

IMPLEMENTATION D'UN ALGORITHME EVOLUTIONNISTE POUR L'ORDONNANCEMENT DE LA PRODUCTION CHEZ UN FOURNISSEUR DANS L'INDUSTRIE TEXTILE

Fabian VARGAS-NIETO

Departamento de Ingeniería Industrial
Universidad del Norte
Km 5 vía Puerto Colombia
Barranquilla, COLOMBIE

Jairo R. MONTOYA-TORRES

Escuela Internacional de Ciencias Económicas y
Administrativas, Universidad de La Sabana
Km 21 autopista norte de Bogotá
Chía, COLOMBIE
jairo.montoya@unisabana.edu.co

RESUME : *Dans cet article, nous étudions le problème d'ordonnancement de la production chez un fournisseur d'étiquettes dans l'industrie textile fabriquées par le procédé de transfert thermique. Le problème est modélisé comme un flowshop hybride à deux étages, avec l'objectif multiple de minimiser le makespan et le nombre de jobs en retard. Nous proposons une procédure de résolution basée sur l'implémentation d'un algorithme évolutionniste bi-objectif. L'analyse expérimentale est effectuée en utilisant des données réelles fournies par l'entreprise partenaire de ce projet. Une comparaison avec la procédure actuellement mise en place dans l'entreprise est également effectuée. Un outil d'aide à la décision sous MSExcel® est fourni à l'entreprise. Les résultats de la mise en place industrielle pendant les derniers mois montrent un impact très positif sur les indicateurs clés de l'entreprise.*

MOTS-CLES : *Ordonnancement, algorithme évolutionniste, makespan, jobs en retard, industrie textile.*

1. PRESENTATION DE LA PROBLEMATIQUE

L'industrie d'étiquettes imprimée par le procédé de transfert thermique est l'une des industries avec la plus grande croissance pendant les dernières années en Colombie (Vargas-Nieto, 2007). Le marché colombien de la mode demande de plus en plus d'étiquettes pour les vêtements avec toute sorte d'informations : taille, style, code à barres, références du fabricant, etc. Dans le même temps, il est demandé plus de flexibilité lors de la fabrication de l'ordre de production de telle sorte qu'il ne soit fabriqué que le nombre d'étiquettes réellement demandées. Ainsi, la taille d'un ordre de production peut dépasser les 500 étiquettes, ce qui complexifie leur fabrication par des procédés traditionnels : étiquette cousue ou imprimée en flexographie. Par conséquent, et en raison d'un plus grand besoin d'informations complémentaires sur l'étiquette, l'impression des étiquettes par le procédé de transfert thermique a gagné beaucoup de terrain en Colombie.

L'entreprise partenaire de ce projet, l'un des plus grands fabricants d'étiquettes dans le pays, a des centres de production en Colombie dans les villes de Barranquilla (dans le nord), Bogotá (la capital, dans le centre du pays), et Medellín et Cali (dans l'ouest). La figure 1 présente une carte du pays et en souligne la localisation des villes citées. Pour des raisons de confidentialité, il n'est pas possible de citer le nom de l'entreprise. Dans chacun de ces centres de production sont traités les ordres de fabrication arrivant chaque jour. Pendant les dernières années, la croissance des ventes a fait que les

délais de livraison ont augmentés de 50%, bien qu'il y ait la capacité suffisante pour traiter toutes les demandes. De plus, l'entreprise a observé une augmentation de presque 250% du délai de livraison pendant les derniers six mois. Cette situation a fait perdre à la compagnie un certain nombre de clients clés. Après avoir analysé les causes de cette situation, nous avons identifié que, lorsque la demande augmentait, le nombre d'ordre de production à ordonnancer augmentait, rendant cette dernière tâche trop complexe pour continuer avec les procédures actuelles d'ordonnancement « à la main ». Ceci nécessite l'utilisation d'un outil informatique pour ordonnancer la production dont les objectifs soient la maximisation de la productivité et la minimisation du nombre d'ordres délivrés en retard. Ces deux objectifs peuvent se traduire par une minimisation simultanée du makespan et du nombre de jobs en retard.

A ce jour, l'ordonnancement de la production dans le département de fabrication d'étiquettes par transfert thermique a été effectué de façon empirique basé sur la connaissance intuitive du superviseur de production de chaque centre de production de la compagnie. Cette façon de faire ne permet pas non plus d'équilibrer la charge de travail entre les différentes machines disponibles dans l'atelier, provoquant ainsi une mauvaise utilisation de la capacité installée. Le superviseur de production doit donc travailler avec un grand nombre d'ordres « urgents » chaque jour.



Figure 1. Carte de la Colombie

Ce travail cherche à proposer une nouvelle procédure que l'entreprise puisse utiliser assez facilement pour ordonnancer la production des lignes de fabrication par transfert thermique. Nous voulons proposer un outil d'aide à la décision permettant d'améliorer les indicateurs clés, tels que la productivité et le niveau de service de la ligne. Ces indicateurs sont traduits par la minimisation du makespan et du nombre de jobs en retard. S'agissant donc d'une approche multi-objectif de décision, cet article propose un algorithme évolutionniste en raison des résultats précédents trouvés dans la littérature sur les bénéfices retrouvés lors de l'application pratique de ces méthodes.

