

TECHNIQUES D’AFFECTATION FLEXIBLE DES RESSOURCES : RÉSOLUTION PAR ALGORITHME GÉNÉTIQUE MONOCRITÈRE

Kouassi Hilaire EDI^{1*}, Tioman AKA², Boko AKA¹, Philippe
DUQUENNE³ et Jean-Marc LE LANN³

¹Laboratoire de Mathématique et Informatique, Université d’Abobo-Adjamé,
02 BP 801 Abidjan 02, Côte d’Ivoire

²Laboratoire de Physique de la Matière Condensée et de Technologie,
Université de Cocody-Abidjan, 21 BP 872 Abidjan 21, Côte d’Ivoire

³LGC - UMR-CNRS 5503, PSI - Génie Industriel, INPT-ENSIACET, 118
route de Narbonne, 31077 Toulouse cedex 4, France

(Reçu le 11 Mai 2010, accepté le 19 Septembre 2010)

* Correspondance et tirés à part, e-mail : edi.hilaire@yahoo.fr

RÉSUMÉ

Dans cet article qui est la suite logique du premier, nous présentons une méthodologie de résolution pour trouver des solutions de la formulation mathématique. Cette méthodologie est basée sur l’utilisation des algorithmes génétiques monocritères qui permettent d’affecter aléatoirement les acteurs sur les différentes charges d’une tâche. Dans cette méthode, nous considérons que les acteurs affectés sur chaque tâche effectuent le même nombre d’heures de travail. Le nombre d’acteurs affectés se fait de manière aléatoire dans la limite du nombre total d’acteurs affectables ; c’est-à-dire, ceux qui ont une efficacité suffisante dans une compétence demandée par la tâche. A partir de ces choix, un calcul est fait pour le travail global périodique de chaque acteur ce qui permet de calculer le nombre global d’heures de l’activité et partant de là son coût.

Mots-clés : *Affectation flexible, polyvalence, modulation du temps de travail, algorithme génétique.*

ABSTRACT

Methods of flexible allocation of resources : resolution by genetic algorithms monocriteria

In this article which is the logical suite of the first, we introduce a methodology of resolution to find resolutions of mathematical formulation. This methodology is based on the use of genetic algorithms monocriteria which allows to affect randomly the actors on the different expenses of a

task. In this method, we consider that the actors affected on every task perform the same number of hours of employment. The number of affected actors is made in a unpredictable way within the limits of the total number of actors concerned; that is to say, those who have a sufficient effectiveness in a competence asked by task. From these choices, a counting is made for the periodical total job of every actor what allows to calculate the total number of the hours of activity and part from there its expense.

Keywords : *Flexible allocation, versatility, modulation of working time, genetic algorithm.*

I - INTRODUCTION

L'objet de la formulation mathématique développée dans l'article précédent est l'obtention d'une solution au problème considéré globalement : opérer des choix d'affectation pour couvrir les charges de travail $W_{i,k}$ des tâches d'une activité, en tenant compte de la polyvalence des acteurs et de la modulation du temps de travail. Des chercheurs ont étudié ces problèmes d'affectation mais sans jamais prendre en compte simultanément les deux leviers de flexibilité que nous étudions dans le cas de notre problème. Pour ces exemples, nous pouvons citer les travaux de [1,2] pour l'étude de la polyvalence et les travaux de [3] pour l'étude sur le temps de travail.

Pour trouver un code capable de traiter notre modèle nous nous sommes appuyés sur les expériences qu'ont vécues les chercheurs sur les travaux récents portant sur les problèmes d'affectation de ressources humaines prenant en compte l'un ou l'autre levier de flexibilité que nous étudions. Ainsi, avec une belle unanimité [2, 4-8], il semble qu'une telle approche est très rapidement limitée par les difficultés de résolution du problème de programmation non linéaire en variables mixtes associé en utilisant les méthodes exactes comme moyen de résolution (limites liées à la capacité des algorithmes, et au temps de calcul) ; son application à des problèmes de dimension industrielle n'est donc pas envisageable sans des moyens informatiques considérables.

Fort de ces expériences de nos prédécesseurs, nous nous sommes tournés vers les méthodes approchées. Ainsi, nous avons développé une méthodologie basée sur l'utilisation des algorithmes génétiques monocritères [7].

II - MÉTHODOLOGIE

II-1. Principe général des algorithmes génétiques

Les algorithmes génétiques monocritères ou mono-objectif sont utilisés pour résoudre des problèmes d'optimisation ne prenant en compte qu'une seule

fonction objectif. Cependant, la mise en œuvre nécessite la définition des procédures et des paramètres associés [5,9,10] : un codage approprié (génotype) qui permet de représenter les variables du problème où chaque possibilité des variables de décision prend le nom de « gène ». Un jeu complet, dans lequel toutes les variables de décision sont représentées avec pour chacune une valeur possible, est un individu : il est représenté par un vecteur dans lequel chacun des gènes nécessaire pour décrire une solution est présent : ce vecteur s'appelle un « chromosome ». On peut ainsi générer un ensemble d'individus (une « population ») aléatoirement, de manière à balayer l'ensemble de l'espace des solutions. Ces individus sont ensuite évalués : on calcule pour chacun la valeur de la fonction objectif (valeur qu'on appelle sa « force » ou son « adaptation »).

Le principe des algorithmes génétiques consiste à faire évoluer la population de génération en génération pour voir comment évolue son adaptation, en cherchant bien sûr à ce que cette adaptation s'améliore. Le passage d'une génération à une autre peut se faire :

- par croisement : on peut procéder à des échanges de gènes de deux individus « parents » de manière à créer deux individus hybrides « enfants » : ce croisement est bien sûr recherché pour des individus (choisis aléatoirement) parmi les plus adaptés, conformément aux lois de l'évolution. Cependant, avec l'esprit de balayer au maximum l'espace des solutions, des individus « inadaptés » peuvent également faire l'objet de ce processus.

- par mutation : un individu « parent » (choisi au hasard) donne un individu « enfant » - mais au passage, un de ses gènes, tiré au hasard, aura vu sa valeur modifiée aléatoirement. Encore une fois, ce phénomène concerne principalement les individus les mieux adaptés, mais pas exclusivement, toujours de manière à être exhaustif dans l'exploration des solutions ;

- certains individus (de préférence les plus adaptés, mais pas seulement ...) sont conservés à l'identique d'une génération à l'autre ;

- enfin, d'autres individus disparaissent (de préférence les moins adaptés, mais pas seulement ...) : de manière à conserver une population constante d'une génération à l'autre, ils sont remplacés par des individus « régénérés », c'est-à-dire générés aléatoirement, comme pour la population originelle.

Pour chaque génération, on retient soigneusement la valeur de l'adaptation du ou des « ind » meilleurs individus, ainsi que la valeur moyenne de l'adaptation de l'ensemble de la population. Et bien entendu, on conserve de génération en génération la liste réactualisée à chaque itération des meilleurs chromosomes.

Le processus peut prendre fin :

- lorsqu'après un certain nombre de générations successives, la meilleure adaptation n'évolue plus,

- lorsqu'après un certain nombre de générations successives, l'adaptation moyenne de la population n'évolue plus.

