la pile, évitant ainsi une utilisation superflue des grands ponts pour dépiler les tôles dans une zone temporaire. Malheureusement, l'inconvénient majeur est qu'il est difficile d'équilibrer chaque côté de l'atelier pour saturer le robot de soudage. L'opération de soudure étant la plus longue, il est important de saturer le robot de soudage.

Dans une séquence par demi îlot, chaque kit a reçu sa destination (numéro du demi îlot) dès le début de la simulation. La charge de travail peut ainsi être répartie pour réaliser un équilibre entre les deux côtés de l'atelier. Notons qu'en cas de problème ou de retard d'un demi îlot, la productivité peut diminuer fortement si nous voulons garder absolument la séquence fixée initialement.

Chaque type de séquençage possède manifestement ces avantages et désavantages, et la simulation peut prédire quelle est la meilleure à utiliser en fonction de divers scénarios.

Le processus de fabrication est le suivant : Préparation de l'îlot (1) – Agrafage (2) – Soudage (3) – Parachèvement (4) – Evacuation (5). Les ouvriers travaillent alternativement sur chaque côté d'un îlot (agrafage d'un côté suivi d'un parachèvement sur l'autre). Donc, pour saturer le robot de soudage, nous devons respecter le schéma donné dans la Figure 2.

Left Side		Right Side
1-2		
3	≥	1-2
4-5-1-2	≤	3
3	≥	4-5-1-2
4-5-1-2	≤	3
3	≥	4-5
4-5		

Figure 2. Condition de saturation du robot

Le modèle développé est très détaillé, la liste suivante (non exhaustive) montre les paramètres modélisés dans la simulation :

- temps d'approvisionnement et d'évacuation (différent pour chaque container)
- temps de changement d'outillage, vitesse et accélération du pont roulant
- temps de changement d'outillage, vitesse et accélération des préhenseurs
- temps d'agrafage par mètre (différent si effectué manuellement ou grâce au préhenseur)
- temps de préhension et de dépose (pour les ponts roulants et les préhenseurs)
- capacité maximale de chaque container
- ...

Chaque temps introduit dans le modèle est une moyenne (avec une distribution, variance et valeurs limites associées) et peut être changé très facilement.

Chaque pièce à souder dans l'atelier est modélisée avec toutes ses caractéristiques (poids, longueur, largeur, ...). Les attributs de chaque pièce sont visualisables directement en cliquant dessus. Toutes ces valeurs proviennent d'une base de donnée Microsoft Access qui a été modifiée par plusieurs sous-routines afin de faciliter les échanges avec notre logiciel de simulation (eM-Plant).

2.4. Résultats

Les résultats présentés dans cet article sont assez arbitraires dans le sens où le modèle de simulation n'a pas encore été calibré avec l'atelier réel mais ils sont indiqués afin de montrer quel genre de résultat peut être obtenu. Trois simulations successives de fabrication de tous les kits sont effectuées, la première afin d'éviter de commencer avec un atelier vide, la

deuxième est la simulation étudiée, la troisième évite de terminer avec un atelier vide. La dernière simulation commence avant la fin de la deuxième et n'est pas complètement terminée, la simulation étant arrêtée dès la fin totale de la deuxième. Tous les résultats montrés concernent donc uniquement la deuxième simulation.

L'avantage principal de la simulation est qu'il est très aisé d'obtenir n'importe quelle statistique. Un résultat intéressant est le temps pris par chaque opération pour chaque kit comme indiqué à la Figure 3.

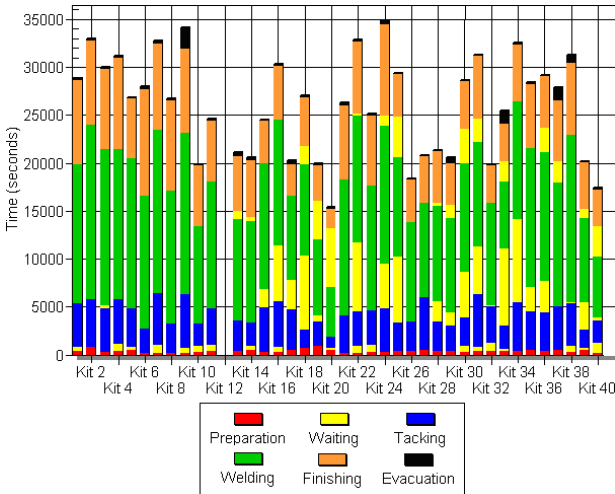


Figure 3. Durée de chaque opération par kit

Un autre résultat intéressant est l'occupation des ponts roulants. Comme prévu, le grand pont est plus utilisé en cas de séquençage par demi îlot (voir Figure 4) mais cette différence n'est pas très importante même si elle peut atteindre plus de 10 % selon les cas.

