tel-00004380, version 1 - 29 Jan 2004

THESE

Présentée par

CHAN Yew Cheong, Peter

Pour obtenir le titre de
DOCTEUR de l'UNIVERSITE
JOSEPH FOURIER – Grenoble 1

Spécialité : Informatique
Formation doctorale : Recherche Opérationnelle

La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes

Date de soutenance : 22 Octobre 2002

Composition du Jury :

Directeur de thèse:	**M. Georges Weil**
Rapporteurs :	**M. Abdelhakim Artiba**
	M. Alain Guinet
Examinateurs:	**M. Jean Charles Pomerol**
	M. Gerd Finké
	M. Eric Jacquet-Lagrèze
	M. Jacques Demongeot

Thèse préparée au sein du Laboratoire TIMC – Institut IMAG

Table des matières

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

Liste des Figures

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

THESE

La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes.

Résumé

Objectifs

L'un des objectifs de cette thèse est la caractérisation de la problématique de la planification des ressources humaines. La planification dans les entreprises est d'une très grande complexité car elle mélange de façon intime la vie sociale et familiale et les contraintes professionnelles à la fois des dirigeants de l'entreprise, des responsables fonctionnels, des responsables de proximité et des salariés. Les volumineux travaux proposés dans ce domaine depuis de nombreuses décennies sont témoins de l'intérêt économique et social que lui accorde la communauté scientifique.

Les objectifs pratiques de la thèse sont l'analyse et l'implantation de méthodes logicielles en s'appuyant sur la Programmation Par Contraintes (PPC). Ces travaux comprennent deux projets de collaboration entre l'Université Joseph Fourier et l'industrie : Gymnaste (avec COSYTEC S.A.) et Equitime (avec EQUITIME S.A.). Un troisième travail MOSAR a été réalisé pour le compte du Ministère de la Justice. Par rapport aux méthodes usuelles, ces travaux permettent de tenir compte des contraintes et préférences individuelles, et de tenir compte des connaissances que détiennent les utilisateurs en leur permettant de participer activement à la planification.

La thèse a aussi permis de nombreuses contributions scientifiques autour du thème Modèle Multi-Niveaux. Or, la législation du travail en vigueur inclut diverses restrictions au niveau journalier, hebdomadaire, mensuel et annuel. Il faut des modèles et méthodes nouvelles pour produire des plannings légaux.

Motivation

Les ressources humaines se trouvent à l'origine du succès de toute entreprise, à partir de 2 personnes. Même à l'échelle d'un pays, les dirigeants de Singapour ont signalé dès les années 1970 que cette nation-île n'a aucune ressource si ce n'est les hommes et femmes qui la constituent. Ce renseignement qui a marqué ma jeunesse, retrouve son écho tous les jours dans la vie professionnelle que je mène depuis plus de dix ans.

Remerciements

J'ai résolu mon premier problème d'affectation du personnel (les conducteurs de métro …) en 1988, lorsque j'étais Consultant en Intelligence Artificielle chez BULL détaché à Singapour. Depuis, j'ai été impliqué dans des projets de planning du personnel pour les surveillants au Grand Louvre, pour les opérationnels à la Bibliothèque F. Mitterrand, pour les gardiens de prisons au Ministère de la Justice, pour les différents corps de métier à Radio France Outre-Mers, pour le personnel de santé (hôpitaux et cliniques) et tout dernièrement pour les sapeurs-pompiers en Rhône-Alpes.

J'ai enfin décidé de publier un mémoire de recherche sur ce thème de prédilection, grâce à l'appui de Georges Weil, qui est devenu mon Directeur de Thèse. Je tiens à le remercier tout spécialement pour m'avoir guidé au cours de ces années de recherche,

malgré le peu de temps que lui laisse le management des affaires de la société qu'il a fondée et qu'il dirige.

Je tiens à remercier les professeurs Artiba et Guinet d'avoir accepté la lourde charge d'être rapporteurs de cette thèse. Je remercie également le jury pour le crédit qu'ils accordent à ce travail.

Je souhaite remercier M. le Professeur Jacques Demongeot de m'avoir accueilli au sein du laboratoire TIMC et m'avoir permis de réaliser cette thèse dans d'excellentes conditions.

Je n'oublierai pas les étudiants et d'autres salariés de la société EQUITIME qui m'ont aidés dans mon travail de recherche. Notamment Rémy Joseph, Soizic Adam et Tahar Zemmouri.

Plan du mémoire

Le mémoire est divisé en trois parties : la première sert d'introduction à la planification avec les chapitres 1 et 2, la deuxième présente les travaux de recherches au chapitres 3, 4 et 5, et la dernière contient la conclusion chapitre 6 et divers annexes.

L'introduction au chapitre 1, met en scène la problématique de la planification des ressources humaines et sa complexité au quotidien dans les entreprises. Il permet de mieux comprendre les différents objectifs de planification. Ensuite, les objectifs de ce travail seront abordés en détail.

Le chapitre 2 proposera un résumé de l'état de l'art en matière de planification, avec un inventaire des modèles et méthodes utilisés au cours des 50 dernières années. On soulignera les avantages et les limites de chaque méthode.

Nous présenterons Gymnaste, un système de génération automatique de planning acyclique au chapitre 3. Ce travail mettra en évidence les modèles individuels en PPC et la résolution via les contraintes globales du système CHIP. Ensuite, nous présenterons Equitime un produit industriel s'appuyant sur des spécifications analogues, mais utilisant un solveur « maison » réalisé en Visual Basic. L'aspect dynamique de la planification sera illustré par l'interface homme-machine, permettant de mêler des pratiques de génération à base de cycles, la génération acycliques et les modifications manuelles.

Le chapitre 4 présentera le système de déroulement automatique des cycles de travail : MOSAR. Afin de rendre les plannings plus humains, ce système permet de relaxer les contraintes de cycle autour des congés annuels, tout en conservant des traits de base du cycle et en respectant les contraintes spécifiées par les utilisateurs. Divers modèles originaux ont été mis conçus et mis en œuvre avec les contraintes globales.

Au chapitre 5, nous présenterons un nouveau type de modèle intégrant plusieurs multi-niveaux d'agrégations simultanément. Effectivement, la législation française agit à plusieurs échelles de temps (journalier, hebdomadaire, mensuel et annuel). Nous présenterons les grandes lignes de cette méthode ainsi que les premiers résultats.

tel-00004380, version 1 - 29 Jan 2004

La planification du personnel

Le chapitre 6 conclut la thèse et présente des indications des suites possibles de travaux. Parmi les annexes, on compte un glossaire, une bibliographie, et organisés par auteur un index des citations et liste des publications citées.

tel-00004380, version 1 - 29 Jan 2004

PREMIERE PARTIE : INTRODUCTION

Cette partie présente l'introduction au mémoire de thèse. Elle se compose des chapitres suivants :

1. La problématique de la planification :
   Le première chapitre présente la problématique analysée, ainsi que nos propositions pour le travail de recherche en le justifiant.
   a. Qu'est-ce que la planification
   b. Qu'est-ce qu'un bon planning
   c. Le contexte statique de la planification
   d. La dynamique de la planification
   e. Nos propositions

2. L'état de l'art : Le deuxième chapitre présente la très riche documentation sur les techniques appliquées dans ce domaine.
   a. La construction des horaires journaliers par PLNE
   b. La construction des horaires journaliers par PPC
   c. La construction de tours acycliques en PLNE
   d. La construction de tours acycliques en PPC
   e. La construction de tours cycliques
   f. Les méthodes de recherche stochastique
   g. Les algorithmes approchés
   h. Les Systèmes interactifs d'aide à la décision
   i. La Courbe de charge et dimensionnement des équipes

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

1 LA PROBLEMATIQUE DE LA PLANIFICATION

Résumé

Ce chapitre est consacré à la présentation de la problématique « planification » d'une part, et d'autre part à la présentation du travail de cette thèse.

Pour mieux cerner ce qu'est la planification ainsi que la complexité inhérente à sa réalisation, nous commençons par donner les différentes définitions sous plusieurs formes. Ensuite nous élargissons les définitions pour traiter le contexte statique autour de la planification, et enfin ses aspects dynamiques dans l'élaboration des plannings.

Nous présentons les objectifs de cette thèse dans le domaine de la planification, ainsi que nos propositions pour résoudre le problème de production d'un planning.

1.1. QU'EST-CE QUE LA PLANIFICATION ?

Afin de rester pérenne dans l'économie globale moderne, toute entreprise doit organiser et planifier le travail de ses salariés. Cela passe par la détermination des capacités de tout un chacun, par le recensement des activités futures et des besoins en personnel. L'entreprise doit satisfaire ces derniers en affectant la bonne personne, à la bonne place au bon moment. Cela sous-entend la satisfaction simultanée des aspects « **JuSTE »** :

- **Ju**ridique : la législation française en matière de droit du travail (durées de travail et de repos) sur différents horizons de temps (journalier, hebdomadaire, mensuel et annuel)
- **S**ocial : répartition équitable des tâches entre salariés, entre hommes et femmes, avec respect des indisponibilités, préférences individuelles et autres souhaits des salariés. Répartition équitable du temps de travail et du repos.
- **T**echnique : les règlements des différents métiers de l'entreprise (prise en compte des compétences et des niveaux requis)
- **E**conomique : Respect des besoins de l'entreprise à chaque moment de l'horizon de planification. Cela se présente comme la meilleure adaptation de l'énergie disponible aux charges à chaque moment de l'horizon. On cherche à ne pas dépenser inutilement cette énergie. Les coûts salariaux peuvent représenter jusqu'à 70% du budget opérationnel dans un établissement de soins [WHC+98].

Afin d'appréhender concrètement la problématique de la planification, voici plusieurs définitions.

Définition opérationnelle de la planification

Pour le planificateur professionnel, la planification est un processus global décrit par le schéma SADT suivant :

tel-00004380, version 1 - 29 Jan 2004

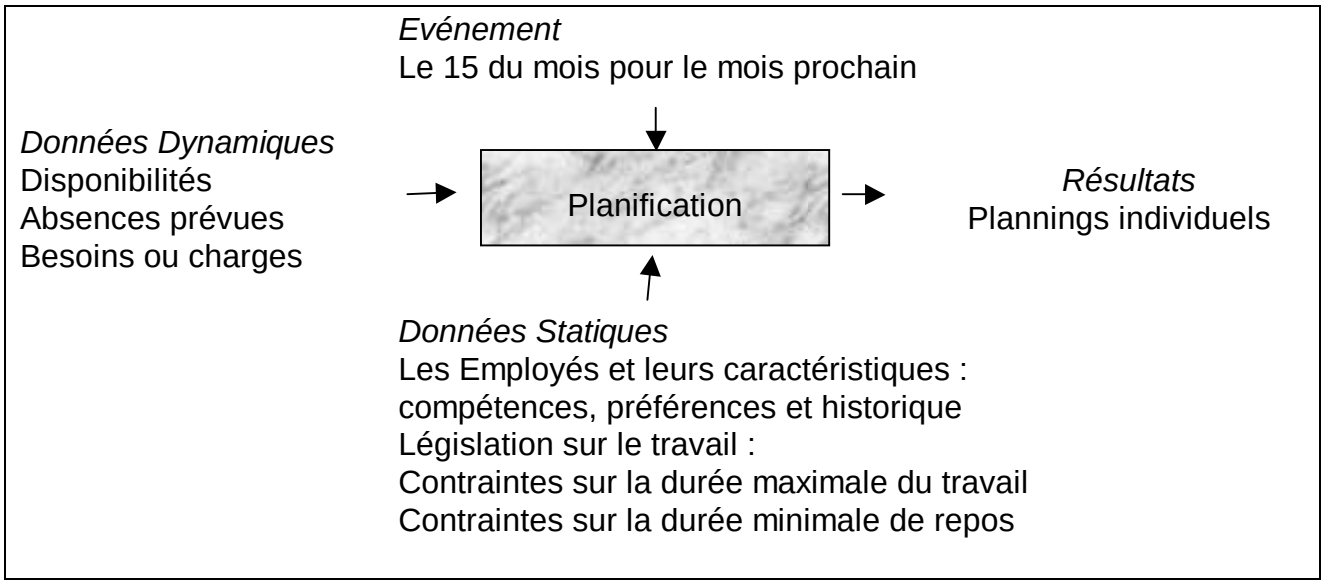


Figure 1.1. Schéma SADT A0 illustrant le problème de la planification

La planification vise à affecter les ressources humaines pour chaque intervalle de temps sur un horizon donné, de telle sorte que les besoins par intervalle soient couverts et que les différentes contraintes soient satisfaites.

Définition mathématique de la planification

La planification peut être définie sous la forme de fonction mathématique qui associe un salarié et un intervalle de temps à une affectation. Voici plusieurs exemples de fonctions de planification à des intervalles de temps de taille différente et d'affectations différentes, compte tenu de la législation en vigueur (Loi Aubry) :

Tableau 1.2. Définitions de différents niveaux de planning.

Au niveau journalier Période = 15, 30 ou 60 minutes	f_j (Salarié, Période) = Activité/Compétence Sur la journée, cette fonction permet de déterminer les heures de présence du salarié (ce qui définit sa vacation) et ses activités de la journée
Au niveau mensuel, Période = 1 jour	f_m (Salarié, Jour) = Vacation Pour le mois (ou la semaine), cette fonction permet de déterminer la suite des vacations du salarié pour la semaine ou le mois
Au niveau annuel Période = 1 semaine	f_a (Salarié, Semaine) = Nombre d'heures de travail Cette fonction permet de déterminer le nombre d'heures travaillées sur l'année

Le tableau ci-dessus sera complété des formules représentant des contraintes au chapitre 5. Comme le décrit [Par98], on remarque la présence de plusieurs modèles distincts à diverses granularité temporelles. Le niveau journalier correspond au problème de construction de vacations et le niveau mensuel à celui de grilles.

tel-00004380, version 1 - 29 Jan 2004

1.2. QU'EST-CE QU'UN BON PLANNING ?

Si un planning est facile à obtenir, un *bon* planning l'est beaucoup moins. S'agissant des personnes, le coût d'un planning n'est pas la somme des coûts horaires des employés. En effet, la planification est rendue beaucoup plus complexe avec les contraintes et souhaits individuels. La satisfaction des salariés est un facteur important sur leur motivation et leur productivité. Il faut aussi tenir en compte les coûts indirects : la formation, les délais d'embauche ou de licenciement, la santé et la sécurité des salariés.

Compte tenu des différents points de vue au sein d'une même entreprise, des contraintes complexes et parfois contradictoires, créer un bon planning implique à la fois une négociation entre le planificateur et les différents employés concernés, et un calcul d'optimisation combinatoire.

Figure 1.3. Un problème multicritère, adapté de [Par98]

Pour le *chef d'entreprise*, le bon planning permet de dimensionner la force de travail au plus juste et d'utiliser ses ressources au bon moment pour fournir le meilleur service au meilleur *coût*. Cela est synonyme de réduction des heures supplémentaires, des coûts liés aux aléas imprévus ou des contrats à durée déterminée.

Figure 1.4. Un bon planning résulte souvent des compromis des différents acteurs

Le *planificateur*, afin de respecter les impératifs de productivité et de demandes des clients, veut disposer d'une souplesse suffisante en main d'œuvre permettant de traiter les imprévus quotidiens.

Le *commercial* lui, exige pour son client une qualité de service irréprochable dans les délais impartis.

tel-00004380, version 1 - 29 Jan 2004

La problématique

Pour le *syndicaliste*, un bon planning doit respecter sans scrupule la législation sur le temps de travail et sur le repos des employés, ainsi que les conventions collectives des différents métiers de l'entreprise.

Le *salarié* recherche une satisfaction sociale de son travail, par rapport à ses contraintes de disponibilités, par rapport à ses préférences ou par rapport à l'équité de traitement. Le bon planning permet au salarié de travailler quand il veut et limite le recours aux horaires atypiques (pour chaque salarié), lui permettant d'organiser son temps libre.

tel-00004380, version 1 - 29 Jan 2004

1.3. LE CONTEXTE STATIQUE DE LA PLANIFICATION

Pour mieux comprendre la problématique de la planification, nous envisageons dans ce paragraphe les différents aspects statiques de la planification.

1.3.1. Où faut-il planifier ?

Le problème du planning est central dans toutes les entreprises disposant des ressources humaines que ce soit dans le secteur de l'agriculture, de l'industrie ou dans le secteur tertiaire. Il est d'autant plus ardu lorsque le travail doit être continu.

- > 8H/J : le repos journalier des salariés à organiser
- > 5J/7 : le repos hebdomadaire à prévoir
- > 35H/Semaine : les congés annuels et l'annualisation à gérer

C'est le cas notamment dans les services *autonomes* suivants :

- Industrie : sidérurgie (industrie dite à feu continu)
- Services dans les hôpitaux : [WP72], [AR81], [BS93]
- Services dans les aéroports : agents au sol [JM88], [Che91], [Jaq+98]
- Services pénitenciers [CW00], casernes de pompiers, commissariats de police et gendarmerie
- Services dans les centres de loisir : Disneyland, etc.

D'autres exemples dans le domaine tertiaire :

- Agents de péage [Dan54], [BP73]
- Opérateurs téléphoniques [Seg74], [HB76], [Kei79]
- Services postaux : [Jar+94]
- Caissiers de supermarchés [Tho88]

En fait, le problème de planning apparaît *systématiquement* dans les situations suivantes :

1. Si le travail doit être assuré pendant plus d'une journée (7H12 si 36H par semaine) il faut prévoir la succession de plusieurs personnes sur le même poste dans la journée. Un outil d'aide est nécessaire lorsque le nombre de postes dépasse la quinzaine, par exemple pour gérer les absences imprévues des salariés (pour cause de maladie, accidents, etc.).

2. Si le travail doit être assuré pendant plus de 35 H par semaine, un outil automatique devient indispensable lorsque le nombre de postes dépasse la trentaine pour gérer la succession de plusieurs personnes dans la semaine, ainsi que les absences imprévues.

tel-00004380, version 1 - 29 Jan 2004

1.3.2. Planifier quoi ?

En se limitant à la planification des ressources humaines (sans prendre en considération les ressources matérielles), il faut décider si l'on planifie les horaires de présence du personnel ou les tâches effectuées par le personnel.

La planification des horaires de présence : Un planning peut être utilisé pour prévoir les horaires de présence du personnel, sans préciser les tâches journalières à effectuer. Dans certains services, on ne peut affecter qu'un certain nombre de personnes pour chaque heure quel que soit le besoin réel, pour des raisons de postes disponibles ou de budget fixe.

Cette pratique se rencontre aussi lorsque la direction souhaite cacher le détail des affectations des tâches, pour des raisons soit de sécurité (afin de déjouer toute complicité des agents avec des éléments relevant de leur poste de travail), soit pour une meilleure souplesse au jour le jour (surtout dans des petites et moyennes entreprises). Il n'existe aucune obligation légale de communiquer l'affectation des tâches au personnel. Pour chaque vacation, on connaît les besoins en nombre de personnes. L'emplacement de l'activité (guichet i, machine j, etc.) est volontairement ignoré pour réduire la complexité du calcul.

Sur un horizon de plusieurs semaines ou mois, la planification se réalise en fonction des poids horaires hebdomadaires. Effectivement, la législation française prévoit une moyenne hebdomadaire qui ne dépasse pas la limite de 38 heures sur tout horizon de 12 semaines glissantes.

La planification des tâches : Dans les entreprises à haute technicité, comportant de nombreux métiers et compétences distincts, lorsqu'un besoin non satisfait se traduit en perte d'affaires, il est souhaitable d'affecter le personnel en fonction des tâches. Cela exige d'une part une décomposition fine des opérations ou des tâches en gammes opératoires, d'autre part, le repérage des tâches que chaque personne est capable d'accomplir.

Le planning se distingue en fonction du type des tâches à planifier :

- Tâches sécables ou non : détermination des pauses et leur durée
- Tâches qui se chevauchent sur 24 H, générant un planning dit « continu ». Dans un planning dit discontinu, on pourra planifier jour par jour.
- Tâches liées à un lieu géographique ou liées à un déplacement géographique : il faut donc considérer le temps de transport ou le temps passé hors dépôt. Exemple des transporteurs par avion, bateau, train, bus ou camion.
- Tâches avec contraintes de succession entre elles, typiquement des problèmes d'ordonnancement dans un atelier.

tel-00004380, version 1 - 29 Jan 2004

1.3.3. Planifier qui ?

La planification anonyme raisonne au niveau du nombre de salariés dans des groupes homogènes identifiés par les modèles appropriés. Ainsi, il est possible que le découpage en groupes prenne en compte les différences en matière de compétences ou de contrats temps. Cette approche permet de prendre en compte un grand nombre de salariés et de profiter au maximum de la synergie entre les salariés. Cependant elle ne tient compte ni des préférences individuelles ni des plannings antérieurs.

La planification individuelle raisonne au niveau de chaque salarié, en s'appuyant sur

- les compétences individuelles
- les contrats temps par ex. plein temps, mi-temps ou autres
- les préférences individuelles pour les horaires et les durées de travail et/ou de pauses
- l'historique du planning afin d'assurer l'équité entre personnes au niveau des horaires désagréables telles que les affectations tôt le matin le lendemain d'un jour d'absence (jour férié, week-end ou congé), ou les affectations tard le soir la veille d'une absence.

Cela exige un calcul plus lourd et limite le nombre de personnes traitées par rapport à la planification anonyme.

1.3.4. L'horizon de planification

Le planning journalier est un planning avec des intervalles de 5 à 30 minutes. Ce type de planning convient à l'affectation des tâches, lorsque le planning est discontinu et lorsqu'ils sont connus de façon précise. A ce niveau, on peut gérer les pauses et déterminer qui et quand.

Le planning hebdomadaire est un planning avec des intervalles allant de 15 minutes à une heure. Ce type de planning est utilisé dans certains pays pour une paie hebdomadaire. On gère le repos hebdomadaire et durée de travail hebdomadaire. Par exemple en France dans le secteur de la santé, on dispose de quatre jours de repos sur la quinzaine travaillée, dont au moins deux consécutifs, dont un dimanche.

Le planning mensuel est un planning avec des intervalles allant d'une heure à un jour. Ce type de planning est utile pour le calcul des coûts pour les besoins de la paie. On peut gérer les jours de repos hebdomadaires, les journées RTT, la mensualisation des heures.

Le planning annuel est un planning avec typiquement des intervalles de journée ou de semaine. Ce type de planning est utile pour l'annualisation des heures de travail et permet de gérer les journées de RTT, les congés annuels ou la formation continue.

tel-00004380, version 1 - 29 Jan 2004

1.3.5. Types de plannings

Il y a deux types de plannings acyclique ou cyclique :

Un planning est acyclique s'il est différent chaque semaine. Pour construire un planning mensuel automatiquement, on dispose de règles qui permettent d'enchaîner les vacations pour une même personne.

Le planning acyclique correspond à une pratique de gestion nécessitée par la demande de service rendu. Il est typiquement utilisé dans le cas d'un besoin non satisfait , ce serait une perte, le client pourrait se diriger vers un autre fournisseur.

Un planning est cyclique si au bout d'une durée P, le salarié retrouve son planning de départ. La période P est généralement mesurée en termes de semaines par exemple 4, 6, 12, 13 ou 17 semaines. Ces cycles hebdomadaires permettent de tenir compte des week-ends. Les cycles à 13 ou à 17 semaines sont qualifiés d'annuel car il s'effectuent un nombre fixe fois par an : 4 fois 13 étant 52 et 3 fois et 17 donnant 51, la dernière semaine se planifie manuellement pour les fêtes de fin d'année. Très souvent ils intègrent des semaines de congés annuels.

Il existe aussi des cycles journaliers définis sur un nombre de jours, qui ne tiennent alors pas compte des week-ends.

Le planning cyclique correspond à une gestion dirigée par l'offre de service rendu. Il est utilisé typiquement dans le cas où le client est captif et ne peut pas consulter un autre fournisseur. En ignorant les besoins ponctuels, cette gestion met l'accent sur une approche à moyen terme. Cela peut être assimilé à la construction d'un planning acyclique sur un horizon égal à la période du cycle. Cependant, puisqu'il sera par la suite déroulé plusieurs fois par an, il faudra envisager avec soin le nombre total des heures, le rythme des journées travaillées et des journées de repos.

Le déroulement systématique du cycle ne permet pas la moindre perturbation extérieure, par exemple des pré-affectations comme les congés annuels ou les formations. Au chapitre 4, nous proposons une étude du déroulement des cycles avec relaxation de contraintes.

La planification des horaires peut être cyclique, car on ne distingue que les heures effectuées et on confond les tâches spécifiques devant avoir lieu. Par contre, la planification des tâches ne peut être qu'acyclique, sauf cas exceptionnel où les tâches sont répétitives d'une semaine sur l'autre.

tel-00004380, version 1 - 29 Jan 2004

1.4. LA DYNAMIQUE DE LA PLANIFICATION

1.4.1. Les phases de planification

Le contexte statique ne suffit pas à expliciter la complexité de la planification. Au-delà des modèles, ce paragraphe détaille la dynamique à travers les différentes phases de planification :

- *Phase Conception : processus à long terme*
  Cette phase couvre les activités dites *stratégiques*, permettant de cerner le problème et de se poser des questions sur le contexte de planification déjà présenté ci dessus § 0. Pendant cette phase, on élabore les différentes façons de travailler, les types de personnel (contrats à durée déterminée / intermittents du spectacle / pigistes) les horaires, les cycles de travail, les méthodes de prévision des besoins, etc.