Cet article est organisé comme suit. La section 2 décrit brièvement le processus de fabrication d'étiquettes par le procédé de transfert thermique, ainsi que le modèle d'ordonnancement utilisé, à savoir un flowshop hybride à deux étages. La section 3 est dédiée à la présentation de la procédure évolutionniste proposée. Cet article étant basé sur nos travaux précédents sur cette problématique, les principaux résultats expérimentaux obtenus précédemment sont présentés dans la section 4. La section 5 est ensuite dédiée à la présentation des résultats de la mise en œuvre industrielle. Enfin, les conclusions et quelques perspectives sont présentées dans la section 6.

2. LE PROCESSUS DE FABRICATION ET LE MODELE D'ORDONNANCEMENT

Le processus de fabrication d'étiquettes est composé de deux étages principaux : impression et découpage. Lors du processus d'impression, les étiquettes peuvent être imprimées sur un seul côté ou recto-verso, en couleur ou en noir et blanc. Plusieurs technologies existent, dans cet article nous ne considérons que des machines d'impression travaillant par transfert thermique. Celles-

ci ne peuvent imprimer qu'une seule couleur par passage. Après avoir été imprimées, les étiquettes doivent être découpées. Cette opération est effectuée par des machines avec une technologie d'ultrason. Le nombre de machines dans chaque étape du processus de fabrication peut varier. Dans le cas particulier de notre partenaire industriel, il y a 4 machines d'impression thermique et 3 machines de découpage.

Nous présentons ensuite quelques caractéristiques du processus de fabrication d'étiquettes par transfert thermique :

- Si une étiquette est composée de plus d'une couleur, il est nécessaire d'effectuer autant de passages que de couleurs.
- Les temps de préparation des machines sont considérés insignifiants par rapport au temps effectif de production.
- Il y a suffisamment d'opérateurs dans l'atelier pour travailler sur toutes les machines.

Le problème d'ordonnancement de la production dans cette usine peut être modélisé comme un flowshop hybride (FSH). Dans le cas général, un atelier de type FSH est composé d'un ensemble de k étages en série, chacun des $s=1, \dots, k$ étages est composé d'un ensemble de m_s machines parallèles identiques. Le stockage entre deux étages successifs est illimité. Le job ou ordre de production est noté j , avec $j=1, \dots, n$ et doit être exécutée à chaque étage sur une machine et une seule. Le temps d'exécution pour chaque job sur chaque étage k est connu et noté p_{sj} . L'objectif est de trouver une séquence (ordre d'exécution) pour l'exécution des jobs sans interruption et de choisir une machine dans chaque étage pour effectuer l'opération sur le job, de façon à optimiser une ou plusieurs fonctions objectif.

Le problème de flowshop hybride dont l'objectif est la minimisation du makespan est connu comme étant NP-complet (Gupta, 1988). Les travaux de recherche dans la littérature ont principalement focalisés sur le problème à deux étages avec minimisation du makespan. Il existe plusieurs méthodes exactes, parmi lesquelles on trouve la programmation linéaire, des méthodes de branch and bound, des algorithmes à performance garantie, et des heuristiques dédiées (Brah et Hunsucker, 1991; Chen, 1995; Haouari et M'Hallah, 1997; Dessouky et al., 1998; Moursli et Pochet, 2000). Des travaux sur l'état de l'art ont été proposés par Chen (1994) et Linn et Zhang (1999).

Dans la littérature, il existe peu de travaux traitant le problème avec plus de deux étages ($k \geq 3$) et une seule fonction objectif. La fonction objectif la plus étudiée est la minimisation du makespan. On trouve des procédures de résolution basée sur des méthodes de branch and bound, programmation dynamique, modifications des heuristiques dédiées au problème avec deux étages, méta-heuristiques, ainsi que des bornes inférieures (Lee et

Vairaktarakis, 1994; Soewandi et Elmaghraby, 2001; Riane et al., 1998; Vignier et al., 1997; Acero et al., 2004). Quelques modèles étudiés considèrent d'autres fonctions objectif ou encore d'autres contraintes: disponibilité des ressources (Allaoui et Artiba, 2006), minimisation du temps total de terminaison (Azizoglu et al., 2001, Guinet et Solomon, 1996), recirculation des jobs (Bertel et Billaut, 2004), contraintes de précedence (Botta-Genoulaz, 2000, Tang et al., 2006), temps d'exécution dépendants de la machine (Sriskandarajah et Wagneur, 1991), stockage intermédiaire limité (Sawik, 2002), interruption de l'exécution des jobs (Djellab et Djellab, 2002), ou encore le traitement de batches (Xuan et Tang, 2007). Des algorithmes basés sur des méta-heuristiques ont également été proposés (Portmann et al., 1998, Jin et al., 2006, Ruiz et Maroto, 2006). Dans le cas du problème multi-objectif, la plupart de travaux se sont focalisés sur la minimisation du makespan et du temps moyen de terminaison des jobs (T'Kindt et Billaut, 2006). D'autres travaux considèrent aussi les mêmes fonctions objectif mais avec d'autres contraintes (Bertel et Billaut, 2004, Morita et Shio, 2005).

