Ordonnancement d'un atelier flow-shop avec écart de performance entre les operateurs

H. Benferroudj
Laboratoire LAP / Dpt génie industriel / Université de Batna
Avenu Chahid Boukhlouf
05000 Batna, Algérie
nawalbf@gmail.com

H.Smadi
Laboratoire LAP / Dpt génie industriel / Université de Batna
Avenu Chahid Boukhlouf
05000 Batna, Algérie
h.Smadi@hotmail.fr

Résumé —**Cet article adresse le problème d'ordonnancement d'un atelier de type flow-shop intégrant l'affectation des opérateurs en vue de minimiser le critère du makespan (Cmax). Tout opérateur a une performance propre qui influence les durées des opérations réalisées, indépendamment de la machine ou de l'opération. Le problème consiste à déterminer à la fois l'affectation des opérateurs et la séquence de passage des tâches (permutation) qui minimisent la durée de réalisation. Nous utilisons deux heuristiques pour résoudre le problème : les algorithmes génétiques et l'algorithme NEH. Une comparaison entre les deux heuristiques conclura ce travail.**

Mots-Clé –Ordonnancement ; flow-shop ; makespan ; opérateur ; performance ; algorithme génétique ; algorithme NEH.

I. INTRODUCTION

Les ressources humaines occupent de nos jours une place de plus en plus importante dans les coûts de production, ce qui justifie d'y accorder une attention particulière. Elles jouent en effet un rôle prépondérant dans la recherche de flexibilité du système de production en réponse à une situation de demande variable. Leur travail se situe en interaction avec les autres ressources, en particulier les machines [1].

Dans la littérature, plusieurs études de recherche, référençant l'ordonnancement des ressources humaines, ont été menées [2, 3, 4]. Cependant, très peut d'entre elles ont considéré le problème avec prise en compte les performances du personnel. Tchommo et al [5] ont réalisé une synthèse bibliographique sur l'ordonnancement simultané de moyens de production et des ressources humaines. Il ressort de cette étude un manque dans la prise en compte des ressources humaines en ordonnancement.

L'article présente une résolution du problème d'ordonnancement d'un atelier de type *flow-shop* intégrant l'affectation des opérateurs. Tout opérateur a une *performance* propre qui influence les durées des opérations réalisées, indépendamment de la machine ou de l'opération. Cet article est organisé comme suit. Après la formulation du problème traité, nous présentons dans la section III deux algorithmes (les *algorithmes génétiques* et l'algorithme *NEH*) proposés pour résoudre le problème .Des résultats numériques sont ensuite présentés, et nous terminons cet article par une conclusion générale.

II. FORMULATION DU PROBLEME

Il s'agit de minimiser la durée totale de réalisation des tâches pour un atelier à cheminement unique (problème de type $Fm\|C_{max}$) avec un nombre d'opérateurs égal au nombre de machines. Tout opérateur e a une *performance* propre ρ_e qui influence les durées des opérations réalisées, indépendamment de la machine ou de l'opération. Le problème consiste à déterminer à la fois l'affectation des opérateurs et la séquence de passage des lots qui minimisent la durée de réalisation.

A. Modelisation

Le modèle en variable mixtes ci-dessous est dérivé du modèle de l'atelier à cheminement unique $Fm\|C_{max}$ présenté par Pinedo [6]. On y ajoute de nouvelles contraintes et on modifie la fonction objectif. Les contraintes liées à séquence des taches sont inchangées.

Affectation des opérateurs

Il s'agit de déterminer sur quelle machine va travailler chaque opérateur. Les m^2 variables de décision de ce problème d'affectation sont $\mathcal{Y}_{e,j}=1$ si l'opérateur e travaille sur la machine j, 0 sinon. Les contraintes liées à l'affectation sont les suivantes :

- Variables de décision binaires:

$$\mathcal{Y}_{e,j}\in\{0\,;1\} \qquad \forall j\in[1\,;m],\ \forall e\in[1\,;m] \tag{1}$$

- Une seule machine par opérateur, un seul opérateur par machine :

$$\sum_{j=1}^{m}\mathcal{Y}_{e,j}=1\ \ \forall e\in[1;m],\ \sum_{e=1}^{m}\mathcal{Y}_{e,j}=1\ \ \forall,j\in[1;m] \tag{2}$$