- *Phase Ordonnancement : processus prévisionnel à court / moyen terme*
  L'élaboration des plannings prévisionnels se fait sur différents horizons : un an, un mois ou une semaine. Cette phase est détaillée au paragraphe 1.4.2.

- *Phase Réaction : processus réactif à court/moyen terme*
  Les horaires réalisés seront archivés ainsi que les absences, pour alimenter le processus de paie et pour l'inspection du travail. Pour boucler la boucle, il faut contrôler l'adhérence au planning prévisionnel et ajuster la prévision des besoins. On tire des conclusions sur le dimensionnement des équipes et au cas échéant, lance des embauches.

Comme nos travaux portent essentiellement sur la phase ordonnancement, celle-ci sera détaillée par la suite. La phase réaction est importante dans la vie de l'entreprise mais elle implique une informatique décentralisée (utilisant des techniques de réseau et base de données) ; elle ne présente pas de difficulté conceptuelle majeure en calcul combinatoire. La phase conception est encore très expérimentale et comporte beaucoup de variantes : elle n'est pas mûre pour une modélisation générique. Néanmoins, nous présenterons l'état de l'art sur le dimensionnement au chapitre 2.

1.4.2. La phase ordonnancement

La *Phase Ordonnancement* est un processus prévisionnel (ou proactif) à court / moyen terme. L'élaboration des plannings se fait à plusieurs niveaux suivant la taille de l'entreprise :

- *Planning théorique*
  Le planning théorique est typiquement obtenu par le déroulement d'un cycle de travail sur une année de façon formelle. On obtient un planning parfaitement équitable qui satisfait des besoins prédéfinis mais qui n'est pas forcément réalisable à cause des absences de type congés annuels ou formation.

- *Planning théorique ajusté*
  Le planning théorique est ajusté pour tenir compte des absences prévues (congés, formation, etc.). Cet ajustement peut être automatique mais les relaxations doivent être spécifiées, cf. [CW00]. Comme ces choix dépendront des stratégies de

tel-00004380, version 1 - 29 Jan 2004

l'entreprise, ces développements ne peuvent pas devenir génériques pour toutes les entreprises.

- *Planning prévisionnel*
  Le planning théorique ajusté est porté à la connaissance des salariés en temps en en heure. Cependant il peut encore subir une ou plusieurs modifications pour tenir compte des aléas ou d'autres imprévus. Le planning prévisionnel doit être archivé pour calculer des primes de prévenance.

- *Planning réalisé*
  Pour compléter la liste des différents type plannings, on appelle les relevés bruts des badgeuses, le planning réalisé brut. Compte tenu des différentes règles de gestion mise en place par l'employeur, ces relevés sont aussitôt modifiés donnant le planning réalisé net. Deux exemples de règles :
  - les présences en dehors des plages admises, non demandées par l'employeur, sont ignorées
  - les arrondis à l'heure : ex. les arrivées entre 7H57 et 8H03 seront arrondies à 8H.

- *Planning validé*
  Il s'agit du planning réalisé officiel, après validation par les cadres de proximité. Cette validation est nécessaire si l'entreprise pratique l'auto-déclaration du temps de présence.

1.4.3. Le processus de planification prévisionnel

La planification consiste à créer un planning sur une période allant d'une semaine à un mois. Les données sont des estimations de la quantité de travail à réaliser par intervalle de temps, obtenu grâce aux statistiques et corrigées pour des événements exceptionnels et ponctuels. Ces chiffres sont traduits en nombre de personnes.

A partir des ces besoins en nombre de personnes, le processus de création d'un planning [BB92] propose de planifier suivant la disponibilité réelle du personnel. La ***construction des vacations*** crée un ensemble de vacations qui permet de couvrir les besoins avec un minimum de surplus de main d'œuvre par rapport aux besoins prévus pour chaque intervalle de temps. Cette construction sera réalisée pour chaque jour de la semaine ou les besoins différents. Elle est souvent réalisée pour un personnel anonyme, donc sans la connaissance des contrats temps ou préférences de chaque salarié.

Connaissant le nombre de vacations dont on a besoin pour satisfaire les charges, il faut ensuite enchaîner les vacations journalières et obtenir un planning hebdomadaire ou mensuel pour chaque salarié. Ce processus est appelé ***construction des tours.*** Il peut être réalisé de façon cyclique ou non.

Afin d'optimiser les ressources, il convient de choisir des intervalles de l'ordre de 15 minutes. A cette granularité, la planification sur un horizon d'une semaine est un problème très complexe. Pour résoudre ce problème combinatoire, les chercheurs proposent des plannings cycliques basés sur un planning parfait de courte durée (par ex. 1 semaine) ou réutiliser des plannings acycliques basés sur des besoins hebdomadaires identiques.

tel-00004380, version 1 - 29 Jan 2004

1.4.4. Le planificateur

Dans la plupart des entreprises, la planification est déléguée et décentralisée. Le cadre fonctionnel, par exemple le directeur de ressources humaines, n'exerce qu'une activité de direction de l'activité de planification et de contrôle des plannings, souvent a posteriori.

Le cadre opérationnel ou le cadre de proximité qui réalise le planning quotidien soit manuellement, soit à l'aide des outils informatiques, est souvent le **chef d'équipe** ou le **superviseur** dans les centres d'appels téléphoniques. Ainsi lorsque la taille de l'équipe ne dépasse pas 30, la planification reste le ressort de ce cadre. Dans une unité de soins de 30 personnes, la planification prend 20-40% du temps de l'infirmière en chef [WHC+98].

Dès que le pool de ressources dépasse la centaine, une personne à plein temps est souvent nommée pour gérer les divers problèmes qui peuvent subvenir. Dans les centres disposant de plusieurs centaines d'employés, les **planificateurs** sont plusieurs et souvent regroupés par métier afin de réaliser des économies d'échelle.

1.4.5. Le facteur humain dans les plannings

Malgré les progrès de la modélisation de plannings, le planificateur a toujours un rôle à jouer parce qu'il détient des connaissances approfondies sur le fonctionnement de l'entreprise ou de l'équipe. Ces connaissances sont difficiles à mettre en évidence car elles peuvent résulter des négociations ou du contexte spécifique à l'entreprise. En général, les fonctionnements spécifiques ne sont pas pris en compte par les logiciels standards et le planificateur doit contrôler le planning manuellement pour en assurer la conformité.

Une collaboration réussie doit permettre au planificateur de participer efficacement à l'élaboration des plannings. L'interface homme-machine joue un rôle primordial : il est indispensable pour le planificateur de comprendre la situation planifiée dans sa globalité et ses détails. Dans la pratique, le planificateur cherche un outil pratique qui peut rendre service dès le début, et ce, sans une formation approfondie.

Le générateur automatique de plannings doit donc admettre des pré-affectations faites manuellement, pour que l'utilisateur puisse réagir avec les plannings automatiques. Très souvent les diverses contraintes applicables sont contradictoires : le planificateur doit choisir les contraintes à assouplir afin de produire des plannings.

tel-00004380, version 1 - 29 Jan 2004

1.5. NOS PROPOSITIONS

1.5.1. Limites des solutions actuelles

Pour créer des plannings, le planificateur ne peut plus se fier à son intuition car il doit satisfaire à la fois la législation, les impératifs de l'entreprise, ses contraintes et les souhaits individuels du personnel.

L'état de l'art présenté au chapitre 2 fera ressortir le fait que les modèles et méthodes utilisés aujourd'hui soient trop rigides par rapport aux processus complexes de la planification. Ces limites sont de plusieurs natures :

L1. Limites des méthodes de calculs

L2. Limites des modèles

L3. Limites de prise en compte du contexte dynamique

On retrouve ces limites dans deux références.

Dans son livre [Bee66] aujourd'hui à sa neuvième édition, S. Beer faisait état de 6 obstacles à la bonne application de la recherche opérationnelle. Ne seront pas considérés les points liés aux chercheurs : (a) Conformisme des chercheurs, enfermés dans leur spécialisation ; (b) Pesanteur de l'attitude scientifique, inapplicable au commerce ; (c) Attitude provoquée par l'autosatisfaction des chercheurs et les conflits entre ces derniers et la direction.

Un logiciel, en plus de la complexité en temps et en espace mémoire, doit gérer une *complexité politique*. Dans tout problème ayant une dimension sociale, la résolution de cette complexité exige le compromis ou la négociation. Le professeur Anthony Aaby [1] propose trois points sur l'axe de la complexité politique : des problèmes bien définis, des problèmes mal définis et des problèmes malicieux[2].

Limites des méthodes de calculs

Les méthodes courantes décomposent la planification en sous-étapes séquentielles et irrévocables. La somme des sous-solutions optimales n'est pas toujours optimale globalement.

[Par98] décompose la planification en deux étapes : la construction de vacations (au niveau journalier avec des périodes d'un quart d'heure) puis la construction des grilles (au niveau mensuel avec des périodes d'un jour).

[Heu96] décompose la construction des grilles en l'affectation des jours de repos puis l'affectation d'étiquettes (ou vacations) pour les jours œuvrés.

[Rot00] distingue deux échelles de décision : ordonnancement *au quart d'heure* sur un horizon de plusieurs semaines, et la planification *gros grain*.

La législation française ne suit pas les règles d'analyse qui permettent aux scientifiques de choisir une (seule) granularité pour résoudre le problème. Effectivement, le choix

[1] http://cs.wwc.edu/~aabyan/ du Walla Walla College, Washington

[2] well-defined problems, ill-defined problems and wicked problems. Voir le glossaire.

tel-00004380, version 1 - 29 Jan 2004

d'une échelle de temps fait, on risque de générer des plannings illégaux par rapport à d'autres échelles de temps, ce qui est inadmissible pour un logiciel commercial.

Ce point a été décrit dans [Bee66] par le conformisme des solutions qui conduit à en rétrécir l'éventail, notamment à cause des cloisonnements interdisciplinaires.

Limites des modèles

Les modèles courants se contentent de calculer le nombre de salariés dans chaque groupe homogène du modèle, voir par exemple [Jar+94], [Bru00]. Ils ne tiennent compte ni des caractéristiques individuelles des salariés telles que leurs compétences, leurs demandes pour convenances personnelles et les préférences des salariés et/ou des managers, ni de l'historique de planning pour générer des plannings équitables.

Aujourd'hui, dans nombreux secteurs le manque en main d'œuvre qualifiée est patent, il est absolument nécessaire de proposer des plannings qui prennent en comptent les individus : les qualifications, les préférences, et l'historique pour générer des plannings équitables. Comme il y a souvent des règles implicites ou des négociations personnelles, ces systèmes ne pourront pas satisfaire tous les planificateurs. Avec ces modèles, la planification des ressources humaines devient *mal-définis* : on ne sait plus dire qu'une solution est « optimale ».

Ce point a été décrit dans [Bee66] par la stéréotypie dans la définition des problèmes, souvent trop partielle pour rendre compte de la réalité.

Limites de la prise en compte du contexte dynamique

Les systèmes courants ne permettent pas aux utilisateurs d'interagir avec le processus de planification. Nous avons vu au paragraphe 1.4 que la planification est un processus dynamique auquel contribuent plusieurs acteurs de l'entreprise. Pour un logiciel commercial, il est capital que l'utilisateur puisse réagir avec le générateur automatique de planning, et maîtriser le planning final qui sera communiqué au personnel.

Les problèmes mal-définis dont la spécification changent, sont des problèmes *malicieux*. Dans certaines entreprises, les plannings sont sujets à aléas en ressources et en charge, et qu'il faut réviser le planning en conséquence.

Ce point a été décrit dans [Bee66] par la stéréotypie des critères de rentabilité, fondé sur une approche statique ou rétrospective plutôt que prospective.

1.5.2. Nos propositions et objectifs

Nous proposons un modèle et une méthode de planification permettant de dépasser les limites résumées ci-dessus, avec un ou plusieurs solveurs exploitant les principes de la Propagation Par Contraintes. [Par98] a déjà démontré que cette approche est valable dans le cas de l'affectation des tours, au niveau mensuel.

O1. Nous proposons des modèles et des méthodes globales, ne découpant pas le processus de planification en étapes de calculs irrévocables. Les différentes étapes ou niveaux de détails temporels (journalier, mensuel et annuel) travailleront simultanément.

tel-00004380, version 1 - 29 Jan 2004

Dans les applications GYMNASTE et EQUITIME, nous utiliserons les **contraintes dites « globales »** permettant de générer des plannings en une seule passe. Au chapitre 5, nous proposerons un modèle pour traiter simultanément les contraintes aux niveaux journaliers, hebdomadaires et annuels.

O2. En tenant compte des caractéristiques individuelles des salariés telles leurs préférences et l'historique de planning, nous proposons un modèle pour créer des plannings de grande qualité. Cependant, le nombre de salariés traité ne pourrait pas dépasser la centaine.

Chaque salarié sera considéré comme une ressource unique, avec ses propres caractéristiques que notre système prendra en compte, en utilisant les techniques basées sur la Propagation Par Contraintes.

O3. Notre système permettra aux utilisateurs non-informaticiens de participer activement à l'élaboration des plannings et d'y ajouter leur savoir-faire.

Nous proposerons des interfaces avancées, permettant aux non-informaticiens de comprendre facilement un planning complexe, et d'y apporter des modifications.

Cette méthode de planification ainsi que les concepts d'IHM se concrétisent dans le produit EQUITIME, qui allie une méthode rigoureuse automatique avec une visualisation du planning permettant au planificateur d'intervenir manuellement si besoin est, afin de faire respecter des contraintes sous-entendues qui font partie de la pratique quotidienne.

1.5.3. Limites du champ d'investigation

Du fait de la grande diversité des situations de planification, ce mémoire est naturellement limité au planning des ressources humaines seules dans le secteur tertiaire et le secteur de l'industrie : on s'intéresse à la couverture des besoins en main d'œuvre.

En particulier, nous nous intéressons aux problèmes de planification des employés, et non à l'emploi de temps hebdomadaire des collèges ou lycées, ni à la planification des cours ou des épreuves (par ex. à l'université). De même, ne sont pas envisagés :

- La planification des ressources matérielles avec des contraintes de capacité des machines et des contraintes de succession de tâches
- La planification du personnel avec des dispositions géographiques (contraintes de distance avec des retours à la base en fin de semaine).

Ce mémoire met l'accent sur les modèles et méthodes de planification proprement dite, mais ne traite ni la prévision des besoins (phase amont) ni le suivi (phase en aval de la planification). On considère que les besoins sont connus : ces prévisions s'appuient sur des lois statistiques.

tel-00004380, version 1 - 29 Jan 2004

2 L'ETAT DE L'ART

Résumé :

Dans la construction de plannings des salariés, si créer un planning optimisé d'une journée est aisé, mais créer un planning pour un mois est beaucoup plus complexe. En plus de la complexité combinatoire, il faut tenir compte de contraintes applicables sur les échelles hebdomadaire et mensuelle.

Afin de découpler la complexité combinatoire, la construction automatisée des plannings émet l'hypothèse que les plannings sont indépendants un jour sur l'autre. Ainsi, le problème se découpe en jours, enchaînes par la suite pour obtenir des plannings hebdomadaires. Les plannings sont élaborés en deux phases : la construction des vacations (ou horaires journaliers) puis l'enchaînement de ces journées dans la construction des tours mensuels. Voir par exemple [Bak76] ou [BM76]

Pour caractériser l'état de l'art en construction des plannings, ce chapitre évoquera d'abord les différents modèles pour chacune des deux phases. Ensuite il étudiera les méthodes de résolution de ces modèles par la Programmation Par Contraintes (PPC) ou par la Programmation Linéaire en Nombres Entiers, PLNE.

Les méthodes des recherches locales (y compris le recuit simulé, la recherche tabou, les algorithmes génétiques) et des algorithmes d'approximation seront également présentées. L'état de l'art sera complété par un rappel des systèmes interactifs d'aide à la décision.

2.1 LA CONSTRUCTION DES HORAIRES JOURNALIERS : MODELES PLNE

Intuitivement, le problème est illustré par la Figure 2.1

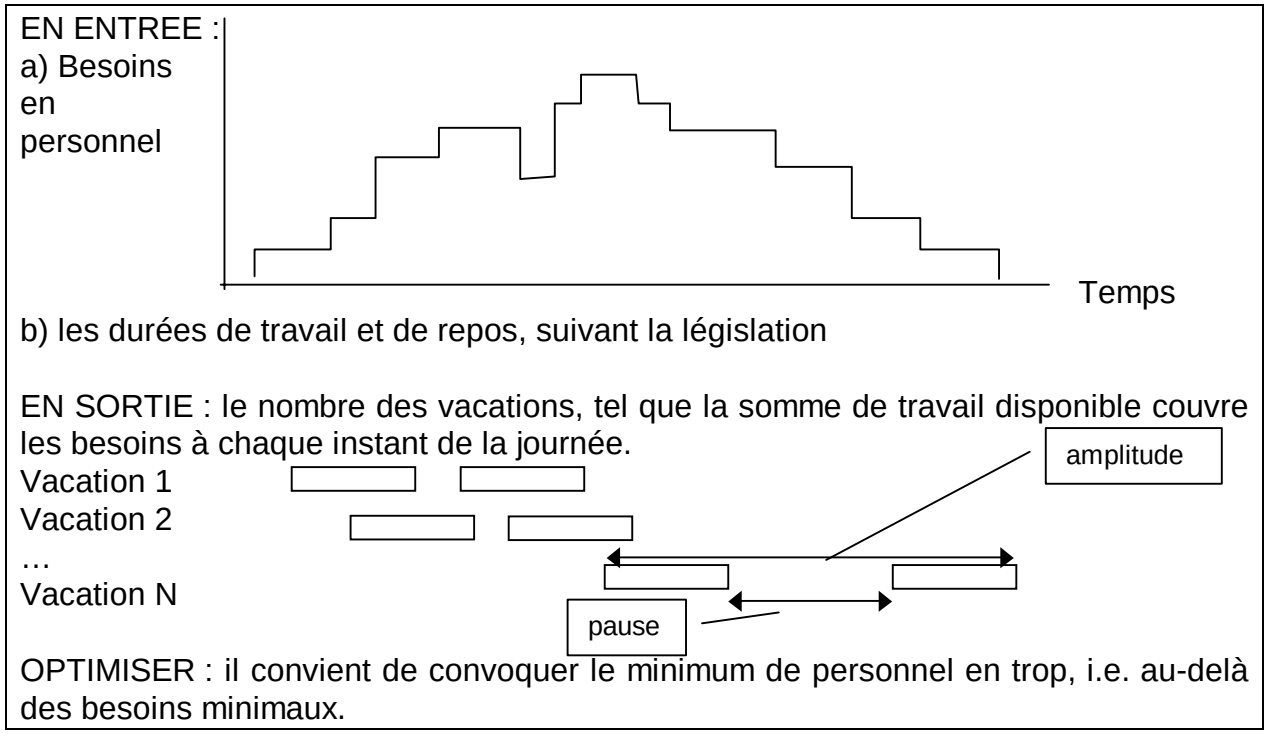


Figure 2.5. Construction des horaires journaliers

tel-00004380, version 1 - 29 Jan 2004

En entrée, on dispose des besoins en personnel : pour chaque intervalle de temps, la courbe de charge indique le nombre de personnes nécessaires. Cette information peut résulter des calculs statistiques sur le nombre d'appels par heure et le temps moyen de traitement par appel sur une période analogue.

En support du processus, on dispose d'un ensemble de règles légales qui définissent la journée de travail, ainsi qu'un ensemble de règles d'usage.

En sortie, on veut savoir comment faire travailler les salariés, soit la définition des horaires de début et fin et des horaires de pause, de telle sorte que la somme de leurs disponibilités couvre au mieux la courbe de charges avec un minimum de surplus.

2.1.1 Le modèle explicite de Dantzig

Le premier modèle de couverture de charge est dû à Dantzig dès 1954 [Dan54]. La construction des vacations se repose sur des modèles de couverture ensemblistes[3], décrit par les équations (1) et (2).

$$\text{Min } z = \sum_{j=1}^{m} C_j X_j \qquad (1)$$

$$\sum_{j=1}^{m} A_{ij} X_j \geq B_i \text{ , I=1,..,n et Xj} \geq 0, \text{ entier} \qquad (2)$$

Où

n=nombre d'intervalles à couvrir dans l'horizon de planification,
m= nombre de vacations valides
Cj = coût lorsqu'un salarié est affecté à la vacation j
A_{ij} = 1 si l'intervalle i est couvert par la vacation j, 0 sinon
B_i = nombre de salaries requis pour l'intervalle i
X_j = nombre de salaries affectés à la vacation j

En amont du traitement par le modèle, l'ensemble des vacations valables est énuméré explicitement par rapport à l'ensemble des règles légales et d'usage. Exemple : la durée totale de travail ne doit pas dépasser 10 heures ; une pause ne débute qu'après 2 heures de travail effectif, etc.

Le modèle compte le nombre des salariés à chaque vacation, de telle sorte que les besoins par intervalle B sont satisfaits, tout en minimisant le coût total des affectations.

Exemple : Avec un seul type de vacation, sur une journée constituée de 9 intervalles d'une heure, les durées de la vacation sont de 4 à 7H

L'exemple comprend des vacations de durées de 4H (vacations a(j)), 5H : b(j), 6H : c(j) ou 7H : d(j), soit un total de 30 vacations.

[3] Generalized Set-Covering Problem GSCP

tel-00004380, version 1 - 29 Jan 2004

Temps:	1	2	3	4	5	6	7	8	9
a(1)	1	1	0	1	1				
a(2)		1	1	0	1	1			
a(3)			1	1	0	1	1		
a(4)				1	1	0	1	1	
a(5)					1	1	0	1	1
b(1)	1	1	0	1	1	1			
b(2)	1	1	1	0	1	1			
b(3)		1	1	0	1	1	1		
b(4)		1	1	1	0	1	1		
b(5)			1	1	0	1	1	1	
b(6)			1	1	1	0	1	1	
b(7)				1	1	0	1	1	1
b(8)				1	1	1	0	1	1
---									
c(1)	1	1	0	1	1	1	1		
c(2)	1	1	1	0	1	1	1		
c(3)	1	1	1	1	0	1	1		
c(4)		1	1	0	1	1	1	1	
c(5)		1	1	1	0	1	1	1	
c(6)		1	1	1	1	0	1	1	
c(7)			1	1	0	1	1	1	1
c(8)			1	1	1	0	1	1	1
c(9)			1	1	1	1	0	1	1
---									
d(1)	1	1	0	1	1	1	1	1	
d(2)	1	1	1	0	1	1	1	1	
d(3)	1	1	1	1	0	1	1	1	
d(4)	1	1	1	1	1	0	1	1	
d(5)		1	1	0	1	1	1	1	1
d(6)		1	1	1	0	1	1	1	1
d(7)		1	1	1	1	0	1	1	1
d(8)		1	1	1	1	1	0	1	1

Figure 2.6. Vacations avec pauses : 30 variables

Quant aux vacations sans pause sur une durée de 9H, il y a 6 vacations de 4H, 5 de 5H, 4 de 6H, 3 de 7H. Au total, on décompte 6+5+4+3+30 = 48 vacations.

Dans la pratique, un très grand nombre de vacations doit être utilisé afin de tenir compte de

- la souplesse inhérente à une charge définie au quart d'heure près
- des journées de longueurs différentes
- différents horaires de début, ou des pauses repas différentes

[Kla73] rapporte un nombre avoisinant 15000 vacations sur une journée, [BJ00] mentionne des milliards de horaires possibles sur une semaine. La génération de toutes les vacations pour la résolution avec ce modèle prendrait un certain temps. Chaque vacation étant représentée par une variable, les systèmes d'équations résultants sont trop grands pour être résolus par les logiciels disponibles aujourd'hui. D'où la motivation de la modélisation implicite des vacations.

2.1.2 Le modèle implicite de Moondra

Moondra fut le premier [Moo76] à proposer un modèle implicite où chaque vacation n'est plus représentée qu'implicitement par le nombre d'agents qui démarrent ou qui terminent au cours d'une intervalle p. Afin de traiter les différences des coûts, on crée des groupes pour les vacations ayant le même coût par période travaillée, la même pause repas et les mêmes restrictions en durée de travail (bornes sur la durée totale de la vacation et la durée travaillée sans interruption).

tel-00004380, version 1 - 29 Jan 2004

Variables de décision :
F(t, p)=nombre d'agents sur une vacation du type t qui se termine à la fin de l'intervalle p
S(t, p)=nombre d'agents sur une vacation du type t qui démarre au début de l'intervalle p

Comme le montre la figure suivante, le nombre d'agents présents à chaque intervalle p est donné par la somme des vacations ayant commencé avant ou pendant cette intervalle S(t, p), moins la somme des vacations déjà achevées. Pour notre exemple, le nombre de variables n'est que 9 par type de vacations de même coût.

Exemple : pour un seul type de vacations, sur 9 intervalles d'une heure. Avec des durées d'une vacation 5 à 7H, on ne commence plus au delà de i=5 donc S(t, p)=0 pour p > 5.

