

Intégration du temps de formation et des desiderata des opérateurs dans un ordonnancement.

VALERIO BOSCHIAN-CAMPANER¹, ROMUALD STOCK²

¹LGIPM, équipe SdP, Ile du Saulcy,
57045 Metz Cedex 01 – France
boschian@univ-metz.fr

²ENIM TECHNOPOLE
Route d'Ars Laquenexy, 57070 Metz - France
rom.stock@gmail.com

Résumé - Dans cet article, nous abordons conjointement un problème d'ordonnancement d'activités de maintenance couplé à un problème de formation et de disponibilité négociable du personnel. L'utilisation de matrices de compétences dynamiques nous permet d'établir une politique de formation du personnel en adéquation avec le respect des objectifs de production de l'entreprise. Cette méthodologie prend en compte le choix ordonné des opérateurs concernant leurs indisponibilités négociables. Après une approche analytique du problème et au vu de la complexité du problème, nous orientons nos recherches vers une méthode méta heuristique qui nous donne rapidement une solution admissible.

Abstract - In this paper a problem of scheduling of maintenance activities coupled with a problem of training and negotiable availability of staff are tackled jointly. The use of dynamic competency matrixes enables us to set up a training policy for the staff in line with the company production targets. This methodology takes into account the ordered choices of operators relating to their negotiable periods of unavailability. After an analytical approach and given the complexity of the problem, our research focuses on a metaheuristic method which rapidly produces an acceptable solution.

Mots clés – Matrice de compétences, maintenance, ordonnancement, disponibilité, formation par apprentissage.

Keywords - Competency matrix, maintenance, scheduling, availability, apprenticeship.

1 INTRODUCTION

Actuellement, nous sommes dans une période de stagnation économique. Beaucoup d'entreprises sont en position de sureffectifs mais doivent dans la mesure du possible éviter les licenciements. Ces mêmes entreprises profitent de cette période de stagnation afin de mieux former leur personnel en attendant la reprise économique. Pour ce faire, les matrices de compétences, statiques ou dynamiques, largement utilisées dans le milieu automobile, peuvent nous indiquer quelles sont les personnes à former et dans quelle spécialité en fonction des objectifs de l'entreprise

Cet article n'a aucunement l'ambition de proposer ce nouveau modèle de fonctionnement de l'entreprise : c'est une tâche complexe, nécessitant des travaux pluridisciplinaires. Nous proposons, plus simplement, de montrer sur un exemple d'ordonnancement des tâches, qu'il est possible d'être attentif à certains desiderata des acteurs tout en préservant les objectifs de performance de l'entreprise. Ainsi, notre contribution a trait à une méthodologie d'ordonnancement qui prend en compte :

- les contraintes de production de l'entreprise (le temps de traitement global des pièces ainsi que la formation de certains opérateurs dans des spécialités prioritaires pour l'entreprise)
- les desiderata des opérateurs comme leurs tâches souhaitées, le choix de leurs périodes de disponibilité ou leurs objectifs personnels de formation.

Les problèmes d'ordonnancement abordés dans la littérature sont nombreux. Nous nous intéressons plus particulièrement aux ordonnancements de projet sous contrainte de ressources, connue sous le nom de RCPSP (Resource-Constrained Project Scheduling Problem), qui ont suscités de nombreux travaux depuis plusieurs décennies. Le principal objectif est la diminution du makespan avec une prise en compte des ressources disponibles et des contraintes de précédence entre les activités [Blazewicz et al., 1983] [Brucker et al., 1999]. D'autres objectifs contribuent à enrichir ce modèle de base : la prise en compte des incertitudes relatives à la production : ressources indisponibles, mauvaise estimation du temps de traitement, tâches interrompues, etc. [Herroelen, 2005] [Aytug, 2005].

Des travaux ont étendu le problème classique du RCPSP dans le cas où nous avons un nombre fini de façon d'exécuter l'activité (la durée de l'activité est fonction du mode d'exécution, les ressources peuvent être renouvelables ou non) [De Reyck et al., 1999] [Van Peteghem et al., 2010]. Cette extension est connue sous le nom de Multi-mode Resource-Constrained Project Scheduling Problem (MRCPSPP).

Dans le « classique » RCPSP, les ressources n'ont qu'une seule compétence, ce qui est contraire à la réalité. L'approche multi-compétence MSPSP (Multi-Skill Project Scheduling

Problem) nous permet de tenir compte de toutes les compétences d'un opérateur, en le faisant participer à plusieurs activités d'un même projet [Morineau, 1996] [Drezet et al., 2008]. Cette extension donne davantage de flexibilité au gestionnaire de production [Burlison et al., 1998]. L'ordonnancement des travaux de maintenance sous contrainte de compétences se retrouve dans de nombreux travaux [Marmier, 2007] [Adzkapka, 2003].

D'autres études justifient la nécessité d'une coopération entre l'ordonnancement de la maintenance et celui de la production [Coudert, 2000] [Kaabi-Harrath, 2008] dans le but d'accroître leurs efficacités respectives.