La durée opératoire réelle $p_{i,j}$ d'un lot i sur une machine j est déterminée à partir de la durée moyenne

$\bar{p}_{ij}$, indépendante des opérateurs, et de la performance ρ_e de l'opérateur affecté à la machine *j*.

$$p_{i,j} = \sum_{j=1}^{m} y_{e,j} . \rho_e . \bar{p}_{ij} \quad \forall, i \in [1 ; n], \forall \in [1 ; m] \qquad (3)$$

Fonction objectif

La valeur du critère C_{max} correspond à l'intervalle entre le début du premier lot sur la première machine et la fin du dernier lot sur la dernière machine :

$$\min_{x,y} C_{max} = \min_{x,y} \Big(\underbrace{\sum_{j=1}^{m-1} \sum_{i=1}^{n} x_{i,1} . p_{i,j}}_{(a)} + \underbrace{\sum_{i=1}^{n-1} I_{m,i} + \sum_{i=1}^{n} p_{i,m}}_{(b)} \Big) \qquad (4)$$

Où

(a) représente la somme des durées opératoires du premier lot, pour les machines 1 à m−1

(b) représente la somme des temps d'inactivité et des durées opératoires sur la dernière machine.

On note que le modèle n'est plus linéaire, compte tenu du fait que les durées opératoires $p_{i,j}$ dépendent désormais des variables d'affectation $y_{e,j}$.

B. Complexeté

Le problème appartient à la classe $\mathcal{NP}$, car pour toute solution pour laquelle $C_{max} \leq C$ est vrai (avec C donné), cela se vérifie en un temps polynomial. De plus, si on considère les cas pour lesquels ρ_e est constant pour tout $e \in [1 ; m]$, le problème se ramène à un problème d'atelier à cheminement unique de type $Fm \| C_{max}$. Par conséquent, le problème est fortement NP-difficile lorsque le nombre de machines est supérieur à 2 [7] pour les variables binaires, la combinatoire est la suivante :

Il ya $n!$ solutions à chaque problème de séquencement (variables $x_{i,k}$), et le problème d'affectation (variables $y_{e,j}$) a $m!$ solutions. Le problème complet a donc $n! \times m!$ solutions.

Lorsque plusieurs opérateurs ont la même performance, le nombre de solutions du problème d'affectation diminue. Les m opérateurs sont séparés en $\mathcal{W}$ catégories de performance :

a: indice des catégories (a=1,2,…, $\mathcal{W}$)

f_a : effectif de la catégorie a, tel que $\sum_{a=1}^{\mathcal{W}} f_a = m$

A l'intérieur d'une catégorie, toutes les affectations sont interchangeables. Le nombre d'affectations du problème est donc :

$$C_m^{f_1} . C_{(m-f_1)}^{f_2} . C_{(m-f_1-f_2)}^{f_3} . \cdots . C_{(m-\sum_{a=1}^{(\mathcal{W}-1)} f_a)}^{f_{\mathcal{W}}} = \frac{m!}{\prod_{a=1}^{\mathcal{W}} (f_a!)}$$

III. Les Methodes De resolution

Les deux heuristiques que nous allons présenter ci-après sont l'algorithme NEH [8] et les algorithmes génétiques [9]. Ce choix fait référence aux travaux de Poonabalam et al [10] où ils ont comparé des heuristiques constructives pour le problème du flow shop et les algorithmes génétiques. Ils sont arrivés à la conclusion que NEH était l'heuristique qui donnait les meilleurs résultats pour l'approche constructive, cependant les algorithmes génétiques étaient encore meilleurs.

A. L'algorithme NEH

L'algorithme est résumé comme suit :

Étape 1 (*La séquence de passage des travaux)*

1) Pour chaque tâche i (i ∈ [1 ; n]), calculer

$$T_i = \sum_{j=1}^{m} \bar{p}_{ij}$$

2) Ordonner dans une liste L les tâches par ordre de la valeur T_i.

3) Prendre les deux premières tâches de la liste L et trouver la meilleure séquence pour ces deux tâches en calculant les deux ordonnancements possibles. Ne plus changer les positions relatives de ces deux tâches lors des prochaines étapes de l'heuristique ; i :=3.