Nombre de personnes disponibles pendant l'intervalle i :
I = 1 : S(1,1)
I = 2 : S(1,1)+S(1,2)
I = 3 : S(1,1)+S(1,2)+S(1,3)
I = 4 : S(1,1)+S(1,2)+S(1,3)+S(1,4)
I = 5 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5)
I = 6 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)
I = 7 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)-F(1,6)
I = 8 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)-F(1,6)-F(1,7)
I = 9 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)-F(1,6)-F(1,7)-F(1,8)

Nombre global de personnes qui débutent et qui finissent
S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5)=F(1,5)+F(1,6)+F(1,7)+F(1,8)+F(1,9)

Pour limiter le minimum durée de travail à 5 périodes pour ceux qui commencent
en P=1 peuvent finir en P=5 : $S(1,1) \geq F(1,5)$
en 1 et 2 peuvent finir en 5 et 6 : $S(1,1)+S(1,2) \geq F(1,5)+F(1,6)$
en 1,2 et 3 peuvent finir en 5,6 et 7 : $S(1,1)+S(1,2)+S(1,3) \geq F(1,5)+F(1,6)+F(1,7)$
en 1,2,3 et 4 peuvent finir en 5,6,7 et 8:
$S(1,1)+S(1,2)+S(1,3)+S(1,4) \geq F(1,5)+F(1,6)+F(1,7)+F(1,8)$

Pour limiter le maximum durée de travail à 7 périodes pour ceux qui commencent
en 1 doivent finir en 5,6 et 7 : $S(1,1) \leq F(1,5)+F(1,6)+F(1,7)$
en 1 et 2 doivent finir en 5,6,7 et 8: $S(1,1)+S(1,2) \leq F(1,5)+F(1,6)+F(1,7)+F(1,8)$

Figure 2.7. Modèle Moondra avec 9 variables

Ce modèle est intéressant historiquement car c'est le premier modèle implicite. Avec les variables F et S, Moondra a proposé un système d'équations permettant de calculer le nombre d'agents présents pendant chaque période, qui doit couvrir les besoins, ainsi que des équations pour limiter la durée maximale et minimale des vacations. Ce modèle ne tient compte que d'une pause de durée fixe et d'une heure de début fixe par vacation.

2.1.3 Le modèle implicite de Bechtold et Jacobs

Le problème des pauses qui peuvent démarrer à des intervalles différents est résolu par [BJ90]. L'équivalence de ce modèle avec celui de Dantzig a été démontrée sous certaines conditions dans [BJ96]. Chaque type de vacation t est associé à une pause pendant une période de travail. Ce modèle utilise des variables B(i) qui donne le nombre de salariés

tel-00004380, version 1 - 29 Jan 2004

sur toutes les vacations dont la pause commence à une période i. Cette approche est limitée à une seule pause repas, dont la durée est la même pour toutes les vacations.

Variables de décision :
X(t) = nombre d'agents dans la vacation de type t
B(p) = nombre total de pauses initiées au début de la période p, quelle que soit la vacation type

L'apport de ce modèle est l'expression des contraintes qui garantissent que les personnes prendront leurs pauses comme prévu. Ces contraintes s'appuient sur l'hypothèse que les pauses ne sont pas imbriquées de façon *extraordinaire*, c'est à dire la fenêtre de pause d'un type de vacation j n'est pas strictement incluse dans celle d'un autre type j1.

2.1.4 Le modèle implicite de Thompson

Un modèle *doublement* implicite est proposé par [Tho95], permettant de tenir compte à la fois des heures de début de vacations de [Moo76] et les pauses repas de [BJ90]. Le nombre d'agents présents à chaque intervalle p est donné par la somme des vacations ayant commencé avant ou pendant cet intervalle S(t, p), moins la somme des vacations déjà achevées et des pauses en cours pendant cette période.

Variables de décision :
F(t, p)=nombre d'agents sur une vacation du type t qui se termine à la fin de la période p
S(t, p)=nombre d'agents sur une vacation du type t qui démarre au début de la période p
B(t, p)=nombre d'agents sur une vacation du type t, en pause commençant au début de p

L'avantage du modèle de Thompson est la souplesse qu'il offre dans l'étendue des vacations concernées, et surtout par la possibilité de trouver des solutions optimales.

tel-00004380, version 1 - 29 Jan 2004

Exemple : pour le même cas présenté ci-dessus
Nombre de personnes disponibles pendant l'intervalle i :

I = 1 : S(1,1)
I = 2 : S(1,1)+S(1,2)
I = 3 : S(1,1)+S(1,2)+S(1,3) -B(1,3)
I = 4 : S(1,1)+S(1,2)+S(1,3)+S(1,4) -B(1,4)
I = 5 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -B(1,5)
I = 6 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5) -B(1,6)
I = 7 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)-F(1,6) -B(1,7)
I = 8 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)-F(1,6)-F(1,7)
I = 9 : S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) -F(1,5)-F(1,6)-F(1,7)-F(1,8)

Le nombre total de personnes qui travaillent égale le nombre total de pauses
S(1,1)+S(1,2)+S(1,3)+S(1,4)+S(1,5) = B(1,3)+B(1,4)+B(1,5)+B(1,6)+B(1,7)

La durée minimale d'une période de travail avant pause est de 2 intervalles :
Le nombre de personnes qui commencent en P=1 est au moins égal au nombre de pauses en P=3 ;
Sinon les pauses en P=3 comptent pour ceux qui commencent en P=2 $S(1,1) \geq B(1,3)$
Ceux qui commencent en P=1 et 2, doivent compter parmi ceux qui sont en pause en P=3 et 4 :

$S(1,1)+S(1,2) \geq B(1,3)+B(1,4)$
Idem : $S(1,1)+S(1,2)+S(1,3) \geq B(1,3)+B(1,4)+B(1,5)$
$S(1,1)+S(1,2)+S(1,3)+S(1,4) \geq B(1,3)+B(1,4)+B(1,5)+B(1,6)$

Pour limiter la durée d'une période de travail après pause à 2 intervalles :

$F(1,9) \geq B(1,7)$
$F(1,8)+F(1,9) \geq B(1,6)+B(1,7)$
$F(1,7)+F(1,8)+F(1,9) \geq B(1,5)+B(1,6)+B(1,7)$
$F(1,6)+F(1,7)+F(1,8)+F(1,9) \geq B(1,4)+B(1,5)+B(1,6)+B(1,7)$

Les restrictions sur la durée maximale d'une période de travail à 4 intervalles ne sont pas nécessaires dans ce cas. Avec une durée maximale de travail à 7 intervalles, une pause repas à 1 intervalle, la durée minimum à 2 intervalles, on obtient une durée maximale à 4.

Figure 2.8. Modèle de Thompson : 14 variables

Un autre modèle implicite proposé par [Ayk96], traite la situation où existent des pauses supplémentaires avant et après la pause repas. Les variables sont les suivantes :

Variables de décision :
U(t, p)= nombre de salariés affectés à une vacation type t dont la première pause courte aura lieu à la période p
W(t, p)= nombre de salariés affectés à une vacation type t dont la pause repas = p
B(t, p)= nombre de salariés affectés à une vacation type t dont la deuxième pause courte = p

tel-00004380, version 1 - 29 Jan 2004

Compte tenu des variables U, W et V, le nombre de personnes disponible au cours de la période p, X'(t, p) est donné par :
X'(t, p) = X(p) si p n'est pas une pause
X'(t, p) = X(p) – U(t, p) si p est la première pause courte
X'(t, p) = X(p) – W(t, p) si p est la pause repas et p-1 n'est pas la pause repas
X'(t, p) = X(p) – W(t, p) – W(t,p-1) si p et p-1 sont les pauses repas
X'(t, p) = X(p) – W(t, p-1) si p-1 est la pause repas et p n'est pas la pause repas
X'(t, p) = X(p) – V(t, p) si p est la deuxième pause courte

Contrainte de couverture de charge (proche du Modèle de Danzig) :
A(t, p) = 1 si la période p est travaillée dans la vacation de type t, = 0 sinon
$\sum_{t=1,n} A(t, p) * X'(t, p) \geq R(p)$ pour chaque période p

Figure 2.9. Modèle d'Aykins [Ayk96]

2.1.5 Le modèle implicite de Jarrah

Le modèle de Jarrah [JBS94] a intégré la modélisation implicite des pauses de [BJ90], avec les algorithmes exacts de positionnement de repos développés dans [BC85], et de plus traite les travailleurs à temps partiel.

Variables de décision :
X(t) = nombre d'agents temps plein dans la vacation de type t
Y(t) = nombre d'agents temps partiel dans la vacation de type t
B(i) = nombre total de pauses initiées au début de la période i, quelle que soit la vacation type

F(j, t) = 1 si la vacation en temps plein type j couvre la période t, 0 sinon
P(j, t) = 1 si la vacation en temps partiel type j couvre la période t, 0 sinon

Figure 2.10. Modèle de Jarrah et al.

Utilise les variables:
p = ratio des salariés en temps plein aux salariés en temps partiel.
Nf = nombre de types de vacations à temps plein
Np = nombre de types de vacations à temps partial
R(t)= besoins en personnel au cours de la période t

F(j, t) et P(j, T) sont utilisés dans la formulation de couverture ensembliste généralisée :

$$\sum_{j=1, Nf} F(j, t) * X(t) + \sum_{j=1, Np} P(j, t) * Y(t) - B(t) \geq R(t) \quad (3)$$

Suite à la résolution du PLNE, il faut affecter les personnes suivant leur contrat temps. Un algorithme glouton est proposé par Jarrah.

2.1.6 Modèle implicite de Cai et Li

A titre de comparaison, nous citons un modèle traitant le personnel à multiples compétences [CL00]. On considère trois populations d'employées qualifiées en 1, qualifiées en 2 et qualifiées en 1 et 2.

tel-00004380, version 1 - 29 Jan 2004

Variables de décision :
X(j) = nombre d'agents de qualification 1 dans la vacation j
Y(j) = nombre d'agents de qualification 2 dans la vacation j
Z1(j) = no. d'agents de qualification 1+2 dans la vacation j faisant le travail qualification 1
Z2(j) = no. d'agents de qualification 1+2 dans la vacation j faisant le travail qualification 2

Contrainte de couverture de charge (Modèle de Danzig) :
A(j, p) = 1 si la période p est travaillée dans la vacation de type t, = 0 sinon
$\sum_{j=1,n} A(j, p) * X(j) + \sum_{j=1,n} A(j, p) * Z1(j) \geq R1(p)$ pour chaque période p, besoins 1
$\sum_{j=1,n} A(j, p) * Y(j) + \sum_{j=1,n} A(j, p) * Z2(j) \geq R2(p)$ pour chaque période p, besoins 2

Figure 2.11. Modèle de Cai et Li [CL00]

L'intérêt de ce modèle est la prise en compte des multiples compétences. Le modèle résultant exige un grand nombre de variables, car il faut considérer toutes les combinaisons de compétences.

Pour résoudre ce modèle, les auteurs proposent une méthode à base de l'algorithme génétique. La durée des intervalles est l'heure ; pour les vacations jour, on peut commencer entre 7 H et 15H et il n'y a qu'une vacation nuit (au total 10 vacations).

2.1.7 Conclusion sur les modèles PLNE

Les modèles en Programmation Linéaire peuvent exprimer les contraintes sur les durées de travail journalier. Les variables en nombres entiers donnent le nombre de salariés qui effectuent un travail caractérisé, ainsi la méthode peut traiter un grand nombre de salariés **homogènes**.

Les travaux ont montré que les modèles implicites peuvent être résolus très efficacement avec la PLNE : de façon générale, les instances donnent des solutions entières optimales.

tel-00004380, version 1 - 29 Jan 2004

2.2 CONSTRUCTION DES HORAIRES JOURNALIERS : MODELES PPC

Ce paragraphe présente des modèles permettant de résoudre le problème du paragraphe précédent avec la Programmation Par Contraintes.

2.2.1 Le modèle explicite de Partouche

Dans ce modèle, la liste de vacations candidates est représentée explicitement comme un objet contenant 3 vecteurs de même longueur : le vecteur des heures de début, le vecteur des heures de fin et le vecteur des durées.

Variables de décision :
X(j) = nombre d'agents pris dans la vacation j, une variable dont le domaine est {1..n}, ce qui assure qu'elle aura une valeur entière
A(j, p) = X(j) si la période p est couverte par la vacation j, = 0 sinon

Contrainte de couverture de charge :
Pour chaque période p, $\sum_{j=1,n} A(j, p)$ doit couvrir les besoins B(p)
Minimiser la somme des excédents e(p) = $\sum_{j=1,n} A(j, p) - B(p)$
Contraintes supplémentaires :
Non-symétrie parmi les agents
Le nombre d'agents est réduit au strict minimum i.e. $\text{Max}_p B(p)$

Figure 2.12. Modèle explicite de Partouche

D'après [Par98], ce modèle est très peu performant en PPC même pour des tests comportant seulement de 10 à 48 agents.

2.2.2 Le modèle implicite de Partouche-Moondra

[Par98] présenta un deuxième modèle implicite de planning individuel avec des variables de début et de fin de vacation, proche de celui de Moondra, ce qui évite d'énumérer toutes les vacations possibles. Pour chaque agent, le système construit sa vacation définie par l'heure de début et de fin.

Variables de décision :
Y(j) = 1 si la vacation j est choisie dans la solution , = 0 sinon
S(j) = heure de début de la vacation j,
F(j) = heure de fin de la vacation j

tel-00004380, version 1 - 29 Jan 2004

Variables dépendantes :
La matrice A(i, j) qui vaudra 1 si l'intervalle i est couvert par la vacation j
A(i, j) = 1 si (Y(j) = 1) et ((S(j) ≤ i) et (F(j) > i))
Les amplitudes des vacations D(j) = S(j) – F(j) dont le domaine est prédéfini.

Contrainte de couverture de charge :
La somme pour chaque intervalle i, $\sum_{j=1,n} A(i, j)$ doit couvrir les besoins B(i) et on cherche à minimiser la somme des excédents $e(i) = \sum_{j=1,n} A(i, j) – B(i)$

Contrainte supplémentaire : Asymétrie
Construire les vacations dans l'ordre chronologique : Pour tout j > 1, S(j) ≥ S(j-1)

Stratégie
Les variables Y, F et S sont affectées dans cet ordre.

Figure 2.13. Modèle implicite de Partouche – Moondra

Avec ce modèle, la première solution est souvent trouvée très rapidement, les résultats sont globalement décevants au niveau de l'optimum. Les résultats suivants ont été publiés pour un PC 486 à 100 M Hz.

	PLNE		PPC 1er		PPC 2ème	
	CPU	Objectif	CPU	Objectif	CPU	Objectif
Test1 10 pers.	<1 s.	2	41 s.	2	4 s.	2
Test 2.1 48 per	<1 s.	171	> 20 min.	Pas de soln.	7 s.	171*+
Test 2.2	<1 s.	211	1 s.	1377*	7 s.	225*
Test 2.3	<1 s.	447		560*	6 s.	474*

Légende : * première solution, + optimum non prouvé

En effet, Partouche propose des plannings individuels (avec 3 variables par individu). Ce type de modèle permet de prendre en compte les particularités individuelles (par ex. préférences et historique). Les modèles planning anonyme utilisent des variables qui donnent le nombre de salariés d'un tel type : ils peuvent traiter des situations avec un nombre plus grand de salariés mais ils ne peuvent pas prendre en compte les préférences et historique.

2.2.3 Notre modèle utilisant la contrainte cumulative

Avec des contraintes locales, les modèles précédents sont peu efficaces. Pour obtenir un maximum de propagation, il est vivement conseillé d'utiliser les contraintes globales. Nous présenterons la contrainte cumulative pour la construction des horaires journaliers. La contrainte cumulative [AB93] a été conçue pour résoudre le problème de l'ordonnancement des tâches où le nombre de ressources disponibles est limité. La définition de la contrainte est donné dans le glossaire.

Il n'est pas évident de savoir comment elle peut servir car l'opérateur de comparaison dans la couverture des charges est supérieure ou égal et non inférieure ou égale.

[SC95] montrait comment utiliser la contrainte cumulative pour modéliser le concept du producteur/consommateur, dans lequel le ou les processus produisent une commodité en stock et des processus qui le consomment, chaque processus ayant lieu à des instants et avec des quantités précises. Voir la Figure 2.14.

tel-00004380, version 1 - 29 Jan 2004

Soit

N le nombre d'événements producteurs P_1, …, P_n,
qui produisent la quantité Q_{Pi} au moment T_{Pi}
M le nombre d'événements consommateurs C_1, …, C_m
qui consomment Q_{Cj} au moment T_{Cj}
Q_0 le niveau initial du stock
$a = \min \{ T_{Cj}, T_{Pi} \} - 1$ et $b = \max \{ T_{Cj}, T_{Pi} \} + 1$

Contrainte de producteur/consommateur :

$$\forall t \in [a, b] \quad Q_0 + \sum_{T_{Pi} \le t} Q_{Pi} \ge \sum_{T_{Cj} \le t} Q_{Cj} \qquad (4)$$

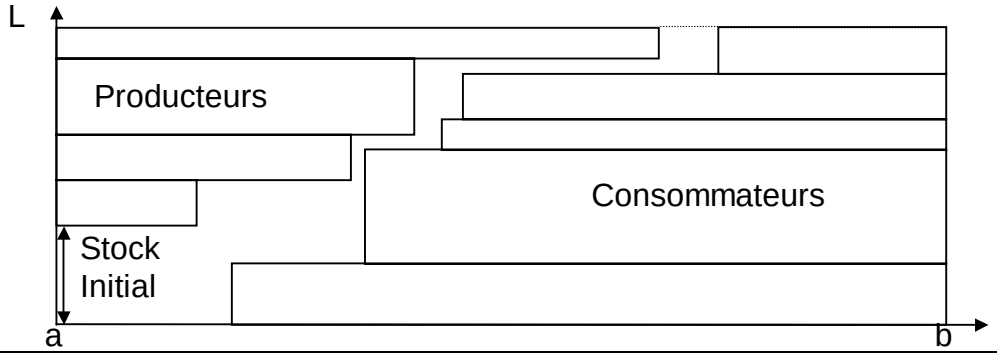
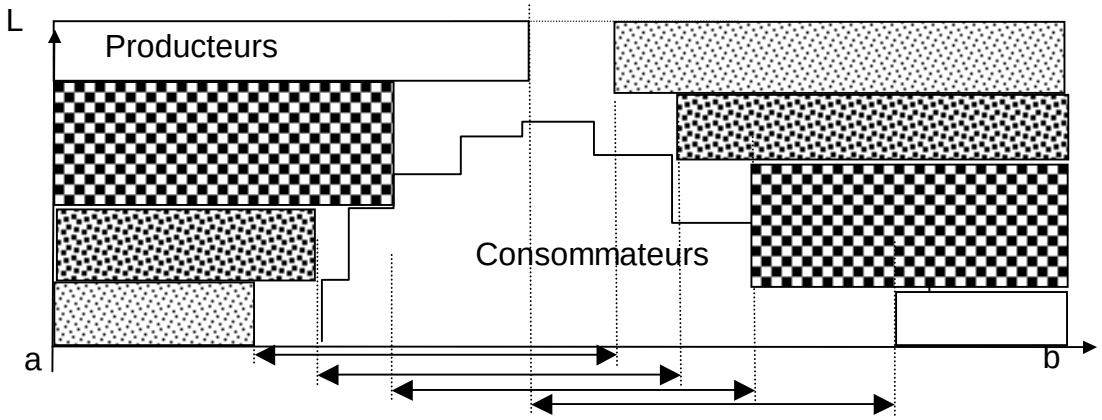


Figure 2.14. Définition de la contrainte Producteur/Consommateur

Le raisonnement suivant a été proposé pour formuler ce problème à l'aide de la contrainte cumulative. Un processus producteur P_i rend disponible la quantité Q_{Pi} de ressource au moment T_{Pi} : on l'exprime avec une tâche cumulative qui occupe la ressource depuis le début jusqu'à T_{Pi}. Inversement, un processus C_j consomme la quantité Q_{Cj} de ressource au moment T_{Cj} : on l'exprime avec une tâche cumulative qui occupe la ressource depuis T_{Cj} jusqu'à la fin de l'horizon de calcul.

Nous nous sommes s'inspiré de ce travail pour créer un modèle de couverture des charges avec la contrainte cumulative où $Q_{Pi} = 1$ et $Q_{Cj} = 1$. Un employé i qui travaille de T_{Di} à T_{Fi}, rend disponible une ressource pendant la durée de sa présence. On l'exprime avec deux tâches : l'une dure depuis le début jusqu'à T_{Di} et l'autre depuis T_{Fi} jusqu'à la fin. La courbe de charges peut simplement se définir par un ensemble de tâches



Début et fin des vacations de chaque groupe d'employés

Figure 2.15. Adaptation à la couverture de la courbe des charges

tel-00004380, version 1 - 29 Jan 2004

2.2.4 Autres modèles implicites

Nous proposons de résumer ici un modèle implicite de construction des vacations, dans le contexte des multiples qualifications. Il sera utilisé dans des développements décrits au chapitre 5.

Variables de décision :
X(e, i) = q où e est un employé, i est un intervalle de temps, et q la qualification de l'employé pendant cet intervalle, ou l'affectation au repos.

Ainsi l'ensemble des vacations n'est pas énuméré avant la résolution des charges de couverture.

Contrainte de couverture de charges :
$\forall i \in 1, \ldots I, \forall q \in$ **Qualifications**, $W_D(i, q) \leq |\{ X(e, i) = q, \forall e \in$ **Employés** $\}|$
Utilisation des primitifs PPC décrits au glossaire : voir [BSK+97b]
among($[N_1, ..., N_Q]$, $[X(e_1, i), \ldots, X(e_N, i)]$, Zeros $_N$, $[q_1, ..., q_Q]$, all)
Minimiser la somme des excédents = $\sum_q \sum_i |\{ X(e, i) = q, \forall e \}| - W(i, q)$

Contrainte sur la durée totale de travail pour un employé e :
Durée totale du travail : among ([Min, Max], $[X(e, 1), \ldots, X(e, I)]$, Zeros $_I$, [repos], all)
on restreint le nombre total de variables prenant la valeur repos

Contrainte sur les pauses repas (après 6H consécutives de travail)
Les contraintes de repos journalier sont implantées dès que la fin de la journée précédente est connue, on peut déterminer le début au plus tôt le lendemain.

Figure 2.16. Autre modèle implicite

Au chapitre 5, nous proposerons une méthode de résolution utilisant ce modèle à multiples qualifications.

2.2.5 Conclusion sur la construction de vacations par la PPC

Les modèles conçus pour la PLNE ne peuvent pas être traités avec la même efficacité en PPC, car les contraintes de type *somme* présentes dans ces modèles sont de nature globale et non locale à un nombre de variables. La propagation locale est très faible, ce qui conduit à la conclusion de Partouche que la PPC n'est pas adaptée pour résoudre ce problème (cf. [Par98]). Nous avons proposé un modèle utilisant la contrainte cumulative permettant de tirer un maximum de propagation.

tel-00004380, version 1 - 29 Jan 2004

2.3 LA CONSTRUCTION DE TOURS ACYCLIQUES : MODELES PLNE

Suite à la constitution d'une courbe de charge journalière, le calcul de dimensionnement de l'équipe et la construction des vacations, nous connaissons le besoin en nombre de personnes à chaque type de vacation pour chaque jour de la semaine. La création de plannings mensuels (ou construction de tours) peut se faire de façon cyclique ou non.

Nous proposons de faire le tour des méthodes usuelles de planification acyclique.

2.3.1 Calcul des Journées Repos. Day-Off Scheduling

Lorsque la durée de travail hebdomadaire d'un salarié ne correspond pas à la semaine œuvrée de l'entreprise, il faut mettre en place le calcul des journées de repos. Ex. la semaine de 5 jours dans des entreprises ouvertes 7 jours par semaine dans le cas d'un salarié à temps plein.

Différents cas de repos hebdomadaires peuvent être envisagés :

- 2 jours de repos par semaine
- 2 jours consécutifs par semaine
- 4 jours de repos sur 2 semaines, dont 2 jours consécutifs
- 4 jours de repos sur 2 semaines, dont 1 week-end

Exemple d'un modèle PLNE pour le calcul des journées repos

Soit les besoins B(p) pour les périodes p=1, …, 7, et l'hypothèse de deux jours de repos consécutifs par semaine pour chaque employé, le programme linéaire correspondant est :

Minimiser $\sum X(p)$

$$\begin{pmatrix} 0&1&1&1&1&1&0 \\ 0&0&1&1&1&1&1 \\ 1&0&0&1&1&1&1 \\ 1&1&0&0&1&1&1 \\ 1&1&1&0&0&1&1 \\ 1&1&1&1&0&0&1 \\ 1&1&1&1&1&0&0 \end{pmatrix} \begin{pmatrix} X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{pmatrix} \geq \begin{pmatrix} B(1) \\ B(2) \\ B(3) \\ B(4) \\ B(5) \\ B(6) \\ B(7) \end{pmatrix}$$

Figure 2.17 Calcul des journées de repos

Des matrices semblables peuvent être proposées pour les autres cas de repos hebdomadaires. [Bak74] a analysé le cas particulier où

1. la somme des besoins est un multiple de 5
2. Le maximum des besoins est inférieur au cinquième de la somme des besoins

Sans la condition 1, il n'est pas possible qu'un effectif entier couvre **exactement** la charge. Sans la condition 2, l'effectif nécessaire pour couvrir la charge maximale engendre immanquablement une couverture totale supérieure à la charge totale.

