RAIRO. Recherche opérationnelle

Alain Hertz
Vincent Robert
Vincent Berthod

Planification des emplois du temps et de la formation au sein d'une grande entreprise

RAIRO. Recherche opérationnelle, tome 34, nº 1 (2000), p. 61-83
<http://www.numdam.org/item?id=RO_2000__34_1_61_0>

RAIRO Operations Research
RAIRO Oper. Res. **34** (2000) 61-83

PLANIFICATION DES EMPLOIS DU TEMPS ET DE LA FORMATION AU SEIN D'UNE GRANDE ENTREPRISE (*)

par Alain HERTZ ([1]), Vincent ROBERT ([1]) et Vincent BERTHOD ([1])

Communiqué par Jacques FERLAND

Résumé. – *Nous décrivons une méthode issue de la Recherche Opérationnelle permettant de planifier l'emploi du temps des collaborateurs d'une grande entreprise. La méthode que nous proposons tient compte aussi bien de la rentabilité immédiate de l'entreprise que de la formation des collaborateurs, le but étant d'assurer une relève au sein de l'entreprise. Afin d'offrir au planificateur un outil simple et rapide d'aide à la décision, nous avons opté pour une approche de type glouton. La méthode a été implantée avec succès dans un important établissement bancaire helvétique.*

Mots clés : Emplois du temps, flot maximum à coût minimum, méthode potentiel-tâche, système d'aide à la décision.

Abstract. – *We describe an O.R. technique which plans the allotment of time of the collaborators of a big company. The proposed method not only considers the immediate profitability of the company, but also the training of the collaborators in order to guarantee the success of the company's rising generation. The proposed method uses a greedy approach and constitutes therefore a simple and fast tool for decision makers. It has been successfully implemented in an important Swiss bank society.*

Keywords: Timetabling, minimum cost flow, critical path method, decision aid system.

1. INTRODUCTION

Alors que la compétitivité mondiale devient plus exacerbée que jamais, et que la rentabilité atteint des seuils encore insoupçonnés à ce jour, nombre d'entreprises se tournent désormais vers l'amélioration non plus des techniques ou du matériel de production, mais de leur potentiel humain. Il semble en effet qu'un des grands défis des décideurs de demain sera de savoir utiliser au mieux toutes les ressources humaines à leur disposition. Ceci est plus particulièrement vrai pour les entreprises de service au premier rang desquelles nous trouvons les grandes administrations aussi bien étatiques

(*) Reçu en juillet 1997.
([1]) Département de Mathématiques, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.

Recherche opérationnelle/Operations Research, 0399-0559/00/01/$ 7.00

que privées (banques, assurances, etc.). Cette valorisation des ressources humaines nécessite la mise en place de politiques permettant d'obtenir des résultats à court terme (rentabilité, condition de travail agréable pour le personnel, etc.) et à long terme (sécurité et pérennité de l'entreprise, expansion, etc.). Pour cela, le responsable des ressources humaines se doit d'avoir à sa disposition des outils performants lui permettant de visualiser sur une période déterminée les conséquences des choix qu'il entend effectuer à un instant précis.

Nous présentons une méthode issue de la recherche opérationnelle permettant d'obtenir une planification des diverses tâches pour les collaborateurs d'une grande entreprise. Nous tenons compte aussi bien de la rentabilité immédiate de l'entreprise, que de la formation des collaborateurs, assurant ainsi une relève au sein de l'entreprise. Notre objectif étant de développer un outil simple et rapide d'aide à la décision, nous avons opté pour une approche de résolution de type glouton.

2. DESCRIPTION DU PROBLÈME

Nous nous plaçons ici dans le contexte d'une entreprise dont les employés sont dirigés par le même planificateur. Les tâches à remplir par cette entreprise sont répétitives et fixées dans un cahier des charges. Chaque tâche doit être assurée en tout temps par un nombre donné de collaborateurs compétents. Un collaborateur ne peut travailler que sur une tâche à la fois et y reste affecté jusqu'à ce que le planificateur en décide autrement.

Dans ce modèle, seules les ressources humaines sont prises en compte. On suppose que les ressources matérielles sont en nombre suffisant et que par conséquent, il n'y a aucune contrainte sur l'équipement requis par chaque tâche.

Chaque collaborateur peut être amené à travailler sur n'importe quelle tâche de l'entreprise. Toutefois, la compétence d'un collaborateur pour une tâche est variable et dépend de son expérience. Son niveau de compétence est une grandeur quantifiable dépendant de sa formation, de son curriculum vitae et principalement du temps qu'il a passé sur une tâche. Il est clair que le niveau de compétence d'un employé pour une tâche augmente en fonction du temps qu'il passe sur cette dernière. On estime qu'un collaborateur est *compétent* sur une tâche dès qu'il peut assumer seul son poste sur cette dernière. Par abus de langage, nous dirons souvent que le collaborateur peut *assurer la tâche.* Si son niveau est inférieur, il doit obligatoirement être encadré par un autre collaborateur compétent. Nous considérons quatre niveaux de compétence : par ordre croissant d'importance, nous avons *débutant, novice, avancé* et *expert*. Un collaborateur est compétent et peut donc assurer une tâche lorsqu'il a dépassé le stade novice sur cette tâche.

La *sécurité à court terme* consiste à assurer toutes les tâches par des collaborateurs compétents. Dès que cette sécurité est assurée, les collaborateurs encore libres sont placés en formation sur des tâches pour lesquelles ils ne sont pas encore compétents. On dit que ces collaborateurs sont des *stagiaires*. Remarquons qu'une personne compétente sur une tâche ne peut acquérir de l'expérience sur cette dernière que si elle assure cette tâche. Le nombre de stagiaires par tâche ne peut dépasser une valeur maximale propre à la tâche concernée. Cette valeur, fixée par le planificateur, doit prendre en compte le nombre de collaborateurs compétents qui travaillent sur la tâche ainsi que la difficulté du travail à accomplir. Chaque personne possède une liste de priorités des tâches pour lesquelles il devrait être formé. Ces priorités dépendent aussi bien des goûts du collaborateur que de la volonté de l'entreprise d'avoir au moins un certain nombre de personnes compétentes par tâche de manière à pouvoir assurer son avenir à long terme. En formant des collaborateurs sur certaines tâches, l'entreprise cherche à assurer sa *sécurité à long terme*. Le but d'une telle politique est donc de disposer d'une équipe apte à répondre à tous les problèmes de suivi des travaux suite à une absence due, par exemple, à une maladie ou un décès.

L'accession au niveau de compétence supérieur intervient après un laps de temps dépendant du niveau courant du collaborateur et de sa situation personnelle. C'est le planificateur qui fixe de manière individuelle le temps que doit passer une des personnes de l'équipe sur un travail avant d'accéder à un niveau de compétence supérieur. Le temps total passé par un collaborateur sur une tâche est égal à la somme des temps morcelés que cette personne a effectués sur cette dernière.

Les tâches assurées par l'entreprise étant répétitives, la ou les personnes affectées à une tâche ne le seront que pour un laps de temps maximum prédéterminé, variable selon le travail, ceci de manière à ne pas lasser le personnel. Le responsable a toutefois la possibilité de forcer un collaborateur à travailler entre deux dates arbitraires sur une tâche particulière. Il n'y a pas dans ce cas de notion de durée maximale.

Pour éviter des changements trop fréquents, la durée minimale d'affectation à une tâche est elle aussi fixée. Les mêmes règles s'appliquent tout naturellement aussi pour le cas de la formation de collaborateurs.