Le cas du flowshop hybride avec minimisation du makespan et minimisation du nombre de jobs en retard n'a pas encore été étudié, au moins à notre connaissance dans la littérature. Dans cet article, nous présentons la conception et mise en oeuvre d'une procédure de résolution du problème de FSH bi-objectif à deux étages basés sur les algorithmes évolutionnistes. Comme expliqué précédemment, cette étude est basée sur une problématique issue de l'industrie textile. En particulier, le problème a été retrouvé chez un fabricant d'étiquettes pour vêtements fabriqués par le processus de transfert thermique. Les données nécessaires pour l'étude expérimentale ont donc été obtenues grâce à la collaboration de notre partenaire.

3. PROCEDURE PROPOSEE

La procédure proposée dans cet article est basée sur nos travaux précédents dans lesquels nous avons présentés la démarche de conception de l'algorithme (Vargas-Nieto et Montoya-Torres, 2007a) et des tests préliminaires (Vargas-Nieto et Montoya-Torres, 2007b). Dans cet article, nous nous intéressons maintenant au paramétrage de l'algorithme (section 4) et aux résultats de la mise en oeuvre industrielle (section 5). Nous présentons maintenant les aspects les plus importants de la conception et du fonctionnement de l'algorithme proposé.

3.1. Préliminaires

La procédure d'ordonnancement proposée est en fait basée sur un algorithme évolutionniste (AE). Il s'agit donc d'une procédure de résolution de problème basée sur les concepts d'évolution des espèces afin de produire de bonnes solutions aux problèmes complexes de type

combinatoire. Les étapes principales d'un algorithme évolutionniste sont illustrées par la figure 2.

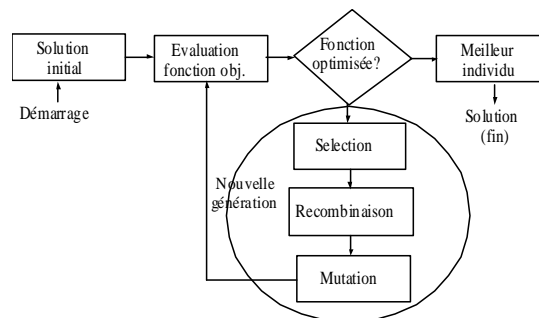


Figure 2. Description générale d'un AE

La différence la plus importante en comparaison avec d'autres méthodes méta-heuristiques (comme la recherche taboue ou le recuit simulé) est que les algorithmes évolutionnistes travaillent avec un ensemble (population) de solutions au lieu de le faire avec une solution unique. Les solutions sont codées dans une chaîne finie de caractères appelés chromosomes et pendant l'exécution de l'algorithme, leur niveau d'adaptation (le *fitness*) est mesuré. En démarrant avec une solution initiale chaque itération génère de nouveaux chromosomes en appliquant les opérateurs de mutation et de *crossover* (Back, 1995 ; Fogel, 1995). L'un des avantages des algorithmes évolutionnistes est leur aptitude à être parallélisé, permettant d'explorer un grand espace de solutions (Sevaux et Dauzère-Péres, 2003).

3.2. Représentation de la solution

Les algorithmes évolutionnistes ont été utilisés depuis quelques années pour résoudre plusieurs types de problèmes d'ordonnancement (Cheng et al., 1996). Dans notre cas, l'algorithme évolutionniste développé utilise la représentation de la solution ou chromosome présenté par un vecteur, comme illustré par la figure 3. Ce vecteur contient le nombre de jobs (1, 2, ..., n) et le nombre de machines d'impression thermique disponibles au moment de l'ordonnancement ($I_1, I_2, I_3, \dots, I_m$). La longueur du chromosome est donc $L=m_1+n$, où m_1 correspond au nombre de machines et n es le nombre de jobs à ordonner. Dans l'exemple de la figure, $L=9$, car le chromosome est composé de 3 machines d'impression et 6 jobs (ordres de fabrication) à ordonner. Ce vecteur définit la séquence d'exécution des jobs sur chaque machine d'impression.

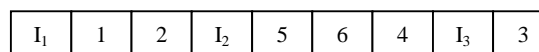


Figure 3. Exemple de chromosome

L'ordre dans lequel les jobs sortent de l'étage d'impression détermine l'ordre d'exécution dans l'étage de découpe. Nous avons donc mis en place une affectation des jobs aux machines de découpe selon la règle FIFO (*First In First Out*) avec une affectation permettant d'équilibrer la charge de travail des machines.

3.3. Crossover et mutation

L'opérateur de *crossover* implémenté dans l'algorithme correspond à un *crossover* uniforme. Il est illustré dans la suite.