Lorsque ces deux conditions sont vérifiées, il peut exister une solution avec l'effectif égal au cinquième de la somme des besoins, qui ne génère aucun sur-effectif.

tel-00004380, version 1 - 29 Jan 2004

[BR78] suggère le problème complémentaire du problème initial qui est en quelque sorte l'image inversée de l'original :

Maximiser $\sum Y(p)$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \\ Y(6) \\ Y(7) \end{pmatrix} \leq \begin{pmatrix} W\text{-}B(1) \\ W\text{-}B(2) \\ W\text{-}B(3) \\ W\text{-}B(4) \\ W\text{-}B(5) \\ W\text{-}B(6) \\ W\text{-}B(7) \end{pmatrix}$$

Figure 2.18 Calcul des journées de repos : matrice inverse

Si W correspond à l'effectif total, W-B(p) correspond au nombre maximum de salariés pouvant être au repos à la période p. Le problème consiste donc à maximiser le nombre d'agents ayant deux repos consécutifs chaque semaine.

L'intérêt de ce modèle réside par la traduction via le programme linéaire en Y d'un problème de couplage maximal dans un graphe constitué d'un cycle simple de 7 nœuds.

Conclusion sur le calcul des journées repos

L'accord des journées repos est un fait éminemment contractuel ou syndical. Dans un milieu où il est difficile de trouver des candidats avec les compétences requises pour étoffer les équipes telles que les équipes de soins médicalisés, il faudra offrir des garanties sur le positionnement des repos.

Ce calcul se fait bien puisque le modèle est très petit.

2.3.2 La construction des tours

La construction des vacations fournit le nombre des vacations à fournir par jour afin de couvrir les besoins. Il n'affecte pas de vacations aux salariés. Il faut alors construire des tours et les affecter aux salariés.

Méthode de résolution utilisant la PLNE

Les variables de décision sont

$X(e, t, k) = 1$ si le jour t, l'employé est affecté à l'étiquette k, = 0 sinon

$X_R(e, t) = 1$ si le jour t est affecté au repos, = 0 sinon

Les contraintes légales sont :

- Contrainte d'unicité : $\sum_k X(e, t, k) \leq 1, \forall (t, e)$
  On place au plus une vacation à chaque jour de l'horizon pour l'employé
- Couverture des besoins : : $\sum_e X(e, t, k) \leq v_{tk}, \forall (t, k)$
- Au plus N vacations k, toutes les P semaines
  Sur toutes les variables t=7*w,…, 7*w + 7*P-1 pour la semaine w

$$\sum_{t = 7*w, \ldots, 7*w+7*P-1} X(e, t, k) \leq N, \forall e, w \qquad (5)$$

tel-00004380, version 1 - 29 Jan 2004

- Au plus C jours successifs de travail : sur C+1 jours consécutifs, on a au moins un repos. $\sum_{i=0, C} X_R (e, t + i) \geq 1, \forall t$ (**6**)
- Repos journalier: si k=0 représente le repos,
  $X(e, t, k) + X(e, t+1, k') \leq 1$ avec $k <> 0, k' < k \ (\forall e, k)$ (**7**)

Afin de réaliser avec la PLNE, une contrainte de transition du type :

si X(e, t, k) = 1 alors X(e, t+1, k') = 1,

[Wil91] propose d'utiliser une variable indicateur δ et une linéaire équation du type : $X - \delta \leq 0$ qui est équivalent à { X=0 ≡ δ=0 & X=1 ≡ δ=1 }

$X(e, t, k) - X(e, t+1, k') \leq 0$

Pour tenir compte des pré-affectations aux employés, il suffit d'ajouter des équations du type X(e, t, k) = 1 dans ce modèle. Si une personne ne peut pas être affectée au code k' : il suffit d'ajouter la contrainte X(e, t, k') = 0.

Méthode de résolution avec des modèles de flot

Afin de produire des tours hebdomadaires, [CGL01] utilise une formulation en PLNE pour combiner les modèles journaliers dans un cadre de flot. Le principe consiste à construire un réseau dans lequel un chemin du nœud source au nœud puit respecte les contraintes suivantes :

T1 : règle sur les jours de repos (ex. au moins un jour doit être un week-end)

T2 : la différence des heures débutantes des jours consécutifs d'un tour n'excède pas une limite donnée (ex. deux heures)

T3 : les tours peuvent être cycliques (ne pas violer la contrainte sur les heures débutantes pour le dernier et le premier jours).

Un chemin entre les nœuds s et f représente un tour qui satisfait les contraintes sur les jours de repos, en visitant 5 nœuds W (jours travaillés obtenu de la construction de vacations journalières) et 2 nœuds F (jours repos).

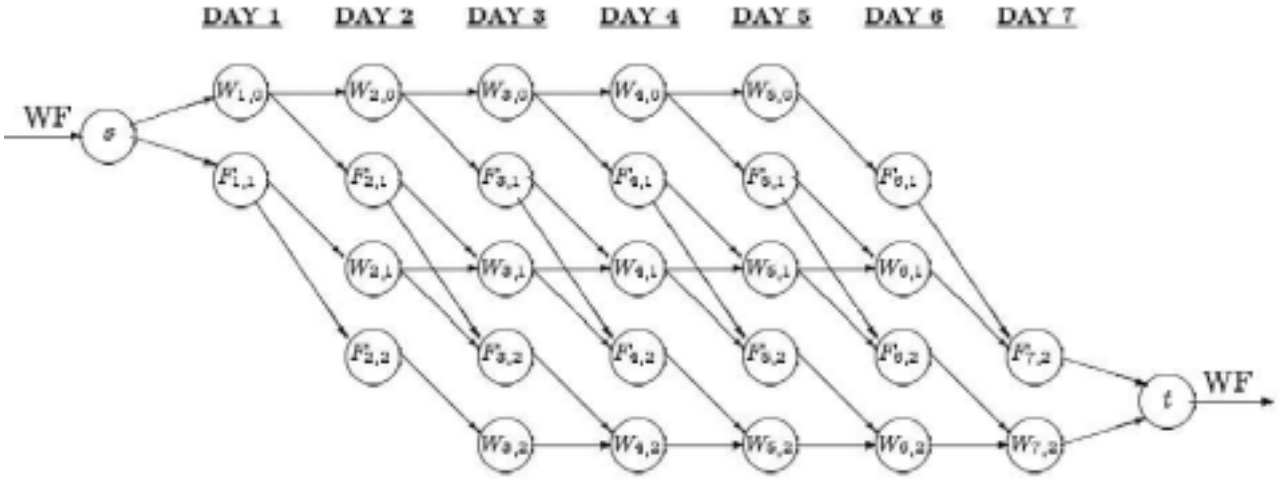


Figure 2.19 Modèle de flot hebdomadaire, repris de [CGL01]

Un arc est représenté entre deux nœuds si la transition correspondante est admise. Par exemple le flot du nœud $F_{3,1}$ to $W_{4,1}$ correspond au nombre des agents qui ont leur premier jour de repos le jour 3 et qui travaillent le jour 4.

tel-00004380, version 1 - 29 Jan 2004

2.3.3 La construction intégrée des vacations et tours

[BJ00] proposent un modèle implicite capable de traiter des intervalles au quart d'heure sur un horizon d'une semaine. Le nombre de variables dans le modèle explicite correspondant serait de l'ordre du milliard ! Il exploite l'idée de « time bands ». Au cours d'une même semaine, un salarié ne doit démarrer sa journée que dans une plage de temps. D'une part, cette condition génère des plannings très appréciés par tous et d'autre part, elle permet une réduction énorme de la combinatoire.

2.4 LA CONSTRUCTION DE TOURS : MODELES PPC

Nous présentons dans ce paragraphe quelques généralités sur la construction de tours avec des modèles PPC. Ce sujet sera traité en détails au chapitre 3.

2.4.1 Modèles Primal et Dual

La construction de tours utilise le modèle suivant : les variables de décision sont des affectations d'un employé un jour donné et les valeurs possibles sont les horaires. Voir par ex. [CGL95], [Heu96] ou [Par98].

X(t, e) = k si le jour t l'employé est affecté à l'étiquette k ou à l'étiquette repos.
Besoins(t, k) = Nombre d'étiquettes nécessaires le jour t

L'avantage de ce modèle est que la contrainte d'unicité est implicitement respectée : chaque salarié a une et une seule affectation par jour. Les outils PPC offrent les différentes contraintes suivantes :

- **all_different** ($[X_1, \ldots, X_N]$), qui exige que toutes les variables X_i aient des valeurs différentes deux à deux.
- **at_most**.(M, $[X_1, \ldots, X_N]$) qui exige que la valeur maximale des variables X_i soit égale ou inférieure à M.
- **cumulative, among** et **sequence** seront présentées dans le glossaire.

La contrainte de charge est modélisée avec la contrainte globale **among** pour le jour j :

among([Min, Max], [X(e_1, i), …, X(e_N, i)], Zeros $_N$, [k], all)

Cet appel spécifie que le nombre de variables dans la liste d'affectations pour le jour j pouvant prendre la valeur k doit être compris entre les bornes Min et Max.

En général, les outils PPC proposent des propagations locales de type '**si … alors**'. Par ailleurs [MGS96] propose une intégration des règles et des contraintes pour résoudre les problèmes d'affectation.

Le modèle dual a été proposé par [Tsa93] où les variables représentent des horaires et les valeurs représentent les salariés:

Y(t, k, i) $\in$ Employés, pour i = 1 , …, Besoins(t, k)
All_different({ Y(t, k, 1), …, Y(t, k, Besoins(t, k)) }), $\forall$ (t, k)

tel-00004380, version 1 - 29 Jan 2004

[MGS96] proposent une application utilisant ces variables, comme le montre le planning suivant :

t	k	i				
Etiquette	Poste	Jour 1	Jour 2	Jour 3	…	Jour N
Matin	Responsable	Qui ?				
Matin	Infirmière DE 1					
Matin	Infirmière DE 2					
Matin	Aide Soignante 1					
Matin	Aide Soignante 2					
Soir	Responsable					
Soir	Infirmière DE 1					
Soir	Aide Soignante 1					
Nuit	Responsable					
Nuit	Infirmière DE 1					

Figure 2.20 Un planning avec des variables dual

Initialement, le domaine des variables contient tous les salariés disponibles ce jour là et ayant les compétences requises. La taille du domaine des variables indique la difficulté d'affectation de cet horaire/poste/jour. Certaines variables peuvent avoir un domaine réduit à une seule valeur : il faut alors réaliser l'affectation afin de conserver la consistance du problème.

2.4.2 La construction simultanée des repos et tours

Pour résoudre en un seul processus le calcul des journées repos et l'enchaînement des vacations sur la semaine ou le mois, les applications Gymnaste et Equitime décrites au chapitre 3 effectuent ce calcul simultané. L'objectif est de proposer des affectations de journées (y compris le repos hebdomadaire) de façon équitable pour tous les salariés.

2.4.3 Une comparaison des modèles PLNE et PPC

Les paragraphes précédents ont présenté des modèles PLNE et PPC pour construire des vacations et des tours. Dans ce paragraphe, nous présentons des observations générales sur ces deux types de modèles.

Richesse sémantique

Les variables PPC sont du type X(e, t) = k, alors que les variables PLNE sont X(e, t, k) = 1 ou 0 (où l'employé e, t représente la date et la vacation k).

Les variables PPC sont plus proches des pensées du planificateur et de sa terminologie. La valeur des variables PPC, est plus riche sémantiquement que les valeurs booléennes des variables PLNE. Ainsi le modèle PPC a besoin de moins de variables, dans le rapport 1 pour |Vacations|. En plus, les contraintes « d'affectation unique » par salarié par jour sont implicitement respectées.

tel-00004380, version 1 - 29 Jan 2004

Avec des variables plus *riches*, la PPC permet de proposer des outils de contraintes très expressifs, qui dépassent les limites des inéquations linéaires de la PLNE. Les algorithmes de consistance ont une sémantique beaucoup plus proche du problème à résoudre que celle de l'algorithme du simplexe par exemple. « Cela tient sans doute aux origines Intelligence Artificielle de la PPC », comme l'écrivait Partouche [Par98].

Avec des variables plus *riches*, la PPC admet des variables dual en construction de tours : elles permettent des déductions supplémentaires, lorsque des contraintes portant sur ces variables sont applicables.

Anonyme ou individuel

Les modèles PLNE sont en général anonymes, et calculent le nombre de salariés dans des groupes homogènes qui sont pris en compte par les modèles. De ce fait, ils sont capables de traiter un grand nombre de salariés. Par ex. des salariés temps plein et à temps partiel, des salariés prenant une pause débutant à midi, etc.

Par contre, les modèles PPC sont en général individuels et ainsi peuvent prendre en compte les spécificités de chaque salarié. Suivant la situation locale de résolution, il est possible de programmer des déductions spécifiques, à la manière des règles.

> Ex : Si Agent 1 travaille le samedi, alors il sera de repos le lundi suivant.
> (uniquement pour cet agent)

Ce constat ne met pas en cause les techniques de résolution, mais découle des origines intellectuels de leurs concepteurs. L'origine IA des concepteurs pratiquant la PPC donne la priorité au traitement des individuels. L'origine de la RO étant liée à l'effort de la guerre, les concepteurs donnent la priorité aux masses.

PLNE : la recherche des solutions entières

Il est connu qu'en général, les solutions sont continues. Pour obtenir des solutions entières, il faut exécuter des algorithmes de recherche. Les matrices uni-modales admettent des solutions entières.

Optimisation

Les modèles PLNE donnent la possibilité d'optimiser une fonction de coût, à l'insu des modèles PPC. Généralement, les chercheurs utilisent des fonctions économiques en termes de salaires ou primes. Cependant, il est notoire que la formulation de cette fonction est très difficile car elle est complexe *politiquement*[4]. Il n'est pas évident de comparer le coût d'un travail de nuit pour une personne très expérimentée au travail d'un débutant le week-end. Cette fonction ne peut pas prendre en compte ni la productivité ni la motivation des individus, simplement parce que le modèle est anonyme.

Pour le problème de programmation des cours à l'université, [KW92] a écrit :

[4] Cela constitue des problèmes mal définis, décrit dans le Glossaire. Ill-defined problems.

tel-00004380, version 1 - 29 Jan 2004

"The objective function approach is generally unsuitable for problems of this complexity due to the difficulty or impossibility of quantifying some of the factors".

Faiblesse de la propagation locale

Caseau et al. écrivaient [CGL95] :

"Timetabling problems are hard to solve with constraint logic programming. After reporting impressive successes for job-shop scheduling, CLP users have tried to address timetabling problems with far less success. The key is that local propagation is the right tool for job-shop scheduling, whereas timetabling scheduling is dominated by the combinations of global constraints that cannot be solved efficiently by local propagation..".

tel-00004380, version 1 - 29 Jan 2004

2.5 LA CONSTRUCTION DE TOURS CYCLIQUES

Connaissant le nombre de vacations par jour, trouvées par la *construction des vacations*, on peut concevoir un cycle qui fournit les dites vacations tout en respectant les règles d'enchaînement des vacations et autres critères.

En déroulant le cycle, on obtient ainsi un planning prévisionnel *théorique*. Ce planning peut être modifié manuellement pour s'adapter aux congés annuels. Cela donne le planning prévisionnel *ajusté*. Dans certaines entreprises, il peut avoir plusieurs niveaux de plannings prévisionnel ajusté afin de tenir en compte des événements initialement non prévus, qui se déclarent au fur et à mesure.

2.5.1 Types de cycles

Les cycles peuvent être regroupés par leurs durées :

- Des cycles journaliers utilisés au ministère de la justice seront cités en exemple : une séquence de N vacations incluant des jours de repos hebdomadaire. Les salariés bénéficient de peu de week-ends sur un an. Ex. Cycle de 5 et de 6 jours :

Cycle 5 jours	Matin	Soir	Nuit	Repos	Repos	
Cycle 6 jours	Matin	Soir	Soir	Nuit	Repos	Repos

  Le repos après le travail de nuit est un repos obligatoire.

- Des cycles hebdomadaires (la période étant un nombre de semaines entières) sont très souvent proposés, car cela permet de caler les repos hebdomadaires sur les week-ends.

  Un Cycle Hebdomadaire de N semaines est défini par une grille de codes horaires sur N semaines. Prenons comme exemple un cycle de 5 semaines proposé par [LNB80], avec une de chaque vacation M=matin, S=Soir, N=Nuit par jour (avec RH=Repos hebdomadaire).

Semaine	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
1	M	M	M	RH	RH	S	S
2	S	S	RH	RH	N	N	N
3	N	N	N	N	RH	RH	RH
4	RH	RH	RH	M	M	M	M
5	RH	RH	S	S	S	RH	RH
M	1	1	1	1	1	1	1
S	1	1	1	1	1	1	1
N	1	1	1	1	1	1	1

Figure 2.21 Cycle hebdomadaire de [LNB80]

  Pour un salarié, son planning se lit de gauche à droite et du haut en bas. Arrivé au dimanche de la dernière semaine, il se poursuit au lundi de la première semaine. Un cycle de N semaines est souvent effectué par au moins N salariés (éventuellement +1 pour les congés annuels). Ainsi, pour un jour de la semaine, il y a un salarié sur chaque ligne, donc le total de présence est 1 pour la vacation M, 1 pour S et 1 pour Nuit. Ceci représente la charge hebdomadaire.

- Un cycle mensuel est difficile à concevoir, car chaque mois n'a pas la même durée.

tel-00004380, version 1 - 29 Jan 2004

- Un cycle annuel est envisageable, car il y a des contraintes légales pourtant sur l'année (heures annualisées, nombre de jours de congés ou RTT). On peut prévoir des semaines hautes et des semaines basses et utiliser des contraintes sur le volant des heures supplémentaires. Un cycle annuel est un cycle hebdomadaire qui s'exécute en un nombre entier de fois par an, par exemple à 13 (4 fois par an) ou à 17 semaines (3 fois, la dernière semaine se planifie manuellement pour les fêtes de fin d'année).

Un cycle peut être qualifié de rotation en avant (selon la séquence matin – soir – nuit) ou en arrière (nuit – soir – matin). Le premier cycle est très utilisé parce qu'il respecte mieux le biorythme des salariés.

Dans la pratique, des adaptations locales sont applicables par exemple pour traiter une surcharge temporaire [Lap99]. S'il faut un S de plus le samedi, le matin du samedi est retardé de 4 H, et la nuit est avancée de 4 H. S'il faut une matinée M et un Soir S de plus du lundi au vendredi, on peut juxtaposer au cycle de 5 semaines de la précédente figure, un deuxième cycle de 2 semaines :

Semaine	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
1	M	M	M	M	M	RH	RH
2	S	S	S	S	S	RH	RH
M	1	1	1	1	1	0	0
S	1	1	1	1	1	0	0

Figure 2.22 Cycle hebdomadaire de [Lap99]

2.5.2 Des contraintes sur les cycles

De nombreux travaux portent sur la conception automatique de cycles par ex. [Tré76], [LNB80], [NB92], [Mus00]. Comme le cycle est vécu sur un horizon relativement long, la qualité du cycle doit être travaillée finement. Les conditions suivantes doient être satisfaites d'après [LNB80] :

1. Un changement de vacation se fait seulement après un repos hebdomadaire
2. Le nombre de jours travaillés consécutifs est limité : au moins 2 ou 3 et au plus 6 ou 7 jours.
3. De la même façon, le repos hebdomadaire doit être d'au moins 2 et pas plus de 6 ou 7 jours.
4. Les périodes longues de travail doivent être succédées de longues périodes de repos.
5. Il doit avoir un maximum de week-ends entiers de repos hebdomadaire
6. Les week-ends en repos doivent être *bien* distribués sur le cycle.

D'autres préférences peuvent se rajouter : le travail de nuit ne doit pas être interrompu par des courtes séquences de travail de jour. Des changements fréquents entre le travail de nuit et de jour sont difficiles à assimiler.

[Mus00] relaxe la contrainte 1, mais rajoute d'autres contraintes:

1. Contrainte sur la succession de trois vacations, définie par un tableau à 3 indices.

tel-00004380, version 1 - 29 Jan 2004

2. La longueur de chaque suite de vacations identiques doit respecter des bornes minimum et maximum (suivant la vacation)
3. La longueur de chaque suite de vacations de travail doit respecter les seuils

Il est certain que la génération acyclique d'un planning ne respecte pas les mêmes impératifs. Un tel planning ne peut pas servir dans le cadre d'un planning cyclique. Un tel planning sur 12 semaines proposé par Partouche n'avait qu'un seul week-end au repos. Des séquences de 6 jours de travail n'ont pas de périodes de repos allongés.

Semaine	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
1	S	S	S	S	N	N	R
2	R	M	M	M	M	S	S
3	R	R	N	N	R	R	M
4	M	M	M	M	R	R	M
5	M	M	R	R	S	S	N
6	N	N	R	R	M	M	M
7	M	M	R	R	M	M	M
8	M	R	R	M	M	M	M
9	M	R	R	R	S	S	R
10	R	M	M	M	M	M	M
11	R	R	M	M	M	M	M
12	R	R	S	S	R	**R**	**R**
M	5	5	4	5	6	5	7
S	1	1	2	2	2	3	1
N	1	1	1	1	1	1	1
R	5	5	5	4	3	3	3

Figure 2.23 Planning de 12 semaines proposé dans [Par98]

2.5.3 La génération de cycles

Un processus semblable à celui décrit ci-dessus a été proposé dans [LNB80], [BW90].

1. Estimation de la taille minimale de l'équipe. Ceci donne la longueur du cycle.

   Diverses formules ont été proposées par plusieurs auteurs sous des hypothèses différentes.

2. Définition des configurations admissibles de journées travaillées + journées au repos

3. En tenant compte des bornes sur le nombre de jours travaillés ou de repos, on identifie les périodes de repos admissibles et on génère toutes les possibilités suivant les jours de la semaine.
   On n'en retient que les possibilités qui permettent de satisfaire les besoins annoncés par jour de la semaine. En évaluant chaque configuration, on favorise celles avec week-ends au repos.

4. Affectation des périodes travaillées

tel-00004380, version 1 - 29 Jan 2004

[BW90] propose un modèle qui intègre les étapes 3-5 en une seule, permettant de trouver des solutions optimales. Ce modèle définit des nœuds pour chaque jour de l'horizon du cycle. Un arc connectant les nœuds j et k correspond à une période de travail ou de repos, débutant le jour j et terminant le jour k. Pour un problème à M codes horaires, il peut avoir M+1 arcs entre deux nœuds quelconques (y compris le code horaire repos).

Le modèle [Mus00] admet des séquences de vacations contenant plus d'un type de vacation. La solution proposée exploite l'énumération implicite, en prenant soin de générer qu'une seule fois des plannings obtenus en faisant une rotation.

2.5.4 Le déroulement de cycles avec relaxation

Aucun des travaux actuels ne traite l'application d'un cycle dans la pratique. Le cas typique est un déroulement théorique, en ignorant toute absence prévue comme les congés annuels.

Le problème posé est celui de la relaxation pratique d'un cycle dont la définition mathématique est connue parfaitement. Ce sera décrit plus en profondeur au chapitre 4.

Presque 20 ans après sa première communication dans la planification cyclique, Laporte [Lap99] déplore que les méthodes actuelles ne permettent pas de donner des plannings qui tiennent en compte des problèmes de relaxation.

tel-00004380, version 1 - 29 Jan 2004

2.6 METHODES DE RECHERCHE STOCHASTIQUE

Un problème de satisfaction de contraintes peut être résolu par des méthodes de recherche arborescente (donc non-déterministe) et stochastique. Ces algorithmes utilisent des heuristiques pour

- Choisir le point de départ
- Sélectionner l'ensemble de points dans l'espace de recherche (le voisinage)
- Choisir le prochain point d'exploration (qui paraît le plus intéressant pour la poursuite de la recherche)

Des méthodes de recherche stochastique sont les algorithmes gloutons, algorithmes génétiques (GA), le recuit simulé ou le tabou. Hormis les GA, ces méthodes maintiennent une seule solution à la fois, et naviguent à petits bonds à travers l'espace de recherche. Un GA peut manipuler plusieurs solutions à la fois et effectue des sauts de portée non-déterminée. Cela donne un caractère non-local aux GA, et le rend plus consistant dans l'obtention des solutions de bonne qualité.

2.6.1 Algorithmes génétiques

Un algorithme génétique GA est un algorithme de recherche locale stochastique qui emprunte le concept de sélection naturelle des espèces pour trouver ainsi des individus le plus apte [Hol76]. Les GA recherchent des solutions optimales, en manipulant des solutions actuelles. Par exemple, ils combinent des éléments de deux bonnes solutions pour en créer une troisième, qui peut représenter un point très éloigné des solutions parents, dans l'espace des solutions.