Vu la complexité du problème, quelques rares travaux donnent des solutions exactes [Sprecher et al., 1998] [Zhu et al., 2006]. D'autres méthodes de résolution, faisant intervenir des heuristiques [Kolisch, 1997] [Lova, 2006] ou des méta heuristiques, nous donnent des solutions admissibles. Parmi cette dernière catégorie, les algorithmes génétiques ont fait l'objet de nombreuses études [Ozdamar, 1999] [Hartman, 2001] [Alcaraz et al., 2003] [Tseng et al., 2009]. Des approches hybrides, branch-and-bound et heuristiques [Cheng et Wu, 2006] donnent des résultats intéressants. Nous trouverons un excellent état de l'art dans les travaux de Hartmann [Hartmann et al., 2010] et de Weglarz [Weglarz et al., 2011].

Notre étude se situe dans le cadre d'un arrêt annuel de production d'une industrie manufacturière. Durant cette période de fermeture (d'une durée que nous assimilerons au makespan), l'équipe de maintenance doit réaliser diverses opérations sur les équipements. Le domaine de la maintenance exige beaucoup de polyvalence au niveau des opérateurs. C'est également dans celui-ci que l'on doit optimiser le nombre de ressources pour rendre économiquement compétitif ce service. La section 2 de cette contribution va introduire la notion de matrices de compétences dynamiques permettant d'analyser l'évolution du niveau de compétences des opérateurs en fonction du type de tâches réalisées. Dans la section 3, nous proposons une modélisation du problème et nous montrerons la difficulté d'une résolution mathématique. Pour contourner cette difficulté, nous faisons appel dans la section 4 à une métaheuristique qui nous donnera une solution admissible. Une application numérique de notre stratégie est développée et commentée dans la section 5. La section 6 de cet article présente les conclusions de ce travail et les extensions en cours de développement.

2 MATRICES DE COMPETENCES DYNAMIQUES.

Ainsi, certaines sociétés profitent de la stagnation économique pour, par exemple, mieux former leurs employés, ce qui permet de répondre à une exigence de compétences accrue liée à la mondialisation et aux progrès technologiques importants de cette décennie. Les matrices de compétences, statiques ou dynamiques, constituent une donnée d'entrée pour l'élaboration du plan de formation du personnel. Dans la plupart des travaux en ordonnancement, les compétences sont définies de manière statique par rapport à l'horizon d'ordonnancement, ce qui est contraire à la réalité. En effet, ce niveau de compétence peut diminuer ou augmenter en fonction des activités de l'opérateur. Ainsi, nous considérons un système de maintenance composé de p opérateurs de maintenance (ressources), ayant des compétences dans s spécialités complémentaires (électrotechnique, mécanique, hydraulique, électricité, ...). Ces opérateurs constituent les ressources nécessaires à l'accomplissement des tâches de maintenance. Le niveau de compétence acquise $C_{i,j}$ (avec $1 \leq i < p$ et $1 \leq j < s$) de

l'opérateur i , dans une spécialité j , peut avoir les valeurs suivantes :

- $C_{i,j} = 0$: Aucune compétence pour réaliser la tâche considérée.
- $1 \leq C_{i,j} < 2$: Niveau encore insuffisant nécessitant l'assistance d'un opérateur de niveau 3 pour réaliser cette tâche.
- $2 \leq C_{i,j} < 3$: Compétence suffisante pour accomplir un travail dans cette spécialité.
- $C_{i,j} = 3$: Forte compétence pour accomplir cette tâche et possibilité de former des opérateurs de niveau 1.

Ces valeurs sont regroupées dans une matrice, appelée matrice des compétences (Figure 1), très utilisée dans l'industrie automobile.

| $p \times s$ | | | | | |
|--------------|-----------|----|----|-----------|--|
| $C_{1,1}$ | $C_{1,2}$ | .. | .. | $C_{1,s}$ | |
| $C_{2,1}$ | $C_{2,2}$ | .. | .. | $C_{2,s}$ | |
| .. | .. | .. | .. | .. | |
| $C_{p,1}$ | $C_{p,2}$ | .. | .. | $C_{p,s}$ | |

Figure 1 : Exemple de matrice de compétences (p opérateurs, s spécialités)

Nous avons ensuite voulu rendre dynamique (ou évolutive) cette matrice des compétences afin qu'elle puisse prendre en compte l'expérience d'un opérateur. Cette expérience peut faire changer de niveau un opérateur grâce à la formation par apprentissage. Dans la littérature scientifique, nous avons remarqué les travaux de Nemhard [Uzuméri et Nemhard, 2000] qui proposent une évolution et une dépréciation des compétences sous forme parabolique. Nous avons choisi de prendre une évolution simplifiée des compétences conformément à la représentation de la figure 2.

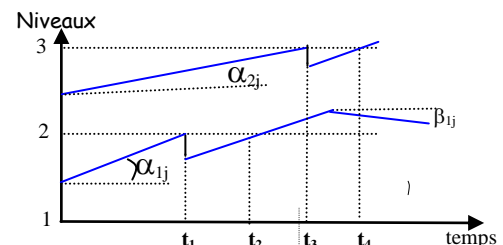


Figure 2. Evolution du niveau de compétence de l'opérateur O_i dans la spécialité j .

Le niveau de compétence de l'opérateur O_i croît avec un taux d'apprentissage α_{ij} pendant qu'il travaille dans la spécialité j et il décroît avec un taux de dépréciation β_{ij} (qui peut-être positif ou nul) dans le cas contraire (Figure 2).