Père 1 :

I ₁	1	2	I ₂	5	6	4	I ₃	3
----------------	---	---	----------------	---	---	---	----------------	---

Père 2 :

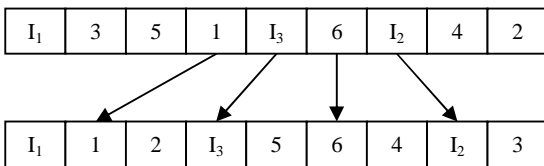
I ₁	3	5	1	I ₃	6	I ₂	4	2
----------------	---	---	---	----------------	---	----------------	---	---

Les gènes sont sélectionnés de façon uniforme à partir des gènes du Père 1 vers le Fils 1 :

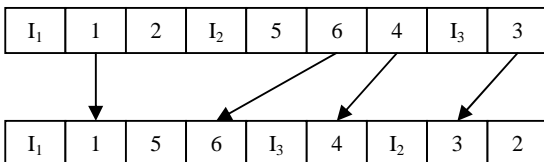
Fils 1 :

I ₁		2		5		4		3
----------------	--	---	--	---	--	---	--	---

Ensuite, les places vides seront complétées avec des gènes provenant du Père 2 qui n'ont pas encore été transmis vers le fils :



Les cas du second fils est similaire. Les gènes du Père 2 sont choisis de façon uniforme, alors que les gènes du Père 1 sont choisis en remplissant les places vides comme expliqué précédemment.



La procédure de mutation est basée sur une stratégie d'échange. Premièrement, un chromosome de l'un des Pères est choisi. Par exemple :

I ₁	1	2	I ₂	5	6	4	I ₃	3
----------------	---	---	----------------	---	---	---	----------------	---

Ensuite, deux gènes de ce Père sont choisis au hasard, disons les gènes I₂ et 6. Ces deux gènes changent leur place. On obtient alors comme résultat le Fils suivant :

I ₁	1	2	6	5	I ₂	4	I ₃	3
----------------	---	---	---	---	----------------	---	----------------	---

3.4. Paramètres

Les paramètres suivants ont été définis. La probabilité de crossover sera successivement de 0.5 et 0.8 selon le plan d'expériences expliqué dans la section 4. La probabilité de mutation aura une valeur fixée de 0.3 pour toutes les expériences. La taille de la population initiale sera de 167 individus, alors que la taille de la population d'individus non dominés sera de 20 individus. La section 4 présente en détail les valeurs des autres paramètres de l'algorithme utilisés lors du plan d'expériences.

3.5. Description de l'algorithme

L'algorithme évolutionniste est le suivant.

Étape 1 : Génération de la population initiale. La génération initiale est générée en utilisant une procédure itérative. Le premier gène de chaque population est choisi dans l'ensemble de machines d'impression disponibles. Les autres gènes sont choisis au hasard à partir des ensembles de machines d'impression et de découpe.

Étape 2 : Calcul des temps d'impression. Les temps d'impression sont calculés, pour chaque job dans chaque solution, comme étant l'instant où l'exécution du job sur l'une des machines d'impression est terminée, selon l'ordre d'exécution défini par le chromosome de la solution pour chaque individu.

Étape 3 : Affectation des jobs aux machines de découpe. Ceci est effectué selon la règle FIFO modifiée. Cette règle considère l'équilibrage de charge entre les machines de découpe au moment de l'assignation d'un job.

Étape 4 : Calcul des temps de fin d'exécution. Pour chaque individu, le temps de fin d'exécution est défini comme l'instant où l'opération d'un job sur la machine de découpe est terminée, selon la séquence définie par le chromosome.

Étape 5 : Calcul du makespan et du nombre de jobs en retard. Le makespan est le temps de fin d'exécution le plus long donné par les

individus de la population. Le nombre de jobs en retard est calculé en comparant les instants de fin d'exécution et la date de livraison promise pour chaque individu.

Étape 6 : *Fitness*. Le *fitness* pour l'individu i est calculé comme $fitness_i = n/N$, où n est le nombre de solutions dans la population dominées par l'individu i , et N est la taille de la population.

Étape 7 : Sélection d'individus. Les individus de la population avec la meilleure valeur de *fitness* sont choisis et placés dans le fichier de solutions non dominées.

Étape 8 : Recombinaison des solutions non dominées. Appliquer les opérateurs de *crossover* et de mutation déjà décrits aux solutions non dominées.

Étape 9 : Sélection de nouveaux individus. Copier les individus de la recombinaison de l'étape 8 et retourner à l'étape 6.

Cette procédure est répétée jusqu'avoir atteint le nombre d'itérations préalablement défini.

3.6. Faisabilité de la solution

Afin de garantir la faisabilité des solutions données par l'algorithme (i.e. chromosomes de chaque itération), le premier gène de chaque chromosome est toujours une machine du stage d'impression. Cela garantit une solution faisable car les jobs seront exécutés sur au moins une machine (celle qui correspond au premier chromosome). Il faut remarquer que selon la distribution des gènes à chaque itération, il est bien possible que dans certains cas toutes les machines aient un ensemble de jobs à exécuter, alors que dans d'autres cas certaines machines n'aient aucun job assigné.

3.7. Pseudo-code

Le pseudo-code général de l'algorithme est le suivant :

```

Begin
  Generate_Initial_Population
  Call Start_EA
  While  $G \leq G_{max}$  do
    Call Generate_Sons
    Call Start_EA
     $G = G + 1$ 
  End While
  Show_Pareto
  Call Select_Solution
End

```

Les sub-routines sont présentées ci-après.