Un diagramme de Gantt (Figure 5) peut aussi être obtenu. Les résultats sont exportés dans un fichier Excel traité par différentes macros VBA afin d'obtenir le diagramme (disponible pour l'atelier en général, ou

pour chaque îlot en particulier). Beaucoup d'autres résultats peuvent être obtenus afin de comprendre très précisément le fonctionnement de l'atelier.

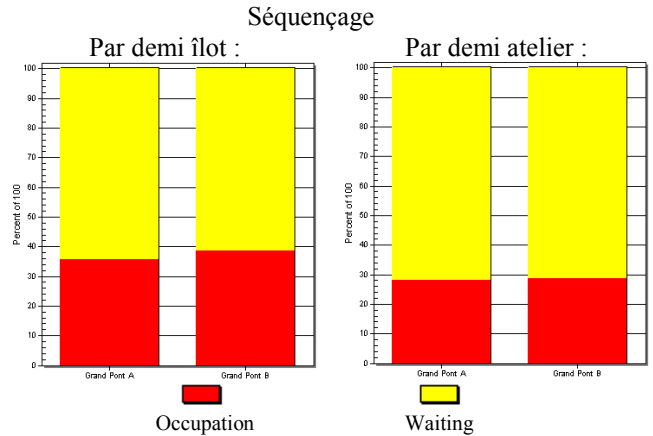


Figure 4. Comparaison de l'occupation du pont

3. ANALYSE STATISTIQUE

3.1. Introduction

Le modèle étant stochastique, une analyse statistique est nécessaire pour valider les résultats. Cette analyse est d'ailleurs indispensable avant la validation et le calibrage du modèle. En effet, l'atelier virtuel est lié à une base de données contenant une multitude de temps unitaires représentant toutes les opérations présentes. L'analyse statistique permet de déterminer quels temps unitaires ont une influence importante sur le temps total de production, et donc quels sont les opérations qui nécessitent une attention toute particulière lors de la modélisation.

Cette analyse va entre autres montrer la faible variance du temps total de production.

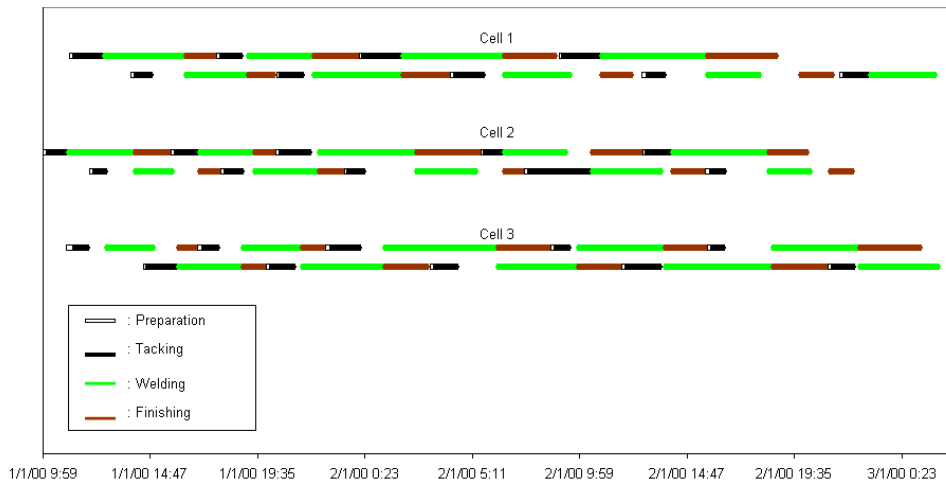


Figure 5. Diagramme de Gantt de l'atelier

3.2. Méthodologie et Résultats

La première étape indispensable est de déterminer la variance totale du processus. Rappelons que l'objectif est de connaître le temps total de production pour réaliser une liste d'assemblages donnée. Après le lancement de 100 simulations successives, nous avons trouvé une variance de 1,37%, ce qui est relativement faible ! L'écart maximal trouvé n'est d'ailleurs que de 3,7%. Cela sera très utile pour l'optimisation où nous éviterons de lancer successivement plusieurs simulations afin de déterminer la moyenne du temps total de production, vu que cette moyenne varie faiblement.