Méthode de base

i. Le codage d'un chromosome : L'ensemble des solutions retenu par GA sous la forme de *chromosomes,* codés comme un vecteur d'entiers non-négatifs. Chaque chromosome est une collection de blocs (appelés des *gènes)* qui constituent une solution. Le codage informatique des solutions doit faciliter les différentes tâches de l'algorithme génétique.

ii. L'opérateur de CROISEMENT : A partir de deux chromosomes parents et d'un masque constitué d'un vecteur de n éléments $\in$ { 0 |1 }, cet opérateur permet de générer un nouveau chromosome en prenant le gène du premier parent lorsque l'élément correspondant dans le masque est 1 et du second lorsqu'il vaut 0.

iii. L'opérateur de MUTATION : A partir d'un chromosome, on modifie une partie de la solution de façon aléatoire. Par exemple, on supprime un nombre de salariés, à partir des éléments sélectionnés aléatoirement (ex. $X_i = X_i - 1$), puis on augmente le nombre de salariés dans les autres gênes, afin de rendre la solution réalisable en terme de compétences disponibles.

iv. L'algorithme

1. On génère une population initiale de 100 membres, de façon aléatoire.
2. On applique des opérateurs de croisement ou de mutation un nombre déterminé de fois. Les chromosomes résultant sont remis dans la population s'ils présentent un intérêt certain. Ceux qui sont « faibles » seront éjectés de la population.
3. Si le critère d'arrêt n'est pas atteint, alors on réitère l'étape 2.

tel-00004380, version 1 - 29 Jan 2004

Problèmes des GA

Epistasies : Le succès de l'application des GA est souvent attribué à la validité de l'hypothèse des blocks de construction[5]. Cette hypothèse stipule que l'opérateur de croisement doit pouvoir combiner des bonnes solutions partielles (ou blocks) pour former des bonnes solutions complètes. Essentiellement l'épistasie est présente si l'aptitude d'une solution n'est pas une combinaison linéaire du taux de l'aptitude de ses éléments. L'épistasie est une mesure de la non-linéarité de la relation entre le codage et la fonction d'aptitude.

Chaque gêne dans un chromosome peut correspondre à une variable dans un CSP. D'après [Lau98], le problème courant est que les différents gènes ne sont pas indépendants. Les contraintes influencent les valeurs simultanées des gènes, ainsi que l'évaluation des chromosomes.

Validité des opérateurs : l'opérateur de croisement est à la base de la genèse de meilleures solutions. Suivant le codage des plannings en blocs, il se peut que le croisement de deux plannings engendre des plannings qui violent des contraintes dures. Par exemple, pour le problème de construction des tours, si un gène représente la suite des affectations de vacations dans un mois pour un salarié, joindre deux bouts de plannings peut résulter en une suite de plannings interdits (ex. Nuit suivi de Matin).

[Eib01] proposent deux méthodes pour résoudre des CSP avec des GA.

- Traiter toutes les contraintes indirectement : la fonction d'aptitude dépend de la violation des contraintes. A chaque itération, la population des solutions devient de plus en plus admissible.

- Traiter toutes les contraintes directement : la violation des contraintes n'affecte pas la fonction aptitude. On y parvient par l'une des approches suivantes :
    - L'élimination des candidats inadmissibles : cette méthode très générale est très peu efficace (elle ressemble à la « génération et test »)
    - Réparation des candidats inadmissibles : lorsqu'elle est possible, cette méthode donne de bons résultats en général, d'après [Eib01].
    - Concevoir des opérateurs qui respectent les contraintes : cette méthode est la plus intéressante mais aussi la plus difficile à réaliser. L'opérateur de mutation paraît bien adapté, car il mélange des chromosomes admissibles des solutions ; le croisement crée de nombreuses violations.

- Traiter une partie des contraintes directement et l'autre indirectement

Diversité des individus dans la population : un GA doit veiller à ce que la population soit suffisamment diversifiée afin de converger vers des bonnes solutions. En conséquence, il faudrait des mesures de similitude ou de diversité. La taille de la population doit être grande.

Résultats de Cai et Li

[5] Building Block Hypothesis

tel-00004380, version 1 - 29 Jan 2004

Une méthode GA [BC96] a été utilisée pour résoudre le modèle de Cai et Li [CL00]. Il y a deux compétences, et le personnel est divisé en 3 classes : ceux qui ne possèdent que la première compétence, ceux la seconde compétence et finalement ceux qui possèdent les deux (dont la disponibilité est supposée inférieure à B). Pour la classe de salarié j=1,2,3, soit n_j le nombre de solutions faisables dans l'ensemble S_j. Pour une solution faisable i, on dénote :

X_i le nombre de salariés du premier groupe affecté aux tâches de la compétence 1,
Y_i le nombre du second groupe affecté aux tâches de compétence 2,
Z'_i le nombre du troisième groupe affecté aux tâches de compétence 1,
Z''_i le nombre du troisième groupe affecté aux tâches de compétence 2,

$$S = \{X_1, X_2, \ldots, X_{n1}, Y_1, Y_2, \ldots, Y_{n2}, Z'_1, Z'_2, \ldots, Z'_{n3}, Z''_1, Z''_2, \ldots, Z''_{n3}\}$$

où n = n1+n2+2n3, le nombre total de solutions, X,Y,Z' sont des gênes. Chaque élément du vecteur représente une des solutions faisables pour chaque type de personnel.

Utilisés dans le contexte de personnel à multiples compétences, sur un horizon d'une semaine, des intervalles d'une heure. Il y a des vacations de nuit (23H-07H) et de jour : 7H-16H, 8H-17H, 9H-18H, 10H-19H, 11H-20H, 12H-21H, 13H-22H, 14H-23H, 15H-0H. La durée des pauses pendant les vacations de jour, est d'une heure après 4 heures de travail. Les horaires hebdomadaires doivent avoir au maximum 3 nuits consécutives, avec une journée de repos compensateur et suivi du repos hebdomadaire. Les vacations journalières sont maximums de 6 jours de suite, suivies du repos hebdomadaire (un jour quelconque de la semaine). Pour chaque salarié, une semaine jour commence toujours à la même heure. Sur une station Sun SPARC, l'implémentation en C prend 8 à 10 minutes pour produire des solutions faisables, avec un total de 593 salariés.

Conclusion

Par rapport aux autres méthodes qui recherchent un chemin à la fois, en considérant plusieurs solutions simultanées le GA est susceptible de trouver des solutions plus intéressantes. Pour la génération de la population initiale de solutions, il faut utiliser des méthodes supplémentaires, par exemple une méthode gloutonne.

Il est difficile de s'assurer qu'à partir de solutions consistantes, les opérateurs produisent des solutions filles consistantes. Plus les contraintes sont complexes, plus l'élimination des solutions inconsistantes prend du temps.

tel-00004380, version 1 - 29 Jan 2004

2.6.2 Algorithmes mimétiques

[Daw76] a introduit l'idée de mime comme une alternative au concept de gène. Non seulement le matériel génétique est transmis d'une génération à la suivante, mais aussi des idées et concepts. Le concept de mime est assimilé comme une unité d'information passée entre des individus. Quand un gène est passé aux descendants, il n'est pas altéré : les individus n'ont aucun contrôle sur leur composition génétique. Par contre, l'individu peut adapter le mime à son propre environnement.

La notion d'un algorithme mimétique a été introduit dans [MN91] pour décrire des algorithmes incorporant la recherche locale (par ex. hill climbing) suite à une mutation. Afin de réduire le temps d'évaluation, les algorithmes exploitent la différence entre les solutions finale et initiale, par ex. [Kra96].

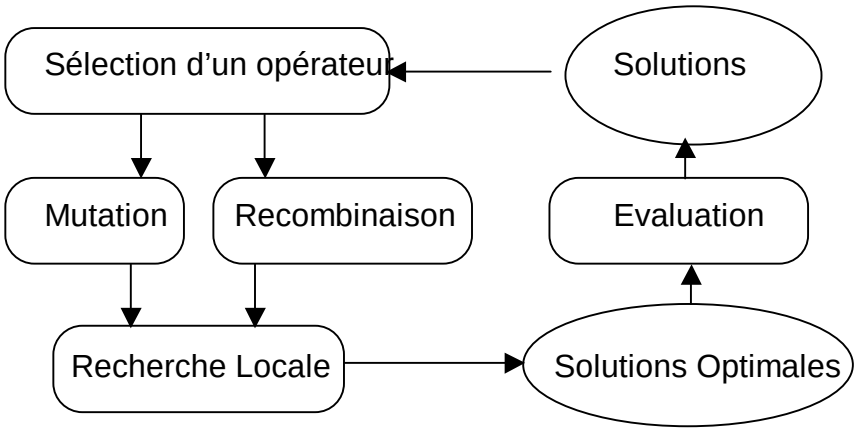


Figure 2.24 Le schéma global d'un algorithme mimétique[6]

2.6.3 Le recuit simulé

Le processus du recuit est le lent refroidissement d'un solide. Le premier algorithme permettant de simuler ce processus a été proposé initialement par [MRT53]. Plus tard, Kirkpatrick *et al.* établissent l'équivalence entre le processus de recuit et la résolution d'un problème d'optimisation combinatoire. L'équivalence est obtenue en faisant une analogie entre l'état des particules et les solutions d'une part, et entre l'état d'énergie et le coût d'autre part.

La méthode de Metropolis génère une séquence d'états qui illustre l'évolution du corps de l'état liquide à l'état solide, en observant l'équilibre thermique à chaque instant. Le prochain état est obtenu par une petite perturbation de l'état courant, générée de façon aléatoire, et un critère d'acceptation. Ainsi une perturbation n'est pas acceptée systématiquement. Cela dépend de la différence d'énergie entre l'état courant et le nouvel état, et la température ambiante.

Une configuration est une solution et une transition de i à j représente une perturbation en configuration i, la transformant en configuration j. Soit Fi et Fj le coût des configurations i

[6] Burke E.K. , Newall J.P. , Weare R.F. , A mimetic algorithm for university exam timetabling, in The Practice and Theory of Automated Timetabling: selected papers from the 1st Int'l. Conf. on the Practice and Theory of Automated Timetabling, Napier University. Lecture Notes in Computer Science Series, Vol. 1153, pp. 241-250, 1995.

tel-00004380, version 1 - 29 Jan 2004

et j, et T représente la température. La probabilité d'accepter la transition de i vers j est définie par le critère d'acceptation de Metropolis :

$$A_{ij}(T) = 1 \quad \text{si } F_i \geq F_j \qquad (8)$$
$$= e^{(Fi - Fj)/T} \quad \text{sinon, avec } T > 0$$

L'algorithme RS classique est une séquence des algorithmes Metropolis qui est évaluée pour une suite décroissante de température. Par conséquent, les caractéristiques globales du RS est que la probabilité d'acceptation des transitions de coût croissant, décroît avec la baisse de température. Quand la température tend vers zéro, la détérioration du coût n'est plus acceptée.

La nature stochastique du RS est décrite par le critère d'acceptation et une fonction de voisinage. Ce dernier définit pour chaque configuration i, l'ensemble de configurations voisines N(i) qui peuvent être atteintes avec une seule transition. Ainsi de deux choses l'une : a) les configurations qui ne sont pas voisines auront une probabilité nulle de devenir la prochaine configuration ; b) les configurations qui sont voisines auront une probabilité non nulle de succéder la configuration courante.

Algorithme du recuit simulé (α, T°, N, P)

```
α = taux de refroidissement entre 0,5 et 0,99
T° = température initiale
N = nombre de paliers de températures
P = nombre d'étapes par palier

Soit
    S = Solution initiale générée de façon aléatoire
    T = T°
    Tant que i < N faire
        J = 0
        Tant que J < P faire  'Une étape
            S' = Générer une transition à partir de S
            Δ E = E_S' – E_S
            Si Δ E ≤ 0 Alors
                S = S' // Accepter la transition
            Sinon
                Proba = exp^(– Δ E/T)
                Si aléatoire(0,1) < Proba Alors
                    S = S'  // Accepter la solution
                Sinon
                    // Rejeter la solution
                Fin Si
            Fin Si
        Fin
        T = α * T
    Fin
    Solution finale = S
```

Figure 2.25 L'algorithme du recuit simulé

Le générateur de nombres aléatoires entre 0 et 1 utilise une loi de distribution uniforme.

tel-00004380, version 1 - 29 Jan 2004

2.6.4 Recherche Locale

Méthode d'échanges locaux

[JBS94] propose un algorithme simple qui construit un ensemble de tours et améliore la qualité en faisant des échanges. Pour chaque jour de l'horizon, les vacations sont triées par ordre croissant de l'heure de début. Un ensemble de tours est généré en tenant compte des repos hebdomadaires. Pour chaque tour, calculer la variance de l'heure début des vacations.

Soit i* le tour dont la variance est maximum sur l'ensemble des tours. Considérons l'ensemble des tours avec au moins une vacation commençant plus tôt que la vacation débutant le plus tard dans i* et avec une vacation commençant plus tard que la vacation débutant le plus tôt dans i*. Pour chacun tour j, identifier la séquence d'échange de vacation permettant de réduire la somme des variances de i* et j. Répéter cette étape un nombre donné de fois, tant que la variance de i* continue à diminuer.

Des variantes de ces méthodes réduisent la différence en heure de début des vacations dans un tour.

Méthode de recherche locale avec redémarrage aléatoire

Les méthodes de recherche locale ont la tendance d'être piégées par des optima locaux. [MK01] propose une méthode permettant de sortir de cette impasse : lorsque le nombre de retour-arrière dépasse un seuil préétabli, la recherche systématique est arrêtée autour de la solution partielle courante, un autre point de l'espace de solutions est généré de façon aléatoire et la recherche locale est redémarrée. Inspiré de [GK98] et incorporant des techniques de *forward-checking* et vérification des consistances, une amélioration de l'ordre d'une grandeur a été rapportée dans [MK01].

2.6.5 Recherche Tabou[7]

Initialement proposé par F. Glover [Glo89], la recherche tabou est un méta-heuristique de recherche locale à partir d'une *solution initiale*, dans un *voisinage prédéfini*. L'algorithme parcourt le voisinage de l'espace de solutions depuis la solution initiale, de façon non-ordonnée. Le voisinage est défini par un ensemble de types de transitions d'état. On ne retient que la solution qui possède une évaluation meilleure à celle obtenue auparavant.

Afin d'éviter de revenir sur une solution déjà visitée, le système maintient une liste de transitions récentes qui sont déclarées interdites, d'où le nom *tabou*. Cette liste fonctionne comme une pile, les éléments seront tabou pendant un nombre K d'itérations. Ainsi, faute de trouver une solution avec une meilleure évaluation, on peut accepter en une qui n'est pas sur la liste tabou.

Cette technique dispose d'un mécanisme permettant de retenir une solution malgré le tabou : un *critère d'aspiration* souvent utilisé est basé sur le taux d'amélioration de la fonction d'évaluation. Si la solution est « très » bonne par rapport à la solution courante, on l'accepte malgré son inscription sur la liste tabou.

[7] Taboo Search

tel-00004380, version 1 - 29 Jan 2004

Un *mécanisme de diversification* est nécessaire pour réduire la possibilité d'être piégé dans un optimum local.

Algorithme de recherche tabou (f, x*)

Entrée : Solution admissible x*
Fonction d'évaluation z* = f(x*)
Résultat : x* est la meilleure solution avec la valeur z*

Soit la solution à l'étape courante k : $x^k := x^*$
Initialiser la liste tabou L ← ∅
Tant que Critère d'arrêt insatisfait
Début
(1) Recherche dans le voisinage de la solution courante
Soit $V(x^k)$ = l'ensemble des solutions voisines de x^k,
obtenues avec les transitions admissibles à partir de x^k
Soit $V_L(x^k) \subseteq V(x^k)$ obtenu en éliminant les solutions interdites par la liste tabou

Certaines solutions ne sont pas éliminées si le critère d'aspiration est satisfait
par exemple si son évaluation est très intéressante par rapport à z*

Choisir la solution $x° \in V_L(x^k)$ qui minimise la fonction f
Si $f(x°) < f(x^*)$ alors faire $x^{k+1} \leftarrow x°$

(2) Gestion de la liste tabou L
Si la liste a atteint la longueur maximale, éliminer l'élément la plus ancienne
Ajouter dans L l'information sur la transition $x^k \rightarrow x^{k+1}$.
Fin

Figure 2.26 L'algorithme de recherche tabou

Pour résoudre le problème de l'emploi du temps, [SS95] utilise le codage matriciel $M_{i,k}$ qui contient le nom de la classe du professeur j à la période k. Les voisinages proposés sont :

- Echanger deux cours pour un même professeur
- Déplacer un cours à une autre période

[CST00] utilise une liste tabou de taille variable. A son entrée dans la liste, à chaque solution est attribué un nombre aléatoire (choisi entre deux valeurs fixes K_{min} et K_{max}) celui ci représente le nombre d'itérations la solution doit rester dans la liste. Le voisinage est défini par le remplacement d'un employé i à la période j effectuant la tâche k, par un autre employé j qui ne travaille pas à la période j.

Problèmes de la recherche tabou

Comme les méta-heuristiques, la recherche tabou est une technique d'optimisation sans contraintes. Effectivement, les transitions d'un état à un autre peuvent engendrer des violations de contraintes, sauf si elles ont été conçues spécifiquement.

2.6.6 Greedy Randomized Adaptive Search Procedures

Proposé la première fois dans [FR95], le méta-heuristique **GRASP** consiste en deux étapes à chaque itération :

tel-00004380, version 1 - 29 Jan 2004

- La construction d'une solution initiale avec une fonction gloutonne aléatoire *adaptative*. La fonction gloutonne prend en compte les choix fait précédemment pour déterminer la prochaine solution.
- L'utilisation d'une procédure de recherche locale pour l'améliorer.

La meilleure solution sur toutes les itérations sera retenue comme solution finale. Bien évidemment, on peut tomber dans un optimum local dans une itération, mais le redémarrage aléatoire permet d'explorer l'espace de solutions pour en trouver une qui soit proche de l'optimum global.

Algorithme GRASP[8](MaxIterations, Seed)

Nombre maximal d'itérations est donné en entrée
Seed permet d'initialiser le générateur de nombres aléatoires

```
Solution ← 0, MeilleurSolution ← 0
Pour K=1, Max_Iterations Faire
    Solution ← GreedyRandomizedConstruction(Seed)
    Solution ← local_Search(Solution)
    MiseAJour (Solution, MeilleurSolution)
Retourner MeilleurSolution
Fin

Procédure GreedyRandomizedConstruction(Seed)
Solution ← 0
Initialiser les générateurs de nombres aléatoires avec Seed
Evaluer la solution
    Tant que Solution est incomplète
        Construire une liste de candidats (de façon adaptative à la solution partielle)
        Sélectionner aléatoirement un élément s de la liste de candidats
        Solution ← Solution ∪ {s}
        Réévaluer le coût incrémental
    Fin Tant que
    Retourner Solution
Fin procédure
```

Figure 2.27 L'algorithme GRASP d'après [RR01]

Dans l'application de cette méthode au problème de l'affectation généralisée[9], la probabilité du choix d'un candidat est en fonction du rapport b_j/a_{ij}, b_j étant la capacité maximale de l'agent j, et a_{ij} la consommation en ressource j par la tâche i si affectée à la ressource. Ce rapport est grand si la tâche i n'utilise qu'une petite partie de la capacité de la ressource j. Le candidat est admis si la contrainte de capacité n'est pas transgressée. Sinon, le premier candidat avec une capacité restante suffisante est pris. Si aucun candidat n'a de la capacité restante, on sélectionne avec une probabilité uniforme. Nous voyons que la méthode est s'adapte aux choix précédents dans la solution courante, et ignore les solutions précédentes.

Au total, des centaines voire des milliers d'itérations peuvent être exécutées, chaque démarrage exploite un point pris de façon aléatoire. Cette technique est conceptuellement

[8] Une liste d'applications GRASP est visible à l'adresse http://www.research.att.com/~mgcr, qui est géré par Rescende.
[9] Generalized Assignment Problem (GAP)

tel-00004380, version 1 - 29 Jan 2004

facile à déployer sur un grand nombre de machines en parallèle. Comme on dispose d’une solution réalisable à chaque itération, on peut arrêter le processus lorsqu’on a besoin d’une solution.

2.6.7 Conclusion sur les méthodes stochastiques

En général l’application de ces méthodes à beaucoup de problèmes de taille réelle montre qu’elles donnent des solutions de bonne qualité dans de nombreux cas. Nous pouvons faire deux remarques à propos de ces méthodes.

D’une part, n’étant pas basées sur le raisonnement logique, elles seraient difficiles à appliquer à la résolution des modèles traitant plus de types de contraintes.

D’autre part, ses résultats semblent découler du grand nombre de calculs de solutions et d’évaluation. Typiquement dans la littérature de recherche sur les GA, on parle de 500 ou 1000 itérations. Donc, elles exigent un temps potentiellement inacceptable pour un produit commercial.

tel-00004380, version 1 - 29 Jan 2004

2.7 LES ALGORITHMES D'APPROXIMATION

La théorie de la complexité (Cook en 1971 et Karp en 1972), postule que tout problème NP complet pourra être résolu dans un temps polynomial en fonction de la taille du problème, si et seulement si tous les autres problèmes NP complets peuvent être résolus dans un temps polynomial. La classe de problèmes NP complets inclut des problèmes pratiques tels que le voyageur de commerce, couverture d'ensemble, programmation en nombres entiers avec des variables limitées. Après des décennies de recherche algorithmique, des générations de chercheurs n'ont pas trouvé de tels algorithmes. Par contre, ils ont trouvé de plus en plus de problèmes de la même classe.

En conséquence, de nombreux chercheurs [Sah77], [Joh74] ont proposé des algorithmes pour résoudre les problèmes NP complets de façon presque optimale : ces algorithmes sont capables de résoudre ces problèmes dont l'évaluation est proche de la valeur optimale. Cette tendance est renforcée par le fait que les problèmes sont souvent définis de manière approximative par rapport à la réalité.
Définition : Un algorithme est appelé algorithme ε-approché pour un problème P si pour toute instance I du problème, F*(I) >0 l'évaluation d'une solution optimale à I, F°(I) >0 est la valeur générée par l'algorithme, et ε est une constante, alors l'erreur relative est :

$$\frac{|F^*(I) - F^\circ(I)|}{F^\circ(I)} \leq \varepsilon \qquad (9)$$

Pour un problème de maximisation, $0 \leq \varepsilon < 1$ et un problème de minimalisation $\varepsilon \geq 1$.

Définition : un schéma d'approximation en temps polynomial PTAS [10] pour un problème de minimisation est une famille d'algorithmes $\{A_\varepsilon ; \varepsilon > 0\}$, tel que pour chaque ε , A_ε est un algorithme $(1+\varepsilon)$ -approché avec une complexité en ε. Pour un problème de maximisation, A_ε est un algorithme $(1-\varepsilon)$ -approché

Sans entrer dans les détails de cette branche d'étude[11], il existe des PTAS pour les problèmes tel que sac à dos, voyageur de commerce sur un plan Euclidien, certains problèmes d'ordonnancement et la couverture d'ensemble.

2.7.1 La couverture d'ensemble[12]

Etant donné un ensemble U d'éléments, une famille F de sous ensembles de U : S_1 , S_2, …, $S_n \subseteq U$, avec des poids associés w_1 , w_2, …, w_n, le problème de couverture de U consiste à trouver l'ensemble $I \subseteq \{1, 2, \ldots, n\}$, dont l'union recouvre les éléments de U.

$$U = \cup_{i \in I} S_i \qquad (10)$$

qui minimise la somme $\sum_{i=1,n} w_i$..

Il y a deux méthodes possibles dans l'énumération de toutes les couvertures I. L'une consiste à construire toutes les collections, et stopper la recherche dès que l'union recouvre U. Cette méthode est qualifiée de l'énumération des sous-ensembles. L'autre

[10] Polynomial time approximation schema PTAS
[11] le lecteur pourra consulter le tutorial de 132 pages [Wil98]
[12] Set Covering Problem

tel-00004380, version 1 - 29 Jan 2004

consiste à augmenter le taux de couverture à chaque étape ; la méthode est qualifiée de l'augmentation de la couverture.

L'un des premiers **algorithmes d'approximation** pour résoudre ce problème a été proposé par [Joh74] avec une complexité de O(n log n). A chaque étape, la technique glouton consiste à ajouter le sous-ensemble S_i qui contient le plus d'éléments non encore couverts.

La proposition de Chvàtal [Chv79] donne une complexité de O(m log n) où $m = \sum_{i=1,n} | S_i |$, le résultat a le poids W qui satisfait la borne supérieure :

$$W \leq W\ opt\ .\ O(\log d), \text{ avec } d = Max\ |\ Si\ | \qquad (11)$$

Ce résultat a été amélioré dans [Hoc82] donnant le poids W

$$W \leq W_{opt}\ .\ f,$$

avec f = le plus grand nombre d'ensembles S_i qui contiennent un élément x quelconque

$$\text{Soit } F_j = \{I \mid e_j \in S_i\}, \text{ alors } f = Max\ |F_j|$$

2.7.2 L'algorithme Yehuda et Even

L'algorithme de [BE81] obtient le même résultat, mais évite l'appel à un algorithme de Programmation Linéaire. Sa complexité est proportionnelle à $m = \sum_{i=1,n} |S(i)|$.