Start_EA

```

Begin
  For  $i = 1$  to  $N$ 
    Compute Population_printing_time
    Determine Cutting_sequences
    Compute Cutting_time
    Compute  $C_{max}$ ,  $ntt$ .
  Next  $i$ 
  Compute fitness.
  Order population by fitness.
  Find Non_dominated
End

```

Generate_Sons

```

Begin
  For  $i = 1$  to Non_dominated
    RandomNum = Rnd(0,1)
    If RandomNum  $\leq P_C$  then
      Apply Crossover
    End if
    If RandomNum  $\leq P_M$  then
      Apply Mutation
    End if
  Next  $i$ 
End

```

Select_solution

```

Begin
  For  $i = 1$  to Non_dominated
    For  $j = 1$  to  $M_i$ 
      Compute Machine_load
    Next  $j$ 
    Find Machine_+_loaded
    Find Machine_-_loaded
    Compute gap ( $i$ )
  Next  $i$ 
  Find min gap( $i$ )
  Final_Solution =  $i$ 
End

```

3.8. Sélection d'une solution unique appartenant au front de Pareto

Après avoir obtenu les solutions données par l'algorithme, le superviseur de production doit ensuite choisir une unique solution de la frontière Pareto pour ordonnancer les ordres de production sur les machines. Afin d'aider ce processus de prise de décision, une dernière étape de plus a été ajoutée à la fin de notre algorithme. Celui-ci utilise une procédure d'équilibrage de la charge de travail sur les machines de chaque étape du processus de fabrication : impression et découpe. L'algorithme calcule la charge de chaque machine pour chaque individu appartenant à la frontière Pareto à la fin des itérations. Ainsi, la solution avec le plus petit écart de charge entre la machine la plus chargée et la machine la moins chargée est choisie comme étant la solution candidate à exécuter sur les machines.

4. RESUME DES EXPERIENCES NUMERIQUES

Dans nos travaux précédents (Vargas-Nieto et Montoya-Torres, 2007a, 2007b), nous avons décrit et effectué en détail des tests expérimentaux permettant de valider l'algorithme. Afin de nous approcher le plus possible de la conception d'un outil d'aide à la décision, l'algorithme évolutionniste a été programmé sous Visual Basic Applications (VBA) de Microsoft® Excel. Ceci permet une plus facile mise en place de la méthode chez notre partenaire en raison de la connaissance et facilite la manipulation des feuilles de calcul. Les tests ont été effectués sur un PC de bureau avec processeur AMD® 2400 Plus, 2.2 GHz et 512 MB de RAM. Sous ces conditions, le temps de calcul nécessaire pour obtenir la solution finale après une centaine de générations est compris entre 10 secondes et 3 minutes. Du point de vu du superviseur de production, ce temps est satisfaisant lors du processus dynamique de prise de décision en temps réel de fabrication car le temps de calcul est compétitif par rapport à la fréquence nécessaire pour ordonnancer la production dans l'usine (pas plus de 2 fois par jour).

Le tableau 1 ci-après présente les paramètres choisis pour effectuer les tests numériques présentés dans nos travaux précédents. Les résultats des tests sont présentés comme illustré dans les figures 4 et 5. A partir des résultats obtenus, nous avons observé qu'avec une probabilité

de *crossover* de 0.8, le front de Pareto est défini beaucoup plus rapidement qu'avec une probabilité de 0.5. Dans le premier cas, nous avons aussi observé qu'à partir de la première itération les solutions obtenues étaient très proches de la frontière de Pareto, alors que dans le second cas il y avait beaucoup de solutions qui en étaient très éloignées. Ces résultats nous ont permis de vérifier les hypothèses présentées dans la plupart de travaux sur les algorithmes évolutionnistes par rapport aux valeurs des probabilités de *crossover* et de mutation lors du paramétrage de l'algorithme. De plus, comme indiqué dans nos travaux précédents, nous avons observé aussi qu'avec un nombre d'itérations supérieur à 100, sont les meilleures solutions obtenues.

Paramètre	Notation	Valeurs
Taille de la population initiale	N	50, 167, 250
Nombre de solutions non dominées	ND	20
Probabilité de <i>crossover</i>	P_c	0.5, 0.8
Probabilité de mutation	P_m	0.3
Nombre de générations	G	1, 5, 10, 20, 100