Dans une spécialité donnée, un opérateur de niveau 1 pourra devenir niveau 2 par la voie de l'apprentissage, en travaillant en qualité d'apprenant, avec un opérateur niveau 3 qui sera son formateur. Dès que son niveau de compétence évoluant de manière linéaire sera obtenu, ce changement de niveau va s'effectuer sous condition de réussite à un contrôle d'aptitude. En cas d'échec (à l'instant t_1), l'apprenant continuera son apprentissage. En cas de réussite (à l'instant t_2), il deviendra niveau 2 et pourra exécuter des opérations dans la spécialité considérée de manière autonome.

L'opérateur de niveau 2 deviendra niveau 3 grâce à son travail en autonomie dans la spécialité considérée. On suppose que

l'évolution de ses compétences va s'effectuer de manière linéaire. Le passage au niveau supérieur est conditionné par un contrôle d'aptitude (instants t_3 et t_4).

Nous supposons que le taux d'apprentissage est plus élevé que le taux de dépréciation des connaissances.

3 FORMULATION MATHÉMATIQUE DU PROBLÈME

Pour résoudre ce problème, nous proposons d'utiliser un modèle de programmation linéaire inspiré des travaux de Manne [Manne, 1960], repris dans de nombreux travaux tels que [Vilcot et al., 2008] que nous avons modifiés afin de tenir compte de la compétence des opérateurs et de la fonction apprentissage.

Notations utilisées :

$R_{i,j}^m$: ensemble des m ressources dont les compétences permettent l'exécution de l'opération j de la tâche i , ($O_{i,j}$)
 $Comp(R_{k,a,b})$: niveau de compétence de la ressource k pour exécuter l'opération $O_{a,b}$.

Les variables utilisées sont:

$t_{i,j}$: date de début de l'opération $O_{i,j}$
 $y_{c,d}^{a,b,k} \begin{cases} = 0 & \text{si } O_{a,b} \text{ est effectuée avant } O_{c,d} \text{ sur la} \\ & \text{ressource } R_k, \\ = 1 & \text{dans le cas contraire.} \end{cases}$

Les contraintes sont les suivantes :

$$\forall i = 1 \dots n_T, \forall j = 1 \dots, n_O : t_{i,j} \geq 0 \quad [1]$$

Cette contrainte indique que les $t_{i,j}$ sont des variables réelles positives

$$\forall i = 1 \dots n_T, \forall j = 1 \dots n_O : t_{i,j+1} \geq t_{i,j} + p_{i,j} \quad [2]$$

Cette équation représente les contraintes de gamme opératoire

$$t_{c,d}^k \geq t_{a,b}^k + p_{a,b} - M \cdot y_{c,d}^{a,b,k} \quad [3]$$

$$t_{a,b}^k \geq t_{c,d}^k + p_{c,d} - M \cdot (1 - y_{c,d}^{a,b,k}) \quad [4]$$

$$\forall a = 1 \dots n_T, \forall b = 1 \dots n_O, \forall c = 1 \dots n_T, \forall d = 1 \dots n_O;$$

$$\forall R_k \in R_{a,b}^m \cap R_{c,d}^m$$

$$R_{a,b}^m = \sum_{t=1}^{t=p} R_{t,a,b} | (Comp(R_{t,a,b}) \geq 2) \cup (\sum_{t=1}^{t=p} \sum_{s=1, s \neq t}^{s=p} R_{t+s,a,b} | (Comp(R_{t,a,b}) = 1), (Comp(R_{s,a,b}) = 3))$$

$$R_{c,d}^m = \sum_{t=1}^{t=p} R_{t,c,d} | (Comp(R_{t,c,d}) \geq 2) \cup (\sum_{t=1}^{t=p} \sum_{s=1, s \neq t}^{s=p} R_{t+s,c,d} | (Comp(R_{t,c,d}) = 1), (Comp(R_{s,c,d}) = 3))$$

Ces deux dernières contraintes [3] [4] traduisent les disjonctions existantes entre toutes les opérations réalisées par les mêmes ressources en introduisant des variables booléennes $y_{c,d}^{a,b,k}$. M représente une grande valeur pouvant être égale à la somme des temps d'exécution de toutes les opérations. Soit $tf_{i,k}$ le temps de formation de l'opérateur i assuré par l'opérateur k . Les fonctions-objectifs à optimiser sont :

$$Cmax \geq t_{i,j} + p_{i,j} \quad \forall i = 1 \dots n_T, \forall j = 1 \dots n_O \quad [5]$$

$$Tf = \sum_{i=1}^{i=p} \sum_{k=1 \& k \neq i}^{i=p} tf_{i,k} \quad [6]$$

Ces deux objectifs étant conflictuels, nous pouvons donc déterminer un ordonnancement qui optimise un des objectifs avec le deuxième objectif prédéfini. Pour des cas relativement simples, et avec une matrice statique des compétences les équations [1] à [6] peuvent être résolues par un solveur commercial du type Cplex. Lorsque le nombre d'opérations devient important, le temps de calcul croît de manière exponentielle. Pour pallier cet inconvénient majeur et pour pouvoir utiliser une matrice dynamique des compétences, nous avons orienté notre recherche vers une méthode méta heuristique, qui pourrait nous fournir une solution jugée satisfaisante en un temps défini par le programmeur.