Algorithme de Behuya et Even ()

Famille de n ensembles $F = \{S_1, \ldots S_n\}$, poids de chaque $S_i = w_i$ nombre real ≥ 0
L'ensemble universel $U = \{e_1, \ldots e_t\} = U_{i=1,n} S_i$, t = nombre d'éléments distincts

Traitement Soit $N = \{1, \ldots, n\}$, $T = \{1, \ldots, t\}$, $F(j) = \{ i \mid e_j \in S_i \} \subseteq N$

Initialisation : $\forall I \in N$, $SW(I) = w_i$
I = 0, J = T
Itération :
Tant que ($J \neq 0$) et soit $j \in J$
Choix d'un S_i :
M = min { SW(I) | $I \in F(j)$ }
Soit i=k pour SW(k) = M
Mise à jour des compteurs
$\forall I \in F(j)$, SW(I) = SW(I) – M
I = I u {k} – choix du sous-ensemble k
$J = J \setminus S_k$ – enlève toutes les périodes couvertes par cet ensemble
Fin

Sortie : $I \subseteq N$, contenant les indices de S choisis, et qui minimise $\Sigma_{i \in I} w_i$

Figure 2.28 L'algorithme de Behuya et Even

Une comparaison de 9 algorithmes pour résoudre le problème de couverture d'ensemble est proposée dans [GW97]. Les meilleurs résultats ont été trouvés par les méthodes suivantes :

tel-00004380, version 1 - 29 Jan 2004

- Algorithme Glouton (Chvàtal) : A chaque itération, on choisit la variable qui apparaît dans le plus grand nombre de contraintes non satisfaites. En cas d'égalité, la variable avec le plus petit indice est choisie.

- Algorithme Glouton-Aléatoire : A chaque itération, on choisit la variable qui procure le plus de gain. En cas d'égalité, on utilise une règle aléatoire pour choisir parmi les candidats. Les calculs sont répétés N=100 fois et la meilleure solution est retenue. Cet algorithme donne les meilleurs résultats dans la comparaison (presque toujours la solution optimale), mais la consommation en temps machine est importante.

2.7.3 Multiples couvertures d'ensemble[13]

La couverture d'ensemble ne permet que de couvrir une seule fois l'ensemble U. La construction de vacations exige la multiples couvertures car à un instant donné, les besoins en personnel sont en général supérieurs à 1.

[PSW97] analyse une généralisation du problème de couverture d'ensemble avec la classe de problèmes dénotée par ILP(k, b). Cela consiste à trouver le vecteur $\mathbf{x} \in \{0,1\}^n$ en minimisant $\sum_j x_j$, soumis aux contraintes $\mathbf{A}\,\mathbf{x} \geq \mathbf{b}$, où **A** est une matrice booléenne m x n avec au plus k fois 1 par rangée, **b** est un vecteur en nombres entiers et b est la plus petite valeur dans **b**. Lorsque k=n, b=1, on retrouve une variante du problème de couverture d'ensemble. Des algorithmes d'approximation avec un rapport $\varepsilon = (k – b + 1)$ ont été proposés, voir [HH86] par exemple.

[PSW97] propose un nouvel algorithme ε-approché pour résoudre ILP(k, b) où

$$\varepsilon = (k – b + 1) (1 – (c/m)^{d}) \qquad (12)$$

avec une petite constante c > 0, d = 1/ (k-b+1), m = nombre de rangées de la matrice **A**. L'algorithme proposé utilise la programmation linéaire comme étape intermédiaire.

2.7.4 Conclusion

Cette revue des méthodes de résolution du problème de couverture d'ensemble, permet de donner une idée des résultats obtenus dans ce domaine. La construction des vacations relève du problème à multiples couvertures d'ensemble. U est l'ensemble des intervalles de temps de la journée ; S_i sont les horaires légaux, compte tenu de la législation sur les durées de travail et de repos. La couverture de l'ensemble U traduit la présence de personnel sur la journée.

Cependant, il y a d'autres contraintes non-couvertes par ce problème :

- La multiple compétence : dans un CALL CENTER, le personnel est souvent compétent pour traiter plusieurs flux d'appels.

- Le temps partiel de certains employés

[13] Multiple Set Covering

tel-00004380, version 1 - 29 Jan 2004

- Les préférences individuelles

Compte tenu de la complexité de ces algorithmes, il n’est pas envisageable d’enrichir le modèle de planification, de tenir compte des préférences ou de l’historique.

tel-00004380, version 1 - 29 Jan 2004

2.8 LES SYSTEMES INTERACTIFS D'AIDE A LA DECISION

Cette thèse a pour finalité la création d'un système logiciel pour la création de plannings. Jusqu'à ce point, nous avons analysé les méthodes techniques pour générer automatiquement des plannings. Or la mise en œuvre de ce type de logiciel n'est pas aussi simple que celle des traitements de textes.

Ce paragraplhe propose d'explorer les modèles conceptuels d'un système logiciel, puis d'un système d'aide interactive à la décision, et enfin d'un logiciel planning.

2.8.1 Le modèle conceptuel d'un système logiciel

Un modèle conceptuel **MC** est une représentation mentale du fonctionnement du système et il indique comment les contrôles de l'interface utilisateur le modifient.

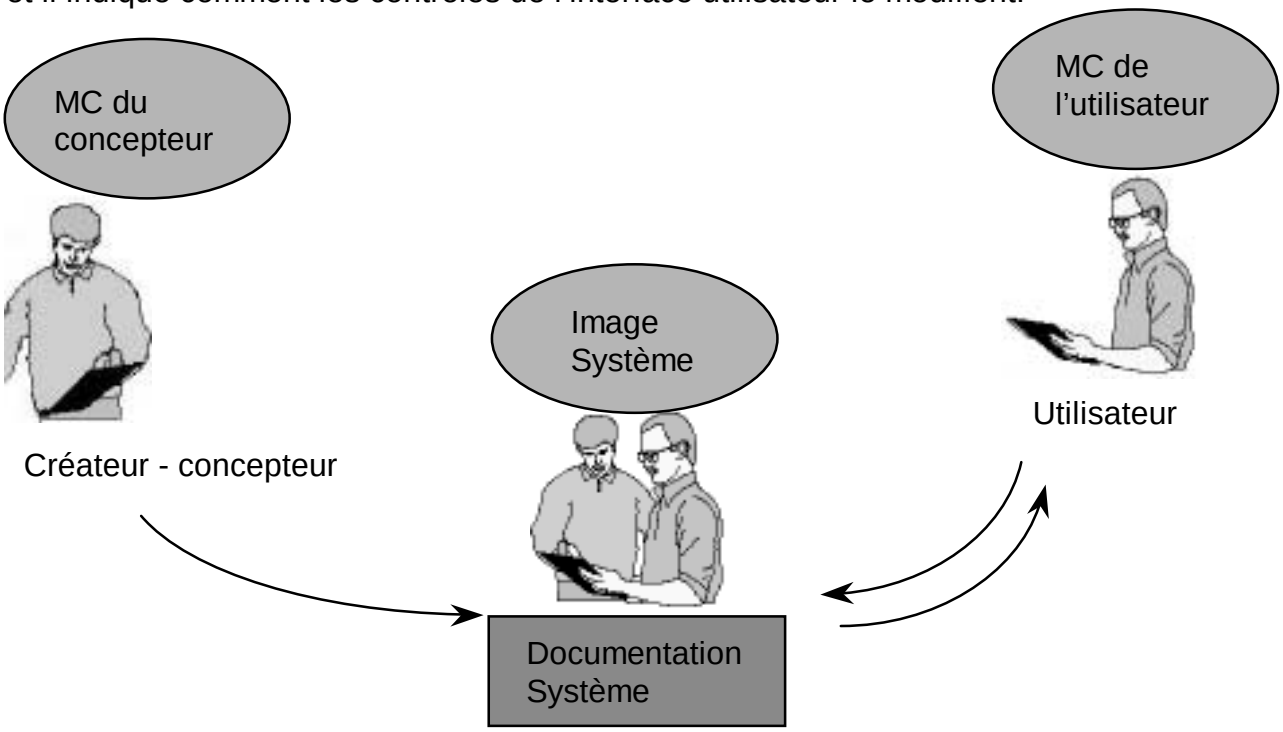


Figure 2.29 Les différents types de modèles conceptuels d'après Norman

Dans la science de communication, [Nor90] proposa trois types de modèles :

1. Modèle concepteur : un modèle conceptuel du créateur du système dans sa globalité.
2. Image Système : la représentation physique externe du système, ce que voit, entend ou ressent tout utilisateur qui interagit avec le système. Le créateur communique à l'utilisateur via l'image système.
3. Modèle utilisateur : un modèle conceptuel de l'utilisateur du système. Il le construit à partir de son expérience, des documents pédagogiques et de la formation qu'il a reçue.

Si l'utilisateur dispose d'un bon modèle conceptuel, il saura prédire les effets de ses actions, donc capable de réussir l'interaction avec le logiciel.

tel-00004380, version 1 - 29 Jan 2004

Afin que l'utilisateur puisse exploiter effectivement tout système, Norman propose que le modèle conceptuel de l'utilisateur soit proche de celui du concepteur [Nor90]. Je cite : "Important image is not the one on the screen but in the users' mind". L'utilisateur crée son modèle conceptuel à partir de son expérience, en lisant la documentation système, ou en suivant les cours. Si les modèles conceptuels de l'utilisateur et du concepteur ne coïncident pas, l'utilisateur fera de nombreuses erreurs et rencontrera des problèmes chaque fois qu'il se servira du logiciel. Il aura besoin de beaucoup de temps pour maîtriser le logiciel et sera frustré.

Un système réussi doit donner des indices visuels quant à leur fonction ou opération. Ils doivent être justes (sans exagération) et proportionnés (liés aux tâches de haut niveau et non des détails d'implantation). L'idéal est de créer des métaphores ou modèles conceptuels.

Une métaphore nous permet de voir des correspondances entre deux choses apparemment désassociées. Elle nous invite à entrevoir deux choses sous un nouvel angle. Pour Aristote le grand philosophe de la Grèce Antique, devenir un maître de métaphore est un but en soi : c'est une capacité qui ne peut s'apprendre des autres, un signe de génie, car une bonne métaphore nécessite une perception intuitive des analogies parmi des phénomènes disparates[14].

Avec des systèmes de plus en plus complexes, l'utilisation d'une seule métaphore peut atteindre ses limites. Une métaphore ne pourrait pas représenter tous les aspects du comportement du système. Plusieurs métaphores peuvent être utilisées conjointement. L'utilisateur doit être encouragé à utiliser des parties relevants de chaque métaphore. Cependant, il convient d'être attentif car de fausses interprétations de la métaphore peuvent engendre des erreurs.

[14] By far the greatest thing is to be a master of metaphor. It is the one thing that cannot be learned from others. It is a sign of genius, for a good metaphor implies intuitive perception of similarity among the dissimilar.

tel-00004380, version 1 - 29 Jan 2004

2.8.2 Les principes d'un SIAD

Un SIAD est un outil informatique qui assiste le décideur tout au long de sa prise de décision. En effet un processus de décision n'est pas entièrement automatisable [Pom92]. Il est nécessaire de garder l'utilisateur dans le processus de décision car il est capable de considérer des critères qui ne peuvent pas être explicitement modélisés dans un système. L'identification des points du processus qui peuvent être automatisés (ou traités automatiquement) constitue une étape importante dans la construction d'un SIAD. Un processus de décision dans la gestion des organisations se compose de quatre phases : [LP89]

- Phase d'information
- Phase de conception
- Phase de choix
- Phase d'évaluation du choix

Le déroulement des phases n'est pas forcément séquentiel : certaines peuvent se dérouler en parallèle. Des retours en arrière peuvent se produire notamment lors de la phase de conception ; l 'élaboration d'un scénario peut nécessiter l'acquisition d'informations supplémentaires.

Si la phase de choix relève du décideur seul, un système d'information a sa place dans les deux phases de préparation à la prise de décision et dans l'évaluation du choix. En effet, la capacité de traitement des informations des ordinateurs permet au décideur, pendant la phase d'information, d'accéder rapidement à des informations brutes ou traitées concernant la situation courante et le champ des manœuvres autorisées par exemple. Cette capacité de traitement de l'information peut être aussi utilisée lors de la phase de conception. Dans cette phase, le système d'information peut fournir des éléments d'évaluation des scénarii à l'aide d'indicateurs calculés à partir de modèles ou de procédures de calcul adaptées. La phase d'évaluation du choix correspond à une évaluation a posteriori du choix du décideur ; cette évaluation permet de corriger les petites erreurs. La détection des erreurs et des aspects à améliorer peuvent être facilités par l'apport d'informations et d'indices calculés par le système d'information [Lév89].

L'association d'un système informatique et d'un décideur permet une symbiose complémentaire. Le décideur de par sa connaissance pratique et son expérience possède un méta-modèle du processus de décision. Le SIAD, par sa capacité de traitement de l'information, l'aide à structurer le modèle. Le décideur contrôle le processus de décision et le SIAD l'assiste en effectuant les calculs standards et répétitifs sur les données. Le processus de décision s'apparente à une recherche heuristique menée par le décideur ; le système jalonne le processus de recherche à l'aide d'indicateurs et d'informations. En fonction de ces informations, le décideur continue l'exploration heuristique des actions possibles ou s'arrête si tout indique que la solution construite rencontre ses buts de façon satisfaisante [LP89].

tel-00004380, version 1 - 29 Jan 2004

2.8.3 Le modèle conceptuel d'un SIAD

Un SIAD se compose de trois modules : un module de dialogue, un module de données et un module des procédures de calcul ou modèles. Le module de dialogue est connecté aux deux autres modules, et constitue l'interface entre l'utilisateur et le reste du système.

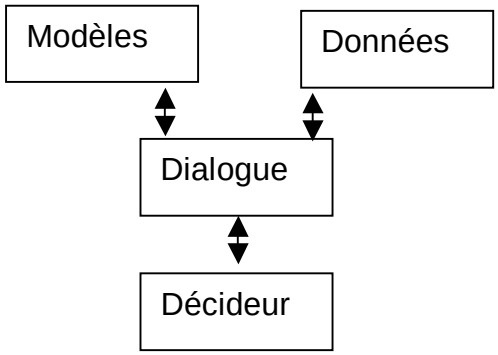


Figure 2.30 Modèle conceptuel d'un SIAD

Le module de dialogue permet au décideur d'accéder aux données et aux modèles du SIAD. Ce dernier utilise ce même module pour communiquer le résultat des manipulations effectuées par le décideur. Les échanges sont d'autant plus favorisés que les représentations des résultats, tout comme le mode de questionnement du système, correspondent aux représentations mentales du décideur. Ainsi, le décideur peut exercer son contrôle et effectuer sa recherche heuristique dans de bonnes conditions.

Le module de données assure la fonction de mémoire : il stocke non seulement les données, de façon permanente (persistante) ou passagère, mais il gère aussi l'enregistrement de données volatiles ainsi que l'effacement de ces mêmes données selon le souhait de l'utilisateur. Ces données volatiles correspondent aux résultats obtenus au cours de traitements de données. Les données que nous avons qualifiées de permanentes sont les statistiques ou autres données qui décrivent la situation courante et passée. Parmi ces données, il peut aussi y avoir des estimations concernant l'évolution de certains paramètres environnementaux.

Le module de modèles contient l'ensemble des procédures de calculs utilisées dans les différents traitements mis à la disposition de l'utilisateur. Il peut s'agir des calculs standards et de procédures de représentation de données.

tel-00004380, version 1 - 29 Jan 2004

2.8.4 Modèles conceptuels d'un logiciel de planning

Pour mémoire, nous rappelons plusieurs modèles conceptuels souvent utilisés pour les plannings muraux. Ces plannings sont des tableaux, dont un axe est l'axe du temps.

- Plannings par tâche : le diagramme de Gantt où le temps occupe l'axe horizontal et une ligne représente typiquement une tâche. Dans des projets avec beaucoup de tâches, on les regroupe en projets et sous projets. La durée totale d'un projet est la durée cumulée des tâches sur le chemin critique du projet.

  On peut y représenter les dates de début et de fin des tâches, ainsi que les relations de précédence entre elles.

  Ce planning permet de suivre l'achèvement d'un projet et déterminer le retard qu'une tâche peut subir avant que le projet ne soit pas affecté.

  Par contre il ne permet pas de suivre les activités de chaque ressource.

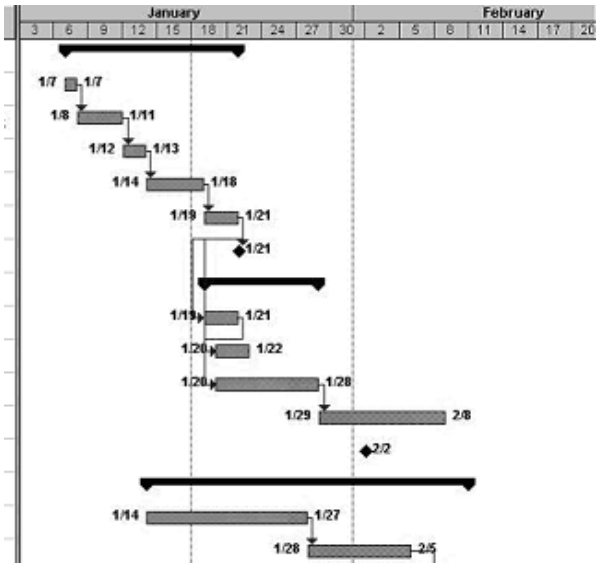


Figure 2.31 Un diagramme de Gantt

- Plannings par ressource : le temps occupe l'axe horizontal et chaque ligne représente le planning de la ressource. Ce dernier peut être une personne, une machine ou une salle.

- Plannings par poste de travail : le temps occupe l'axe horizontal et chaque ligne représente le planning du poste.

tel-00004380, version 1 - 29 Jan 2004

2.9 COURBE DE CHARGE ET DIMENSIONNEMENT

La phase de calcul de charge sert à fournir des données sur la charge de travail prévue sur chaque intervalle de temps. Comme on en déduit les besoins en personnel, elle est nécessaire à toute planification des horaires du personnel.

Ensuite, il faut calculer la taille des équipes capables de faire face aux charges. Nous rappelons les travaux courants permettant de dimensionner les équipes.

Comme cela se fait dans l'esprit de la planification en gestion de production, nous poursuivons notre étude sur l'application du dimensionnement à des fins de planification du personnel.

2.9.1 La courbe de charge

La planification des horaires du personnel prend comme entrée le besoin en nombre de personnes pour assurer un niveau de service donné. Or fréquemment, la donnée disponible est la quantité de travail ou service pendant chaque intervalle de temps. Pour retrouver le besoin en nombre de personnes, une loi linéaire ou règle de trois est souvent utilisée.

Dans les centres d'appel, [Seg74] rappelle la notion du *niveau de service* dans la forme « pas plus de α% des clients en attente de plus de β unités de temps ». En supposant qu'un état d'équilibre statistique est atteint au cours de chaque intervalle de temps, avec une arrivée de clients suivant une loi de Poisson avec le taux λ, les temps d'attente suivent une distribution exponentielle décroissante avec une moyenne de la durée de traitement $1/\mu$ donnant une charge $a = \lambda/\mu$. Le nombre d'opérateurs s est le plus petit entier tel que :

$$\alpha \geq C(s,a)\, e^{-\beta\mu(s-a)} \qquad (13)$$

où C est la loi d'Erlang C

$$C(s,a)=\frac{(a^s/s!)*(s/(s-a))}{\sum_{k=0}^{s-1} a^k/k! + (s/(s-a))} \qquad (14)$$

Les besoins en personnel pour chaque intervalle de temps, relèvent souvent des statistiques. Il est généralement traité par une méthode probabiliste, alimentée par les historiques des demandes (de nature probabiliste), éventuellement corrigée par la présence des événements exceptionnels sur la période à étudier. Une distribution probabiliste p peut représenter le nombre de salariés r dont on a besoin.

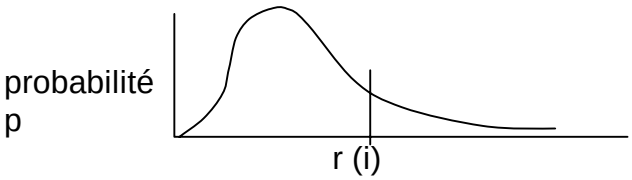


Figure 2.32. Besoins dans l'intervalle i considéré

Le décideur doit aussi estimer l'absentéisme et augmenter la demande pour pouvoir y faire face. Sinon, il faut prévoir d'autres méthodes d'organisation pour le traiter (équipes volantes, temps partiels). Le décideur doit décider quelle proportion de cette demande à traiter effectivement.

tel-00004380, version 1 - 29 Jan 2004

2.9.2 Le dimensionnement de l'équipe

Avant de pouvoir planifier l'emploi du temps d'une équipe, il est nécessaire de connaître sa taille. Dans la littérature scientifique, de nombreux auteurs proposent des méthodes de calcul de la borne inférieure avec des règles de travail contraignantes. Dans le même temps, ils proposent des algorithmes permettant de faire des plannings sur un nombre réduit de semaines ; ces plannings peuvent alors être utilisés en plannings cycliques. Les algorithmes sont très rapides en exécution car ils exploitent les structures particulières du problème et les résultats sont de bonne qualité. Dans la suite du paragraphe, seules les bornes inférieures de la taille de l'équipe sont présentées.

a. Cas d'une seule vacation (demandes par jour de la semaine)

Burns et Carter ont été les premiers à proposer une méthodologie basée sur le calcul de plusieurs bornes inférieures, la taille étant le maximum de ces bornes M [BC85]. Ils ont considéré le cas d'une seule vacation par jour et le besoin est n(d), d=1 pour le dimanche, …, 7 pour le samedi. W = nombre total de salariés, n = Max(n(1), …, n(7)).

Méthode de Burns et Carter

La taille d'une équipe dépend de plusieurs facteurs :

1. Charge de week-end : le nombre de salariés disponible en moyenne chaque week-end doit être suffisant pour respecter les besoins du week-end.

   Sur B semaines, chaque salarié est disponible (B-A) week-end. (B-A)W= nombre de personne-jour disponible pour le week-end, Bn = maximum des besoins de week-end. (B-A)W ≥ Bn, d'où

   $W \geq \lceil B\, n / (B-A) \rceil$ (**15**)

   Où $\lceil x \rceil$ = le plus petit entier x égal ou supérieur à x.
2. Charge totale de la semaine : le nombre de salarié-jour par semaine doit être suffisant pour supporter la charge totale de la semaine. Si le salarié ne travaille que 5 jours par semaine :

   $5W \geq \sum_{j=1,7} n(j)$, donc $W \geq \lceil 1/5 * \sum_{j=1,7} n(j) \rceil$ (**16**)
3. Charge maximale journalière : le nombre de salariés disponible doit être suffisant pour respecter le pic des besoins une journée donnée. $W \geq \mathrm{Max}_j (n(j))$

Il faut une équipe dont la taille W égale le maximum des trois bornes inférieures.

b. Cas des multiples vacations (demandes jour de la semaine/week-end)

[Hun94] propose une méthode de calcul de la taille minimale d'une équipe. Chaque jour il y a P vacations (ex. P=3 pour le jour, soir et nuit) qui peuvent se chevaucher afin d'absorber un pic de besoins ou pour assurer une meilleure communication entre les postes.