Tableau 1. Valeurs des paramètres pour les tests

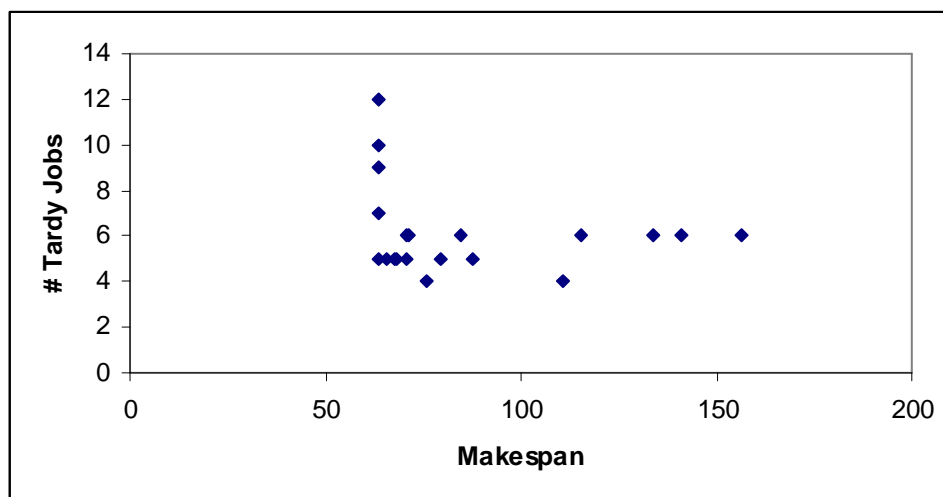


Figure 4. Exemple de solution initiale avec $P_c=0.8$

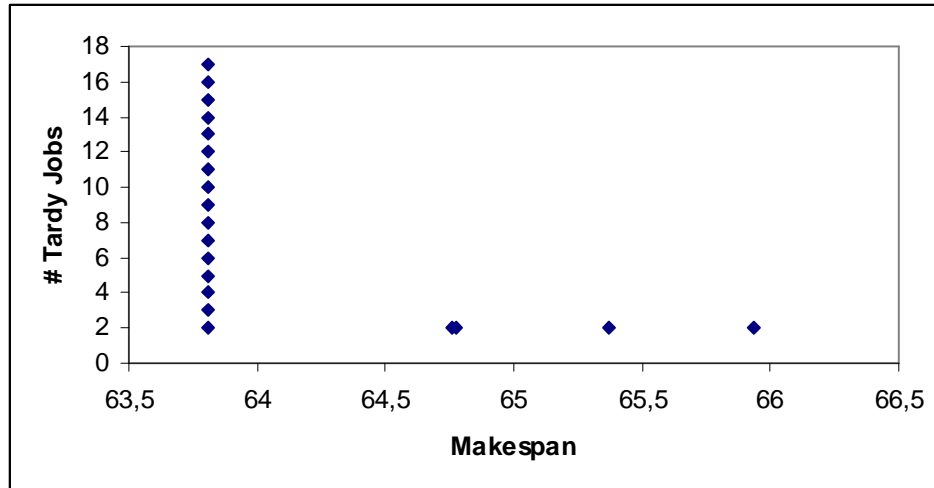


Figure 5. Exemple de solution après 100 itérations avec $P_c=0.8$

4.1. Sélection d’une solution unique : comparaison avec la procédure actuelle d’ordonnancement

Après avoir obtenu le front de Pareto, nous devons appliquer la procédure décrite dans la section 3.6 afin de choisir une seule solution qui sera ensuite implémentée dans l’atelier. Lors de l’exécution de l’algorithme, nous avons observé une amélioration des deux indicateurs étudiés pour une séquence d’exécution choisie selon la méthode décrite. La valeur du makespan donnée par la procédure proposée est en moyenne réduite de 25% par rapport à la procédure actuelle d’ordonnancement, alors que le pourcentage de jobs en retard est diminué de 75%.

4.2. Comparaison avec d’autres procédures de la littérature

Afin d’approfondir l’analyse de la procédure proposée, nous avons comparé les résultats obtenus avec l’algorithme évolutionniste par rapport à d’autres méthodes disponibles dans la littérature. En particulier, les méthodes utilisées pour effectuer cette comparaison sont ceux proposées par le logiciel académique Lekin® Scheduler (Pinedo, 2002) : Shifting Bottleneck (SB) générale avec minimisation du makespan (SB/Cmax), et ses extensions permettant de minimiser le retard maximum (SB/Lmax) et le retard total (SB/TT). Ces deux dernières fonctions objectif ont été choisies comme étant des indicateurs clés pour mesurer le niveau de service car la procédure SB n’a pas une fonction permettant de minimiser directement le nombre de jobs en retard.

Afin de calculer la variation en pourcentage de la méthode proposée par rapport à chacune des procédures du logiciel Lekin® Scheduler, nous avons utilisé l’équation (1) :

$$\%dev = \frac{(Z_{heur} - Z_{prop})}{Z_{heur}} \tag{1}$$

où Z_{heur} correspond à la valeur de la fonction objectif (makespan ou nombre de jobs en retard) donnée par la méthode SB respective et Z_{prop} correspond à la valeur de la fonction objectif obtenue avec la méthode évolutionniste proposée. Afin d’illustrer les résultats obtenus avec ces procédures, le tableau 2 présente une comparaison des valeurs solutions obtenues pour un jeu de donnée particulier.

Nous avons observé que la valeur du makespan est la même en appliquant la procédure évolutionniste ou la méthode SB/Cmax. Cependant, par rapport au nombre de jobs en retard, notre procédure améliore cet indicateur de 78%. En comparant avec la procédure SB/Lmax, la valeur du makespan donnée par notre algorithme est 13% meilleure ; l’indicateur du nombre de jobs en retard est 325% meilleur. Enfin, en comparaison avec SB/TT, les résultats montrent que la méthode évolutionniste proposée est meilleure en 16% et 50%, respectivement pour la valeur du makespan et le nombre de jobs en retard.