Une autre contrainte consiste à assurer une sorte de « permanence » sur certaines tâches. Pour cela, on peut adjoindre à certaines tâches des groupes de collaborateurs compétents parmi lesquels on désire qu'il y en ait toujours au moins un de présent sur la tâche concernée. Par exemple, supposons que l'on veuille imposer qu'au moins un des collaborateurs de l'ensemble $\{C_1, C_2, C_3\}$ soit présent sur une tâche particulière ; ceci signifie que si une seule personne doit assurer cette tâche, elle sera obligatoirement choisie parmi ces trois collaborateurs. Si par contre la tâche requiert plus de personnel, on choisira au moins l'un des trois collaborateurs précités, ainsi que d'autres collaborateurs si nécessaire. Cette contrainte de *permanence* apparaît dans le cas où l'on dispose de collaborateurs très qualifiés pour certaines tâches importantes. Ceci est particulièrement vrai dans le cas de travaux impératifs requérant une réponse rapide et précise (par exemple, pour des actes médicaux d'urgence ou des opérations en bourse). En obligeant au moins un des membres du groupe à être là, on est donc sûr d'avoir toujours un spécialiste sous la main. De plus, on dispose quand même d'une plus grande souplesse que dans le cas de l'affectation obligatoire que nous avions mentionné précédemment.

Les absences des collaborateurs peuvent être de deux types : les absences *flexibles* et les absences *fixes*. Une absence est dite flexible si elle est de courte durée et imprévisible (par exemple, pour cause de maladie ou de deuil). Dans ce cas, les autres membres de l'équipe sont à même d'effectuer la tâche laissée vacante en sus de leur travail, de sorte que nous ne nous y intéresserons pas. Les absences fixes, par contre, sont des absences planifiées de longue date, telles que les vacances ou les séjours hospitaliers. Ces absences sont longues et connues à l'avance. Dans ce cas, il peut être nécessaire de réorganiser le travail pour continuer à assurer la sécurité à court terme. Une absence fixe est considérée comme un changement d'affectation. On peut en quelque sorte voir cette absence comme une tâche fictive sur laquelle on pourrait rester au minimum zéro jour et au maximum une éternité. Un collaborateur absent y serait obligatoirement affecté pendant toute la durée de son absence. Ceci fait qu'à son retour le collaborateur peut très bien être réaffecté à son précédent travail, et dans ce cas, le nombre de jours passés sur cette tâche repart à zéro.

Le travail du planificateur consiste donc à trouver une affectation au jour le jour permettant d'assurer un suivi correct de toutes les tâches et tenant compte de toutes les contraintes énoncées précédemment. Il est encore important de noter que nous ne considérons pas ici de pondération sur les travaux à effectuer en fonction par exemple de l'attention qu'ils requièrent.

3. CONTRAINTES DU PROBLÈME

Nous décrivons ci-après les contraintes du problème que nous classons en deux catégories. Une contrainte est dite forte si elle doit absolument être respectée. On parle, par contre, de contrainte *faible* lorsque celle-ci peut être violée, à quelques rares occasions.

Contraintes fortes

(a) Un collaborateur ne peut être affecté qu'à une seule tâche par jour.

(b) Le nombre requis de collaborateurs compétents sur une tâche doit toujours être respecté.

(c) Certaines tâches nécessitent la présence d'au moins une personne faisant partie d'un groupe de collaborateurs donné.

(d) Les affectations obligatoires demandées par le planificateur doivent être respectées.

(e) Les indisponibilités des collaborateurs doivent être prises en compte.

(f) Chaque tâche ne peut accueillir qu'un nombre limité de stagiaires. Le planificateur fixe lui-même ce maximum pour chaque tâche.

Contraintes faibles

(g) Respect des nombres minimal et maximal de jours consécutifs que peut passer un collaborateur sur une tâche.

(h) Les priorités de formation des collaborateurs devraient être respectées.

(i) Le nombre de collaborateurs compétents que l'entreprise souhaite avoir pour chaque tâche devrait être atteint à long terme.

Une affectation des collaborateurs aux tâches est dite *admissible* si toutes les contraintes fortes sont satisfaites.

4. FORMALISME

Soit $C = \{C_1, ..., C_n\}$ un ensemble de collaborateurs et $T = \{T_1, ..., T_t\}$ un ensemble de tâches devant être accomplies par ceux-ci. Notons N_{ij} le niveau de compétence du collaborateur C_i sur la tâche T_j. Nous considérons ici le cas général où les collaborateurs peuvent être de K niveaux distincts de compétence, le premier étant le plus faible, le dernier le plus fort. Le niveau k_0 $(1 \leq k_0 \leq K)$ est le niveau à partir duquel une personne n'a plus besoin d'être supervisée et est donc considérée comme compétente.

Nous noterons TEMPS_{ijk} le temps nécessaire au collaborateur C_i pour passer du niveau k au niveau supérieur $k+1$ sur la tâche T_j. Par convention, nous poserons $\mathrm{TEMPS}_{ijK=\infty}$ puisqu'une personne C_i de niveau maximal K sur la tâche T_j ne peut pas passer à un niveau supérieur sur cette tâche.

Considérons p différents niveaux de priorité pour la formation des collaborateurs sur les tâches. Notons P_{ij} l'entier appartenant à $\{1, ..., p\}$ qui représente la priorité de formation de C_i sur T_j. On préférera donc affecter C_i en formation sur T_j plutôt que sur $T_{j'}$ si $P_{ij} > P_{ij'}$. La priorité est traitée ici comme une notion absolue et non pas relative, c'est-à-dire que nous ne créons pas une hiérarchie entre les priorités d'un collaborateur, mais nous leur attribuons simplement une valeur absolue. Ainsi un collaborateur peut très bien avoir deux tâches avec une même valeur de priorité. La personne peut alors être placée de manière équivalente sur chacune des deux tâches. Nous avons retenu cette approche, en fixant $p = 2$, étant donné qu'en règle générale, un collaborateur ne désire être formé que sur un nombre relativement limité de tâches, les autres lui important peu.

Nous utiliserons de plus les notations suivantes :

- CN_j l'ensemble des collaborateurs compétents pour la tâche T_j,
- CO_j la liste des collaborateurs dont un au moins doit obligatoirement être présent à tout instant sur T_j.
- TC_i l'ensemble des tâches pour lesquelles le collaborateur C_i est compétent,
- R_j le nombre de personnes compétentes nécessaires à chaque instant pour l'exécution de T_j,
- F_j le nombre maximal de stagiaires pouvant être placés en formation sur la tâche T_j,
- O_t l'ensemble des paires (i, j) tel que le collaborateur C_i doit obligatoirement travailler sur la tâche T_j à l'instant t,
- L_j le nombre de collaborateurs compétents que l'entreprise désire à long terme pour la tâche T_j.
- MIN_j (respectivement MAX_j) le nombre minimal (respectivement maximal) de jours consécutifs pouvant être passés par un collaborateur sur la tâche T_j.

5. RÉSOLUTION DU PROBLÈME

Il existe sur le marché de nombreux logiciels permettant de planifier les emplois du temps du personnel d'une entreprise. Ces logiciels ne tiennent cependant pas compte de la formation continue des employés, et plus généralement de la sécurité à long terme de l'entreprise. C'est la raison pour laquelle nous avons décidé de développer nos propres méthodes.