Pendant la semaine, il faut au moins D(j) personnes sur la vacation j et au moins E(j) pour le week-end. En général, D(j) ≥ E(j) pour j=1,…,P. Chaque salarié ne travaille qu'une

tel-00004380, version 1 - 29 Jan 2004

seule vacation par jour et doit avoir au moins un repos avant de changer de vacation. Sur deux semaines consécutives, un salarié doit avoir 3 repos en une semaine et 4 repos sur l'autre semaine. Il doit aussi avoir au moins A week-ends de repos sur B, avec 0 <= A <= B. La durée maximale de travail ininterrompue est 5 jours. La méthode proposée ne distingue pas les différentes demandes suivant les jours de la semaine.

c. Cas des hiérarchies de compétences

Une situation fréquemment rencontrée est une hiérarchie de compétence. Au sommet de la hiérarchie, le niveau 1 est le plus qualifié : une telle personne peut remplacer toute autre personne du même niveau et également une personne aux niveaux inférieurs. Emmons et Burns sont les premiers à proposer [EB91] de calculer la taille de l'équipe dans ces conditions, tout en proposant un coût minimal. Les contraintes sont une seule vacation et les demandes sont confondues pour tous les jours de la semaine, y compris le week-end.

d. Cas des hiérarchies de compétences et multiples vacations

[Nar00] montre comment traiter le cas des multiples vacations et distingue les demandes des jours de la semaine et du week-end. Les variables sont :

Nombre de catégories = K,
Le nombre de salariés de la catégorie k = w(k)
Le besoin du week-end, pour la catégorie k et la vacation j = n(j, k)
Le besoin en semaine = d(j, k)
La taille cumulée des catégories entre 1 et k, $W(k) = \sum_{i=1,k} w(i)$
Le besoin cumulé des catégories entre 1 et k, $D(j, k) = \sum_{i=1,k} d(j , i)$

Par clarté d'écriture, on écrit $n(., k) = \sum_{j} n(j, k)$ et $d(., k) = \sum_{j} d(j, k)$

tel-00004380, version 1 - 29 Jan 2004

Méthode de Narasimhan [Nar00]

1. Borne inférieure basée sur la charge du week-end pour chaque catégorie : L1(k)

Le nombre de salariés de la catégorie k w(k) disponible le week-end doit satisfaire les besoins n(j, k). Pour que chaque salarié dispose de A week-end tous les B semaines, sur un cycle de B semaines, on a

$$(B - A)\, w(k) \geq B\, n(., k) \qquad (17)$$

Or, $w(k) \geq L1(k)$ par définition de la borne inférieure, d'où

$$L1(k) = \lceil B\, n(., k) / (B\text{-}A) \rceil \qquad (18)$$

2. Borne inférieure basée sur le besoin total de chaque catégorie : L2(k)

Le nombre total de salariés * jour par semaine disponible (quelle que soit la catégorie) doit être suffisant pour répondre aux besoins totaux de la semaine pour cette catégorie. Comme chaque salarié ne travaille que 4 jours par semaine,

$$4\, w(k) \geq 5\, d(., k) + 2\, n(., k) \qquad (19)$$

or, $w(k) \geq L2(k)$ par définition de la borne inférieure,
d'où $L2(k) = \lceil 1.25\ d(., k) + 0.5\, n(., k) \rceil$

La taille de la catégorie k égale au moins le maximum des deux bornes inférieures:

$$w(k) \geq \text{Max}\ \{ L1(k), L2(k) \} \qquad (20)$$

3. Borne inférieure basée sur le besoin cumulatif : L3(k)

Le nombre cumulatif des vacations disponibles des catégories 1 à k doit être au moins égal au nombre total de vacations demandées dans les catégories 1 à k

$$4\, W(k) \geq 5\, D(., k) + 2\, n(., k) \qquad (21)$$

d'où

$$W(k) \geq \lceil 1.25\, D(., k) + 0.5\, n(., k) \rceil \qquad (22)$$

Par ailleurs, le nombre cumulé de salariés des catégories 1 à k doit être au moins égal au nombre cumulé de 1 à k-1 plus le nombre w(k) :

$$W(k) = \text{Max}\ \{ W(k\text{-}1) + \text{Max}(L1(k), L2(k)) \} \qquad (23)$$

$L3(k) = \text{Max}\ \{ L3(k\text{-}1) + \text{Max}(L1(k), L2(k)), \lceil 1.25\, D(., k) + 0.5\, n(., k) \rceil \}$, si $k > 1$
$L3(1) = \text{Max}\ \{ L1(1), L2(1) \}$

e. Conclusion

Ces calculs sont basés sur des hypothèses rigides. En présence des contraintes supplémentaires, par exemple avec des contraintes de transition entre les vacations (pour des questions de repos) et les différents besoins par jour de la semaine, ils peuvent être utilisés pour proposer des bornes inférieures de la taille des équipes.

tel-00004380, version 1 - 29 Jan 2004

2.9.3 Le dimensionnement des vacations

L'art du dimensionnement peut être appliqué plus finement qu'au niveau de l'équipe : calculer le nombre de personnes dans chaque type de vacation (ou horaires types sur une journée). Cette possibilité a donné suite à toute une série de travaux sur les modèles implicites de construction de vacation, présentés au paragraphe § 2.1.

On constate que la complexité théorique du problème à résoudre est réduite de façon substantielle. Cela est dû au fait que les vacations ne sont pas décrites explicitement. Cependant, une fois ces nombres obtenus, il faut associer les heures de début et les pauses aux vacations et générer des plannings définitifs pour chaque salarié (Voir § 2.3 Construction de Tours Acycliques).

2.9.4 Le dimensionnement tout au long de la résolution

Enfin l'art du dimensionnement peut être appliqué tout au long du processus de la résolution, plus particulièrement dans le cas des algorithmes dits constructifs, où la solution est construite progressivement. Dans la résolution de problèmes de planification de grilles, cette approche a été proposée dans [CGL95].

Analyse de Caseau

Le problème consiste à affecter à chaque personne p à chaque jour w, à un code horaire a (ex. M= matin, S= soir, N= nuit, R= repos, J= jour), annoté par Affectation(p, w) = a. Les contraintes sont définies par les bornes Min / Max suivantes :

Par code horaire a par jour w, le total des affectations des agents doit être compris entre des bornes suivants (|Z| étant la cardinalité d'un ensemble Z) :

$$\text{Min}[w, a] \leq |\{ p \in \text{Personnes}, \text{Affectation}(p, w) = a\}| \leq \text{Max}[w, a]$$

Par code horaire a par personne p, le total des affectations doit être compris entre des bornes

$$\text{Min}[p, a] \leq |\{ w \in \text{Horizon}, \text{Affectation}(p, w) = a \}| \leq \text{Max}[p, a]$$

Soit N(a) = le nombre d'affectations au code a, sur l'ensemble des personnes et sur tout l'horizon. Par dimensionnement, N(a) doit être inférieur à la somme respectivement des maximum des affectations par personne et par jour au code a. De même il doit être supérieur à la somme respective des minima des affectations par personne et par jour. D'où :

$$\sum_{p \in \text{Personnes}} \text{Min}[p, a] \leq N(a) \leq \sum_{w \in \text{Horizon}} \text{Max}[w, a] \quad (\mathbf{24})$$

$$\sum_{w \in \text{Horizon}} \text{Min}[w, a] \leq N(a) \leq \sum_{p \in \text{Personnes}} \text{Max}[p, a] \quad (\mathbf{25})$$

Caseau propose les compteurs supplémentaires suivants :

Num[p, a] = no. d'affectations au code a pour la personne p sur l'horizon
Num[w, a] = no. d'affectations au code a pour le jour w sur l'ensemble des personnes
Pos[p, a] = no. d'affectations possibles au code a pour la personne p sur l'horizon
Pos[w, a]= no. d'affectations possibles au code a pour le jour w sur l'ensemble des personnes

[CGL95] montre que

tel-00004380, version 1 - 29 Jan 2004

Num[p, a] ≤ Max[p, a] Num[w, a] ≤ Max[w, a] le no. affecté doit être inférieur au max.
Min[p, a] ≤ Pos[p, a], Min[w, a] ≤ Pos[w, a], le no. possible doit être supérieur au min.

Caseau raffine la condition de dimensionnement par les équations suivantes :

$$\sum_{p \in \text{Personnes}} \max(\text{Num}[p,a], \text{Min}[p,a]) \leq N(a) \leq \sum_{w \in \text{Horizon}} \max(\text{Pos}[w,a], \text{Max}[w,a]) \quad (\mathbf{26})$$

$$\sum_{w \in \text{Horizon}} \max(\text{Num}[w,a], \text{Min}[w,a]) \leq N(a) \leq \sum_{p \in \text{Personnes}} \max(\text{Pos}[p,a], \text{Max}[p,a]) \quad (\mathbf{27})$$

Cette condition peut être utilisée pour élaguer les branches potentiellement infructueuses dans la construction de solution, à réévaluer chaque fois que Num ou Pos sont modifiés. Elle constitue une méthode de consistance globale.

Pour ce même problème, Caseau propose une condition supplémentaire applicable sur les compteurs Min et Max, liée au fait que le nombre total de jours ou de personnes est connu :

- Pour chaque code a et pour chaque personne,

$$\text{Max}[p,a] \leq |W| - \sum_{a' \neq a} \text{Min}[p,a'] \text{ et } \text{Min}[p,a] > |W| - \sum_{a' \neq a} \text{Max}[p,a'] \quad (\mathbf{28})$$

- Pour chaque code a et pour chaque jour,

$$\text{Max}[w,a] \leq |P| - \sum_{a' \neq a} \text{Min}[w,a'] \text{ et } \text{Min}[w,a] > |P| - \sum_{a' \neq a} \text{Max}[w,a'] \quad (\mathbf{29})$$

2.9.5 Conclusion sur le dimensionnement

Etape essentielle dans la constitution des équipes, le dimensionnement peut aussi être exploité pour détecter des insuffisances au cours de planification. Les propositions de Caseau permettent de réduire la complexité des problèmes combinatoires, et a fortiori des problèmes de planification des ressources humaines.

Nous retenons ces leçons dans le cadre de nos recherches au chapitre 5.

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

DEUXIEME PARTIE : RECHERCHE

Cette partie présente les travaux de recherche entrepris dans le cadre de la thèse. Elle se compose des chapitres suivants :

3. Calcul des tours acycliques en PPC : Gymnaste & EQUITIME
    a. La génération automatique des plannings
    b. Les spécifications de contraintes types
    c. La modélisation avec les contraintes globales
    d. Les contraintes redondantes
    e. EQUITIME V3 : génération sans retour arrière
    f. L'interface utilisateur
    g. Les limites des approches précédentes

4. Le déroulement des cycles autour des pré-affectations
    a. Généralités
    b. La définition du problème
    c. Le modèle et son implantation
    d. La recherche de solutions
    e. Nos résultats et conclusions

5. Les modèles à multiples niveaux d'agrégation
    a. La législation française en matière de durées de travail et de repos
    b. Les modèles de base
    c. Les modèles multi-niveaux
    d. Nos conditions nécessaires en multiples qualifications
    e. Nos méthodes heuristiques
    f. Nos résultats et conclusions

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

3 LE CALCUL DES TOURS EN PPC

Résumé

Dans ce chapitre, nous présenterons nos travaux de recherche sur le calcul des tours sur un horizon mensuel en utilisant la programmation par contraintes. Après un rappel des contraintes types, nous montrons dans un premier temps l'utilisation des contraintes globales AMONG et SEQUENCE du système CHIP pour résoudre ce problème au sein du projet GYMNASTE-COSYTEC. Puis nous traitons un problème similaire à l'aide des nos propres algorithmes rédigés en Visual Basic (projet EQUITIME).

Nous présentons aussi les différentes interfaces conçues pour faciliter l'interaction entre le planificateur et le générateur automatique de planning.

3.1 LA GENERATION AUTOMATIQUE DES PLANNINGS ACYCLIQUES

Le calcul des tours est défini comme l'affectation des étiquettes (par exemple matin, après-midi ou nuit) à chaque membre de l'équipe, pour chaque jour de l'horizon de planification, afin de couvrir des besoins exprimés en nombre de salariés par étiquette par jour. A la différence des plannings cycliques, ces besoins peuvent être différents chaque jour de la semaine et d'une semaine sur l'autre.

Ceci correspond à l'affectation des vacations ou des horaires journaliers. L'activité précise des salariés heure par heure, n'est pas l'objet de l'affectation.

3.1.1 Modèle PPC

Nous proposons un modèle pour l'affectation d'étiquettes aux agents, avec la satisfaction des différentes contraintes types qui seront définies au paragraphe § 3.2. Une variable est créée pour chaque agent pour chaque jour de planification. Par exemple la variable `X(e,t)` correspond à l'unique affectation de vacation (ou code horaire) de l'employé e le jour t. Le domaine de ces variables est constitué des différentes vacations admissibles pour cet employé à ce jour.

3.1.2 Justification de la technique PPC

Le modèle de la PPC est très bien adapté pour modéliser le calcul des tours, compte tenu de la distance entre les contraintes du problème et les contraintes disponibles dans l'outil PPC. La PPC permet de formaliser des contraintes comme :

- Un salarié ne doit pas avoir un jour de repos isolé (ou repos sec).
- Un salarié ne doit pas travailler un jour sur le week-end.

La PPC est une approche constructive : elle avance en réalisant des affectations de valeurs (codes horaires) aux variables (agent/jour). Afin de réduire l'espace de recherche, la PPC utilise des méthodes de vérification de consistance[15].

[15] forward checking et look-ahead [Mac77]

tel-00004380, version 1 - 29 Jan 2004

Cette approche permet de traiter des cas de plannings partiels, par exemple suite aux affectations manuelles et interactives qui tiennent compte des situations exceptionnelles (ex. formation, congés annuels ou autres obligations de service ou personnelles). Ces situations peuvent être issues des négociations.

3.2 LES SPECIFICATIONS DES CONTRAINTES TYPES

Nous avons publié les différentes classes de contraintes agissant sur un planning acyclique dans [WHC+98]. Elles ont été reprises avec modifications dans les projets Gymnaste et EQUITIME.

3.2.1 Les propriétés contextuelles

Les différentes classes de contraintes possèdent des propriétés contextuelles concernant leur validité :

Les dates de validité : date début – date fin

L'équipe dans laquelle la contrainte est applicable

Les agents concernés : tout le monde ou une catégorie (avec d'éventuelles exceptions), un individu, etc.

Activation ou désactivation : une contrainte peut être désactivée temporairement pendant la mise au point du planning

Ces propriétés permettent aux contraintes d'être exprimées de façon très succincte et ergonomique. Ces expressions sont stockées dans la base de données telles quelles. Au moment du lancement, le solver crée une instance de toutes les contraintes pour tous les salariés concernés par le planning à réaliser.

Cette méthode permet l'application immédiate des contraintes aux salariés, y compris les nouveaux venus, lorsque la contrainte est valable pour tous. L'inconvénient vient des exceptions (ex. tous sauf M.Durand), lorsque ces derniers ne font plus partie de l'équipe.

On décrit les classes de contraintes GYMNASTE ainsi que les paramètres pour s'approcher de la réglementation en rigueur. A et B sont deux agents quelconques, Début et Fin sont des dates, et S1, S2 et S3 sont des codes horaires.

Les différentes classes de contraintes possèdent une propriété commune :
La nature des contraintes : positive/négative, obligation/préférence

Dans les paragraphes suivants, les exemples de contraintes sont des textes générés par le logiciel suite à la phase de paramétrage.

tel-00004380, version 1 - 29 Jan 2004

3.2.2 Les contraintes de charge

On peut générer un planning si l'on dispose des données suivantes :

- L'horizon de calcul
- Les codes horaires à affecter (elles constituent le domaine des variables)
- Les besoins bruts par code horaire par jour de l'horizon
- Les candidats à considérer pour la planification, en fonction du mode de planification (manuelle ou automatique) du salarié, des dates de service de ce dernier, sa qualification, etc.

Ainsi le besoin net en nombre de salariés par code horaire par jour de l'horizon de planning est obtenu des besoins bruts en déduisant les pré-affectations des salariés. Le planning doit couvrir au moins les besoins nets, tout en tenant compte des autres contraintes décrites dans les paragraphes suivants.

3.2.3 Les contraintes d'affectation ponctuelle

Cette contrainte sert à définir une pré-affectation à une date donnée, même très éloignée dans l'horizon de planning. Par extension, si on spécifie plusieurs codes horaires ce jour, cela indique les différentes possibilités retenues.

Format :

- Le [Date], [Agent] est affecté[Préférence] aux [Codes Horaires]

Exemples :

```
o  Le X, A travaille toujours        les tranches [S3, S1]
o  Le X, A travaille si possible     les tranches [S3, S2]
o  Le X, A travaille si possible pas les tranches [S2, S1]
o  Le X, A travaille jamais          les tranches [S3, S1]
```

Pour tous les salariés concernés par le planning, le domaine des variables est défini par l'ensemble des besoins du planning. Dans le produit EQUITIME, cette expression permet d'indiquer une priorité aux salariés pour ces codes horaires.

3.2.4 Les contraintes de disponibilité

Cette contrainte spécifie les différentes vacations admises pour cet agent ce jour.

Format :

- [Agent] [Préférence] travaille les [Codes Horaires] les jours [Ji] de [Début] à [Fin]

Exemples :

```
o  A travaille toujours    les tranches [S1, S2] les jours [j1] de Début à Fin
o  A travaille si possible les tranches [S1, S3] les jours [j1] de Début à Fin
o  A travaille si possible pas les tranches [S2] les jours [j2] de Début à Fin
o  A travaille jamais      les tranches [S3]     les jours [j1] de Début à Fin
```

tel-00004380, version 1 - 29 Jan 2004

Dans le produit EQUITIME, cette contrainte spécifie les jours de semaine où le salarié doit effectuer ces codes horaires. Cette disponibilité donne une priorité supplémentaire aux affectations.

3.2.5 Les contraintes de vacation due

Cette contrainte est conçue pour imposer un nombre donné de vacations à un salarié donné (ex. 4 repos hebdomadaires sur 15 jours, ou 7 RH si la personne ne travaille que la nuit.

Format :

- [Agent] doit faire des [S1, S2 et S3] au moins N fois sur P semaines de [Début] à [Fin]

Exemples :

```
o  A doit faire des S1 au moins   N fois sur P semaines de Début à Fin
o  A doit faire des S1 exactement N fois sur P semaines de Début à Fin
```

Afin de créer des plannings équitables, l'utilisateur peut compter les affectations réelles (éventuellement via un outil supplémentaire) et imposer le complément aux salariés.

3.2.6 Les contraintes de transition

Ces contraintes permettent de spécifier des suites obligatoires ou interdites, traduisant le règlement sur le repos journalier. Si une infirmière travaille une vacation de nuit, elle ne doit pas travailler le lendemain matin et l'après-midi. Si elle travaille un après-midi, elle ne doit pas travailler le lendemain matin.

Format simple :

- Pour [Agents], [Préférence] [S1] suivi de [S2] de [Début] à [Fin]

Exemples :

```
o  Pour A, toujours        S1 suivi de S2, de Début à Fin
o  Pour A, si possible     S1 suivi de S2, de Début à Fin
o  Pour A, si possible pas S1 suivi de S2, de Début à Fin
o  Pour A, jamais          S1 suivi de S2, de Début à Fin
```

Le produit EQUITIME réalise une forme étendue de cette contrainte: si un tel code horaire sur un tel jour de la semaine, alors un autre code horaire sur un autre jour de la semaine.

3.2.7 Les contraintes de répartition

Cette contrainte spécifie la répartition de certains codes horaires sur un horizon donné.

tel-00004380, version 1 - 29 Jan 2004

Format simple :

- Pour [Agents], [Contrainte], N [Shift] consécutive de [Début] à [Fin]

Exemples :

```
o  Pour A, toujours        N S1 de suite, de Début à Fin
o  Pour A, si possible     N S1 de suite, de Début à Fin
o  Pour A, si possible pas N S1 de suite, de Début à Fin
o  Pour A, jamais          N S1 de suite, de Début à Fin
```

Avec le produit EQUITIME, on peut spécifier les bornes inférieures et supérieures sur la suite. Le produit permet aussi de spécifier tous les codes travaillés (sans distinction), ce qui est important pour générer des plannings dont le temps total de travail ne dépasse pas les limites légales.

3.2.8 Les contraintes de composition

Cette contrainte permet la constitution d'équipes (pour tuteur et élève). Elle peut aussi générer des plannings pour des salariés qui ne peuvent pas travailler ensemble.

Format simple :

- [Agent1] travaille [Contrainte] avec [Agent2] de [Début] à [Fin]

Exemples :

```
o  A travaille toujours        avec B, de Début à Fin
o  A travaille si possible     avec B, de Début à Fin
o  A travaille si possible pas avec B, de Début à Fin
o  A travaille jamais          avec B, de Début à Fin
```

Dans le produit EQUITIME, cette contrainte n'a pas été réalisée car peu utilisée dans la pratique.

3.2.9 Critère d'optimisation

Les logiciels Gymnaste et EQUITIME ont été conçus sans critère à optimiser. Dans la pratique, la planification est souvent basée sur de multiples critères, mais la pondération relative des critères varie selon :

- L'acteur concerné
- La période concernée (haute saison/promotions/période estivale, etc.)
- Le contexte (les délais de notification, etc.)
- Le passé (questions d'équité)
- Négociation inter-personnelle (échanges de plannings entre deux salariés avec entente préalable)

Néanmoins, il est possible d'utiliser ces solveurs pour la simulation de plannings étant donné un nombre de salariés. Ainsi, on peut trouver le nombre minimum de salariés nécessaires pour répondre à un ensemble de besoins par vacation par jour sur un horizon mensuel.

tel-00004380, version 1 - 29 Jan 2004

3.3 LA MODELISATION AVEC LES CONTRAINTES GLOBALES

Dans ce paragraphe, on décrit l'implantation des contraintes du paragraphe précédent en utilisant les contraintes globales `among` et `sequence` du système CHIP, dont la sémantique est rappelée dans le glossaire. Dans les exemples, on utilisera la syntaxe prolog de CHIP. Dans les deux contraintes, `Zeros(R)` est une liste de R zéros indiquant qu'on ne veut aucun déplacement sur les valeurs des variables V, et le paramètre **all** spécifie l'exploitation par cette contrainte de tous les événements de changement de domaine.

3.3.1 L'affectation des codes horaires

Soit J le nombre de jours et I le nombre total de salariés.

Pour le salarié i le jour j, la variable est `X(e, j)` :

- L'ensemble de variables correspondant au salarié e est `X(e)` (pour tous les jours de l'horizon).
- L'ensemble des variables pour tous les salariés correspondant au jour j est `X(j)`.
- L'ensemble de toutes les variables est dénoté par X (pour tous les salariés et tous les jours).

Les valeurs possibles de ces variables sont :

- `Vm` la valeur numérique de la vacation du matin,
- `Ve` celle du soir,
- `Vn` celle de la nuit et
- `Vr` la valeur du repos hebdomadaire

Typiquement, les valeurs sont `Vr =0, Vm=1, Ve=2, Vn =3`.

3.3.2 Les contraintes de charge

On applique la contrainte `among` sur l'ensemble des agents pour chaque jour de l'horizon. Par exemple pour avoir entre `Min` et `Max` affectations à la valeur Vn = NUIT au jour j :

```
among([Min, Max], X(j) , Zeros, [Vn], all).
```

Plus globalement, pour traiter l'ensemble des valeurs V_i simultanément, on peut utiliser la contrainte suivante qui permet de donner plus de déduction :

`among([`N_1`, ..., `N_N`], [X(`e_1`,j), …, X(`e_N`, j)], Zeros `$_N$`, [V1, ..., Vn], all)`

Il suffit de récupérer l'ensemble des charges pour poser cette contrainte.

3.3.3 Les contraintes d'affectation ponctuelle et de disponibilité

Les variantes obligatoires de la contrainte d'affectation ponctuelle sont traduites directement par l'enlèvement de valeurs du domaine des variables des agents concernés sur les jours en question :

Le [Date], [Agent] est affecté[Préférence] aux [Codes Horaires]
[Agent] [Préférence] travaille le(s) tranches [Codes Horaires] les jours [Ji] de [Deb] à [Fin]

tel-00004380, version 1 - 29 Jan 2004

Alors que la contrainte d'affectation ponctuelle ne s'applique que pour les jours en question, la contrainte de disponibilité est appliquée pour chaque jour mentionné de la semaine, sur l'horizon de planification. Les variantes préférentielles sont implantées avec des points de choix.

Le produit EQUITIME prend en compte les priorités de ce type des contraintes au niveau de l'algorithme d'affectation. Dans le cadre de la PPC, il n'est pas possible de prendre en compte directement les priorités entre contraintes.

3.3.4 Les contraintes de vacation due

La contrainte de vacation due restreint le nombre total des vacations pour un agent i donné, sur tout l'horizon, notamment pour imposer une équité dans le planning. Par exemple, sur une suite quelconque de J variables, le nombre de variables ayant la valeur nuit = `Vn` doit être compris entre `Min` et `Max` :

```
among([Min, Max, J], X(e), ZerosJ, [Vn], all)
```

Nous remarquons qu'il y a peu de différence sémantique entre la contrainte de vacation due et la contrainte globale among. Dans la pratique, among est très peu efficace en tant que telle, sauf pour le cas exact où Min=Max, car elle doit porter sur nombreuses variables. Des contraintes supplémentaires doivent être utilisées pour s'assurer un bon taux de propagation. Le paragraphe 3.4 traite ce sujet plus en détail.

3.3.5 Les contraintes de transition

Nombre de contraintes légales peuvent être modélisées par une contrainte de transition entre deux variables successives pour un salarié i. Par exemple, l'affectation nuit ne doit pas être suivie par l'affectation matin ou après-midi. Ainsi pour toute séquence de deux variables X_1 et X_2, il y a exactement 0 occurrence des patterns spécifiés par l'argument 4 de la contrainte séquence, d'où le paramètre [0,0,2] dans l'appel suivant :

```
sequence([0,0,2], X(e), ZerosJ,[[[sum,1,#=,[Vn]],[sum,1,#=,[Vm,Ve]]]],all)
```

Le premier terme `[sum,1,#=,[Vn]]` spécifie que la somme de la première variable X_1 de la séquence est égale à V_n (c'est-à-dire l'affectation de la première variable dans une séquence de deux variables est V_n). Le second terme `[sum,1,#=,[Vn]]` spécifie que la somme de la variable X_2 de la suite est égale à l'une des valeurs V_m ou V_e. Ces deux termes constituent le seul pattern dans l'appel. La contrainte séquence généralement prend une liste de patterns.

La contrainte séquence peut être appliquée afin d'interdire l'affectation de repos (la valeur V_r) avant ou après l'affectation de travail. La séquence de trois variables est décrite par le pattern :

```
P = [[sum,1,#=,[Vr]], [sum,1,#=,[Vm,Ve,Vn]], [sum,1,#=,[Vr]]]
```

Lorsque plusieurs patterns sont applicables aux même variables (p.ex. au même salarié) et lorsqu'il ne doit avoir aucune occurrence, le même appel de la contrainte séquence peut être utilisé, ainsi à une meilleure propagation et une meilleure utilisation de la mémoire.

tel-00004380, version 1 - 29 Jan 2004

Sur une journée j, soit `Njm` les besoins de la vacation matin, `Nje` (les besoins de soir) et `Njn` (nuit). La contrainte de charge est spécifiée par :

```
XNjm :: Njm..J, XNje :: Nje..J, XNjn :: Njn..J,
among ( [XNjm, XNje, XNjn], Xj, Zeros, [[Vm],[Ve],[Vn]], all).
```

Cet appel contraint qu'il y a exactement `XNjm` variables parmi les variables `Xj` prendront la valeur `Vm`, `XNje` variables la valeur `Ve`, et `XNjn` variables la valeur `Vn`. Il s'agit de la variante `multi-among` de la contrainte globale `among`.