L'étape suivante consiste à déterminer l'influence de chaque temps unitaire sur le temps total de production. Pour ce faire, il est important de comprendre comment sont déterminées les variables aléatoires dans notre modèle. A chaque distribution aléatoire est attachée un nombre appelé *stream*. Ce nombre sert de base au calcul qui va déterminer la valeur aléatoire et va donner un autre nombre utilisé alors pour déterminer une autre valeur lors du prochain appel. Si nous effectuons la simulation plusieurs fois successivement sans modifier d'une simulation à l'autre toutes ces valeurs de *stream*, les résultats seront à chaque fois identiques ! Cette propriété est très intéressante car elle permet au programmeur de déboguer plus facilement le programme. En effet, si un cas particulier pose problème mais ne se produit que très rarement, le programmeur aurait de fortes difficultés à localiser le problème.

Dans notre modèle, chaque opération unitaire (temps de préhension, temps de set-up, ...) possède sa distribution mais aussi un *stream* unique. Changer le *stream* d'une seule opération sans modifier celui des autres permet donc de voir l'influence de cette opération sur le processus global. En effet, si nous modifions le *stream* du temps d'agrafage des pièces, chaque agrafage de pièce prendra un temps différent de celui de la simulation initiale, mais tous les autres temps resteront inchangés (le parachèvement des assemblages ne changera pas, et pareillement pour toutes les autres opérations). Il est donc important d'avoir un *stream* différent pour chaque opération, sans quoi l'interprétation des résultats serait plus délicate.

Chaque opération possède un temps moyen et une distribution associée (donnés par le chantier). Pour voir l'influence de cette opération, nous effectuons la simulation plusieurs fois en faisant varier cette moyenne : 60% de la valeur donnée, puis 70 %, 80% et ainsi de suite jusque 140%. Remarquons que pour chaque valeur fixée, la simulation est lancée une dizaine de fois et les valeurs enregistrées sont des moyennes. Pour chaque valeur différente de moyenne de l'opération unitaire étudiée, nous obtenons alors une valeur du temps total de production. Si ce dernier varie de manière significative, cela indique qu'il est nécessaire de porter beaucoup d'attention à la valeur moyenne de l'opération,

ainsi qu'à sa modélisation. Cela peut permettre de renseigner aux responsables de l'atelier quels sont les processus à améliorer en priorité.

La Figure 6 montre l'influence du temps de parachèvement unitaire sur le temps total de production. L'abscisse indique le coefficient que nous avons multiplié à la valeur moyenne de l'opération de parachèvement.

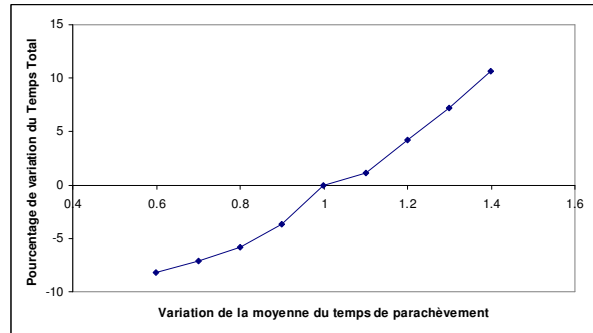


Figure 6 Influence du Parachèvement

D'une manière très claire, ce temps de parachèvement est très influant, vu qu'une variation de 40% de sa moyenne peut modifier le temps total de production de 10%. Cette valeur est nettement supérieure à la variance du temps total de production (1,37%).

Pour la plupart des autres opérations, leur influence est nettement moins évidente. Cela peut notamment être observé à la Figure 7 qui montre l'influence du temps d'agrafe manuel.

Une variation de -40% conduit à une variation du temps total de production de 0,6%, ce qui est très faible et reste en dessous de la variance de ce temps total ! De plus, une augmentation de la moyenne de 140% conduit à un temps de production moins élevé, ce qui montre bien que ces résultats sont dus uniquement au caractère aléatoire du temps total de production, et ne sont pas réellement influencés par la moyenne de cette opération unitaire.