Diverses techniques de résolution peuvent être envisagées. Une approche possible serait l'utilisation d'algorithmes de propagation de contraintes, en formulant le problème en termes de satisfaction de contraintes [Tsa93, Bou95]. Les deux défauts suivants nous ont incités à choisir d'autres techniques.

- Cette approche aurait pour objectif de déterminer une solution satisfaisant toutes les contraintes (a),...,(i). De telles solutions idéales n'existent cependant pas en pratique, et il est important que le planificateur ait la possibilité de décider de manière dynamique si telle ou telle contrainte peut être relaxée. Ce n'est pas un ordinateur qui peut prendre de telles décisions à sa place.
- Les problèmes de satisfaction de contraintes sont généralement résolus à l'aide de techniques de recherche exhaustive ; ces méthodes peuvent être très coûteuses en temps de calcul. Le planificateur désire cependant pouvoir comparer très rapidement divers scénarios, et mettre en évidence les conséquences de ses choix.

C'est également pour des raisons de temps de calculs prohibitifs qu'il ne nous a pas semblé approprié d'utiliser une technique de recherche locale [Ree93] (tel que le recuit simulé ou la recherche tabou) pour résoudre ce problème d'optimisation sous contraintes.

Pour répondre au mieux aux besoins du planificateur, nous avons décidé de mettre en place une approche itérative, de type glouton qui détermine une première affectation admissible pour le premier jour, et la conserve jusqu'au jour où l'une des contraintes fortes, ou la contrainte faible la moins prioritaire (contrainte (g)) n'est pas satisfaite ; une nouvelle affectation est alors recalculée et le processus est répété jusqu'à la fin de l'horizon de planification.

Plus précisément, supposons que l'on soit intéressé à affecter les collaborateurs aux tâches entres les dates D_{init} et D_{fin}. Au jour $D_0 = D_{\text{init}}$, nous déterminons une première affectation admissible A_0 des collaborateurs aux tâches. Partant du jour D_0, nous déterminons ensuite la première date D_1 ($D_1 > D_0$) à partir de laquelle au moins l'une des contraintes fortes (a),...,(f), ou la contrainte faible (g) est violée. À l'instant D_1, il est donc souhaitable de modifier l'affectation A_0 des collaborateurs aux tâches. Nous déterminons une nouvelle affectation admissible A_1 en tenant compte du temps passé par les collaborateurs sur les tâches entre les dates D_0 et D_1. Ceci étant fait, nous calculons la prochaine date D_2 ($D_2 > D_1$) à laquelle l'affectation courante A_1 devra être remise en cause, et déterminons une nouvelle affectation A_2, et ainsi de suite. Ce processus est réitéré jusqu'à ce que la date de changement D_k courante soit chronologiquement supérieure à la date de fin D_{fin}.

La recherche d'une affectation A_k, valable dans l'intervalle $[D_k, D_{k+1}[$, est effectuée en deux temps. Afin de satisfaire la sécurité à court terme, nous tentons tout d'abord de déterminer une affectation satisfaisant les contraintes fortes (a), ..., (e), ainsi que la contrainte faible (g). Puis, nous planifions les stages de formation en tenant compte des contraintes (a),(e),(f) ainsi que de toutes les contraintes faibles.

Afin de satisfaire les contraintes fortes et la contrainte faible (g) du problème lors de la détermination de l'affectation A_k, nous serons parfois amenés à reconsidérer la solution partielle courante et à décaler dans le temps certaines dates D_i ($i < k$) de changements d'affectation. Si de tels décalages ne permettent pas d'obtenir une affectation A_k satisfaisant les contraintes (a), ..., (g), nous tolérerons alors la violation de la contrainte faible (g).

5.1 Recherche d'une affectation satisfaisant la sécurité à court terme

Pour satisfaire la sécurité à court terme, il suffit d'affecter des collaborateurs compétents à chaque tâche, en tenant compte des contraintes du problème. La recherche d'une telle affectation A_k, valable dès l'instant D_k, peut être formulée mathématiquement de la manière suivante. Soit I l'ensemble des collaborateurs supposés être présents au temps D_k. Pour chaque collaborateur $C_i \in I$, nous considérons l'ensemble TC_i des tâches pour lesquelles il est compétent. Pour un collaborateur $C_i \in I$, nous noterons :

- $T(i)$ la tâche à laquelle il est affecté avant le changement d'affectation de l'instant D_k.
- Δ_i le temps qu'a passé C_i sur $T(i)$ depuis sa dernière modification d'affectation.

Le coût c_{ij} d'affectation d'un collaborateur $C_i \in I$ à une tâche $T_j \in TC_i$ est calculé de la manière suivante :

$$c_{ij} = \begin{cases} \text{MIN}_{T(i)} - \Delta_i & \text{si} \quad T_j \neq T(i) \quad \text{et} \quad \Delta_i < \text{MIN}_{T(i)} \\ \Delta_i - \text{MAX}_{T(i)} + 1 & \text{si} \quad T_j = T(i) \quad \text{et} \quad \Delta_i \geq \text{MAX}_{T(i)} \\ 0 & \text{sinon.} \end{cases}$$

De cette manière, nous espérons satisfaire la contrainte faible (g). En effet, nous favorisons le changement d'affectation du collaborateur C_i lorsque celui-ci est déjà affecté à la tâche $T(i)$ depuis $\mathrm{MAX}_{T(i)}$ jours. Par contre, nous essayons d'empêcher le changement d'affectation de C_i si ce collaborateur n'a pas passé suffisamment de temps sur $T(i)$ depuis son dernier changement d'affectation.

Le problème à résoudre peut alors être formulé de la manière suivante :

$$
\begin{array}{llll}
\min & \displaystyle\sum_{C_i \in I} \sum_{T_j \in TC_i} c_{ij} x_{ij} & & \\
\text{sc} & \displaystyle\sum_{T_j \in TC_i} x_{ij} \le 1 & C_i \in I & \text{(a)} \\
 & \displaystyle\sum_{i \mid T_j \in TC_i} x_{ij} = R_j & 1 \le j \le t & \text{(b)} \\
 & \displaystyle\sum_{C_i \in CO_j} x_{ij} \ge 1 & 1 \le j \le t & \text{(c)} \\
 & x_{ij} = 1 & (i,j) \in O_{D_k} & \text{(d)} \\
 & x_{ij} \in \{0,\ 1\} & C_i \in I,\ T_j \in TC_i &
\end{array}
$$

où $x_{ij} = \begin{cases} 1 & \text{si le collaborateur } C_i \in I \text{ est affecté à la tâche } T_j \in TC_i \\ 0 & \text{sinon.} \end{cases}$

Les contraintes (a), ..., (d) sont exactement celles décrites dans la section 3. Les contraintes (e) et (g) de la section 3 sont implicitement prises en compte par les variables x_{ij} et les coût c_{ij}.

Ce problème d'affectation classique est facile à résoudre. En effet, s'il existe une affectation admissible, alors la solution optimale peut être obtenue à l'aide de la recherche d'un flot maximum à coût minimum dans un réseau [For62, Gon90]. Remarquons tout d'abord qu'il est facile de satisfaire la contrainte (d) puisqu'il suffit de forcer l'affectation d'un collaborateur C_i sur une tâche T_j si celui-ci doit obligatoirement être affecté à T_j au temps D_k. On peut ainsi réduire d'une unité le nombre R_j de personnes nécessaires à l'exécution de T_j et se ramener à un problème avec un collaborateur de moins. Nous supposerons donc par la suite que la contrainte (d) est satisfaite et qu'il nous reste à déterminer une solution pour les collaborateurs dont l'affectation n'est pas forcée.