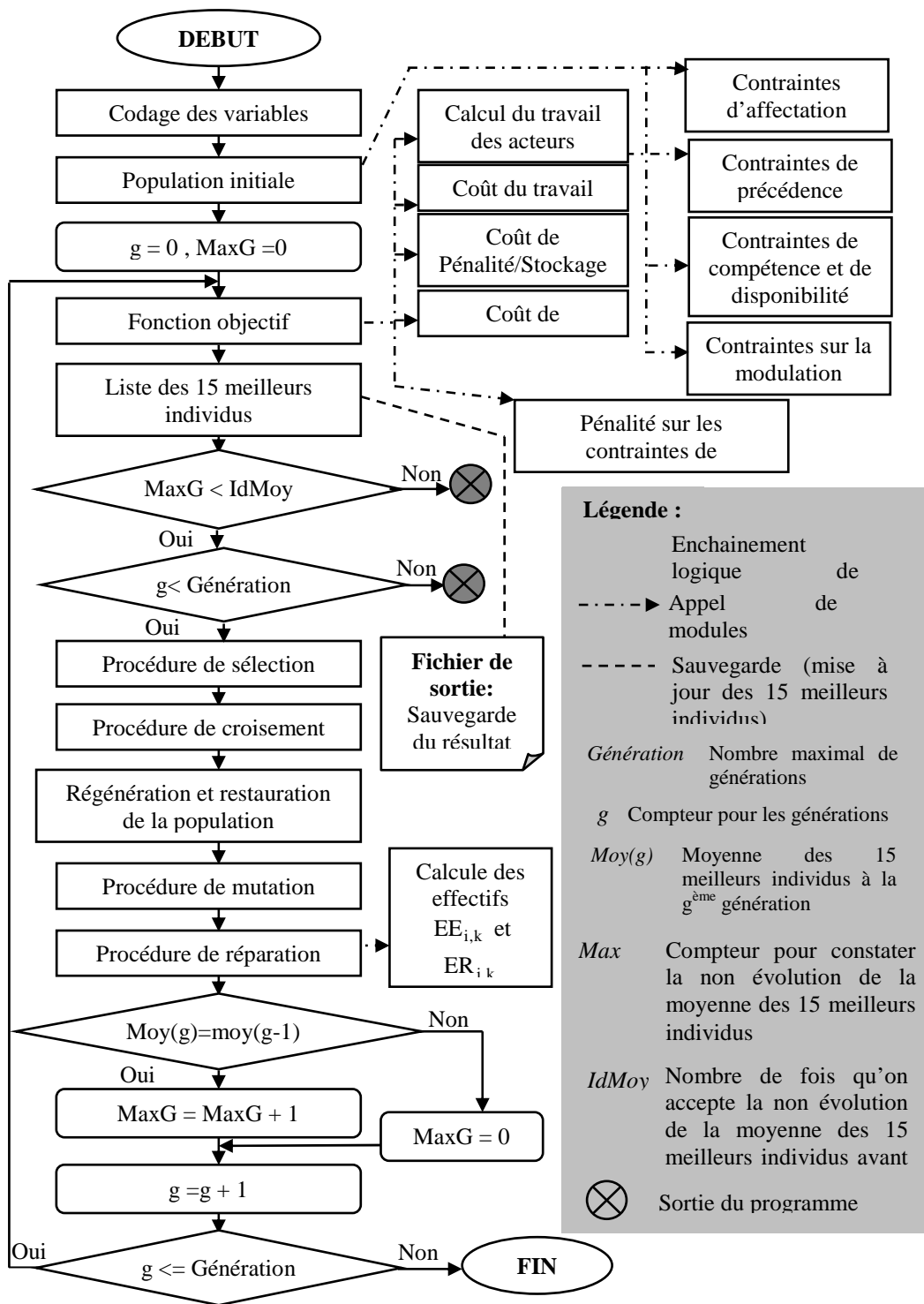


Figure 1 : Approche d'exploration par algorithme génétique monocritère

Dans l'un ou l'autre de ces cas, on peut estimer avoir suffisamment exploré l'espace des solutions et constaté que toute recherche d'une solution meilleure que celles déjà trouvées devient illusoire. Le processus peut également s'arrêter arbitrairement au bout d'un nombre suffisamment grand de solutions : la recherche est alors non concluante.

L'algorithme génétique que nous avons mis en place est représenté sur la *Figure 1*.

II-2. Cas de l'algorithme génétique monocritère

II-2-1. Codage de la solution

Un individu ou chromosome sera « constitué » de deux sous chromosomes (chaque variable de décision fait référence à un sous chromosome) :

- Sous chromosome 1 : variable d'affectation $\sigma_{a,k,i}$ (elle sera 1 si acteur affecté et 0 sinon),
- Sous chromosome 2 : variable de durée des tâches $d_{i,k}$ (variable entière).

II-2-2. Génération de la population initiale

Notre population initiale IP est générée de manière aléatoire tout en respectant certaines contraintes qui ne peuvent absolument pas être violées :

- les contraintes d'affectation et de compétence pour la génération du sous chromosome 1,
- les contraintes de durée pour la génération du sous chromosome 2 :

$$D_i^{\min} \leq d_{i,k} \leq DR_i^{\max} \quad (1)$$

Afin de respecter les contraintes de durée des tâches et les données issues de l'ordonnancement initial, nous définissons DR_i^{\max} qui représente la durée maximale réelle que peut prendre une tâche :

$$DR_i^{\max} = \min(D_i^{\max}, D_i + MT_i) \quad (2)$$

II-3. Étude de faisabilité

II-3-1. Description de l'étude

Avant la mise en œuvre de l'algorithme génétique, notre méthodologie générale prévoit une étude de faisabilité pour étudier la compatibilité entre la charge de travail requise par une activité à réaliser, et la capacité de l'entreprise dans l'horizon de planification considérée. Cette étude consiste à utiliser les limites maximales des contraintes pour montrer que le problème posé ne peut pas trouver une solution satisfaisante qui ne violerait pas certaines des contraintes. Une telle validation permettrait l'arrêt de la résolution du problème avant de démarrer des processus calculatoires longs et fastidieux.