4 ALGORITHMES GENÉTIQUES

Les algorithmes génétiques 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 [Holland J.H., 1975] et repris dans de nombreux travaux [Goldberg, 1989] [Rendres, 1995] vont tenter de reproduire ce modèle d'évolution afin de trouver des solutions admissibles à un problème donné.

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

Figure 3. Principe d'un AG

Les principales étapes d'un algorithme génétique sont (Figure 3) :

Une génération d'individus que nous avons choisis de manière aléatoire. Chaque individu est composé d'une matrice de fabrications et d'une matrice des disponibilités des opérateurs, toutes deux générées aléatoirement (Figure 4)..

$$\begin{bmatrix} \dots & \dots & R_2 + R_3, R_3 & \dots \\ \dots & R_1 & \dots & \dots \\ R_1 + R_2, R_2 & \dots & \dots & \dots \\ \dots & \dots & \dots & R_2, R_3, R_1 + R_3 \end{bmatrix} \begin{bmatrix} Ch_x^1 \\ Ch_y \\ Ch_z^p \end{bmatrix}$$

Figure 4. Génération aléatoire d'un individu

Dans la première matrice, chaque ligne représente une tâche, chaque colonne représente un ordre chronologique de

traitement. Pour chaque opération de cette tâche, un choix aléatoire parmi les ressources disponibles est réalisé. Dans la deuxième matrice, chaque élément Ch_j^i représente le choix aléatoire j de l'opérateur i pour ses périodes d'indisponibilité. Dans l'exemple simplifié de la figure 4, la tâche T_1 sera réalisée en troisième position, la première opération sera réalisée en binôme par les ressources R_2 et R_3 , la seconde par la ressource R_3 de manière autonome. La figure 5 nous montre l'aspect évolutif des ressources disponibles grâce aux matrices de compétences dynamiques ainsi que l'adaptabilité de la matrice des disponibilités en fonction des desiderata des opérateurs. La matrice des disponibilités fournit la liste des opérateurs (autonomes ou binômes) compétents pour réaliser une opération donnée. Après chaque opération, elle est mise à jour afin de tenir compte d'un éventuel changement de niveau d'un opérateur.

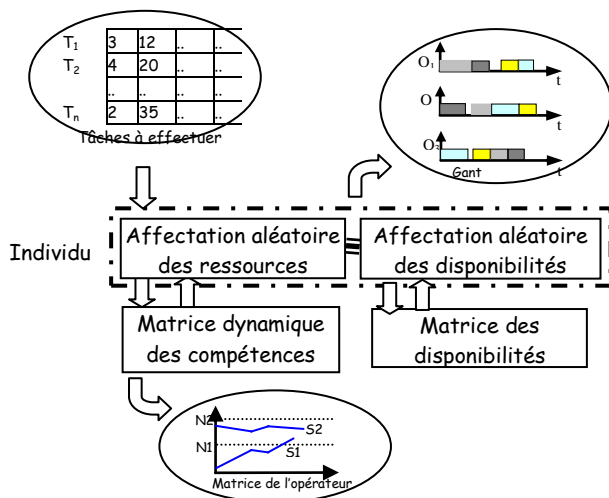


Figure 5. Caractéristiques d'un individu

Le croisement d'individus. Le croisement permet de créer de nouveaux individus. On choisit aléatoirement deux individus (parents) et on croise leurs matrices de fabrication et leurs matrices de disponibilités afin d'obtenir deux nouveaux individus (enfants) susceptibles d'améliorer les résultats. Chaque enfant va comprendre une partie des gènes d'un des parents et le complément de l'autre parent.

La mutation. Cette opération permet d'apporter une diversification de la population complémentaire à celle qui est apportée par l'opération de croisement. Nous utilisons trois types de mutation :

- pour une opération sélectionnée aléatoirement, nous remplaçons l'opérateur par un autre opérateur compatible dans la spécialité requise.
- Nous inversons la chronologie de deux tâches.
- Nous modifions le choix des périodes d'indisponibilité d'un opérateur

Nous avons choisi une probabilité dynamique de croisement τ définie par

$$\tau = \tau_{\max} \times \frac{N_{sa}}{N_{sa} \max}$$

avec

τ_{\max} , la probabilité maximale de mutation

N_{sa} , le nombre courant d'itérations de l'AG sans amélioration.

$N_{sa} \max$, le nombre maximum d'itérations sans amélioration.

L'évaluation de l'individu. Cette évaluation va se faire sur plusieurs critères que sont la valeur du makespan, le temps de formation des opérateurs et la satisfaction de l'opérateur (période de disponibilité, tâches ou formation souhaitées). Le gestionnaire de production va combiner les diverses fonctions objectifs pondérés en une seule que l'algorithme devra maximiser.

Sélection. Cet opérateur est chargé de définir les individus qui vont être dupliqués pour constituer la nouvelle population et servir de parents.