Le réseau dans lequel nous déterminons un flot maximum à coût minimum est construit de la manière suivante.

- On associe tout d'abord un nœud, noté C_i, à chaque collaborateur $C_i \in I$ ainsi que trois nœuds notés Tj, T'_j et T''_j à chaque tâche $T_j \in T$.
- Soit T_j une tâche et C_i un collaborateur compétent pour T_j. Si $C_i \in CO_j$ alors on crée un arc entre le nœud C_i et le nœud T_j. Si par contre $C_i \notin CO_j$ alors on relie C_i au nœud T'_j. Un arc reliant un nœud C_i à un nœud T_j ou à un nœud T'_j est de coût c_{ij} et de capacité 1.
- Nous rajoutons une source s reliée à chaque nœud C_i par un arc de coût 0 et de capacité 1.
- Pour chaque tâche $T_j \in T$, nous relions chacun des nœuds T_j et T'_j au nœud T''_j. L'arc reliant T_j à T''_j est de coût 0 et de capacité $|CO_j|$ alors que celui reliant T'_j à T''_j est de coût 0 et de capacité $R_j - 1$.
- Finalement nous rajoutons un puits p et relions chaque nœud T''_j à p par un arc de coût 0 et de capacité R_j.

Un exemple d'un tel réseau est représenté dans la figure 1 (les nœuds T'_j sans prédécesseur ne sont pas représentés). Dans cet exemple, nous avons 5 collaborateurs et 3 tâches. Nous supposons avoir les données décrites dans le tableau 1 ci-dessous.

TABLEAU 1

i	$T(i)$	Δ_i
1	T_1	10
2	T_2	40
3	T_1	10
4	T_2	15
5	T_3	17

j	CN_j	CO_j	MIN_j	MAX_j	R_j
1	$\{C_1, C_2, C_3, C_5\}$	$\{C_1, C_5\}$	5	30	2
2	$\{C_2, C_3\}$	$\{C_2, C_3\}$	15	40	1
3	$\{C_3, C_4, C_5\}$	$\{C_3, C_4, C_5\}$	20	35	1

Tous les arcs sont de coût nul sauf celui reliant C_5 à T_1 (car il manque encore 3 jours à C_5 pour avoir passé $\text{MIN}_3 = 20$ jours sur T_3) et celui reliant C_2 à T_2 (car C_2 a déjà passé $\text{MAX}_2 = 40$ jours sur T_2).

Les triangles pleins correspondent aux unités d'un flot maximum à coût minimum qui circule dans le réseau. Lorsqu'une unité de flot circule entre un nœud C_i et un nœud T_j ou T'_j, cela signifie que le collaborateur C_i est affecté à la tâche T_j. Nous décidons donc d'affecter C_1 et C_2 sur T_1, C_3 sur T_2, C_5 sur T_3, alors que C_4 est libre et peut donc être placé en formation.

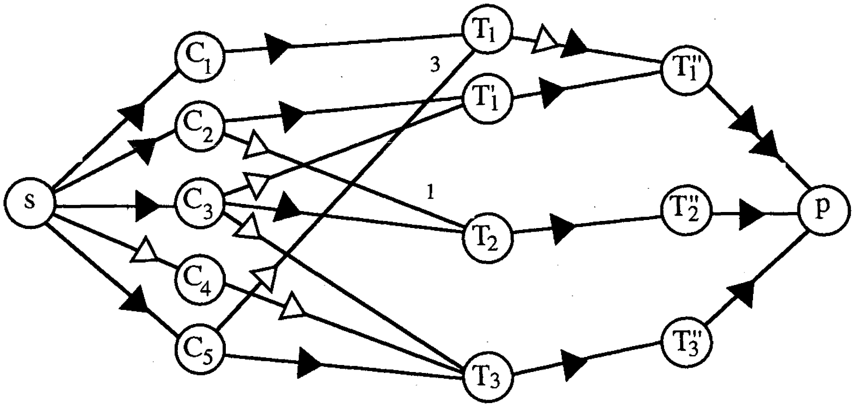

Figure 1. – Réseau associé au problème de la recherche d'une affectation satisfaisant la sécurité à court terme.

Si le flot maximum sature tous les arcs entrant dans le puits p, alors l'affectation correspondant à ce flot satisfait toutes les contraintes du problème. En effet,

- chaque collaborateur n'est affecté qu'à au plus une tâche,
- exactement R_j personnes sont affectées à chaque tâche T_j,
- au plus $R_j - 1$ collaborateurs $C_i \notin CO_j$ sont affectées à une tâche T_j, ce qui nous assure que la contrainte (c) est bien satisfaite.

Il peut arriver que les compétences des collaborateurs présents ne permettent pas d'assurer la sécurité à court terme. Une telle situation est caractérisée par un flot maximum de valeur inférieure à la somme des R_j $(j = 1, ..., t)$. Il se peut également que la seule façon d'assurer la sécurité à court terme requiert une violation de la contrainte (g). Ce cas se traduit par un flot maximum de coût minimum strictement positif. Nous verrons dans la section 6 comment modifier les décisions prises avant la date D_k, afin de tenter de respecter la contrainte (g).

5.2 Recherche d'une affectation satisfaisant la sécurité à long terme

Dans cette section, nous décrivons comment déterminer une affectation des collaborateurs encore libres à des tâches de formation à un instant D_k donné. Soit I' l'ensemble des collaborateurs présents au temps D_k et qui ne sont pas affectés à une tâche pour satisfaire la sécurité à court terme.

Pour chaque collaborateur $C_i \in I'$, nous considérons l'ensemble J_i des tâches pour lesquelles il n'est pas compétent, c'est-à-dire $J_i = \{T_j \in T \mid T_j \notin TC_i\}$. Chaque collaborateur C_i de I' peut donc être mis en formation sur chaque tâche T_j de J_i. Comme dans la section précédente, nous notons $T(i)$ la tâche à laquelle C_i est affecté avant le changement d'affectation de l'instant D_k, alors que Δ_i représente le temps passé par C_i sur $T(i)$ depuis sa dernière modification d'affectation.

Le coût c'_{ij} d'affectation d'un collaborateur $C_i \in I'$ à une tâche $T_j \in J_i$ est défini comme la somme pondérée de trois termes, ceci afin de prendre en compte trois objectifs : le respect des priorités de formation, le respect des objectifs de formation de l'entreprise, le respect des nombres minimal et maximal de jours consécutifs que peut passer un collaborateur sur une tâche.

Le premier terme est égal à $\max\{0, |CN_j| - L_j + 1\}$. Il est positif uniquement si le nombre de collaborateurs compétents pour T_j égale ou dépasse le nombre souhaité par l'entreprise pour sa sécurité à long terme. On favorise ainsi le respect de la contrainte (i).

Le deuxième terme prend en compte les priorités de formation des collaborateurs. Il est défini comme étant égal à $p - P_{ij}$, (où p est le niveau de priorité maximum). Une affectation de coût minimum favorise ainsi le respect de la contrainte (h).

Finalement, le troisième terme est égal au coût c_{ij} défini dans la section précédente et prend donc en compte la contrainte (g).