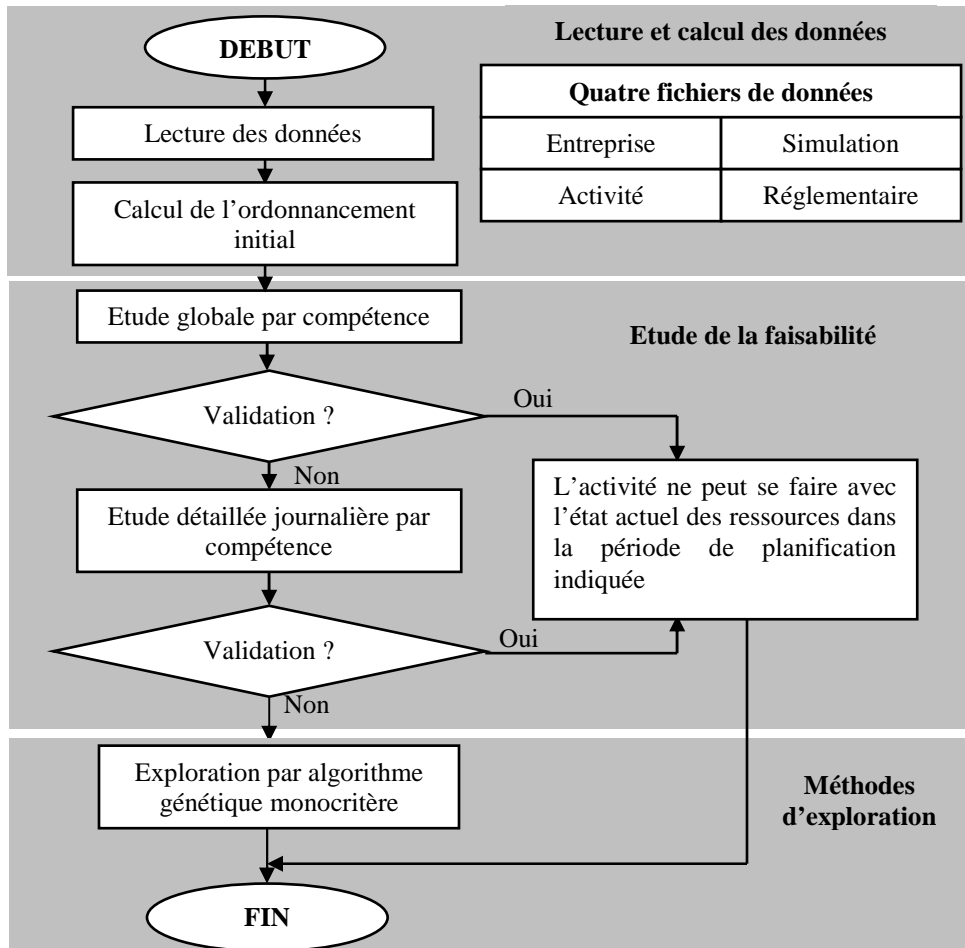


Figure 2 : Ordinogramme général de résolution

Dans la description de l'étude de faisabilité, nous appliquerons l'exemple d'application suivante : nous considérons une entreprise dans laquelle on recense $K=4$ compétences et $A=10$ acteurs, tous polyvalents, disposant chacun d'une compétence principale (avec une efficacité de $\theta_{a,k}=1$), et de compétences supplémentaires pour lesquelles $\theta_{\min} \leq \theta_{a,k} \leq 1$ (**Tableau 1**) ; l'entreprise doit exécuter une activité composée de $I=10$ tâches dont les données sont représentées sur le **Tableau 3**. Le **Tableau 2** fournit les différentes données sur le temps de travail. Dans cet exemple, nous supposons que tous les acteurs sont rémunérés au même taux horaire U (en unité monétaire : *um*) et que les heures supplémentaires sont majorées de $u=25\%$ par rapport aux heures normales. Les liens entre les tâches sont de type Fin-Début et les délais $\alpha_{i,n}$ sur ces liens sont tous nuls.

Tableau 1 : Données sur l'entreprise

N° de l'acteur a	$\theta_{a,k}$ (adimensionnel)			
	k = 1	k = 2	k = 3	k = 4
1	0,8	1	0	0,5
2	1	0	0,8	0
3	0	0,6	0	1
4	0,7	0	1	0,6
5	0	1	0,7	0
6	0,9	0	0	1
7	1	0,8	0	0,6
8	0	0,7	1	0
9	1	0,8	0	0,5
10	0	0,9	1	0

Tableau 2 : Données règlementaires

DMaxMod	39 heures
DMaxJ	10 heures
DMaxS	48 heures
DMax12S	44 heures
C _{s0}	35 heures
HSA	180 heures
DSA	1600 heures
u	0,25
NJS	5 jours
U	11 um

Tableau 3 : Données sur l'activité

N° de la tâche i	D _i	D _i ^{min}	D _i ^{max}	w _{i,k} (heures)				N° Tâches successeurs	Type de lien	$\alpha_{i,n}$
				k = 1	k = 2	k = 3	k = 4			
1	4	2	6	0	60	0	50	2 – 3 – 4	F-D	+0
2	5	3	7	45	68	0	0	3 – 5 – 7	F-D	+0
3	4	3	7	0	63	45	35	5 – 6	F-D	+0
4	7	5	10	53	0	60	0	6 – 9	F-D	+0
5	4	2	6	0	65	0	60	7 – 8	F-D	+0
6	3	1	5	60	0	35	0	8 – 9	F-D	+0
7	5	3	7	35	56	0	40	10	F-D	+0
8	5	3	8	0	0	47	50	10	F-D	+0
9	4	2	5	0	45	26	0	10	F-D	+0
10	3	2	4	35	30	35	30	---	---	---

D'après l'ordonnancement initial établi avec les durées standards D_i (**Figure 3**), la durée fixe contractuelle est : L=25 jours qui sera pour nous la durée contractuelle et nous adopterons la partie flexible $\beta= 5$ jours par rapport à cette durée contractuelle.

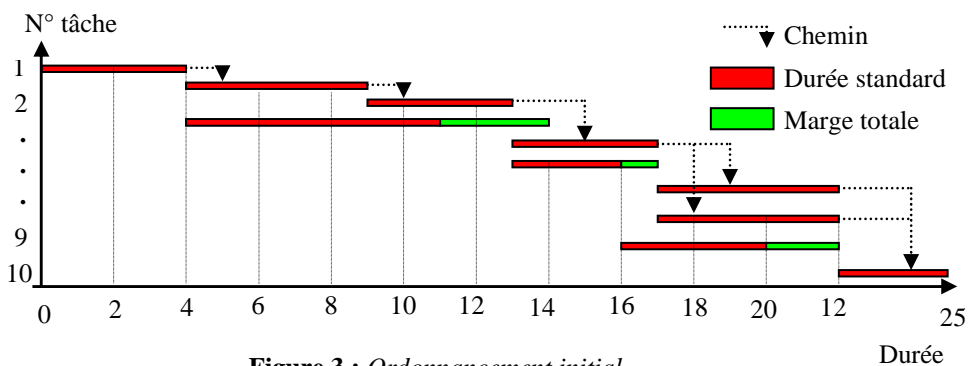


Figure 3 : Ordonnancement initial

Pour ne pas payer de pénalités de retard, ni supporter de coût de stockage, il faut que la durée réelle LV soit comprise entre:
 $L - \beta \leq LV \leq L + \beta \Rightarrow 20 \leq LV \leq 30$ jours. (3)

Cet exemple d'application servira à présenter notre méthodologie de manière à rendre la lecture plus facile. Cet exemple est volontairement simple, beaucoup trop pour prétendre être représentatif d'une application industrielle réelle. Cependant, une application de taille importante est décrite dans la thèse [8] plus conforme à ce qui peut se rencontrer dans « le monde réel ». Pour l'étude de faisabilité, nous considérerons deux aspects :

- cas avec prise en compte de la polyvalence des acteurs,
- cas sans polyvalence des acteurs.

II-3-2. Lecture et calcul des données

Les données nécessaires à la résolution du problème sont regroupées selon leur nature dans quatre fichiers différents :

- le fichier « entreprise » regroupe l'ensemble des données concernant l'entreprise : essentiellement l'inventaire des compétences recensées et des acteurs, notamment leurs efficacités respectives sur les compétences (cf. **Tableau 1**),
- le fichier « règlementaire » regroupe l'ensemble des données issues des lois sur la modulation du temps de travail (cf. **Tableau 2**),
- le fichier « activité » regroupe l'ensemble des données liées à l'activité à réaliser (cf. **Tableau 3**),
- le fichier « simulation » regroupe l'ensemble des paramètres de simulation et de résolution : par exemple les critères d'arrêt. Certains de ces jeux de paramètres dépendent de la méthode de résolution choisie. Par exemple, dans le cas de l'algorithme génétique, il faut définir la taille de la population initiale, les probabilités de croisement et de mutation, etc.

A partir de ces données, on procède rapidement à un ordonnancement initial (**Figure 3**), afin de calculer :

- Les dates de début initiales de chaque tâche dd_i ,
- La durée de l'activité, qui deviendra par défaut sa durée fixe contractuelle L,
- La marge totale de chaque tâche MT_i

Le calcul de l'ordonnancement initial est basé sur les durées standards D_i des tâches (ces durées correspondent à une façon de procéder standard dans l'entreprise).