4) Prendre la tâche se trouvant en i^{ieme} position dans la liste L et trouver la meilleure séquence en plaçant cette tâche dans les i positions possibles parmi les tâches déjà placées

5) Si (i < n) alors i := i+1 ; aller à l'étape 4. Sinon STOP : La séquence est trouvée.

Étape 2 (*L'affectation des opérateurs*)

L'affectation $\widehat{A}f$ des opérateurs est déterminée à l'aide d'une *règle intuitive* qui consiste à affecter les opérateurs, pris dans l'ordre de performance décroissante, aux machines, prises dans l'ordre décroissant de la charge. $\sum_j \bar{p}_{i,j}$.

Les deux étapes de l'heuristique sont indépendantes et peuvent donc être réalisées dans n'importe quel ordre. Outre sa simplicité, un avantage majeur de la règle d'affectation est qu'elle ne nécessite pas de quantification de la performance, action délicate qui se révèle fortement subjective et sujette à variations.

Règle d'affectation des opérateurs

La figure1 présente les valeurs du critère C_{max}, pour un cas à 20 tâches et 6 machines. Ces valeurs sont obtenues en balayant l'ensemble des affectations et pour chaque affectation, en déterminant la séquence S (p_{ij}) à l'aide de NEH.

Un élément important pour l'évolution de C_{max} est la charge réelle $\sum_i p_{i,j}$ sur la machine la plus chargée (appelée µ). Les affectations sont classées en abscisse suivant cette règle. La courbe du haut représente les valeurs du critère et celle du bas, les valeurs de $\sum_i p_{i,\mu}$.

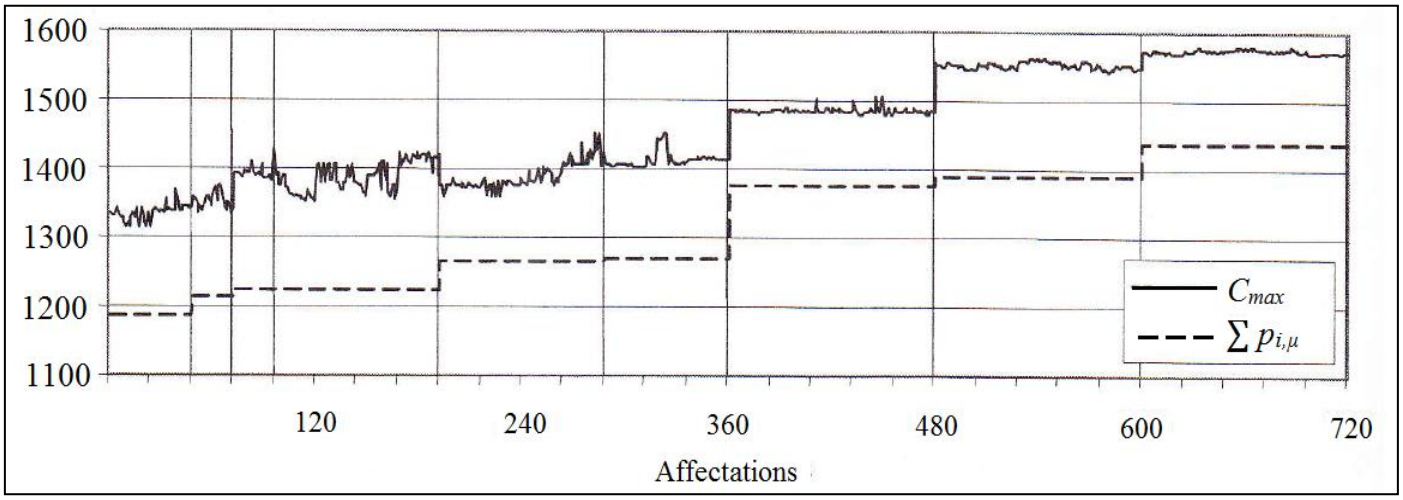

Figure 1: Valeurs de C_{max} classées suivant $\sum_i p_{i,j}$ sur la machine la plus chargée.