En considérons trois poids strictement positifs α_1, α_2 et α_3, permettant de donner plus ou moins d'importance aux trois contraintes susmentionnées, nous obtenons la définition suivante :

$$c'_{ij} = \alpha_1 \max\{0, |CN_j| - L_j + 1\} + \alpha_2 (p - P_{ij}) + \alpha_3 c_{ij}$$

$$\text{où} \quad c_{ij} = \begin{cases} \mathrm{MIN}_{T(i)} - \Delta_i & \text{si} \quad T_j \neq T(i) \quad \text{et} \quad \Delta_i < \mathrm{MIN}_{T(i)} \\ \Delta_i - \mathrm{MAX}_{T(i)} + 1 & \text{si} \quad T_j = T(i) \quad \text{et} \quad \Delta_i \geq \mathrm{MAX}_{T(i)} \\ 0 & \text{sinon.} \end{cases}$$

Le problème à résoudre peut alors être formulé de la manière suivante :

min	$\sum_{C_i \in I'} \sum_{T_j \in J_i} c'_{ij} x_{ij}$		
sc	$\sum_{T_j \in J_i} x_{ij} \leq 1$	$C_i \in I'$	(a)
	$\sum_{i \mid T_j \in J_i} x_{ij} \leq F_j$	$1 \leq j \leq t$	(f)
	$x_{ij} \in \{0, 1\}$	$C_i \in I', T_j \in J_i$	

$$\text{où} \quad x_{ij} = \begin{cases} 1 & \text{si le collaborateur } C_i \in I' \text{ est affecté à la tâche } T_j \in J_i \\ 0 & \text{sinon.} \end{cases}$$

Comme à la section 5.1, ce problème peut facilement être résolu en utilisant une méthode de recherche d'un flot maximum dans un réseau. Nous décrivons brièvement ce réseau.

- On associe tout d'abord un nœud, noté C_i, à chaque collaborateur $C_i \in I'$ ainsi qu'un nœud noté T_j à chaque tâche $T_j \in T$.
- Entre tout collaborateur C_i et toute tâche $T_j \in J_i$, nous créons un arc de coût c'_{ij} et de capacité 1.
- Nous rajoutons une source s reliée à chaque nœud C_i par un arc de coût 0 et de capacité 1.
- Finalement nous rajoutons un puits p et relions chaque nœud T_j à p par un arc de coût 0 et de capacité F_j.

En reprenant l'exemple de la section précédente, nous remarquons que le seul collaborateur pouvant être placé en formation est le collaborateur C_4. Comme il est compétent pour la tâche T_3 uniquement, il peut être mis en formation sur la tâche T_1 ou T_2. En supposant que $F_1 = F_2 = 2$, on obtient le réseau de la figure 2.

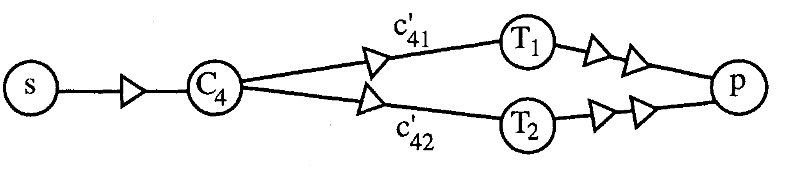

Figure 2. – Réseau associé au problème de la recherche d'une affectation satisfaisant la sécurité à long terme.

5.3 Détermination d'une date de changement d'affectation

Dans les deux sections précédentes, nous avons montré comment il est possible de déterminer une affectation des collaborateurs aux tâches en satisfaisant les sécurités à court et à long terme. Une telle affectation est valable dès l'instant D_k et il s'agit maintenant de déterminer la date à laquelle l'affectation en question va devenir caduque. La politique que nous avons choisie consiste à conserver une affectation tant qu'aucune des contraintes (d), (e) ou (g) n'est violée. Pour calculer la date $D_k + 1$ à laquelle une modification d'affectation devra obligatoirement avoir lieu, nous déterminons donc le minimum parmi les six dates suivantes :

- la date de début de la prochaine absence fixe d'un collaborateur;
- la date de fin d'une absence fixe d'un collaborateur;
- la date du prochain dépassement du nombre maximum de jours consécutifs sur une tâche;
- la date de début d'une affectation obligatoire d'un collaborateurs sur une tâche, pour autant que cette tâche n'est pas celle à laquelle il est actuellement affecté;
- la date de fin d'une affectation obligatoire d'un collaborateur sur une tâche T_j, pour autant que la durée totale que ce collaborateur a passé sur T_j dépasse le temps maximum pouvant être passé sur cette tâche;
- la date D_{fin} de fin de planification.

On pourrait évidemment trouver d'autres critères de changement d'affectation, par exemple en considérant la date à laquelle un collaborateur aura passé $(\text{MIN}_j + \text{MAX}_j)/2$ jours consécutifs sur une tâche T_j.

Pour l'exemple traité dans les sections précédentes, en supposant qu'aucun collaborateur n'est absent entre les dates D_{init} et D_{fin} et que le planificateur n'a émis aucun vœu particulier d'affectation d'un collaborateur à une tâche, nous obtenons $D_{k+1} = \min\{D_{\text{fin}}, D_k + 18\}$ car il reste encore 18 jours au collaborateur C_5 avant que celui-ci ait passé $\text{MAX}_3 = 35$ jours sur la tâche T_3.

Nous déterminons ensuite une nouvelle affectation (grâce aux techniques décrites aux Sects. 5.1 et 5.2) qui sera valable de la date D_{k+1} jusqu'à une date D_{k+2} qu'il s'agit de calculer. Ce processus est répété jusqu'à ce que la date D_{fin} de fin de planification est atteinte.

6. RÉAJUSTEMENT DE LA PLANIFICATION

Lorsqu'on désire déterminer une nouvelle affectation A_k à une date D_k, il se peut que la seule façon d'assurer la sécurité à court terme nécessite la violation de la contrainte (g) portant sur le respect du nombre minimal ou maximal de jours consécutifs passés sur une tâche. Dans un tel cas, le flot maximum décrit à la section 5.1 est nécessairement de coût strictement positif. Nous montrons dans cette section comment on peut espérer remédier à un tel problème en réajustant les dates de certaines affectations. À cette fin, nous utilisons la méthode *potentiel-tâche* telle qu'elle est présentée dans [Roy70, Cor90].

À l'ensemble $T = \{T_1, ..., T_t\}$ des tâches, nous ajoutons une tâche T_{t+1} correspondant aux absences fixes des collaborateurs. En d'autres termes, lorsqu'un collaborateur a prévu une absence à une certaine période, nous dirons qu'il est affecté à la tâche T_{t+1} durant cette période. Nous construisons un graphe $G = (V, E)$ de la manière suivante.

Ensemble V de sommets

À chaque changement d'affectation d'un collaborateur, nous associons un sommet, noté (i, j_1, j_2, r), indiquant que le collaborateur C_i passe de la tâche T_{j_1} à la tâche T_{j_2} à la date D_r. De plus, nous rajoutons un sommet source s correspondant à la date D_{init} de début de planification, ainsi qu'un sommet p correspondant à la date D_k.

Arcs liés à la contrainte (g)

Pour tenir compte de la contrainte (g) sur les durées minimum et maximum de travail sur une tâche, nous plaçons des arcs « aller et retour » entre les sommets correspondant à deux changements d'affectation consécutifs d'un même collaborateur. Soient (i, j_1, j_2, r) et (i, j_2, j_3, s) deux de ces sommets. En d'autres termes, C_i passe de la tâche T_{j_1} à la tâche T_{j_2} à la date D_r, et reste sur T_{j_2} jusqu'à D_s, date à laquelle il est affecté à la tâche T_{j_3}. Nous associons un poids de valeur MIN_{j_2} à l'arc allant de (i, j_1, j_2, r) à (i, j_2, j_3, s), et un poids de valeur $-\mathrm{MAX}_{j_2}$ à l'arc de sens opposé. Ces poids indiquent que le collaborateur C_i ne peut quitter T_{j_2} pour se consacrer à T_{j_3} qu'après MIN_{j_2} jours au moins et MAX_{j_2} jours au plus.