II-4. Affectation par Algorithmes génétiques monocritères

II-4-1. Calcul de la fonction objectif

II-4-1-1. Intégration des contraintes dans le calcul de la fonction objectif

On peut, avec les algorithmes génétiques, se laisser la latitude de négliger certaines contraintes, en pénalisant la fonction d'évaluation en cas de violation. Elle permet de considérer de manière globale une fonction d'adaptation constituée de la fonction objectif, à laquelle on ajoute la somme des valeurs absolues des violations de contraintes. Lorsque toutes les pénalités sont nulles, cela signifie que toutes les contraintes sont satisfaites.

Dans la résolution du modèle par les algorithmes génétiques, outre les contraintes de modulation du temps de travail, les autres contraintes doivent obligatoirement être respectées pour chaque individu de la génération. Pour prendre en compte la violation des contraintes de modulation (par exemple le travail journalier d'un acteur peut dépasser les 10 heures autorisées), nous avons introduit une fonction pénalité F^5 dans la fonction objectif pour handicaper les individus qui ne respectent pas ces types de contraintes.

Dans le cadre de cet algorithme la fonction objectif devient :

$$F = F1 + F2 + F3 - F4 + F5 \tag{4}$$

Le **Tableau 4** ci-dessous dresse, dans le cas de l'exemple traité, les valeurs de pondérations utilisées :

Tableau 4 : Valeur de pondération en cas de violation des contraintes de modulation

Violation contrainte journalière	6000
Violation contrainte hebdomadaire	3500
Violation contraintes sur une période flottante de douze semaines consécutives	2000

Si nous appelons ind un individu de la population, sa fonction objectif sera notée $F^{(ind)}$.

II-4-1-2. Calcul du travail global journalier de chaque acteur

La détermination du coût salarial de l'activité passe par l'identification du travail de chaque acteur affecté. Pour une charge $W_{i,k}$, nous supposons que les $ER_{i,k}$ acteurs affectés effectuent le même nombre d'heures de travail quelle que soit l'efficacité de chacun d'eux. De ce fait, le travail total d'un acteur a affecté à une charge $W_{i,k}$ sera de :

$$\omega_{a,k,i} = \left(\frac{W_{i,k}}{EE_{i,k}} \right)_{a \in ER_{i,k}} \text{ avec } EE_{i,k} = \left(\sum_{a \in ER_{i,k}} \theta_{a,k} \right)_{\theta_{a,k} \geq \theta_{min}} \tag{5}$$

Ce qui donne un travail journalier sur une durée $d_{i,k}$ pour la charge $w_{i,k}$:

$$\omega_{a,k,i,j} = \frac{\omega_{a,k,i}}{d_{i,k}} \quad (6)$$

En fonction des dates de début dd_i des tâches, le travail global journalier $\omega_{a,j}$ de chaque acteur sera calculé pour servir de base aux calculs des coûts des heures supplémentaires, de la flexibilité résiduelle.

$$\omega_{a,j} = \sum_{k=1}^K \sum_{i=1}^I \omega_{a,k,i,j} \quad \forall a \quad \forall j \quad (7)$$

Aussi, l'application des contraintes de précédence permet d'identifier la durée réelle LV de l'activité afin de calculer d'éventuels coûts de stockage (immobilisation) ou de pénalité de retard.

II-4-1-3. Evaluation des individus

La phase d'évaluation consiste à calculer la « force » d'adaptation de chaque individu de la population (c'est-à-dire son adaptation aux contraintes de l'environnement dans l'esprit de la comparaison avec un processus évolutif naturel). Les algorithmes génétiques sont mis en œuvre pour maximiser une fonction objectif [10] ; Or la démarche de résolution de notre problème consiste à minimiser une fonction objectif. Il est donc nécessaire de transformer le problème de telle sorte que les minima de la fonction correspondent aux individus les plus forts. Ainsi, les travaux de [10] permettent d'associer à la fonction objectif $F^{(ind)}$ de chaque individu ind une constante aussi grande possible C_{MAX} pour donner une nouvelle fonction $F_{ab}^{(ind)}$. C'est cette procédure qu'on utilise pour notre modèle :

$$F_{ab}^{(ind)} = C_{MAX} - F^{(ind)} \quad (8)$$

On appellera cette fonction la « force absolue » des individus. Cette méthode permet de s'affranchir des problèmes de signe de la fonction.

II-4-2. Définition des paramètres et des procédures

II-4-2-1. Les critères d'arrêt de l'algorithme

Comme dans tout algorithme itératif, la mise en œuvre d'un algorithme génétique nécessite la définition d'un critère permettant d'arrêter l'exploration de l'ensemble - solution. Pour ce modèle, nous en avons défini deux :

- Le premier critère concerne la constatation de la non évolution de la moyenne de la fonction objectif. En effet, la moyenne sur la fonction objectif ne se fait pas sur tous les individus de la population. Nous sauvegardons dans

un fichier les quinze meilleurs individus de l'ensemble des générations déjà explorée. Si nous appelons g le numéro d'une génération, cette condition d'arrêt concerne la moyenne sur les 15 meilleurs individus à chaque génération, notée $Moy(g)$, où les 15 meilleurs individus sont mis régulièrement à jour. Si nous constatons une non - évolution de cette moyenne pendant un certain nombre de générations $IdMoy$ fixé à l'avance, alors l'exploration s'arrête.

- Le deuxième critère concerne un nombre maximal de générations noté « Génération » aussi fixé à l'avance. Ce deuxième critère est mis en œuvre dans le cas où le premier n'est pas atteint. Cela permet juste d'arrêter une recherche qui ne semble pas aboutir.

Dans le cas de l'exemple d'application que nous traitons, $IdMoy = 100$ et Génération = 8000.

II-4-2-2. Sélection des individus

L'objectif de la sélection est de permettre aux individus sélectionnés de reconstituer la population pour la génération suivante par les processus de croisement, de mutation et/ou régénération [5]. La procédure de sélection que nous avons utilisée est celle opérant avec la « roulette Goldberg » [10]. Elle permet de sélectionner de manière aléatoire les individus ayant la fonction d'adaptation la plus élevée, tout en n'interdisant pas la sélection d'individus moins bien adaptés. Chaque individu occupe un secteur de la roue proportionnel à sa force relative $F_{relative}(ind)$, correspondant à sa force absolue $F_{ab}(ind)$ rapportée à la somme totale des forces absolues de tous les individus de la génération [10] :

$$F_{relative}(ind) = \frac{F_{ab}(ind)}{\sum_{ind=1}^{IP} F_{ab}(ind)} \quad \text{avec} \quad \sum_{ind=1}^{IP} F_{relative}(ind) = 1 \quad (9)$$

La sélection se fait avec une probabilité de survie P_{surv} qui est complémentaire à la probabilité de croisement P_{crois} . Cependant, pour permettre un « rafraîchissement » de la nouvelle génération, nous introduisons une probabilité de régénération P_{reg} ; qui permet de régénérer de nouveaux individus comme dans le cas de la population initiale mais avec un taux plus faible que celui du croisement (choix arbitraire).

$$P_{surv} = 1 - (P_{crois} + P_{reg}) \quad (10)$$

La reconstitution des individus d'une génération g à une génération $g+1$ se fait selon la **Figure 4** suivante :