5 APPLICATION NUMERIQUE.

5.1 Données numériques.

Nous avons choisi d'illustrer les démarches proposées précédemment avec les valeurs numériques suivantes :

- Les activités de maintenance sont définies dans une benchmark (abz5) définie par Adams [Adams et al., 1988] dans laquelle nous avons remplacé le numéro de la ressource par le numéro d'une compétence.
- La matrice des compétences dynamique initiale (4 opérateurs, 5 spécialités) que nous avons adoptée est la suivante :

| p x s | | | | |
|-------|-----|-----|-----|-----|
| 3.0 | 1.9 | 0 | 3.0 | 0 |
| 2.0 | 0 | 1.9 | 0 | 2.0 |
| 0 | 3.0 | 3.0 | 0 | 1.9 |
| 1.9 | 0 | 0 | 1.9 | 3.0 |

- L'algorithme génétique est caractérisé par la taille de la population (T_{pop} individus), un croisement mono point ($p=1$), son taux de mutation (0,005), une sélection par roue de la fortune et un critère d'arrêt qui est le nombre d'itérations (nbit).

5.2 Avantages d'une matrice des compétences dynamique.

L'utilisation de matrices de compétence dynamiques nous permet de réaliser une formation ciblée des opérateurs dans le but d'atteindre les objectifs de production fixés (makespan). Cette formation, réalisée par apprentissage en binôme comprenant un formateur (niveau 3) et un apprenant (niveau 1), permet d'augmenter la flexibilité des opérateurs.

Pour pouvoir visualiser les résultats de nos simulations, nous avons développé une application qui nous permet de générer un diagramme de Gant des tâches. Cette application utilise les représentations suivantes : Opération j de la tâche i réalisée en autonomie (a), en qualité de formateur (b) ou en qualité d'apprenant (c). Nous avons aussi la possibilité de représenter les indisponibilités (§5.3) qui peuvent être fixes (d) ou mobiles(e).

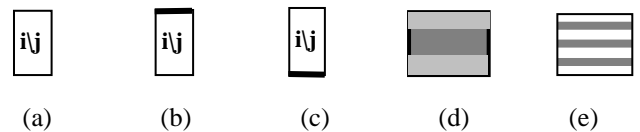




FIGURE 6. Simulation Compétences Dynamiques\Compétences Statiques

La première tâche (7\1) de la figure 6 nous montre la réalisation de l'opération n°1 de la tâche n°7. Cette activité réalisée dans la spécialité S_3 , nous montre que l'opérateur O_1 (niveau de compétence initial de 3.0) formera l'opérateur O_4 (compétence initiale de 1.9) pendant la réalisation de cette tâche. Ainsi, la compétence de l'opérateur O_4 va évoluer et celui-ci deviendra autonome dès que son niveau de compétence sera égal à deux, validé par un contrôle d'aptitude. Sur cet exemple de comparaison d'une solution des compétences dynamiques\compétences statiques, on constate que dans la solution dynamique, l'algorithme s'est efforcé de former les opérateurs pendant les premières tâches afin de les rendre autonomes pour la suite. Ainsi, on obtient une diminution du makespan de 20% par rapport au makespan réalisé avec une matrice statique.

5.3 Gestion des desiderata des opérateurs.

Nous pouvons classer les indisponibilités des opérateurs en trois catégories :

- les indisponibilités fixes correspondent aux périodes prévisibles et fixées pendant lesquelles l'opérateur ne peut pas réaliser sa mission (congés annuels, périodes de formation). Ces périodes sont fixées au début de l'algorithme et prises en compte ensuite lors des simulations afin de ne pas affecter d'opérations à un opérateur durant son indisponibilité.

- les indisponibilités mobiles négociables sont liées aux desiderata des opérateurs comme par exemple les souhaits de congés. Chaque opérateur a la possibilité d'émettre une liste ordonnée de préférences pour ses congés. L'algorithme s'efforcera de satisfaire ses souhaits (fonction désirabilité) tout en respectant les contraintes de production.

- les indisponibilités mobiles non négociables, correspondent à des périodes de réduction du temps de travail, de chômage partiel, etc. Ces indisponibilités seront gérées par l'algorithme comme des tâches virtuelles affectées

spécifiquement à l'opérateur concerné. Ainsi, lors des simulations, ces indisponibilités mobiles seront automatiquement placées dans les périodes les moins pénalisantes pour la production.

Pour trouver une solution qui satisfasse au mieux tous les critères de production et les desiderata des opérateurs, nous proposons d'utiliser la notion de désirabilité (Harrington, 1965) et la représentation graphique de Derringer (Derringer et Suich, 1980) qui nous semble bien adaptées à cette problématique. Connaissant le makespan minimum représentant la valeur cible, le gestionnaire de production impose une limite supérieure pour traiter l'ensemble du lot de pièces (Figure 7). De même, connaissant la valeur maximum du temps de formation des opérateurs (valeur cible), le gestionnaire va fixer une limite inférieure acceptable (figure 8). Si la réponse obtenue est égale à la valeur cible, on dira que le désir partiel de l'expérimentateur pour cet objectif vaut 1 (objectif atteint à 100%). Si la valeur obtenue est supérieure ou égale à la limite supérieure, on dira que le désir partiel de l'expérimentateur pour cet objectif vaut 0 (objectif atteint à 0%). Entre ces deux points, nous proposons une évolution linéaire. Sur le même principe, nous adoptons une évolution à niveaux pour les différents souhaits d'indisponibilités des opérateurs (figure 9). Dans notre application numérique, chaque opérateur propose trois souhaits pour ses périodes de congés classées par préférence (Figure 10). Le désir partiel d_{op} de tous les opérateurs sera égal au produit du désir de chacun des opérateurs.