Arcs liés aux contraintes (b) et (c)

Afin de garantir la sécurité à court terme, certaines réaffectations doivent impérativement avoir lieu en même temps sous peine d'obtenir des solutions pour lesquelles certains postes ne sont plus assurés. De manière générale, si des collaborateurs compétents quittent la tâche T_j à une date D_r et sont remplacés par d'autres collaborateurs compétents, il est indispensable que ces modifications soient effectuées de manière synchrone. Ainsi, si la date de ces modifications est avancée, elle doit l'être pour toutes les modifications en question. Cette contrainte se traduit dans notre modèle potentiel-tâche de la manière suivante. Pour chaque date D_r $(1 \leq r \leq k)$ et chaque tâche T_j $(1 \leq j \leq t)$, nous créons un circuit composé d'arcs de poids 0 et reliant tous les sommets de type (i, j, j', r) et (i, j', j, r) tels que $T_j \in TC_i$.

Arcs tenant compte des affectations avant la date D_{init}

À la date D_{init} de début de planification, chaque collaborateur C_i est affecté à une tâche $T(i)$ depuis Δ_i jours (cf. sect. 5.1). Le sommet source s doit donc être relié à tous les sommets de type (i, j, j', r) tel que D_r est la première date à partir de D_{init} à laquelle C_i est affecté à une tâche autre que $T(i)$. Nous introduisons un arc de poids $\text{MIN}_j - \Delta_i$ reliant la source à un tel sommet, et un arc de poids $\Delta_i - \text{MAX}_j$ dans l'autre sens. De tels arcs permettent de tenir compte de l'historique des affectations avant la date D_{init}.

Arcs liés aux contraintes (d) et (e)

Certaines affectations nous sont imposées par les contraintes (d) et (e). Pour en tenir compte, nous rajoutons quelques arcs dans le graphe décrit ci-dessus. Considérons un collaborateur C_i devant obligatoirement être affecté à une tâche T_j entre les dates δ_1 et δ_2. Si $D_{\text{init}} \leq \delta_1 \leq D_k$ et $T(i) \neq T_j$, alors il existe nécessairement une date D_r à partir de laquelle C_i est affecté de manière ininterrompue sur la tâche T_j jusqu'à une date supérieure ou égale à δ_2. Soit $(i, j' j, r)$ le sommet du graphe associé à ce début d'affectation sur T_j. Nous relions ce sommet à la source s par un arc de poids $D_{\text{init}} - \delta_1$. Nous imposons ainsi que cette affectation doit avoir lieu au plus tard à la date δ_1. Si C_i ne reste pas sur T_j jusqu'à la date D_k, il existe alors un sommet (i, j, j'', r') tel que $\delta_2 \leq D_{r'} < D_k$ indiquant que C_i est affecté à une nouvelle tâche $T_{j''}$ à partir de la date $D_{r'}$. Afin d'imposer que C_i reste sur T_j jusqu'à la date δ_2, nous relions la source au sommet (i, j, j'', r) par un arc de poids δ_2-D_{init}.

Il est important de mentionner que lorsqu'un collaborateur doit obligatoirement être affecté à une tâche T_j entre les dates δ_1 et δ_2 alors que $\text{MAX}_j < \delta_2 - \delta_1$, on retire l'arc de poids $-\text{MAX}_j$ reliant (i, j, j'', r') à (i, j', j, r) car les contraintes (d) et (e) sont plus fortes que la contrainte (g) que nous devons obligatoirement violer dans ce cas.

Arcs limitant l'horizon de planification à la date D_k

Nous relions au puits p chaque sommet associé au dernier changement d'affectation d'un collaborateur avant la date D_k. Ces arcs sont de poids zéro.

Nous n'avons pas tenu compte jusqu'ici des absences fixes ou des affectations obligatoires prévues après la date D_k. Ainsi, si nous proposons de nouvelles dates postérieures à D_k pour l'affectation d'un collaborateur C_i à une tâche T_j, rien ne nous assure que C_i n'est pas absent ou qu'il ne doive pas obligatoirement être affecté à une autre tâche durant cette période. Pour éviter ce problème, nous imposons que tous les changements d'affectation représentés dans le graphe doivent avoir lieu au plus tard à l'instant D_k. Pour ce faire, nous rajoutons un arc aller-retour entre la source et le puits. L'arc allant de s à p est de poids $D_k - D_{\text{init}}$, et l'arc retour est de poids $D_{\text{init}} - D_k$. La contrainte (a) est implicitement prise en compte dans la structure du graphe, alors que les contraintes (f), (h) et (i) n'apparaissent pas car elles sont liées à la formation du personnel.

La méthode potentiel-tâche [Roy70, Cor90] nous assure que si le graphe G décrit ci-dessus ne contient pas de circuit de longueur positive, alors nous pouvons déterminer une solution satisfaisant toutes les contraintes en planifiant par exemple chaque changement d'affectation (i, j, j', r) à sa date « au plus tard », c'est à dire à la date $D_k - D_{\text{init}} - L$, où L est la longueur du plus long chemin reliant le sommet (i, j, j', r) au puits. La recherche des plus longs chemins des sommets au puits peut par exemple être effectuée à l'aide de l'algorithme de Bellman [Bel58]. Pour plus de détails sur ce point, le lecteur est invité à consulter [Roy70, Cor90].

À titre d'illustration, considérons l'exemple décrit à la section 5.1. et supposons que le collaborateur C_4 doive s'absenter 53 jours après la date D_{init}. En appliquant les techniques décrites aux sections 5.1, 5.2 et 5.3, on pourrait par exemple obtenir la planification décrite dans le tableau 2, valable jusqu'à la date $D_{\text{init}} + 53$.

TABLEAU 2
Planification violant la contrainte (g).

	$[D_0 - D_1[$	$[D_1 - D_2[$	$[D_2 - D_3[$	$[D_3 - \ldots[$
	de 0 à 19 jours après D_{init}	de 20 à 49 jours après D_{init}	de 50 à 52 jours après D_{init}	dès le 53ᵉ jour après D_{init}
C_1	T_1	$\underline{T}_2$	T_1	T_1
C_2	T_1	T_2	T_1	T_1
C_3	T_2	T_1	T_2	T_2
C_4	$\underline{T}_1$	T_3	T_3	Absent
C_5	T_3	T_1	$\underline{T}_2$	T_3

Les affectations soulignées (par exemple $\underline{T}_2$) représentent les affectations des collaborateurs en formation. Par exemple, le collaborateur C_4 est en formation sur la tâche T_1 les 19 premiers jours de la planification.

On remarque que l'affectation valable dès le 53ᵉ jour après D_{init} nous oblige à changer le collaborateur C_5 de la tâche T_2 à la tâche T_3 alors que ce dernier n'a pas encore effectué $\text{MIN}_2 = 15$ jours consécutifs sur cette tâche. Il y a donc violation de la contrainte (g). Pour remédier à cela, nous construisons le graphe représenté dans la figure 3.

À la date D_1, on devrait en fait avoir 3 circuits, l'un reliant les collaborateurs C_1, C_2, C_3 et C_5 pour la tâche T_1, un autre reliant C_2 et C_3 pour la tâche T_2 et un troisième reliant C_4 et C_5 pour la tâche T_3. Ces trois circuits peuvent en fait être ramenés à un seul circuit imposant que tous les changements d'affectation ayant lieu en D_1 doivent être effectués simultanément.