Algorithme	Type	Cmax	Jobs en retard
AE proposé	Bi-objectif	63.7	4
General SB	Mono-objectif	63.7	18
General SB/Lmax	Mono-objectif	71.7	17
General SB/TT	Mono-objectif	74.0	6

Tableau 2. Comparaison de la méthode évolutionniste proposée avec Lekin® Scheduler pour un ensemble de données avec 20 jobs

5. IMPLEMENTATION CHEZ LE FABRICANT

A partir des résultats précédents, nous avons paramétré l'algorithme et avons effectué un suivi continu du pourcentage de livraison des ordres de fabrication pendant l'année 2007 (entre les mois de janvier et novembre). L'évolution de cet indicateur est présentée dans la figure 6. La ligne horizontale correspond à l'objectif général (pour toutes les lignes de l'entreprise) d'arriver à 80% de livraisons à temps. Le mois d'avril 2007 (colonne 200704) est signalé comme étant le mois où la procédure proposée dans cet article a été mise en place. En prenant ce point-ci comme référence, nous observons une très mauvaise valeur de l'indicateur avant la mise en place de l'algorithme. Le pourcentage de livraison à temps est inférieur à 50% avant la mise en place de la méthode. Après le mois d'avril, l'implémentation de l'algorithme proposé a permis que le pourcentage d'ordres délivrés à temps s'approche de 80% en juillet et soit supérieur à cet objectif industriel pour les mois suivants. En comparaison avec les autres lignes de production de l'entreprise, la procédure évolutionniste proposée a permis de s'approcher et même dépasser leurs performances.

6. CONCLUSIONS ET PERSPECTIVES

Dans cet article, nous avons étudié le problème d'ordonnement de la production chez un fabricant d'étiquettes pour vêtements. Le système étant modélisé comme un flowshop flexible à deux étages, le problème bi-objectif de minimisation du makespan et du nombre de jobs en retard est résolu en implémentant un algorithme évolutionniste. Des données fournies par l'entreprise nous ont permis de développer des expériences numériques et de paramétrer l'algorithme. La mise en place dans l'entreprise s'est effectuée entre les mois d'avril et novembre 2007, montrant une amélioration importante des indicateurs clés de la ligne de fabrication.

Au niveau de perspectives, une première extension naturelle du travail serait de considérer d'autres contraintes dans le modèle, telles que le stockage limité, ou encore un modèle avec plus de deux étages. D'autre part, en termes de fonctions objectif considérées, il est bien connu que, si la minimisation du nombre de jobs en retard est un indicateur du niveau de service pour l'entreprise, celui-ci peut conduire à des retards très importants (dates de livraison très supérieure aux dates promises). Un aspect intéressant pourrait ainsi être la mise en place d'une borne supérieure pour le retard autorisé des jobs.

REMERCIEMENTS

Les auteurs remercient la collaboration des membres de l'équipe production de l'entreprise partenaire du projet, ainsi que le soutien de l'Universidad del Norte et de l'Universidad de La Sabana.

REFERENCES

- Acerro, M., J.R. Montoya-Torres, C.D. Paternina-Arboleda (2004). Scheduling jobs on a k -stage flexible flow shop. In: A. Oulamara, M.C. Portman (eds.) *Proceedings of the Ninth International Workshop on Project Management and Scheduling*, p. 242-245.
- Allaoui H., A. Artiba (2006). Scheduling two-stage hybrid flow shop with availability constraints. *Computer & Operation Research* 33, 1399-1419.
- Azizoglu M., E. Cakmak, S. Kondakci (2001). A flexible flowshop problem with total flow time minimization. *European Journal of Operational Research*, 132, 528-538.
- Bertel S., J.C. Billaut (2004) A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research*, 159, 651-662.

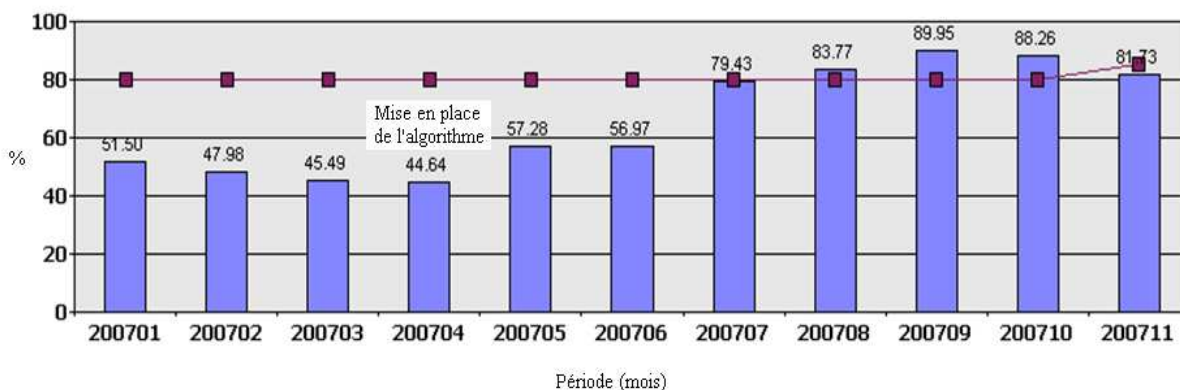


Figure 6. Evolution du pourcentage de livraison à temps des ordres de fabrication entre janvier et novembre 2007