Chacun de ces objectifs partiels (makespan, temps de formation, souhaits des opérateurs) va nous fournir un désir partiel d_i (avec $0 \leq d_i \leq 1$) pondéré. Ces différentes pondérations vont être fixées par le gestionnaire de production en fonction de ses objectifs prioritaires. Nous définissons une fonction désir global D que nous allons maximiser afin de satisfaire au mieux les différents critères.

$$D = \sqrt[w]{d_m^{w_m} \cdot d_{tf}^{w_{tf}} \cdot d_{op}^{w_{op}}} \text{ avec } w = \sum w_i$$

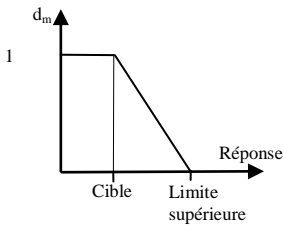


Figure 7. Désirabilité de la fonction makespan

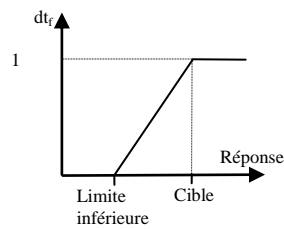


Figure 8. Désirabilité de la fonction temps de formation

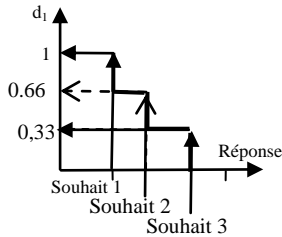


Figure 9. Désirabilité de l'opérateur i (avec $1 \leq i \leq 4$)

La figure 10, nous donne les différentes desirabilités obtenues en faisant varier les poids des différents objectifs. Les valeurs cibles et limites V_c/V_l ont été fixées à 2000\2800 pour le makespan et 2000\600 pour le temps de formation. Nous remarquons que plus le poids d'un objectif est élevé, plus nous nous rapprochons de la valeur cible.

| | Cas n°1 | Cas n°2 | Cas n°3 |
|------------------------------------|---------|---------|---------|
| Poids du Makespan : | 4 | 1 | 1 |
| Poids Temps de formation : | 1 | 4 | 1 |
| Poids des opérateurs : | 1 | 1 | 4 |
| Makespan (u.t.) : | 2265 | 2438 | 2434 |
| Temps de formation (u.t.) : | 1482 | 1995 | 1591 |
| Désir partiel (D_p) makespan : | 0,67 | 0,45 | 0,45 |
| D_p Temps de formation : | 0,63 | 1 | 0,7 |
| D_p souhait opérateurs : | 0,9 | 0,9 | 1 |
| Désir global D : | 0,69 | 0,86 | 0,82 |

Figure 10. Désirabilité en fonction du poids de chacun des objectifs

La figure 11, ci-dessous, nous montre l'évolution de la simulation pour le Cas n°2. Dans le cadre d'un ralentissement d'activité, on a privilégié le temps de formation (Poids=4) par rapport au makespan (Poids=1).

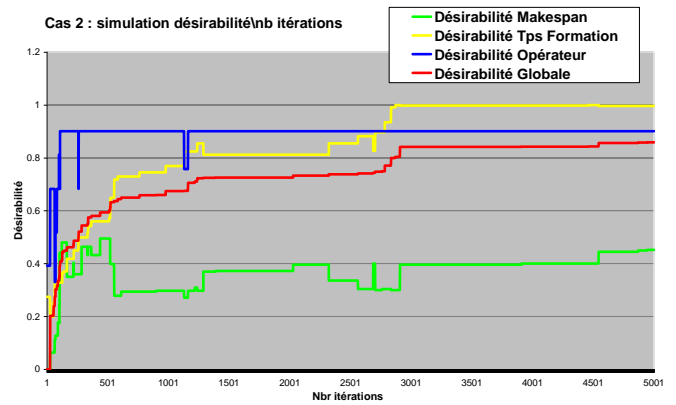


Figure 11. Variation des desirabilités lors de la simulation l'indice de satisfaction globale D

Pour la simulation du cas N°3, on a souhaité privilégier les desiderata des opérateurs (poids = 4). Après simulation, l'algorithme propose une solution, représentée sur le Gant de la figure 13 avec une desirabilité opérateur de 1. Ainsi, tous les souhaits opérateurs de rang 1 (figure 12) ont été satisfaits tout en conservant un makespan (2434 u.t.) acceptable pour l'entreprise.

| | Souhait 1 | Souhait 2 | Souhait 3 |
|------------|-----------|-----------|-----------|
| Opérateur1 | 1000/150 | 800/100 | 400/100 |
| | 1500/150 | 1200/100 | 1000/100 |
| | | 1600/100 | 1400/100 |
| Opérateur2 | 1000/300 | 1000/150 | 800/100 |
| | | 1500/150 | 1000/100 |
| Opérateur3 | 1000/300 | 500/150 | 400/100 |
| | | 1500/150 | 800/100 |
| Opérateur4 | 1000/300 | 800/150 | 800/100 |
| | | 1200/150 | 1000/100 |
| | | | 1400/100 |