En planifiant chaque changement d'affectation le plus tard possible, nous imposons une nouvelle solution où seule la date D_2 est modifiée. En effet, le plus long chemin du sommet (1,1,2,1) au puits est de longueur 33 alors que celui du sommet (5,1,2,2) au puits est de longueur 15. Ceci nous impose donc que $D_1 - D_{\text{init}} = 53 - 33 = 20$ et $D_2 - D_{\text{init}} = 53 - 15 = 38$. La solution réajustée est représentée dans le tableau 3.

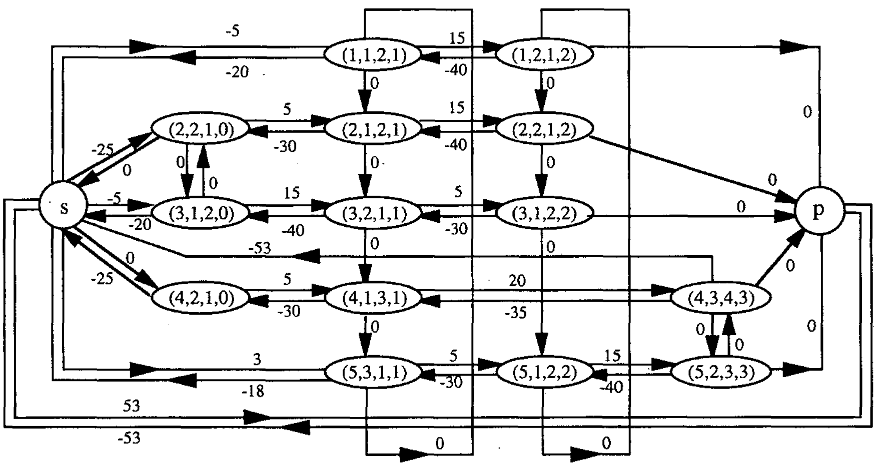

Figure 3. – Exemple de réseau potentiels-tâches pour le réajustement de la planification.

Il peut malheureusement exister des circuits de longueur positive dans le graphe que nous avons construit. Si tel est le cas, il est alors impossible de respecter toutes les contraintes (a), ... ,(e) et (g). Nous nous permettons dans un tel cas de violer la contrainte faible (g). Soit C un tel circuit de longueur positive et soient $(i, j'j, r)$ et (i, j, j'', r') deux sommets consécutifs sur ce circuit, choisis de manière quelconque. Nous remplaçons les poids $\text{MIN}_{j'}$ et $-\text{MAX}_{j'}$ de l'arc aller-retour entre ces deux sommets par les poids

TABLEAU 3
Planification réajustée.

	$[D_0 - D_1[$ de 0 à 19 jours après D_{init}	$[D_1 - D_2[$ de 20 à 37 jours après D_{init}	$[D_2 - D_3[$ de 38 à 52 jours après D_{init}	$[D_3 - \ldots[$ dès le 53e jour après D_{init}
C_1	T_1	$\underline{T}_2$	T_1	T_1
C_2	T_1	T_2	T_1	T_1
C_3	T_2	T_1	T_2	T_2
C_4	$\underline{T}_1$	T_3	T_3	Absent
C_5	T_3	T_1	$\underline{T}_2$	T_3

$D_{r'} - D_r$ et $D_r - D_{r'}$ respectivement. Nous imposons donc que la durée de l'affectation de C_i sur T_j soit de durée exactement égale à celle prévue dans la solution que nous tentons de réajuster.

Nous recherchons alors à nouveau les longueurs des plus longs chemins entre les sommets du graphe et le puits. Si nous devions détecter un nouveau circuit de longueur positive, il suffirait alors d'appliquer à nouveau la procédure décrite ci- dessus. En répétant ce processus, nous finirons par obtenir un graphe sans circuit de longueur positive puisque nous nous rapprochons pas à pas de la solution qu'on essaie de réajuster.

Le processus de réajustement que nous avons adopté peut donc être résumé de la manière suivante.

Algorithme de réajustement

(1) Construire le graphe G potentiel- tâche associé à la situation demandant réajustement.
(2) **Si** G ne contient aucun circuit de longueur positive
Alors
Planifier chaque changement d'affectation au plus tard. STOP.
(3) **Sinon**
Soit C un circuit de longueur positive dans G, et soient (i, j', j, r) et (i, j, j'', r) deux sommets consécutifs sur ce circuit.
Imposer dans G que le collaborateur C_i soit affecté sur T_j entre les dates D_r et $D_{r'}$.
Retourner à (2).

7. QUELQUES REMARQUES COMPLÉMENTAIRES

7.1 Absences imprévues de courte durée

Les absences de courte durée n'entraînent généralement que peu de modifications de l'affectation courante. Dans le meilleur des cas, il est possible de choisir un collaborateur compétent et en formation pour l'assigner à la place de la personne manquante.

Il peut cependant arriver qu'aucun collaborateur ayant les compétences pour remplacer la personne absente soit en formation. Dans ce cas, il faut effectuer une réaffectation. Celle-ci peut être facilement déterminée en résolvant un problème de flot similaire à celui décrit dans la section 5.1. Lors d'une telle réaffectation de courte durée, la seule contrainte importante est d'assurer la sécurité à court terme tout en essayant de modifier le moins possible l'affectation courante. Pour cela, nous mettons un faible coût (par exemple nul) sur les arcs reliant chaque collaborateur à la tâche à laquelle il était actuellement affecté, et un coût plus important (strictement positif) pour les autres tâches pour lesquelles il est compétent. Dans de telles situations, nous faisons donc abstraction des contraintes faibles. Évidemment, ce type de décision doit être entériné par le planificateur lui-même.

7.2 Modification des niveaux de compétence

À la section 5.3, nous avons décrit la façon de déterminer la date D_{k+1} à partir de laquelle une affectation A_k devient caduque. Cette date D_{k+1} est définie comme étant égale au minimum d'un ensemble de six dates. Lorsqu'un collaborateur est mis en formation sur une tâche à partir d'une date D_k, il se peut qu'il ne lui restait que peu de jours à passer sur cette tâche avant d'atteindre le niveau de compétence k_0. Nous aurions ainsi pu considérer une septième date pour la détermination de D_{k+1} : la date à laquelle un collaborateur en formation sur une tâche devient compétent pour cette tâche.

À l'inverse, lors d'une phase de réajustement (*cf.* Sect. 6), il se peut qu'une date de changement d'affectation soit avancée dans le temps et qu'un collaborateur ne soit alors plus compétent pour une tâche à laquelle il est affecté pour assurer la sécurité à court terme. L'exemple ci-après donne une illustration du problème.

Considérons le cas d'un collaborateur C_i compétent pour la tâche T_1 mais n'ayant que le niveau $k_0 - 1$ pour la tâche T_2. Supposons qu'il ne manque à C_i que 60 jours pour devenir compétent sur T_2. Supposons de plus que $\text{MAX}_1 = \text{MAX}_2 = 60$, $\text{MIN}_1 = 50$ et $\text{MIN}_2 = 40$. Considérons les affectations décrites dans le tableau 4.

Les 60 premiers jours de la planification, C_i est mis en formation sur T_2. Il devient ainsi compétent pour cette tâche et peut y être affecté à partir de la date D_3 pour assurer la sécurité à court terme.