Les affectations sont groupées par ensembles de même valeur de $\sum_i p_{i,\mu}$ appelés paliers. Un palier $\mathcal{P}_{e_\mu,\mu}$ est l'ensemble des affectations qui contiennent le même couple (e_μ,µ) et pour lesquelles $\max_j (\sum_i p_{i,j}) = \sum_i p_{i,\mu}$. On note que l'affectation $\hat{A}f$ suivant la règle intuitive même à une valeur qui se trouve dans le premier palier, palier de plus petite valeur de $\sum_i p_{i,\mu}$.

B. L'algorithme Génétique

Les algorithmes génétiques (AG) sont une solution à notre problème. Ils ont la particularité de s'inspirer de l'évolution naturelle des espèces. Chaque espèce s'adapte à son environnement évolutif. Le vocabulaire utilisé est similaire à celui de la génétique. Nous aurons une population d'individus, chaque individu est représenté par un chromosome constitué de plusieurs gènes. Certains individus se reproduisent et donnent naissance à de nouveaux individus possédant ses propres chromosomes, d'autres disparaissent.

Les algorithmes génétiques issus des travaux de Holland [11] et repris dans de nombreux travaux [12] [9] vont tenter de reproduire ce modèle d'évolution afin de trouver des solutions admissibles à un problème donné.

Les principales étapes d'un algorithme génétique sont :

Début :

Génération aléatoire d'une population d'individus

Répéter

Evaluer chaque individu en fonction des objectifs
Sélectionner les individus pour le croisement
Croiser les individus sélectionnés
Muter certains individus

***Jusque** : critère(s) de fin*

Retourner le meilleur compromis

Fin

Le AG commence par la création d'une solution de départ population initiale générée d'une manière aléatoire. Chaque solution (chromosome) est évaluée et son coût (fonction objectif) est calculé. Le mécanisme de production permet de sélectionner les parents. Ensuite, une nouvelle génération est crée en utilisant les opérateurs de reproduction (croisement et mutation).

1. Représentation des chromosomes

La représentation des chromosomes pour le problème d'ordonnancement de production qu'est le flow-shop et l'affectation des travailleurs est montré dans la figure 2.

Afin de démontrer la représentation des chromosomes, nous prenons un exemple de neuf tâches et trois machines.

Dans la figure2, les neuf premiers gènes représentent l'ordonnancement et les trois gènes suivants représentent l'affectation des opérateurs aux machines (sur quelle machine va travailler chaque opérateur). La longueur du chromosome est donc égale à $n+m$ (n : nombre de tâches, m : nombre de machines).

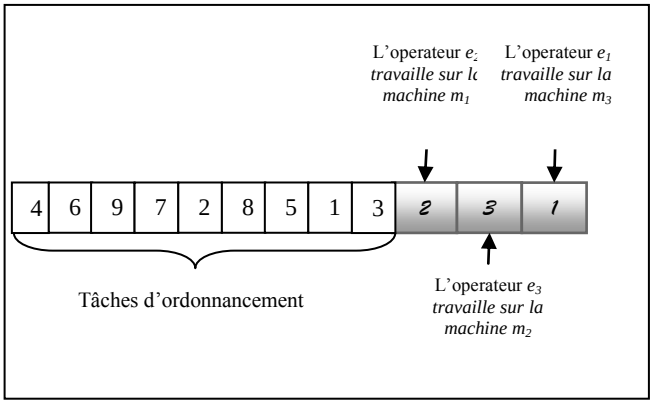

Figure 2. Représentation de chromosome

2. *Sélection*

La sélection de reproduction et de remplacement garde les meilleures solutions à travers leurs fonctions objectives. Cette sélection est donc élitiste. Nous avons mis en place la sélection est N/2 élitisme (N la taille de la population), elle consiste à sélectionner les N/2 meilleures solutions dans la population courante.

La sélection de remplacement concerne l'évolution de la population d'une génération à une autre. La taille de la population est fixée à N. Donc seulement N meilleures solutions sont gardées pour la population suivante. Elles sont choisies parmi les N+N/2 solutions (N parents de la population courante et N/2 enfants obtenus par la reproduction). Cette sélection est donc élitiste. Il est à noter que les individus doublons sont supprimés à cette étape afin de maintenir une diversité suffisante de la population.