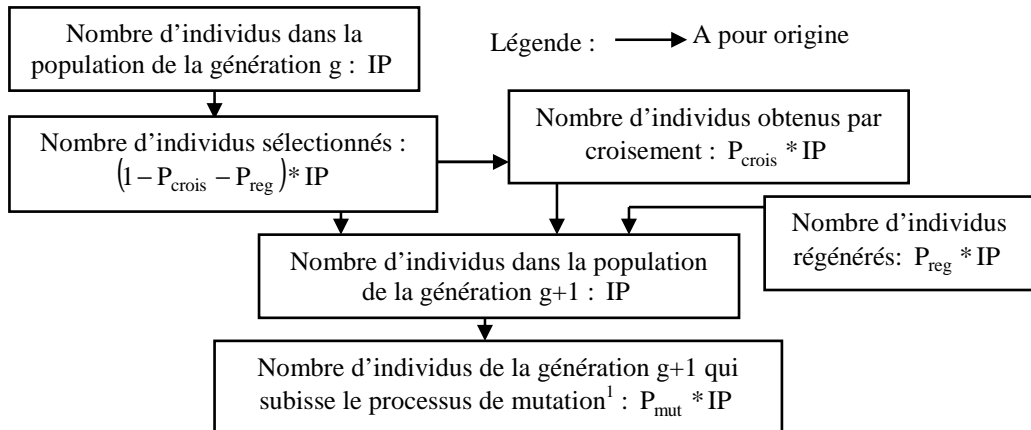


Figure 4 : Reconstitution des individus de la génération g à la génération $g+1$

Le nombre d'individus sélectionné sera de : $IP * P_{surv}$. Et c'est sur cette population sélectionnée qu'on appliquera le processus de croisement afin d'avoir une partie de la population suivante.

Cette application a été testée sur un ordinateur de type « Biprocesseur intel Xeon 2,4 GHz avec 2,5 Go de Ram ». Le codage a été réalisé sur le logiciel de programmation « Visual C++ ».

Pour cette application, les données de simulation suivantes ont été utilisées :

- les probabilités de croisement, de régénération et de mutation ont été fixées respectivement à $P_{crois} = 0,65$; $P_{reg} = 0,10$; $P_{mut} = 0,20$,
- l'étude porte sur une population de : $IP = 200$ individus,
- l'algorithme est stoppé pour non-évolution de la moyenne au bout de $IdMoy = 100$ générations. Cette moyenne concerne les 15 meilleurs individus de l'ensemble des générations déjà explorées (condition d'arrêt),
- le nombre maximum de génération est fixé à : Génération = 8000 (condition d'arrêt pour non convergence).

III - RESULTATS ET DISCUSSION

III-1. Étude de faisabilité

L'étude de la faisabilité permet en principe d'étudier la possibilité pour l'entreprise, compte tenu des effectifs présents, de réaliser l'activité dans la durée impartie L ; plus exactement pour cette étude, cela permet de mettre en évidence une impossibilité de réalisation, sans préjuger de la faisabilité réelle. L'objectif est d'éliminer toute idée de non faisabilité de l'activité en utilisant les limites maximales des contraintes. Cette étude se fait à deux niveaux distincts : dans un premier temps, nous procédons à une étude globale de la

disponibilité et de la charge sur chacune des compétences et pour l'ensemble de la durée L ; si aucune conclusion ne peut être émise quand à la non - faisabilité, nous nous livrons, dans un deuxième temps, à une étude détaillée.

III-1-1. Étude globale par compétence

Dans le calcul global, nous considérons que tous les acteurs sont disponibles. Nous prenons chaque compétence indépendamment l'une de l'autre. Pour chacune d'elles, les acteurs sont pris en compte en fonction de leur efficacité.

III-1-1-1. Calcul de la disponibilité globale maximale équivalente des acteurs

Nous entendons par disponibilité globale, l'équivalent du travail global réalisable par les acteurs dans la limite de la loi sur la modulation du temps de travail en tenant compte de l'efficacité. Cette disponibilité globale par compétence peut varier tout en gardant l'effectif des acteurs constant. Si nous appelons $Q_{k,L}$ cette disponibilité maximale des acteurs ayant une efficacité suffisante dans la compétence k pendant la période L , on peut définir $Q_{k,L}$ comme le produit de la durée maximale du travail hebdomadaire DM_{MaxS} , de l'effectif équivalent de la compétence considérée EE_k et du nombre de semaines qu'on a dans la période L , cela nous donne :

$$Q_{k,L} = DM_{MaxS} * EE_k * Ent \left[\left(\frac{L-1}{NJS} \right) + 1 \right], \quad \forall k \tag{11}$$

$$EE_k = \left(\sum_{a=1}^{A_k} \theta_{a,k} \right)_{\theta_{a,k} \geq \theta_{min}}, \quad \forall k$$

Dans cette équation, $\left(\sum_{a=1}^{A_k} \theta_{a,k} \right)_{\theta_{a,k} \geq \theta_{min}}$, représente l'effectif équivalent des acteurs affectables dans la compétence k , et NJS le nombre de jours considérés dans une semaine.

III-1-1-2. Charge globale par compétence

Pour chaque tâche, sont connues les charges requises $w_{i,k}$ sur toutes les compétences. De ce fait, nous calculons la charge globale par compétence w_k avec la relation suivante :

$$w_k = \left(\sum_{i=1}^{I_k} w_{i,k} \right), \quad \forall k \tag{12}$$

où I_k correspond au nombre des tâches mobilisant la compétence k .

III-1-1-3. Compatibilité globale : Validation

Globalement, pour que l'activité ne soit pas faisable, il faut que pour une compétence donnée on ait :

$$w_k \geq Q_{k,L} \quad \forall k \quad (13)$$

Une fois que l'équation 3 est vérifiée pour au moins une compétence k , la résolution ne peut plus continuer dans la mesure où, comment qu'on s'y prenne, et en utilisant au maximum les temps de travail tolérés, il ne pourra y avoir suffisamment de ressources pour fournir globalement la charge nécessaire. Dans le cas contraire, aucune conclusion ne peut être émise : nous passons alors à une étude plus détaillée.

III-1-1-4. Cas de l'exemple d'application

Nous présentons sous forme graphique (**Figure 5**) la disponibilité maximale pour chaque compétence $Q_{k,L}$ dans le cas avec ou sans polyvalence, et la charge globale w_k de l'entreprise (ces grandeurs sont exprimées en heures).

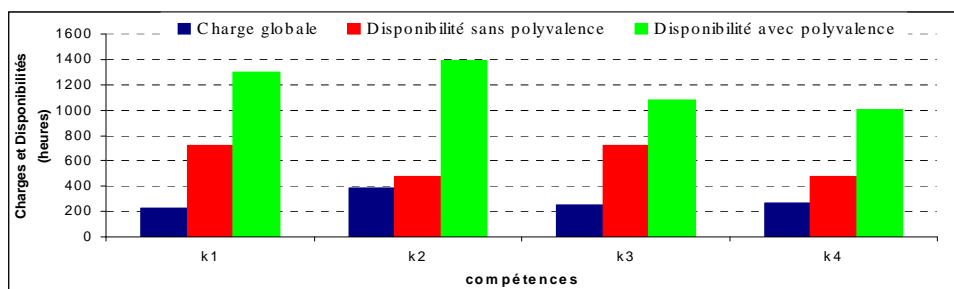


Figure 5 : Etude globale par compétence

Pour cet exemple, aucune conclusion ne peut être émise quant à la non faisabilité dans la mesure où avec ou sans polyvalence, la disponibilité globale équivalente pour chacune des compétences dépasse la charge globale respective. Nous ne pouvons à ce stade pas encore conclure à la non-faisabilité, mais rien ne justifie d'arrêter les calculs à ce stade (dans le cas sans polyvalence, on notera toutefois que la compétence $k2$, pour laquelle nous avons presque égalité entre la charge et la disponibilité, suscite quelques inquiétudes). En conclusion, une étude plus détaillée s'avère nécessaire.