Supposons maintenant que C_i doivent s'absenter au 140^e jour après D_{init}. La contrainte (g) sera donc violée si nous conservons ces affectations puisque C_i n'aura passé que 30 jours consécutifs sur T_2. En appliquant la technique de réajustement décrite à la section 6, nous modifions les affectations et obtenons la solution décrite dans le tableau 5.

TABLEAU 4

	$[D_0 - D_1[$	$[D_1 - D_2[$	$[D_3 - \dots[$
	de 0 à 59 jours après D_{init}	de 60 à 109 jours après D_{init}	dès le 110^e jour après D_{init}
C_i	$\underline{T}_2$	T_1	T_2

TABLEAU 5

	$[D_0 - D_1[$	$[D_1 - D_2[$	$[D_3 - \dots[$
	de 0 à 49 jours après D_{init}	de 50 à 99 jours après D_{init}	dès le 100^e jour après D_{init}
C_i	$\underline{T}_2$	T_1	T_2

La contrainte (g) est bien vérifiée dans ce cas. Cependant, entre le 100^e et et 110^e jour après D_{init}, C_i n'est pas compétent pour T_2 puisqu'il n'a accompli que 50 jours de formation au lieu des 60 requis. La méthode potentiel-tâche de la section 6 ne peut malheureusement pas prévenir ce genre de problèmes. En effet, nous n'imposons que des bornes supérieures et inférieures sur le nombre de jours qu'un collaborateur passe sur une tâche et nous n'avons donc aucune maîtrise sur le nombre exact de jours qu'il passera sur cette tâche. Notre stratégie consiste à inviter le planificateur à refaire très périodiquement des simulations, et à n'utiliser qu'une partie de ces résultats. En agissant de cette manière, on donne plus d'importance à la sensibilité du planificateur. Cela permet entre autres d'ajuster périodiquement le niveau des collaborateurs en fonction de leurs compétences réelles, ce que le programme ne saurait faire.

8. REMARQUES FINALES ET CONCLUSIONS

La méthode présentée dans cet article a été développée et testée pour la gestion du personnel dans un important établissement bancaire helvétique. Un programme a été réalisé dans le langage C++, sur un ordinateur PC 386-33 MHz sous le système d'exploitation Windows. Grâce à une interface utilisateur conviviale, le planificateur est à même de tester rapidement le jeu de données qu'il propose. À titre d'exemple, une planification sur 18 mois des emplois du temps d'une vingtaine d'employés nécessite un temps de calcul de l'ordre de une à deux minutes. La qualité des solutions obtenues ainsi que la possibilité d'apprécier à long terme la politique mise en place aujourd'hui ont séduit plus d'un planificateur.

Le but de notre méthode est avant tout de fournir un outil d'aide à la décision qui soit simple à utiliser et qui tienne compte le mieux possible des contraintes du problème. Le rôle d'un tel logiciel n'est pas tant d'obtenir la meilleure solution possible au prix de longues heures de calcul, mais plutôt de permettre à l'utilisateur de disposer d'un simulateur rapide et performant, lui permettant de tester plusieurs jeux de données et plusieurs stratégies. Étant donné le nombre et la diversité des contraintes à prendre en compte dans la réalité, il est illusoire de vouloir toutes les traiter à la fois. D'ailleurs, une grande partie de ces contraintes sont difficilement formulables, et dépendent en droite ligne de facteurs psychologiques liés à la personnalité du planificateur. De même, l'application stricte d'une planification à long terme nous semble relever de l'utopie. L'homme n'étant pas une machine, la décision finale ne doit donc pas appartenir à l'ordinateur.

Notre approche par une série de simulations successives correspond tout à fait à l'approche habituelle des planificateurs. Ces derniers planifient en principe à court terme, tout en réfléchissant sur le long terme. Ainsi, le responsable d'une unité va effectuer plusieurs simulations sur les trois ou quatre prochains mois, tout en sachant pertinemment qu'il ne conservera que les trois premières semaines de la simulation. L'avantage pour lui est de comprendre comment son équipe va évoluer sur le long terme, si cette dernière est suffisamment étoffée, quelles sont les tâches qui risquent de poser problème dans un avenir proche, etc. Bref, il va acquérir une certaine sensibilité. Il est aussi important pour lui de répéter relativement souvent ces simulations de manière à ce que les données fournies soient le plus proche possible de la réalité.

Notre approche a donc été développée dans cette optique. Ses deux atouts sont la vitesse de simulation et les bons résultats obtenus. Évidemment, chaque médaille a son revers. La vitesse est surtout due au fait qu'il s'agit d'une approche de type glouton, c'est-à-dire que nous ne remettons jamais en cause les affectations que nous avons effectuées auparavant. Tout au plus, nous pouvons modifier après coup les dates auxquelles ces dernières ont lieu.

Un autre problème réside dans la relative « myopie » dont souffre notre méthode. En effet, notre vision de l'avenir est basée sur l'immédiat, c'est-à-dire que nous n'intégrons pas les événements futurs, même s'ils sont inéluctables. Par exemple, il ne vaut vraiment pas la peine de changer quelqu'un de tâche s'il part un jour après en congé. Dans ce cas particulier, il vaut mieux directement nommer son successeur sur la tâche un jour avant. Ceci n'est pas forcément facile puisque soumis à interprétation. Quel est le nombre minimum de jours à effectuer depuis une affectation avant le début d'une absence fixe ? Est-on prêt à violer une autre contrainte pour cela, par exemple celle limitant la durée d'une affectation sur une tâche ? Ces questions n'amènent pas de réponses toutes faites. La politique correcte va dépendre du type de travail, de l'avis du planificateur, ainsi que du personnel. Il est important de laisser le planificateur agir à sa guise dans ce genre de situations en proposant, et c'est ce que nous avons fait, une série de facilités pour bloquer, rallonger, ou reprendre une simulation à partir d'un point précis. De même, nous suggérons de laisser le planificateur régler des cas particuliers tels que des collaborateurs « trop » compétents qui ne seraient plus jamais placés en formation, même s'ils ont des lacunes importantes sur quelques tâches minoritaires.

En résumé, nous avons développé un outil de gestion et formation du personnel avec comme principal objectif d'adapter l'outil à l'homme plutôt que l'inverse, comme c'est souvent l'usage pour ce genre de problèmes.

RÉFÉRENCES

[Bel58] R. E. BELLMAN, *On a routing Problem*, Quart. Appl. Math. 1958, *16*, p. 87-90.

[Boi95] P. BOIZUMAULT, Y. DELON et L. PÉRIDY, *Constraint Logic Programming for Examination Timetabling*, J. Logic Programming, 1995, p. 1-17.

[Cor90] T. H. CORMEN, C. E. LEISERSON et R. L. RIVEST, *Introduction to Algorithms*, MIT Press, Cambridge, MA, USA, 1990.

[For62] L. R. FORD et D. R. FULKERSON, *Flows in Networks*, Princeton University Press, Princeton, New Jersey, USA, 1962.

[Gon90] M. GONDRAN et M. MINOUX, *Graphes et Algorithmes*, Éditions Eyrolles, Paris, France, 1990.

[Ree93] C. R. REEVES, *Modern Heuristic Techniques for Combinatorial Optimization*, Blackwell, Oxford, 1993.

[Roy70] B. ROY, *Algèbre moderne et Théorie des Graphes*, Tome 2, Dunod, Paris, France, 1970.

[Tsa93] E. TSANG, *Foundations of Constraint Satisfaction*, Academic Press, 1993.