3. *Opérateur de croisement*

L'opérateur de croisement classique à un seul point est mis en place pour combiner les solutions de l'ensemble des parents. L'ensemble des N/2 parents sélectionnés est partitionné de manière aléatoire en N/4 couples. Le point de croisement est aussi choisi de manière aléatoire à chaque génération de l'algorithme dans l'intervalle $[1,\ldots, n+m]$. Chaque couple subit une opération de croisement avec une probabilité de croisement. Les couples qui ne subissent pas une opération de croisement sont maintenus.

4. *Opérateur de mutation*

Dans notre étude, chaque solution peut subir une permutation aléatoire. On lançant une variable aléatoire k entre 1 et $n+m$ on peut avoir deux types de mutation. On peut avoir une mutation sur les gènes de 1 à n (les gènes correspondant à l'ordonnancement) comme le montre la figure 4.

Dans ce cas, la mutation concerne uniquement deux gènes k_1 et k_2 ($k_1 \neq k_2 = 1,\ldots, n+m$) du chromosome. Les deux gènes sont sélectionnés aléatoirement à chaque génération par une simple distribution uniforme.

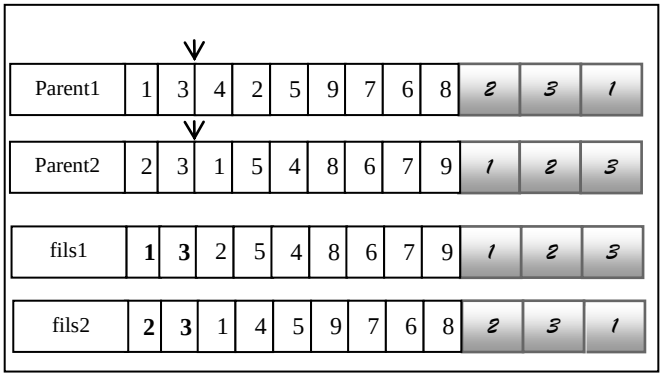

Figure 3. Opérateur de croisement

Comme on peut avoir une mutation sur les gènes $n+1$ et $n+m$ (les gènes qui correspondent aux affectations des opérateurs) et dans ce cas le gène est régénéré aléatoirement.

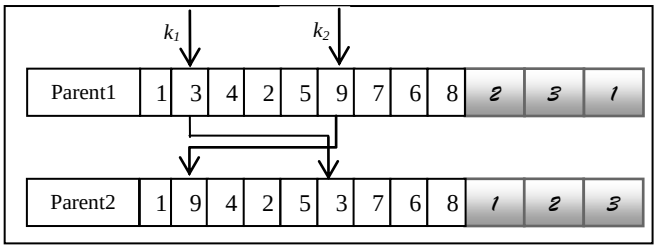

Figure 4. Opérateur de mutation

5. *Paramètres*

Les probabilités de mutation et de croisement sont Fixées d'une manière classique ($P_{crossover}$=0.95; $P_{mutation}$=0.05). En effet, ce choix a été motivé par les résultats donnés par quelques simulations préliminaires qui ont montre que ces grandeurs doivent être judicieusement choisies. Les figures 5, 6 et 7 montrent des tests de réglage de ces probabilités sur un exemple de taille 20×6 et une population de 100 individus. De telles simulations prouvent qu'un taux élevé de mutation conduit à une difficulté de convergence traduite par l'obtention d'un extremum local. Par contre, un choix de taux de mutation faible permet de réaliser un bon compromis entre exploration et optimisation.

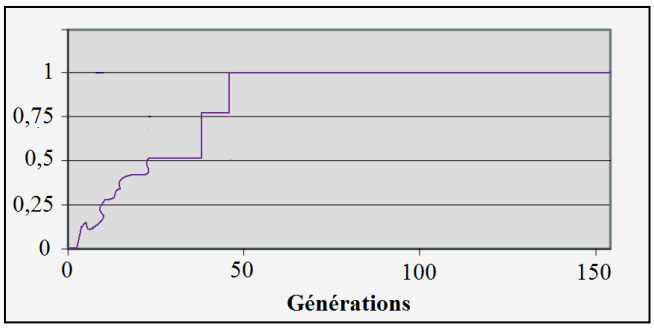

Figure 5. Probabilité de convergence vers la solution optimal avec $P_{crossover}$=0.95 et $P_{mutation}$=0.05