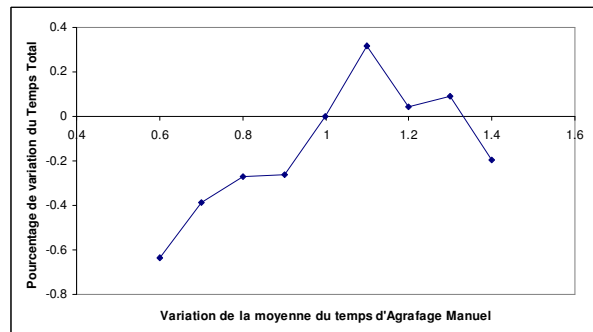


Figure 7 Influence du temps d'Agrafage Manuel

Une précision moindre dans la modélisation de cette opération ne devrait donc probablement pas prêter à conséquences. Attention néanmoins au fait qu'un ensemble d'opérations qui ne semblent pas avoir

d'influence peut quant à lui être influent ! De plus, des grossières erreurs d'estimation ne peuvent pas être permises : les figures précédentes n'étudient une variation « que » de 40%, et si la calibration n'a pas été faite avec le modèle réel, les choix initiaux des temps moyens peuvent parfois être beaucoup plus mauvais que prévu. Pour plus de sécurité, la plage d'étude peut être augmentée (au dépend du temps de calcul), tout dépend des objectifs recherchés par l'utilisateur.

Si le modèle comporte beaucoup de temps unitaires, une analyse détaillée de chacun de ces temps peut être fastidieuse et il est intéressant de pouvoir avoir un ordre d'idée de quelles sont les opérations à étudier en priorité. Une méthode consiste à comptabiliser dans la simulation le nombre de fois que chaque opération a lieu, et de multiplier ce nombre par le temps moyen de l'opération. Nous obtenons ainsi une valeur qui permet de classer les opérations par ordre d'importance. Cette méthode ne reste cependant qu'indicative, puisque certains temps peuvent être totalement « cachés », c'est le cas par exemple du temps d'évacuation de certaines bennes (notamment celles placées dans la zone STK). Une variation de 40% du temps moyen d'évacuation de ces bennes amène à une variation du temps total de production de 0% ! Cette opération est donc totalement cachée ce qui montre bien que la méthode précédente ne doit être utilisée que de manière indicative.

4. OPTIMISATION DU MODELE

4.1. Introduction

Comme dit précédemment, l'objectif est de trouver les deux meilleures séquences qui minimisent le temps total de production. Il est crucial de définir clairement ce qu'est notre fonction objectif (ou encore fitness). Pour éviter de commencer avec un atelier vide, la première simulation de tous les kits est effectuée sans enregistrer les statistiques. Cet enregistrement n'est effectué que lors de la deuxième simulation. Le problème avec cette période de warm-up est que chaque demi îlot va démarrer à enregistrer les statistiques à un différent moment.

A la fin de la simulation, certains kits de la troisième simulation sont également fabriqués, ce qui influe sur la fitness. Nous prendrons donc la somme du temps de fabrication de chaque demi îlot de la deuxième simulation divisée par 3 (car nous avons 3 îlots). Ainsi, nous obtenons cependant un temps par côté de l'atelier, la fitness sera donc la somme de ces 2 temps.

Si nous effectuons différentes simulations avec le même séquençage, nous obtenons une variation de la fitness d'environ 1,4% ce qui est très bas (avec un intervalle de confiance de 90%) ! Cela peut s'expliquer par deux raisons :

- la plus longue opération est le temps de soudage qui est le seul temps fixe,
- nous avons beaucoup de temps morts entre les opérations successives : ces temps d'attente agissent dès lors comme des tampons, les variations des durées des opérations étant absorbées dans ces temps d'attente.

Cette caractéristique est très importante et sera exploitée pleinement pour l'optimisation.

A cause de la complexité du modèle et vu le type d'optimisation à réaliser (optimisation d'un séquençage), les algorithmes génétiques sont les outils les plus appropriés pour résoudre le problème. L'algorithme utilisé est inclus dans le logiciel de simulation eM-Plant. Différents types d'algorithmes génétiques existent et ils requièrent généralement de définir un grand nombre de paramètres (types de mutation, de croisement, ...). Le chapitre suivant va expliquer leur fonctionnement.