- Botta-Genoulaz V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*, 64, 101-111.
- Brah S.A., J.L. Hunsucker (1991). Branch and Bound algorithm for the flow shop with multiple processors. *European Journal of Operation Research*, 51, 88-99.
- Chen, B. (1994). Scheduling multiprocessor flowshops. In: *Advances in Optimization and Approximation*, Kluwer, Dordrecht.
- Chen, B. (1995). Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at one stage. *Journal of the Operational Research Society* 46, 234-244.
- Cheng R., M. Gen, Y. Tsujimura. 1996. A tutorial survey of job-shop scheduling problems using genetic algorithms—I. Representation. *Computers & Industrial Engineering*, 30, 983-997.
- Dessouky, M., M. Dessouky, S. Verma (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research* 109, 620-631.
- Djellab H., K. Djellab (2002). Preemptive Hybrid Flowshop Scheduling problem of interval orders. *European Journal of Operational Research*, 137, 37-49.
- Guinet A., M. Solomon (1996). Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, 34, 1643-1654.
- Gupta, J. (1988). Two stage hybrid flow shop scheduling problem. *Journal of the Operations Research Society* 38, 359-364.
- Haouari, M., R. M'Hallah (1997). Heuristic algorithms for the two-stage hybrid flowshop problem. *Operations Research Letters* 21, 43-53.
- Jin, Z., Z. Yang, T. Ito (2006). Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *International Journal of Production Economics*, 100, 322-334.
- Lee, C.Y., G. Vairaktarakis (1994). Minimizing makespan in hybrid flowshops. *Operations Research Letters* 16, 149-158.
- Linn, R., W. Zhang (1999). Hybrid flow shop scheduling: A survey. *Computers and Industrial Engineering* 37, 57-61.
- Morita H., N. Shio (2005) Hybrid Branch and Bound Method with Genetic Algorithm for Flexible Flowshop Scheduling Problem. *JSME International Journal Series C*, 48, 46-52.
- Moursli, O., Y. Pochet (2000). A branch and bound algorithm for the hybrid flow shop. *International Journal of Production Economics* 64, 113-125.
- Pinedo, M. (2002). *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Englewood Cliffs, New Jersey.
- Portmann M.C., A. Vignier, D. Dardilhac, D. Dezalay (1998). Branch and bound crossed with GA to solve hybrid flowshops. *European Journal of Operational Research*, 107, 389-400.
- Riane, F., A. Artiba, S.E. Elmaghraby (1998). A hybrid three-stage flexible flowshop problem: Efficient heuristics to minimize makespan. *European Journal of Operational Research* 109, 321-329.
- Ruiz R., C. Maroto (2006). A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. *European Journal of Operational Research*, 169, 781-800.
- Sawik T. (2002) An exact approach for batch scheduling in flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 36, 461-471.
- Sevaux, M., S. Dauzère-Pères (2003) Genetic algorithms to minimize the weighted number of late jobs on a single machine. *European Journal of Operational Research*, 151, 296-306.
- Soewandi, H., S.E. Elmaghraby (2001). Sequencing three-stage flexible flowshops with identical machines to minimize makespan. *IIE Transactions* 33, 985-993.
- Sriskandarajah C., E. Wagneur (1991). Hierarchical control of the two processor flow-shop with state dependent processing times: complexity analysis and approximate algorithms *INFOR, Canadian Journal of Information Systems and Operational Research*, 29, 193-205.
- Tang L., H. Xuan, J. Liu (2006). A new Lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time. *Computers & Operations Research*, 33, 3344-3359.
- T'Kindt V., J.C. Billaut (2006) *Multicriteria Scheduling: Theory, Models and Algorithms*. Springer Verlag. Segunda Edición.
- Vargas-Nieto, F. (2007) *Diseño e implementación de un modelo de programación de operaciones bi-objetivo basado en algoritmos evolutivos aplicado a la industria de marquillas estampadas por transferencia térmica*. Tesis de Maestría en Ingeniería Industrial, Universidad del Norte. Barranquilla, Colombia.
- Vargas-Nieto F., J.R. Montoya-Torres (2007a). Design and implementation of a bi-objective EA-based scheduling strategy: Case study in thermal-printed label manufacturing. *Proceedings of the International Industrial Simulation Conference (ISC-2007)*, p. 219-222. Delft, The Netherlands.
- Vargas-Nieto F., J.R. Montoya-Torres (2007b). Scheduling a thermal-printed label manufacturing plant using an evolutionary algorithm. *Proceedings of the 19th International Conference on Production Research (ICPR-19)*, CD-ROM. Valparaiso, Chile.
- Vignier, A., P. Commandeur, C. Proust (1997). New lower bound for the hybrid flowshop scheduling problem. *Proceedings of the 1997 IEEE Conference on Emerging Technologies and Factory Automation*, 446-451.

- Wu J., C.F. Chien 2008. Modeling semiconductor testing job scheduling and dynamic testing machine configuration. *Expert Systems with Applications*, (In Press), Corrected Proof, Available online 20 July 2007.
- Xuan H., L. Tang (2007). Scheduling a hybrid flowshop with batch production at the last stage. *Computers & Operations Research*, 34, 2718-2733.