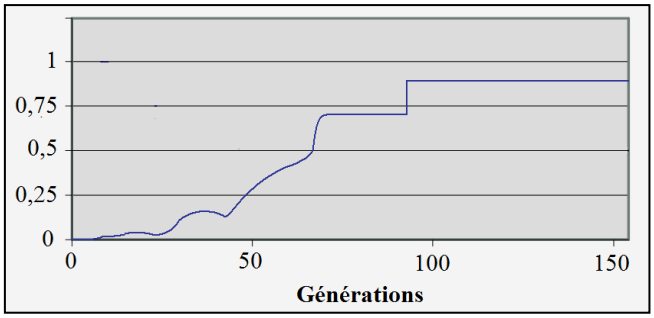

Figure 6. Probabilité de convergence vers la solution optimal $P_{crossover}$=0.6 et $P_{mutation}$=0.4

IV. TESTS ET RESULTATS

Les instances utilisées pour tester les méthodes sont créées d'une manière aléatoire. Les paramètres sont génères de la manière suivante :

- Le nombre de tâches *n* varie dans {5 ; 1000},
- le nombre de machines *m* varie dans [4;50],
- Les durées opératoires sont dans [1;10].

La figure 8 présente une étude comparative entre les résultats des deux heuristiques.

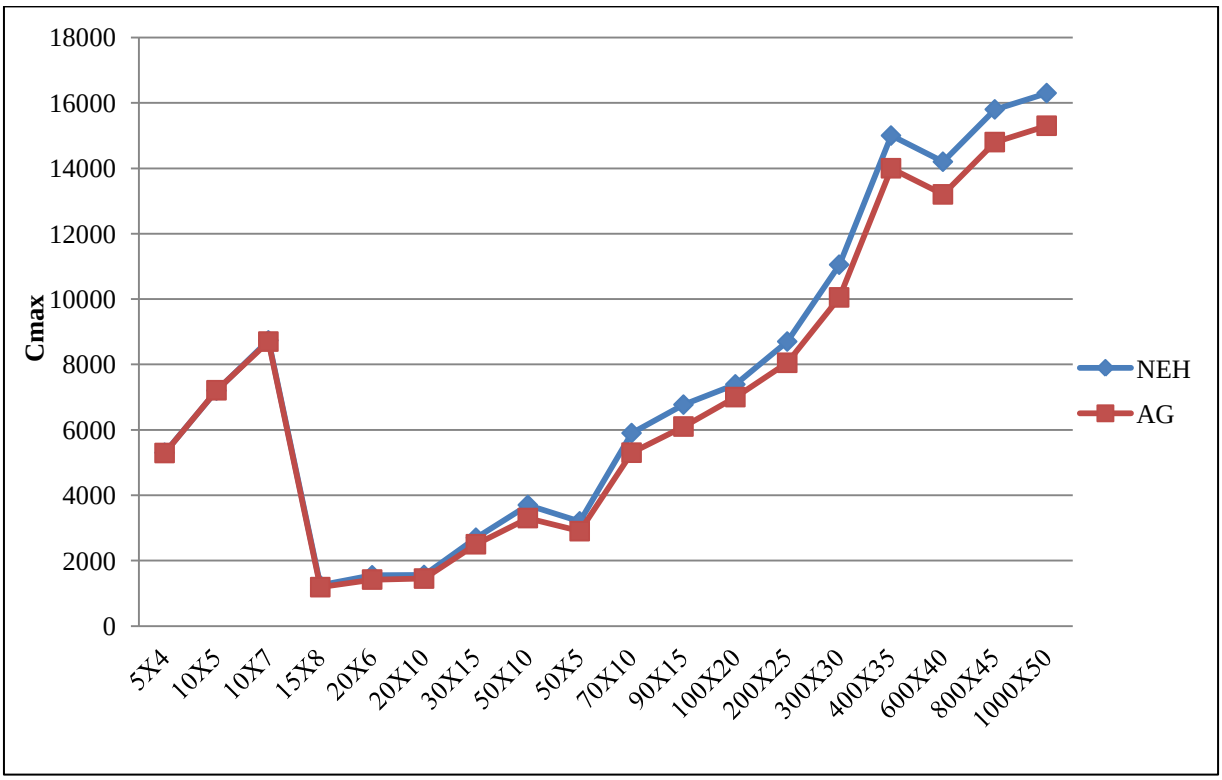

Figure 8. Résultats d'ordonnancement

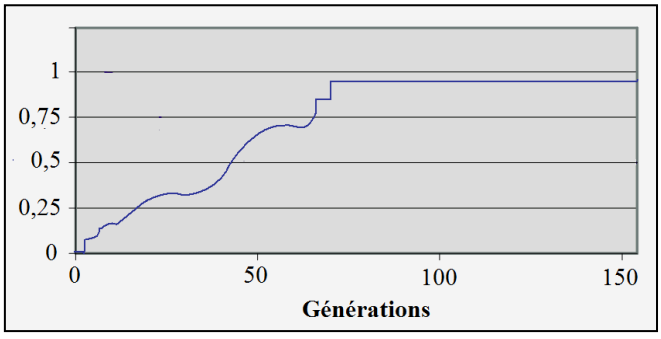

Figure 7. Probabilité de convergence vers la solution optimal avec $P_{crossover}$=0.8 et $P_{mutation}$=0.2

Les AG ont fourni des résultats intéressants sur les différents tests effectués, mais d'un écart faible par rapport aux meilleurs résultats trouvés par l'algorithme NEH. Cependant, dans les cas (400×35, 600×40, 800×45, 1000×50) les algorithmes génétiques ont étaient nettement meilleurs que l'algorithme NEH.

L'algorithme génétique que nous avons proposé semble bien adapté aux problèmes de grande taille. Donc, on remarque bien l'efficacité de l'optimisation par algorithme génétique pour la résolution du problème d'ordonnancement avec écart de performance entre les opérateurs.

V. CONCLUSION

Dans cet article, nous avons étudié un problème d'ordonnancement d'un atelier à cheminement unique avec écart de performance entre les opérateurs pour minimiser le critère du makespan.

Nous avons proposé deux heuristiques (l'algorithme génétique et l'algorithme NEH) et nous avons effectué une étude expérimentale pour évaluer leur performance. Les résultats expérimentaux ont confirmé les bonnes performances de l'algorithme génétique.