4.2. Description de l'algorithme génétique et des ses paramètres

4.2.1. Introduction

Les algorithmes génétiques sont des procédures de recherche automatisées basées sur le principe de l'évolution naturelle et de l'hérédité. Beaucoup d'ouvrages et d'articles traitent du problème (Birk et Harries, 2003), (Coley, 1999), (Davis, 1991), (Goldberg, 1989), (Haupt R.L. et Haupt S.E., 1998), (Man *et al.*, 1999).

L'espace de recherche du problème est représenté par une collection « d'individus » qui sont caractérisés par des « chromosomes ». Un set d'individus au même instant est appelé une population. Chaque individu est composé d'un set de valeurs (ou gènes) qui définit ses caractéristiques. Un code binaire (soit 0 ou 1 pour chaque gène) est généralement utilisé pour les chromosomes. A chaque individu est associé une fitness qui va influencer sur sa probabilité de survivre et de se reproduire. Après l'initialisation, chaque génération produit des nouveaux enfants en fonction des opérateurs de croisement et de mutation définis par l'utilisateur. Le principe est basé sur la sélection naturelle qui suppose que les enfants ont une chance significative de garder les meilleures caractéristiques de leurs parents, les individus les plus faibles n'étant pas sélectionnés pour la reproduction. Au cours des générations, la population évolue sans cesse, la moyenne de fitness de tous ses individus devant donc s'améliorer au cours du temps. Après plusieurs générations, la solution peut finalement être obtenue (ou approchée). Différents paramètres doivent être choisis pour rendre efficace l'algorithme et vont être décrits dans les points suivants.

4.2.2. Opérateur de mutation

Deux différents types de mutations peuvent être utilisées : l'inversion de séquence ou la mutation entre deux gènes. Pour une inversion, une zone d'inversion est

choisie aléatoirement et la séquence des gènes en son sein est inversée (voir Figure 8).

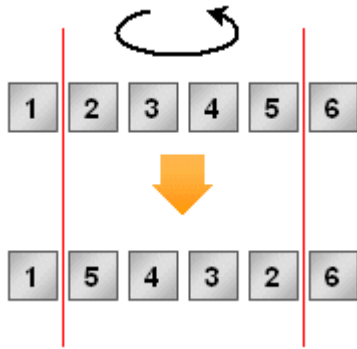


Figure 8. Mutation de gènes

Un autre opérateur de mutation consiste à échanger aléatoirement 2 gènes (voir Figure 9).

Il est également possible de définir comment l'algorithme doit échanger les positions en utilisant une probabilité d'occurrence.

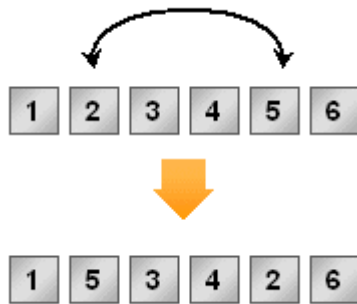


Figure 9. Simple échange entre 2 gènes

4.2.3. Croisement

Contrairement aux opérateurs de mutation et d'inversion, l'opérateur de croisement est appliqué aux deux chromosomes. Il échange des éléments entre eux. Initialement, deux points d'intersection sont choisis et les éléments entre ces points sont intervertis.

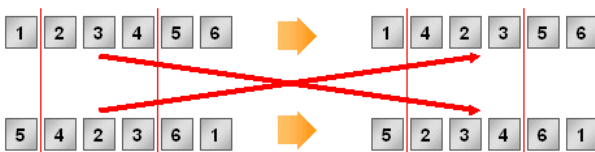


Figure 10. Croisement entre 2 individus

L'utilisation de l'opérateur de croisement va conduire à de meilleures solutions car il y a une haute probabilité que de bonnes zones soient préservées et vont donc être reproduites dans un nombre croissant de solutions. Nous pouvons aussi différencier deux types différents d'opérateur de croisement, l'un essayant de garder la position relative des gènes, l'autre la position absolue.

4.2.4. Sélection des parents

Deux différents types de sélection sont disponibles : une sélection déterministe et une sélection aléatoire.