III-1-2. Étude journalière par compétence

Pour cette poursuite de l'examen, nous utilisons l'ordonnancement initial des tâches qui fournit pour chacune, les dates de début initiales dd_i et la marge

totale MT_i . En vue de détailler la compatibilité entre la charge et la disponibilité, nous relâchons certaines contraintes du problème (pour les besoins de ce test) :

- On affecte à chaque tâche, la plus grande durée possible, qui sera la plus grande des deux valeurs suivantes : la durée maximale que peut prendre l'activité selon ses données intrinsèques (cf. **Tableau 3**), et la durée standard augmentée de la marge déterminée précédemment par l'ordonnancement initial (cf. **Figure 3**), Cette durée sera égale au $\max(D_i^{\max}, D_i + MT_i)$. Ceci a pour but de minimiser les charges journalières des tâches : comme pour l'étude précédente (globale), il s'agit moins de démontrer la faisabilité que de mettre en évidence une impossibilité de réaliser l'activité envisagée. Ainsi, l'incapacité de l'entreprise à fournir des charges minimisées autant que faire se peut mènera à la conclusion que toute recherche ultérieure de solution est illusoire.

- En affectant aux tâches leur durée maximale, nous relâchons les contraintes de précédence pour toujours rester dans l'horizon de planification considéré (nous considérons que chaque tâche est indépendante donc les dates de début initiales restent inchangées). Ainsi le rallongement de la durée des tâches reporte une partie de sa charge sur les dates auxquelles se déroulent ses successeurs. Cela permettra dans le cas d'une non compatibilité journalière d'une charge et de la disponibilité de tirer les conclusions d'une impossibilité de réalisation de l'activité dans l'horizon de planification L ; dans la réalité des faits, les contraintes de précédence sont obligatoirement respectées.

III-1-2-1. Expression de la charge journalière maximale par compétence

A partir des dates de début de l'ordonnancement initial on détermine $P^{k,j}$, l'ensemble des tâches qui sont exécutables le jour j et mobilisant la compétence k . Nous faisons l'hypothèse que la charge $w_{i,k}$ de chaque tâche est uniformément répartie sur sa durée (durée qu'on vient de fixer à la valeur maximale qu'elle peut prendre).

La charge journalière globale pour une compétence est donnée par la relation suivante :

$$w_{k,j} = \sum_{i \in P_{j,k}} \frac{w_{i,k}}{\max(D_i^{\max}, D_i + MT_i)}, \quad \forall k \forall j \quad (14)$$

Après relâchement des contraintes de précédence et l'attribution des durées maximales aux tâches, nous identifions pour un jour j et pour une compétence k les charges $w_{k,j}$ des tâches qui sont susceptibles d'être réalisées.

III-1-2-2. Expression de la disponibilité journalière maximale par compétence

Lors du calcul global (cf. équation 11), nous avons déterminé $Q_{k,L}$. De la même manière, pour chaque compétence, si nous appelons $Q_{k,j}$ sa disponibilité journalière maximale, on a :

$$Q_{k,j} = \frac{Q_{k,L}}{L}, \quad \forall j \quad (15)$$

Dans cette relation, nous supposons que $Q_{k,j}$ est identique pour chaque jour j . En procédant de cette manière, pour déclarer l'activité non faisable, il suffit que pour un jour j donné, il existe une compétence k pour laquelle la disponibilité journalière maximale est inférieure à la charge journalière maximale, soit :

$$\exists k / Q_{k,j} \leq w_{k,j}, \quad \forall j \quad (16)$$

Comme pour l'analyse globale, si ce test est vérifié, on conclura à la non-faisabilité de l'activité. Dans le cas contraire, aucune conclusion définitive ne peut être émise à ce stade : nous devons alors continuer la résolution du problème par l'exploration par algorithmes génétiques monocritères.

III-1-2-3. Cas de l'exemple d'application

Nous montrons le résultat de ce calcul sous forme graphique : la **Figure 6** donne la disponibilité maximale journalière et la **Figure 7** livre les charges maximales. Pour une question de clarté, nous avons séparé la représentation des deux grandeurs.

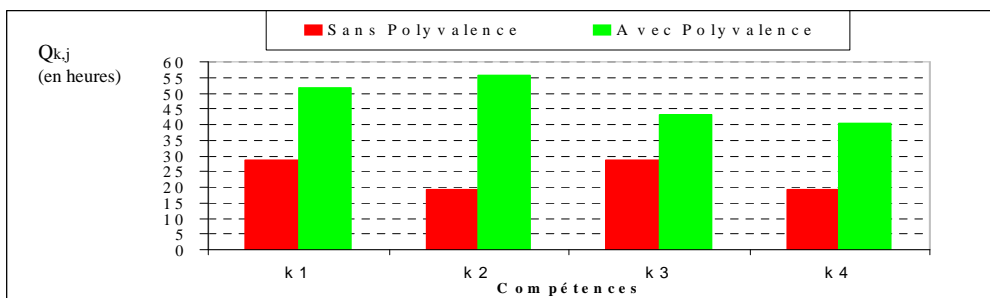


Figure 6 : Disponibilité maximale journalière équivalente par compétence

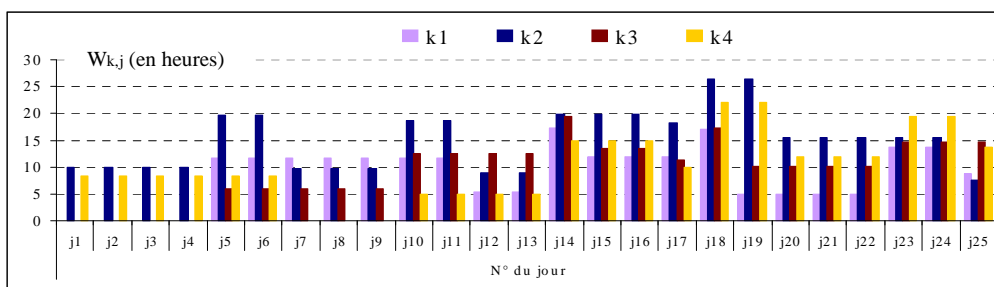


Figure 7 : Charge maximale journalière par compétence

Analyse du cas sans polyvalence

Pour cet exemple, aucune conclusion ne peut être émise pour les compétences k_1 et k_3 car leurs disponibilités respectives dépassent largement la charge. Par contre, les charges nécessaires sur les compétences k_2 et k_4 ne peuvent être assurées avec la disponibilité actuelle de l'entreprise pour les jours 18 et 19 (**Figure 7**) : Donc, sans avoir recours à la polyvalence, cet exemple d'application ne peut être exécuté par l'entreprise.

Analyse du cas avec polyvalence

Pour cette option, les charges journalières respectives sur les compétences sont largement en dessous des disponibilités journalières. Aucune conclusion ne peut être émise quand à la non faisabilité. Une étude d'exploration s'avère nécessaire pour trouver d'éventuelles solutions.

III-2. SIMULATIONS

Pour apprécier l'application de l'algorithme génétique au problème que nous traitons, nous avons procédé à une dizaine de simulations. Le **Tableau 5** suivant donne les résultats sur les temps de calcul, nombres de générations, durées réelles totales de l'activité, ainsi que les valeurs de la fonction objectif du meilleur individu par génération en fonction du critère d'évaluation de l'exemple d'application sur six simulations.