Figure 12. Souhaits de congé des opérateurs (date début/durée)

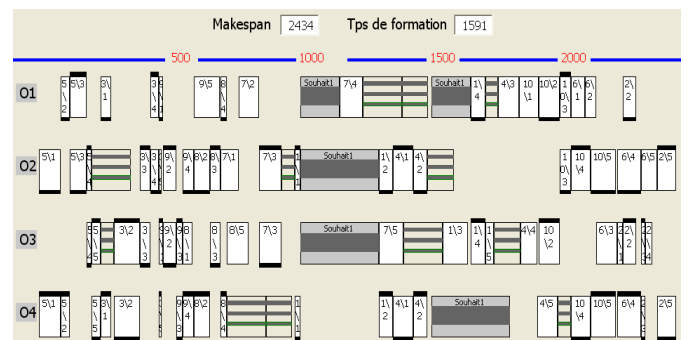


Figure 13. Diagramme Gant Simulation cas n°3

Sur le même principe, nous proposons de montrer la possibilité de satisfaire des souhaits opérateurs au niveau des opérations qu'ils souhaiteraient réaliser. L'algorithme génétique va s'efforcer d'affecter ces opérations (Figure 14) aux opérateurs demandeurs.

| Choix des opérations préférentielles | | | | | |
|--------------------------------------|-----|-----|-----|-----|-----|
| Opérateur1 | 0\2 | 4\4 | 6\1 | 9\2 | 9\3 |
| Opérateur 2 | 1\4 | 4\0 | 6\4 | 2\2 | 4\1 |
| Opérateur 3 | 8\4 | 7\2 | 0\3 | 5\1 | 7\4 |
| Opérateur 4 | 1\0 | 3\2 | 4\0 | 5\3 | 8\3 |

Figure 14. Souhaits des opérations préférentielles des opérateurs (n° tâche\n°opération)

La figure 15 indique les différentes désirabilités obtenues lors de la simulation en prenant en compte le souhait des opérateurs au niveau des opérations préférentielles (poids de 4, équitablement réparti entre tous les opérateurs).

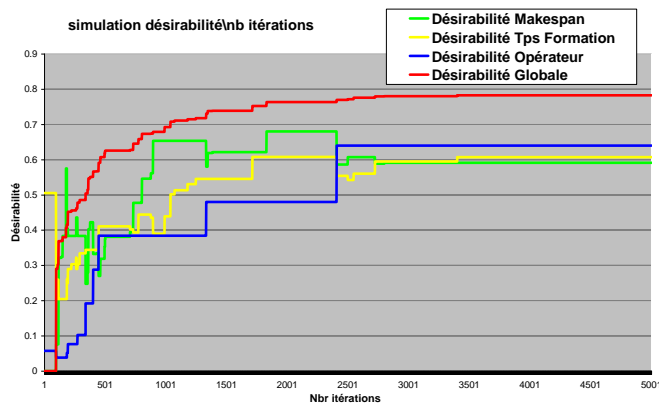


Figure 15. Variation des désirabilités.

De la même façon, une autre possibilité que nous avons testée concerne le choix par les opérateurs des spécialités sur lesquelles ils souhaitent être formés. Il est bien sûr aussi possible de combiner ces différents desiderata opérateurs lors des simulations.

6 CONCLUSIONS ET PERSPECTIVES

Dans cet article, nous avons étudié conjointement une méthodologie d'ordonnancement des tâches qui prend en compte les contraintes de l'entreprise ainsi que les desiderata des opérateurs. L'utilisation de matrices de compétences dynamiques nous permet d'établir une politique de formation du personnel en adéquation avec les objectifs de production de l'entreprise.

Après une approche analytique du problème, basé sur une matrice statique des compétences, et au vu de sa complexité, nous orientons nos recherches vers une méthode méta heuristique basée sur les algorithmes génétiques.

Afin de tenir compte des objectifs multicritères conflictuels, nous avons utilisé une méthode de pondération d'objectif qui consiste à combiner les n fonctions-objectifs en une seule. Cette méthodologie prend en compte les préférences du gestionnaire de production, ainsi que le choix ordonné des opérateurs concernant leurs indisponibilités négociables et les tâches souhaitées. Cette méthodologie, qui est un outil d'aide à la décision pour résoudre des problèmes d'ordonnancement multicritère, permet au gestionnaire de production d'exprimer, de manière claire et conviviale, ses préférences en rapport avec les buts fixés pour chaque objectif et les résultats obtenus.

Une extension de ces travaux qui prend en compte la sous-traitance est actuellement en cours de développement.

7 REMERCIEMENTS

Les auteurs remercient Nathalie Sauer pour ses suggestions et commentaires qui ont permis d'améliorer la version initiale de ce document.