Pour une sélection déterministe, les parents sont sélectionnés aléatoirement en fonction de leur fitness (principe de sélection de la roue). Pour cette raison, les individus avec une bonne fitness vont être choisis plus régulièrement comme parent pour créer la nouvelle génération. Mais les individus avec une faibles fitness ont toutefois une chance d'être sélectionné comme parent.

Pour une sélection aléatoire, la valeur de la fitness n'est pas utilisée. Chaque individu a la même probabilité d'être utilisé comme parent.

4.2.5. Sélection des enfants

Ce paramètre permet de sélectionner les membres utilisés pour la génération suivante. Nous avons quatre possibilités :

- Utiliser uniquement les enfants et sélectionner le meilleur (*Iof2*)
- Utiliser les enfants et les parents et sélectionner le meilleur (*Iof4*)
- Utiliser les enfants et les parents et effectuer la sélection aléatoirement mais proportionnellement à la fitness des individus (*Prob*)
- Sélectionner aléatoirement un individu sans tenir compte de la fitness (*Random*)

4.2.6. Référence de la Fitness

La valeur de la fitness peut être évaluée grâce à deux méthodes différentes.

- Référence Absolue : dans ce cas la fitness est évaluée par rapport au zéro absolu. Au début de l'optimisation, cela produit de bons résultats grâce à la variance des individus dans la génération. Mais la convergence est très lente en cas de génération homogène.

- Référence Relative : ici la fitness est évaluée par rapport au plus mauvais individu. La sélection naturelle est donc maintenue même en cas de population homogène.

4.3. Résultats

4.3.1. Introduction

L'objectif est de trouver le meilleur séquençage de kits qui minimise le temps total de production. Comme expliqué précédemment, il y a quelques contraintes sur ces séquençages car les assemblages appartenant au même panneau doivent être réalisés successivement. Dès lors, certains kits doivent être groupés ensemble et nous ferons l'hypothèse que l'ordre de fabrication de ces groupes est fixé par l'utilisateur et ne peut être changé par le processus d'optimisation. Le processus

d'optimisation final consiste donc à déterminer la meilleure séquence de kits au sein de chaque groupe.

Les résultats montrés ci-après sont nos premiers résultats et quelques développements doivent encore être faits pour améliorer l'optimisation, mais ils montrent déjà que des gains significatifs peuvent être obtenus.

4.3.2. Résultats pour un séquençage par demi atelier

La formulation dans ce cas est assez évidente, nous avons 6 chromosomes, un pour chaque groupe, les quatre premiers pour le séquençage d'un côté de l'atelier, les deux derniers pour le second côté (cela dépend évidemment de la liste initiale d'assemblages à optimiser). Remarquons que ce nombre est directement lié à la liste initiale d'assemblages à réaliser et à optimiser. Rappelons que dans ce séquençage, nous ne fixons pas à l'avance l'emplacement des kits dans l'atelier.

La Figure 11 indique l'évolution de la fitness (meilleure, moyenne, pire) au cours des générations avec les paramètres utilisés suivants :

- Taille de la population : 20
- Nombre de générations : 10
- Fitness référence : absolue
- Sélection des parents : Déterministe
- Clonage de la meilleure sélection
- Sélection des enfants : 1of4
- Croisement : conservation de la position absolue
- Mutation : simple échange entre 2 gènes aléatoires (probabilité d'occurrence de 0,1)

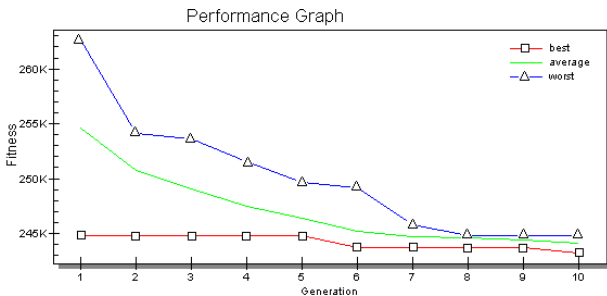


Figure 11. Evolution de la Fitness

Ces résultats semblent bons car nous avons un gain entre la moyenne de la première génération et la meilleure fitness à la fin de 4,7%, et la moyenne de la dernière génération est également très bonne. Malheureusement, nous devons être prudents car nous tendons à obtenir des populations fortement homogènes. La courbe de la moyenne ainsi que de la pire solution ne s'améliorent pas grâce à l'efficacité de l'algorithme mais car les individus qui constituent ces courbes deviennent de moins en moins différents (tendent vers la meilleure). Par exemple, si nous évaluons la fitness d'un nombre important de séquençages choisis aléatoirement (sans aucune optimisation), nous trouvons pratiquement la même meilleure fitness ! Des améliorations doivent donc

encore être faites même si cette méthode donne une bonne séquence de départ sans effort spécifique.