REFERENCES

[1] P. Baptiste, A. Hait, F. Soumis, "Gestion de production et ressources humaines," Presses Internationales Polytechnique, Montréal, Canada, 2005.

[2] A. Haït, et P. Baptiste, " Ordonnancement d'atelier et ressources humaines : affectation des opérateurs dans flowshop," soumis au journal européen des systèmes automatisés, 2005.

[3] M. Cheurfa, "Gestion des ressources humaines en production cyclique," Thèse de doctorat, École des Mines de Saint Etienne, France, 2005.

[4] J.L. Tchommo, "Ordonnancement simultané de production et des ressources humaines," Mémoire de maîtrise, École Polytechnique de Montréal, Canada, 2004.

[5] J.L. Tchommo, P. Baptiste, A. Hait, F. Soumis, "Etude bibliographique de l'ordonnancement simultané des moyens de production et des ressources humaines," Congrès International de Génie Industriel (GI), 2003.

[6] M. Pinedo, "Scheduling: Theory, Algorithms, and Systems", 2nd edition, Prentice-Hall, 2002.

[7] M. Garey, D.S. R., Johnson et R. sethi, " The complexity of flowshop and jobshop scheduling," Mathematics of Operations Research, t.1, n° 2, p.117-129, 1976.

[8] N. Nawaz, E. Jr. Enscore and I. Ham, "A heuristic algorithm for the m machine, n job flow shop sequencing problem," Omega, 11, 1983.

[9] D.E. Goldberg, "Genetic algorithms in search, Optimisation and Machine Learning," Addison-Wesley, Mass., 1989.

[10] S.G Poonabalam, P. Aravindan and S. Chandrasekaran, "Constructive and improvement flow shop scheduling:an extensive evaluation," production planning &control,vol.12,no.4,335-344, 2001.

[11] J.H. Holland, "Adaptation in natural and artificial system," Ann Arbor, The University of Michigan Pres, 1975.

[12] C.R Reeves, "A genetic algorithm for flow shop sequencing," Computers and Research, 22:5–14, 1995.