8 REFERENCES

- Alcaraz J, Maroto C. et Ruiz R., (2003) "Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms", *Journal of the operational Research Society*, 54 (6), pp.614-626.
- Adzakpa Kossi-Pélopé et Adjallah Kondo-Hloindo, (2003) "A new effective heuristic for the intelligent management of the preventive maintenance tasks of the distributed systems", *Advanced Engineering Informatics*, n°17, pp. 151-163.
- Aytug H., Lawley M., McKay K., Mohan S. et Uzsoy R. (2005), Executing production schedules in the face of uncertainties: a review and some future directions. *European Journal of Operational Research*, 161(1), pp. 86-110.
- Bellenguez-Morineau Odile, (2006) Méthodes de résolution pour un problème de gestion de projet avec prise en compte de compétences, Thèse de Doctorat, Université de Tours, novembre
- Blazewicz J., Lenstra J.K. et Rinnoy Kan A.H.G., (1983) Scheduling projects subject to resource constraints: classification and complexity, *Discrete Applied Mathematics*, n°5, pp. 11-24.
- Brucker P., Drexl A., Morhing R., Neumann K. et Pesch E., (1999) Resource Constrained Project Scheduling Problem: Notation, classification, models and methods, *European Journal of Operational Research*, n°112, , pp. 3-41.
- Cheng X. et Wu C., (2006) Hybrid algorithm for complex project scheduling, *Computer Integrated Manufacturing Systems*, 12(4), pp. 585-9.
- Coudert T., (2000) Apport des systèmes multi-agents pour la négociation en ordonnancement : application aux fonctions production et maintenance. PhD thesis, Laboratoire Génie de Production, ENI de Trarbes.
- De Reyck B. et Herroelen W. (1999), The Multi-Mode Resource-Constrained Project Scheduling Problem with generalized precedence relations, *European Journal of Operational Research*, n° 119, pp. 538-556.
- Derringer G. et Suich R., (1980) Simultaneous Optimization of several Response Variables. *Journal of Quality Technology*, 12(4), pp. 214-219.
- Drezet L.-E. et Billaut J.-C., (2008) A project scheduling problem with labour constraints and time-dependent activities requirements, *International Journal of Production Economics*, vol. 112, pp. 217-225.
- Goldberg D.E., Genetic algorithms in search, (1989) Optimisation and machines Learning. Addison Wesley, Mass.
- Harrington E. C. Jr, (1965) The Desirability Function. *Industrial Quality Control*, 21 (10), pp. 494-498.
- Hartman S., Project scheduling with multiple modes: a genetic algorithm, (2001) *Annals of operations Research*, 102, (1-4), pp. 11-135.
- Hartmann Sonke et Briskorn Dirk, (2010) A survey of variants and extensions of the resource-constrained project scheduling problem, *European Journal of Operational Research*, 207, pp. 1-14.
- Herroelen W. et Leus R., (2005) Project scheduling under uncertainty - survey and research potentials. *European Journal of Operational Research*, n°165(2), pp. 289-306.

- Holland J.H., (1975) Adaptation in natural and artificial system. Ann Arbor, The University of Michigan Pres.
- Kaabi-Harrath, (2008) Contribution a l'ordonnancement des activités de maintenance dans les systèmes de production, thèse de l'université de Franche-Comté,
- Kacem I., (2005) Minimising the sum of the completion time with availability constraint: comparaison of branch and bound method and integer linear programming. *IESM'05*, Marrakech, Maroc, pp. 16-19.
- Kolisch R. et Drexl A., (1997) Local search for nonpreemptive multi-mode resource constrained project scheduling. *IIE Transactions*, 29, pp. 987-999.
- Lova A., Tormos P., et Barber F., (2006) Multi-Mode Resource Constrained Project Scheduling: Scheduling Schemes, Priority Rules and Mode Selection Rules. *Inteligencia Artificial*, 30, pp. 69-86.
- Marmier R., (2007) Contribution à l'ordonnancement des activités de maintenance sous contrainte de compétences: une approche dynamique, proactive et multicritère, Thèse de doctorat de l'Université de Franche-Comté.
- Nembhard D.A. et Uzumeri M.V., (2000) Experimental learning and forgetting for manual and cognitive task, *International Journal of Industrial Ergonomics*, Vol. 25, N°4, pp. 315-326.
- Ozdamar L., (1999) A genetic algorithm approach to a general category project scheduling problem, *IEEE Transactions on system, Man and Cybernetics – Part C: Applications and Reviews* 29 (1), pp. 44-59.
- Renders J.M., (1995) Algorithmes génétiques et réseaux de neurones, Editions Hermes.
- Sprecher A. et Drexl A., (1998) Multi-mode resource-constrained project scheduling with a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, 107, pp. 431-450.
- Tseng L.Y. et Chen S.C., (2009) Two-phase genetic local search algorithm for the multi-mode resource-constrained project scheduling problem. *IEEE Transactions on Evolutionary Computation*, n° 13 (4), pp. 848-857.
- Van Peteghem V. et Vanhoucke M., (2010) An genetic algorithm for the preemptive and nonpreemptive Multimode-Ressource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, n°201(2), pp. 409-418.
- Weglarz Jan, Jozefowska Joanna, Mika Marek et Waligôra Grzegorz, (2011) Project Scheduling with finite or infinite number of activity processing modes – A survey, *EJOR* n° 208, pp. 177-205.
- Zhu G., Bard J., et Tu G., (2006) A Branch-and-Cut Procedure for the Multimode Resource-Constrained Project-Scheduling Problem, *Journal on Computing*, 18(3), pp. 377- 390.