Nous pouvons observer à la Figure 12 ce qui se passe si nous essayons d'éviter l'homogénéisation de la population (en prenant par exemple une sélection des enfants 1of2). La convergence est nettement moins évidente.

Nous avons néanmoins toujours un gain d'environ 5% en comparaison avec la moyenne des individus de la première génération. Nous pouvons aussi observer une légère amélioration de la moyenne mais qui n'est pas très marquée.

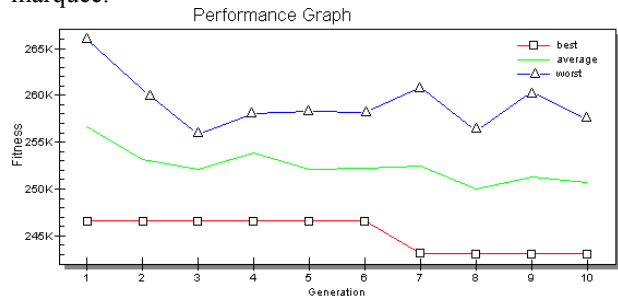


Figure 12. Evolution de la fitness (Sélection 1of2)

Différentes optimisations en changeant les paramètres ont été effectuées, mais nous ne pouvons montrer tous les résultats ici. Le séquençage par demi atelier ne semble néanmoins pas trivial à optimiser.

4.3.3. Résultats pour un séquençage par demi îlot

L'optimisation de ce type de séquençage semble plus logique car il est plus aisé d'équilibrer les charges de travail entre les deux côtés de l'atelier (et donc de garder des bonnes parties de séquençage d'individu en individu). C'était moins le cas dans le séquençage précédent car les résultats étaient plus délicats à analyser : la même séquence pouvait mener à des résultats relativement différents puisque pour plusieurs simulations à priori identiques, les kits ne seront pas toujours fabriqués au même endroit. En effet, les demi îlots ne se libéreront pas toujours au même moment, le processus étant stochastique.

Néanmoins, ce séquençage est plus difficile à adapter pour l'algorithme d'optimisation et nous avons ici besoin de 14 chromosomes pour constituer un individu (les explications détaillées ne seront pas données ici).

Avec les paramètres suivants, nous obtenons les résultats indiqués à la Figure 13.

- Taille de la génération : 20
- Nombre de générations : 10
- Fitness Référence : Absolue
- Sélection des parents : Déterministe
- Clonage de la meilleure sélection
- Sélection des enfants : 1of4
- Croisement : conservation de la position absolue

- Mutation : simple échange entre 2 gènes aléatoires (probabilité d'occurrence de 0,1)

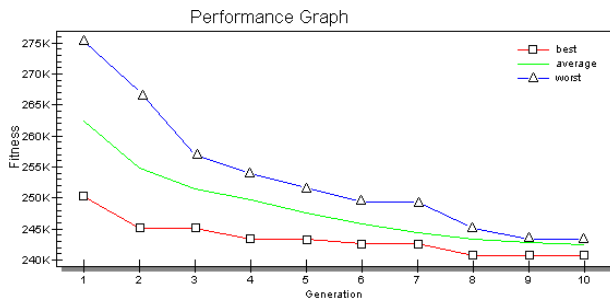


Figure 13. Evolution de la Fitness – Séquençage demi flot

Les mêmes remarques peuvent être réalisées ici : la convergence de la moyenne et de la pire solution montrent une homogénéisation de la population. A la première génération, la moyenne possède une valeur plus grande que celle du séquençage par demi atelier. C'est tout à fait conforme à nos attentes vu qu'ici le grand pont est plus sollicité, et que vu que les séquences sont choisies aléatoirement, il n'y a pas de raison pour que l'équilibre entre les 2 côtés de l'atelier soit meilleur (ce qui, pour rappel, est pourtant l'avantage de ce séquençage). L'équilibre est meilleur si la séquence est choisie judicieusement par l'utilisateur. Ce n'est pas le cas ici, mais il est fort probable comme dit précédemment que des bonnes parties du séquençage soient gardées par l'algorithme pour finalement tendre vers cet équilibre.