Tableau 5 : Récapitulatif des résultats de simulations de l'algorithme génétique

N° Simulation	Temps de calcul (en secondes)	nombre de générations	Valeur de la fonction objectif	Fonction F5 : violation des contraintes	Durée LV de l'activité (j)
1	76	613	7 833,151	0	29
2	144	1 055	9 325,489	0	29
3	102	573	9 897,428	0	31
4	155	709	6 834,938	0	30
5	84	686	7 674,276	0	30
6	114	849	7 646,877	0	30
7	150	1 196	7 275,291	0	30
8	59	467	7 715,61	0	29
9	74	377	7 748,657	0	29
10	122	1 023	7 220,69	0	30

Dans cet ensemble, nous exposons la meilleure solution : celui de la *simulation n°4* qui présente la meilleure valeur de la fonction objectif. Ainsi, les durées et les dates de début sont présentées sur le **Tableau 6** suivant :

Tableau 6 : Durées et date de début des tâches

N° des tâches i	d _i (en jours)	d _{i,k} (en jours)				dd _i
		k = 1	k = 2	k = 3	k = 4	
1	4	0	4	0	4	0
2	4	3	4	0	0	4
3	7	0	4	5	7	8
4	10	10	0	5	0	4
5	5	0	4	0	5	15
6	4	4	0	3	0	15
7	5	5	3	0	5	20
8	6	0	0	3	6	20
9	5	0	3	5	0	19
10	4	3	2	4	4	26

Pour la vérification de la contrainte de durée : $D_3^{\min} \leq d_3 \leq D_3^{\max}$ nous prenons l'exemple de la tâche 3 dont la durée globale de réalisation est de : $d_3 = \max(d_{3,k})_{k=1 \text{ à } 4} = 7$ jours. A partir des données de l'activité nous savons que sa durée standard est $D_3 = 4$ jours, sa durée maximale $D_3^{\max} = 7$ jours et sa durée minimale de $D_3^{\min} = 3$ jours. Aussi, à l'aide de l'ordonnancement initial on obtient sa marge totale $MT_3 = 0$ jour donc la durée maximale que doit prendre cette tâches est de : $DR_3^{\max} = \max(D_3^{\max}, D_3 + MT_3) = 7$ jours. Compte tenu de ces informations, nous voyons bien que la tâche 3 vérifie bien la contrainte de durée ($3 \leq d_3 \leq 7$). Cependant, les durées sur les compétences sont différentes. Ainsi, les acteurs affectés sur la charge $W_{3,2}$ peuvent être libérer au bout de 4 jours de travail alors que d'autres charges de la même tâche continuent d'être exécutées ($W_{3,3}$ dure 5 jours et $W_{3,4}$ dure 7 jours). L'analyse est la même pour les autres tâches. Cependant, elles respectent toutes les contraintes de durées définies.

En ce qui concerne les contraintes de précédence, nous donnons un exemple : le **Tableau 6** donne les dates de début des tâches. Dans les données activités, la tâche 3 a pour successeurs les tâches 5 et 8 et pour antécédent la tâche 2. Le résultat fournit par le **Tableau 6** indique que la date de début de la tâche 3 est de $dd_3 = 8$ avec une durée de $d_3 = 7$ jours, ce qui donne une fin de $dd_3 + d_3 = 8 + 7 = 15$. Or les dates de début de ses successeurs sont : pour la tâche 5 on a $dd_5 = 15 \geq dd_3 + d_3$ et pour la tâche 8 on a $(dd_8 = 20) \geq (dd_3 + d_3)$. Pour sa tâche antécédente $dd_3 \geq (dd_2 + d_2 = 4 + 4 = 8)$. Nous remarquons que les contraintes de précédente sont tout à fait respectées.

A partir des données du **Tableau 6** on déduit la durée réelle d'exécution de l'activité $LV = \max(dd_i + d_i)_{i=1 \text{ à } 10} = 30$ jours

Les effectifs affectés compétence par compétence sont représentés sur le **Tableau 7** ci-dessous :

Tableau 7 : Effectif réel et équivalent affecté par tâche et par compétence

N° des tâches	ER _{i,k} (effectif réel)				EE _{i,k} (effectif équivalent)			
	k = 1	k = 2	k = 3	k = 4	k = 1	k = 2	k = 3	k = 4
1	0	4	0	4	0	3,7	0	3,1
2	5	5	0	0	4,7	4,5	0	0
3	0	5	2	3	0	4	1,8	2,5
4	4	0	4	0	3,4	0	3,5	0
5	0	7	0	2	0	5,8	0	2
6	4	0	3	0	3,7	0	2,8	0
7	2	5	0	2	1,9	4,5	0	1,6
8	0	0	3	5	0	0	2,8	3,6
9	0	6	3	0	0	4,8	2,7	0
10	3	5	3	4	2,6	4,4	2,8	2,6

Dans ce tableau, on remarque que pour la tâche 3, l'effectif réel affecté sur la charge $w_{3,2}$ est de : $ER_{3,2} = 5$ acteurs ce qui représente un effectif équivalent de : $EE_{3,2} = 4$. Ce qui signifie qu'il ya eu une affectation des acteurs dont la valeur de l'efficacité de certains est inférieure à la valeur nominale.

Ce qui donne une productivité globale de réalisation de :

$$\tau_{opt} = \frac{\sum_{i=1}^{10} \sum_{k=1}^4 EE_{i,k}}{\sum_{i=1}^{10} \sum_{k=1}^4 ER_{i,k}} = \frac{79,8}{93} = 0,8580 \tag{17}$$

$$\omega = \sum_{a=1}^{10} \sum_{j=1}^{30} \omega_{a,j} = 1307,43 \text{ heures}$$

Pour un travail global de : , alors que la charge (qui correspond à un minimum incompressible) est de

$$w = \sum_{i=1}^{10} \sum_{k=1}^4 w_{i,k} = 1128 \text{ heures} \tag{18}$$

La fonction objectif et ses composantes sont présentées dans le **Tableau 8** :

Tableau 8 : Fonction objectif (AG)

F1 (coût effectif sans heures supplémentaires)	14 316,53
F2 (coût des heures supplémentaires)	66,68
F3 (coût de stockage ou de pénalité de retard)	0
F4 (coût de flexibilité résiduelle)	7 548,28
F5 (coût de violation de contraintes de modulation)	0
F = F1 + F2 + F3 – F4 + F5	6 834 , 938

La comparaison des valeurs que prennent les différentes composantes du coût illustre le poids des pénalités en cas de violation de contraintes (F5) : si ces violations sont autorisées dans l’exploration des solutions, elles n’en sont pas moins fortement pénalisées.