Le gain entre la meilleure fitness et la moyenne de la première génération est maintenant de 8,4% (et de 3,6% entre la meilleure fitness de la première et de la dernière génération) ! Il semble donc que l'optimisation a été plus efficace avec ce type de séquençage. C'est d'ailleurs confirmé après exécution d'un grand nombre de séquences aléatoires (sans optimisation). La meilleure solution obtenue dans ce cas est 249K, ce qui n'est pas aussi bon que nos résultats obtenus avec une optimisation.

En conclusion, même si des améliorations au niveau de l'optimiseur et/ou de la méthodologie doivent encore être effectuées, il est très intéressant de choisir les résultats de l'optimisation comme séquence de départ (gain de plus de 8% par rapport à une séquence aléatoire), séquence qui peut être modifiée manuellement par l'utilisateur après une analyse plus détaillée du comportement du système.

5. CONCLUSIONS ET PERSPECTIVES

La simulation est évidemment un outil très puissant et peut être utilisée de manière très intéressante même sans l'utilisation d'un logiciel d'optimisation. La principale difficulté est le temps de développement qui peut être assez important, particulièrement dans notre modèle qui est très détaillé. Mais lorsque ces développements sont

effectués, des gains et des informations importants concernant la productivité de l'atelier peuvent être effectués.

Les premiers résultats obtenus montrent que le couplage entre la simulation et un outil d'optimisation peut être fait et nous encourage à poursuivre nos investigations. Pour avoir une estimation des gains réels pouvant être réalisés, une calibration entre le modèle virtuel et l'atelier doit cependant être faite. Les résultats dépendent aussi fortement de la variance du temps de production. Plus elle est faible, plus la méthode utilisée sera efficace. Si la variance devient trop élevée, il faut alors effectuer plusieurs simulations et prendre la moyenne comme valeur de la fitness. Dans ce cas, l'inconvénient majeur est le temps de calcul qui augmente de manière très conséquente. Néanmoins, l'optimisation n'est finalement pas très difficile à développer comparé au temps nécessaire pour réaliser le modèle virtuel et donne rapidement des résultats intéressants. Il est à parier que l'optimisation et la simulation seront de plus en plus couplées au cours des années à venir.

Après quelques améliorations de cette optimisation, la prochaine étape à franchir est d'automatiquement optimiser les kits. Nous partirons alors d'une liste d'assemblages, et la simulation et l'optimisation détermineront les groupements adéquats de ceux-ci pour former des kits. Cette optimisation est beaucoup plus complexe car l'espace de conception est ainsi fortement augmenté, et de plus nous avons une contrainte additionnelle : l'utilisation de l'espace au sol dans la zone de travail. Chaque assemblage a ses dimensions propres et il n'est donc pas aisé de déterminer automatiquement quelles vont être les assemblages d'un kit.

REMERCIEMENTS

Les auteurs tiennent à remercier l'Université de Liège pour les fonds de recherche «Crédit Impulsion», la Région Wallonne (RW) et le Fond Social Européen (FSE). Des remerciements spéciaux aux Chantiers de l'Atlantique, ALSTOM Marine, Saint-Nazaire, France pour l'ensemble des données fournies qui ont permis de travailler sur un atelier réel et plus spécifiquement à M. François Ellie.

REFERENCES

- Banks J., John S.Carson and Barry L.Nelson. *Discrete-Event System Simulation*, Second Edition. Prentice Hall, Upper Saddle River, New Jersey.
- Birk L. and Harries S., 2003. *Optimistic: Optimization in Marine Design*, Mensch & Buch, Berlin.
- Coley D.A., 1999. *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific, Singapore.

Davis L., 1991. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

Goldberg D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc.

Haupt R.L. and Haupt S.E., 1998. *Practical Genetic Algorithms*, John Wiley & Sons, Inc., New York.

Man K.F., Tang K.S. and Kwong S., 1999. *Genetic Algorithms*, Springer-Verlag, London.

Steinhauer D., 2003. *The Virtual Shipyard – Simulation in production and logistics at Flensburger*, 2nd international EuroConference on computer and IT applications in the marine Industries COMPIT'03, Hamburg.

Tecnomatix, eM-Plant v.7.0, *User's Guide*.