La **Figure 8** présente la flexibilité résiduelle de chaque acteur après affectation. Elle relève le taux de préservation de la disponibilité de chaque acteur à la fin de la réalisation de l’activité :

$$\tau_{FlexR_a} = 1 - \frac{\sum_{s=1}^{Ent[(LV-1)/NJS]+1} O_{a,s}}{Ent[(LV-1)/NJS]+1} \quad (19)$$

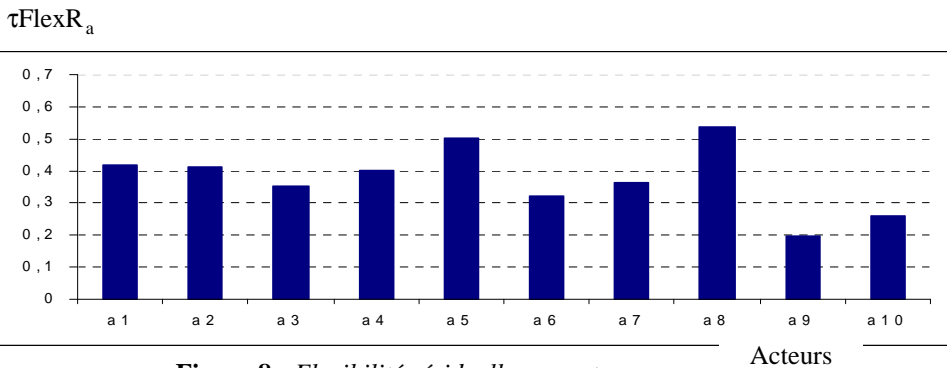


Figure 8 : Flexibilité résiduelle par acteur

Le travail journalier par acteur est donné par les **Figures 9, 10** et **11** ci-dessous :

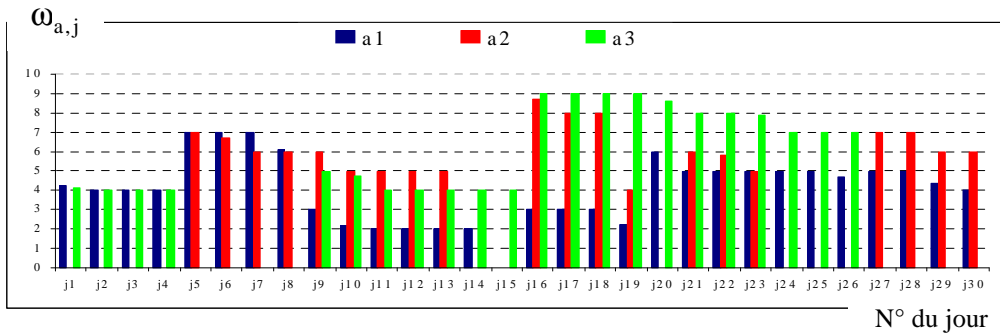


Figure 9 : Travail journalier par acteur (a1, a2, a3)

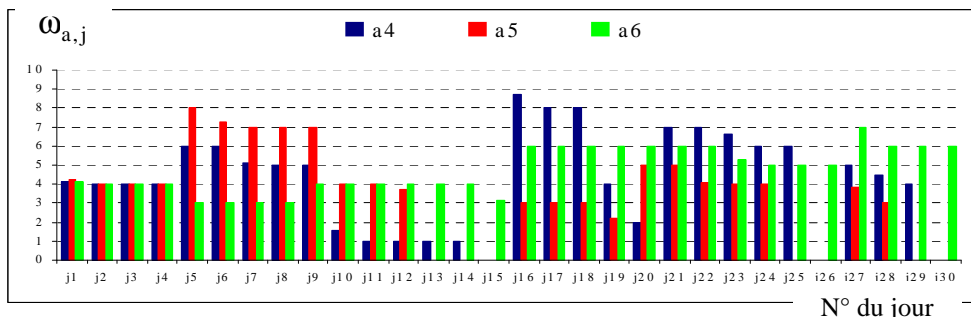


Figure 10 : Travail journalier par acteur (a5, a6, a7)

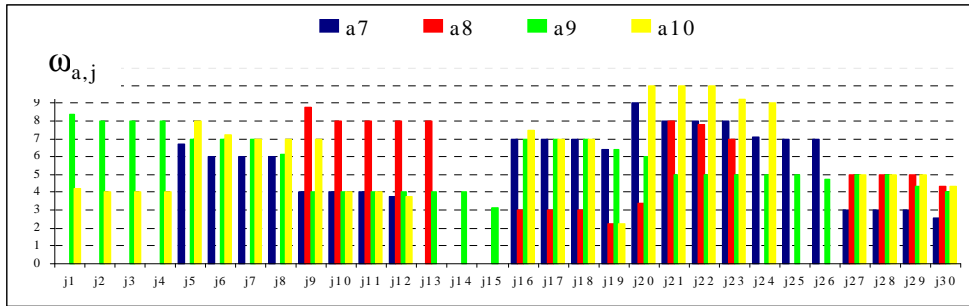


Figure 11 : Travail journalier par acteur (a7, a8, a9, a10) N° du jour

L'analyse de ces travaux journaliers nous indique qu'aucune contrainte journalière n'est violée quelque soit l'acteur et quelque soit le jour : $\omega_{a,j} \leq D_{MaxJ} = 10$ heures, $\forall a \forall j$. Aussi, la sommation des travaux des acteurs que nous avons faits ne montre aucune violation des contraintes hebdomadaires.

IV - CONCLUSION

L'utilisation des algorithmes génétiques pour la résolution du problème d'affectation des ressources avec prise en compte de la polyvalence permet l'obtention d'une solution acceptable. Tout d'abord la solution ne viole aucune contrainte de modulation du temps de travail (cf. **Figures 9, 10 et 11**), ensuite elle fournit un ordonnancement final des tâches qui respecte la durée calendaire définie (**Tableau 6**, pas de coût d'immobilisation ni de pénalité de retard). Il est cependant probable que d'autres jeux de données pour le même problème ne permettront pas d'obtenir la même solution. De plus, le temps de calcul pour cet exemple permet de réaliser plusieurs simulations, afin de se placer dans les meilleures conditions d'obtention d'une bonne solution, en utilisant ainsi les propriétés de convergence stochastique des algorithmes génétiques. Cette étude permet donc de valider l'utilisation d'algorithmes génétiques pour l'obtention d'une affectation des ressources humaines avec prise en compte de la polyvalence.

RÉFÉRENCES

- [1]- O. BELLENGUEZ, "Méthodes de résolution pour un problème de gestion de projet multi-compétence", Thèse de doctorat, université de Tours, Novembre (2006).
- [2]- M. YOSHIMURA, Y. FUJIMI, K. IZUI, S. NISHIWAKI, "Decision-making support system for human resource allocation in product development projects", International Journal of Production Research, Vol. 44, P. 831 - 848, March (2005).

- [3]- H. KANE, “*Étude de l’ajustement de la capacité à la charge pour une gestion quantitative des ressources humaines en production*”, Thèse de doctorat de l’Institut National des Sciences Appliquées de Lyon, (2001).
- [4] - M. BENNOUR, “*Contribution à la modélisation et à l’affectation des ressources humaines dans les processus*”, Thèse de doctorat de l’Université de Montpellier II, France, Décembre (2004).
- [5] - A. R. DIETZ, “*Optimisation multicritère pour la conception d’ateliers discontinus multiproduits : aspects économique et environnemental*”, Thèse de doctorat de l’Institut National Polytechnique de Toulouse, (2004).
- [6]- E. Vidal, “*Stratégie de planification pour la flexibilité opérationnelle d’un atelier à cheminement unique*”, Thèse de doctorat, Institut National Polytechnique de Toulouse, (2000).
- [7] - K. H. EDI, P. DUQUENNE, J. M. LELANN, “*Influence de la polyvalence dans l’affectation flexible des acteurs sur la planification des activités industrielles : résolution avec un algorithme génétique*”, 7^{ème} congrès international de génie industriel, Trois-Rivières, Canada, 5–8 Juin (2007).
- [8] - K. H. EDI, “*Affectation flexible des ressources dans la planification des activités industrielles : prise en compte de la modulation d’horaires et de la polyvalence*”, Thèse de doctorat de l’Université Paul Sabatier de Toulouse, France (2007).
- [9] - D. E. GOLDBERG, “*Real-coded Genetic Algorithms, virtual alphabets and blocking*”, Complex Systems, pp. 139-167 (1991).
- [10]- D. E. GOLDBERG, “*Genetic Algorithms in Search, Optimization and Machine Learning Reading*”, Addison Wesley (1989).