Nt = nb total de variables / de jours

- Pour agent X, *toujours* S1 suivi de S2, de XX à YY
  Chaque fois S1 est affectée S2 suit dans la période
  [0,0,Nt], C1= [card,#=,[BegST],1,#=,[1]], C2= [card,#\=,[EndST],1,#=,[1]]

- Pour agent X, *toujours* S1 suivi de S2, *au moins N fois* de XX à YY
  Chaque fois S1 est affectée, S2 suit au moins N fois dans la période. Si N est trop élevé, on cherche beaucoup, car la contrainte est contraire à la stratégie d'uniformité d'affectation
  [N,Nt,2], C1= [card,#=,[BegST],1,#=,[1]], C2= [card,#=,[EndST],1,#=,[1]]

- Pour agent X, *si_poss* S1 suivi de S2, de XX à YY
  Après une affectation S1, on affectera S2 (si possible=au moins 0 fois)
  [0,Nt/2,2], C1= [card,#=,[BegST],1,#=,[1]], C2= [card,#=,[EndST],1,#=,[1]]

- Pour agent X, *si_poss_pas* S1 suivi de S2, de XX à YY
  Après une affectation S1, S2 sera une valeur si_possible_pas pour l'affectation suivante
  Idem pour *jamais,* au moins 0 fois et au plus 1

- Pour agent X, *jamais* S1 suivi de S2, de XX à YY
  Après une affectation S1, la valeur S2 sera prélevée de l'affectation suivante
  [0,0,Nt], C1= [card,#=,[BegST],1,#=,[1]], C2= [card,#=,[EndST],1,#=,[1]]

3.3.6 Les contraintes de répartition

Ces contraintes permettent de distribuer des jours de travail ou de repos, dans un cadre légal ou un accord syndical. Par exemple, il est interdit de travailler plus de 5 jours de suite. Ainsi pour toute suite de 6 variables dans `X(e)`, il y a entre 1 et 5 variables ayant la valeur repos. Cette condition est obtenue par la contrainte suivante :

```
among ([1,5,6],X(e), Zeros, [Vr], all).
```

Lorsqu'il y a trois valeurs dans le premier argument, l'utilisation de la contrainte `among` utilise la première et seconde valeurs comme les bornes inférieures et supérieures du nombre d'occurrences, la troisième valeur étant la taille de la séquence de variables consécutives à considérer parmi les variables dans la liste X.

Inversement, la règle de 5 jours de travail, pour toute suite (consécutive) de 6 variables, une valeur de travail (ex. V_m, V_e, V_n) apparaît entre 0 et 5 fois.

```
among([0,5,6], X(e), Zeros, [Vm, Ve, Vn], all).
```

Ces deux contraintes sont une première approximation aux contraintes de travail. La première ignore toutes les autres vacations de repos (telles que les RTT ou Congés). La

tel-00004380, version 1 - 29 Jan 2004

seconde ignore toutes les autres vacations légales de travail (ex. Formation ou Délégation syndicale).

La contrainte `among` peut s'utiliser pour interdire la suite de 4 vacations de nuit. Donc, sur chaque suite de 4 variables, il y a au plus 3 affectations de nuit :

```
among([0,3,4], X(e), Zeros, [Vn], all).
```

- Pour agent X, *toujours* Nb S1 consécutives, de XX à YY
  [0,0,Nb], génération de patterns de taille Nb+1, interdits.
  Ne pas faire pour Nb > 6 jours de travail consécutifs ! Entre deux blocs de N affectations consécutives il y a une autre affectation, imposée par une contrainte portant sur Nb+1 variables. Si S1=tous alors alerte 2 (pas traitable).

- Pour agent X, *si_poss* Nb S1 consécutives, de XX à YY
  Idem pour la variante *toujours*, au moins 0 fois et au plus Nb fois

- Pour agent X, *si_poss_pas* Nb S1 consécutives, de XX à YY
  Idem pour la variante *jamais,* au moins 0 fois et au plus Nb-1 fois

- Pour agent X, *jamais* Nb S1 consécutives, de XX à YY
  [0,0,Nt], [card,#=,[S1], Nb, #=,[Nb]]
  Si S1 est 'tous', alors on a [card,#>=,[1], Nb, #=,[Nb]]

3.3.7 Les contraintes de composition

- Agent X travaille *toujours* avec Agent Y, de XX à YY

Attention: cette contrainte est très forte, car ces agents travailleront tous les jours ensemble !
[NbDays,NbDays,3], [[range,2,#=,0],[sum,1,#>=,[0]]]

- Agent X travaille *toujours* avec Agent Y, *au moins N fois* de XX à YY

Attention le nombre de jours entre XX et YY doit être supérieur à N ! Comme la stratégie de recherche ne prend pas cette contrainte en compte, cela peut être lent pour N grand, si X et Y sont pris n'importe comment. Préférable de les mettre en haut de l'écran (les premiers) ou faire avec N=0, puis ré-affecter.
[N,NbDays,3], [[range,2,#=,0],[sum,1,#>=,[0]]]

- Agent X travaille *si_poss* avec Agent Y, de XX à YY
  On interprète cette contrainte par « X travaille toujours avec Y au moins 0 fois »
  [0,NbDays,3], [[range,2,#=,0],[sum,1,#>=,[0]]]

- Agent X travaille *si_poss_pas* avec Agent Y, de XX à YY
  Idem pour *jamais,* au moins 0 fois et au plus 1

- Agent X travaille *jamais* avec Agent Y, de XX à YY
  [0,0,3], [[range,2,#=,0],[sum,1,#>=,[0]]]

tel-00004380, version 1 - 29 Jan 2004

3.3.8 Recherche de solution

Nous utilisons une stratégie d'énumération en une passe en affectant à la fois les différentes vacations et les vacations de repos, pour une catégorie de personnel à la fois. L'énumération se fait jour par jour afin de satisfaire d'abord le minimum spécifié de la charge journalière, ensuite respecter la charge moyenne des agents (si trop de repos).

```
Pour chaque jour J
    Pour chaque vacation, tant qu'il y a des besoins non satisfaits,
        Rechercher le candidat de la catégorie,
        équilibrant le total des heures effectuées
```

La détermination du repos hebdomadaire est réalisée simultanément. Pour chaque jour, l'énumération affecte exactement les besoins demandés de chaque vacation. Les autres variables sont laissées libres afin de ne pas sur-contraindre le problème inutilement. Lors de la deuxième passe, ces variables libres sont affectées afin de respecter les autres contraintes.

Les contraintes peuvent être spécifiées de façon positive ou négative et de façon obligatoire ou préférentielle. De par leur nature, les contraintes préférentielles ne peuvent pas réduire l'espace de recherche. L'effet préférentiel est obtenu par programme via les points de choix, à créer après la pose des contraintes obligatoires.

tel-00004380, version 1 - 29 Jan 2004

3.4 LES CONTRAINTES REDONDANTES

Etant donné la généralité des contraintes globales, la quantité de propagation obtenue dans certains cas est insuffisant. Afin d'éviter de chercher des solutions dans des branches mortes de l'arbre de recherche, il faut augmenter la propagation et détecter des contradictions au plus tôt. Pour cela, on dispose de deux méthodes basées sur une étude plus approfondie du problème. La première consiste à définir des bornes plus strictes sur les variables à domaine, et la seconde est d'appliquer des contraintes redondantes.

Définition : Une contrainte c est redondante par rapport à un ensemble donné de contraintes C lorsqu'elle est une conséquence logique de C.

3.4.1 Charge journalière

La contrainte de charge journalière minimale décrite ci-dessus peut être enrichie. La borne supérieure du nombre de chaque type de variable (XN_{jm}, XN_{je}, ou XN_{jn}) peut être déduite en présence des autres valeurs, car le nombre total est connu. Pour simplifier la notation, les bornes supérieures des variables dans le passage suivant, sont notées entre parenthèses[16].

```
XNjm :: Njm..(J-Nje-Njn) ,
XNje :: Nje..(J-Njm-Nje),
XNjn :: Njn..(J-Njm-Nje),
among( [XNjm, XNje, XNjn], Xj, Zeros, [[Vm],[Ve],[Vn]], all).
```

3.4.2 Charge totale

Ni les contraintes de charge journalière ni les contraintes de répartition de vacations ne peuvent assurer à elles seules qu'assez de main d'œuvre soit disponible pour couvrir la charge sur l'horizon de planification sur toutes les vacations. Ici, nous proposons une contrainte redondante applicable sur toutes les variables pour chaque paire de variables (salarié-jour).

$S_m = \sum_{j=1,J} N_{jm}$
$S_e = \sum_{j=1,J} N_{je}$
$S_n = \sum_{j=1,J} N_{jn}$
IJ is I*J,
$XS_m :: S_m..IJ$,
$XS_e :: S_e..IJ$,
$XS_n :: S_n..IJ$,
among([XS_m, XS_e, XS_n], X, Zeros, [[V_m],[V_e],[V_n]], all).

Cette contrainte utilisant la variante multiple `among` spécifie que chaque valeur `Vm`, `Ve`, `Vn` doit apparaître au moins `Sm`, `Se` ou `Sn` fois (respectivement) sur l'ensemble de toutes les variables. Ainsi si le nombre de salariés-jours disponible est insuffisant pour couvrir la charge totale (donc non affecté préalablement à une valeur autre que `Vm`, `Ve` or `Vn`), cette contrainte échouera immédiatement dès la pose.

[16] en réalité, elles doivent être calculées avant la déclaration de domaine

tel-00004380, version 1 - 29 Jan 2004

Dans ce cas, on sollicite l'utilisateur par un diagnostic détaillé de la situation : il pourra alors enlever les pré-affectations telles que RTT ou Repos Compensatoire, ou réduire les besoins en personnel.

En réalité, les bornes supérieures des variables XS peuvent être déterminées avec une plus grande précision, en conjonction avec la contrainte sur le nombre minimal de jours de repos par salarié (ex. `Sr`) sur l'horizon de planification :

```
XSm :: Sm..(IJ-Se-Sn-Sr),
XSe :: Se..(IJ-Sm-Sn-Sr),
XSn :: Sn..(IJ-Sm-Sn-Sr),
XSr :: Sr..(IJ-Sm-Se-Sn),
among( [XSr, XSm, XSe, XSn], X, Zeros, [[Vr],[Vm],[Ve],[Vn]], all).
```

Ce raisonnement peut être appliqué à la contrainte de charge minimale pour chaque jour j, telle que la valeur V_r doit apparaître au plus $J-X_{jm}-X_{je}-X_{jn}$ fois.

3.4.3 Contraintes de charge

La contrainte sur le nombre de vacations qu'un salarié doit effectuer sur une période de temps (ex. vacations de repos) est très difficile à respecter lorsque l'intervalle est grand, car il y peu de déduction sur les affectations des premières variables. Ceci se traduit par la contrainte globale suivante :

```
among([7,8,14], Xi, Zeros, [Vr], all).
```

Sur une période de 14 jours, le nombre de variables qui peut prendre la valeur Vr doit être compris entre 7 et 8 fois.

Le taux de propagation ainsi obtenu n'est pas très fort. Cependant, utilisant la méthode décrite dans [DHS+88], s'il y a 7 jours de repos sur 14 jours, alors sur tout intervalle de 14-7+1= 8 jours, si les 6 autres jours sont au repos, alors il doit avoir au moins 1 repos dans l'intervalle. Ce raisonnement appliqué sur 14 jours, on obtient six contraintes, où Xe(k) est un ensemble de huit variables débutant le jour k :

```
among([1,8,8], Xe(k), Zeros, [Vr], all), for k=1,6
```

L'effet combiné est en effet, la sémantique de l'appel `among` suivant, où le nombre de séquences (ici 8) est strictement inférieur au nombre total de variables Xi (ici, 14) :

```
among([1,8,8], X(e), Zeros, [Vr], all).
```

Ce raisonnement peut être appliqué aux 5 séquences de 9 jours consécutives, avec au moins 2 repos, jusqu'au 4 séquences de 10 jours avec au moins 3 repos, etc.

tel-00004380, version 1 - 29 Jan 2004

3.5 EQUITIME V3 : GENERATION SANS RETOUR ARRIERE

Nous décrivons ici le développement d'un système de génération de planning, destiné à être générique et pour un très large public.

3.5.1 Notre motivation

Notre approche est caractérisée par les éléments suivants :

- Algorithme d'affectation de vacations aux employés, de façon heuristique, constructive et incrémentale
- Algorithme bâti sans les outils commerciaux PPC et sans retour-arrière

Au cours d'une phase de recherche (1990-1995), le projet Gymnaste V0 a utilisé les outils ILOG afin de vérifier l'adéquation de la PPC par rapport à la problématique de la planification. Lors de la phase d'industrialisation initiale (1996-1999), le projet Gymnaste V1 a analysé l'apport des contraintes globales avec l'outil CHIP, dans une approche logicielle standard.

La recherche arborescente avec retour arrière chronologique a donné des performances satisfaisantes lors des essais.

Avec le projet EQUITIME V3, nous avons opté pour des outils de développement plus traditionnels pour les raisons suivantes :

- Performance
- Ergonomie
- Intégration dans l'environnement Windows et dans la suite Microsoft Office

Pour cette version EQUITIME, nous avons choisi de réaliser un moteur de propagation de contraintes sans retour-arrière, en nous appuyant sur des les heuristiques habituels de choix des variables ou des valeurs. De même, nous préférons une approche heuristique à l'approche d'optimisation pure.

D'une part, ce choix était motivé par le succès des algorithmes de type glouton. Nous avons vu au chapitre 2 le succès emporté par les algorithmes approchés et la méta-heuristique GRASP.

D'autre part, nous constatons que les contraintes globales qui s'appliquent sur une bonne partie des variables de décision (sinon toutes) sont mal gérées par la technique de propagation des contraintes. Or, des règles algorithmiques sont capables de les traiter plus efficacement que les fonctions de coût, voir [MGS96]. Par exemple la distribution uniforme de travail le week-end est obtenue en utilisant une règle pour trier les employés selon le nombre décroissant du nombre de week-ends affectés. C'est identique à l'approche utilisée dans EQUITIME V3. Il faut cependant vérifier la faisabilité des affectations. En effet les contraintes peuvent facilement détruire l'équilibre crée par la règle, si les affectations sont faites de façon gloutonne et sans retour arrière.

tel-00004380, version 1 - 29 Jan 2004

3.5.2 Les choix généraux du système

Affectation puis propagation

Un tel système peut être réalisé de la façon suivante :

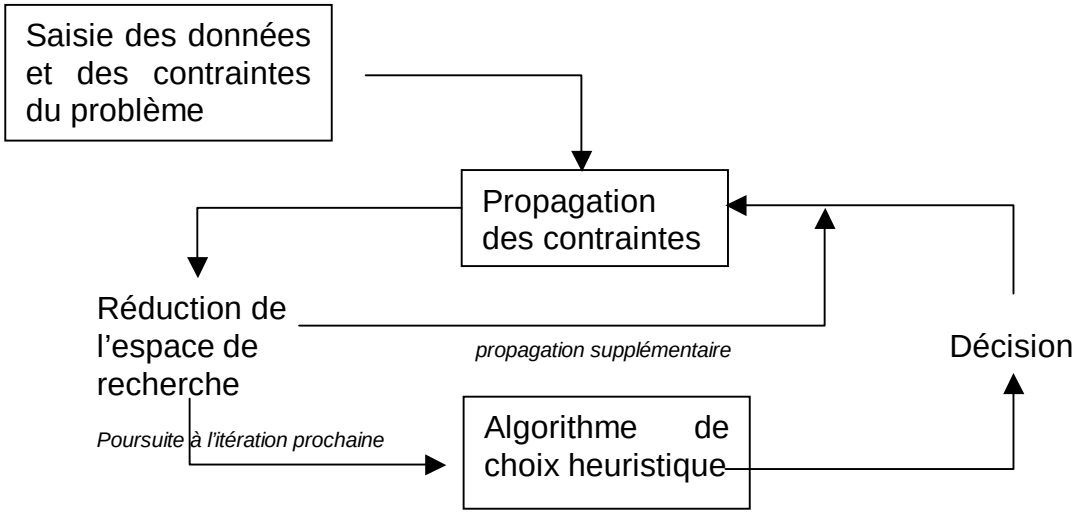


Figure 3.33. Le schéma général du moteur de planification

Ce schéma est inspiré du paradigme de la Programmation par contrainte, avec des variables à domaine fini. Voir [Hen97] pour une description détaillée.

Algorithme de choix heuristique

L'algorithme effectue les choix suivants :

- Choix d'une variable : les heuristiques habituelles incluent le principe « fail-first », jour par jour, de gauche vers la droite.
- Choix d'une valeur, représentant les horaires à affecter aux salariés : habituellement, les besoins sont résolus par ordre croissant. L'idée est de satisfaire d'abord les besoins les plus petits.

La décision consiste à statuer sur l'ensemble des affectations possibles, et l'effectuer.

Propagation des contraintes

Ensuite, il convient de vérifier la cohérence par rapport aux contraintes énoncées, ce qui constitue la phase de propagation des contraintes. La réduction de l'espace de recherche s'en suit. Les cas qui se présentent sont les suivants :

- Lorsque le domaine d'une variable se trouve réduit à une seule valeur, le système déduit que cette variable doit prendre cette valeur et une phase supplémentaire de propagation est nécessaire.
- Lorsque le domaine d'une variable se trouve vide, le système déduit qu'il y a une contradiction dans le système et il faudrait effectuer un retour-arrière.
- Sinon, on poursuit l'énumération des variables jusqu'à leur épuisement.

tel-00004380, version 1 - 29 Jan 2004

Propagation puis affectation

Un autre schéma est possible. Afin d'éviter des retours arrières inutiles, pour une variable candidate et une valeur candidate, on s'assure par avance qu'il n'y aura pas de contraintes transgressées. On fera l'affectation uniquement dans le cas de non-transgression, évitant ainsi le retour-arrière. Cette approche est inspirée de l'optimisation du retour arrière en Prolog, où on distingue le retour arrière profond avec dépilement des substitutions, du retour arrière superficiel, par exemple pour chercher une clause qui s'unifie avec le but courant.

Itération orientée problème ou orientée technologie

L'algorithme du Backtrack effectue des itérations sur les variables du problème à résoudre (voir le glossaire). Appliqué à la planification, cela consiste à trouver des vacations ou horaires pour chaque membre de l'équipe pour chaque jour du planning. Nous considérons que cette approche orientée technique, mène à une sur-affectation inutile des ressources.

Nous avons préféré une approche orientée problème : l' algorithme effectue des itérations sur le besoins à résoudre. Ainsi, si les ressources sont en surnombre ponctuellement, certaines salariés ne seront pas affectées. Cela se traduit par des variables non-instanciées. L'utilisateur a la possibilité de compléter le planning manuellement avec des journées de formation ou de repos.

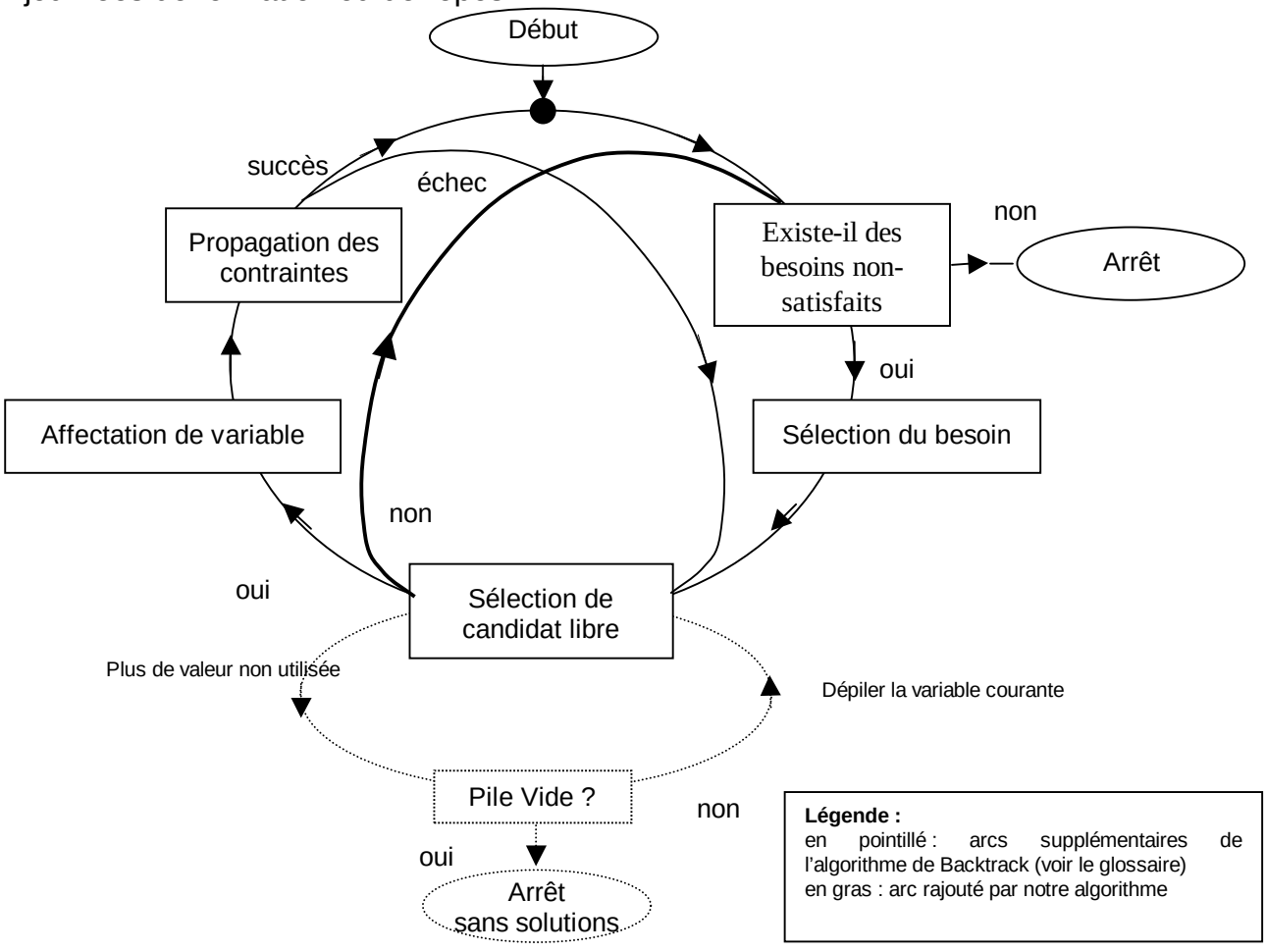


Figure 3.34. Le schéma algorithmique d'EQUITIME

tel-00004380, version 1 - 29 Jan 2004

3.5.3 Exploitation de la symétrie

La symétrie des affectations est très souvent exploitée dans la résolution des PPC pour réduire la complexité de la recherche de solutions par la réduction de la taille des domaines. Ainsi, dans le coloriage des graphes, lorsqu'il faut utiliser une couleur non encore exploitée, toutes les couleurs sont symétriques.

La symétrie permet à partir d'une solution d'en déduire d'autres, en échangeant les valeurs de deux variables ou plus. [Smi01] définit la méthode comme la partition de toutes les affectations possibles en classes d'équivalence. Si l'algorithme montre qu'il n'y a pas de solution avec un membre d'une classe, alors il est futile d'explorer les autres membres de la classe.

Au sein de l'algorithme d'affectation, nous utilisons la symétrie de la façon suivante : en général, un besoin d'étiquettes s'exprime en plusieurs personnes. Lorsqu'on affecte une étiquette de même type à un salarié, on confond tous les besoins. Ainsi on ne fait pas de retour-arrière pour un deuxième besoin du même type.

Nœud représentant une variable
Arcs représentant des valeurs possibles

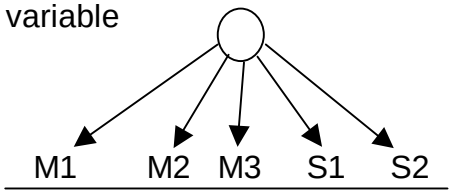


Figure 3.35. Nœud disjonctif dans un arbre de recherche : exemple d'une variable ayant comme domaine {M, S} et les besoins du jour sont 3 matins et 2 soirs

3.5.4 Les algorithmes de base

Variables à domaine fini

Les variables à domaine fini sont des instances de la classe ClsVariable. Les différentes valeurs possibles sont stockées dans un tableau dynamique d'entiers : le nombre d'éléments du tableau est fixe lors de l'initialisation de la variable. Les valeurs sont :

- 0 pour indiquer que le salarié est affecté à un travail hors planification (donc ne contribue pas à satisfaire les besoins de l'équipe).
- 1 pour indiquer que le salarié est au repos.
- 2, etc. pour indiquer que le salarié est affecté à un horaire pour lequel il y a un besoin de l'équipe.

La valeur de chaque élément du tableau est initialisée à –1 : c'est une valeur possible de la variable. Si une valeur est enlevée du domaine, l'élément correspondant du tableau est affecté à 0 ou 1. Les méthodes essentielles de ces variables sont :

- **Peut Instancier (Valeur)** retourne vraie si cette valeur est dans le domaine
- **Peut Travailler** retourne vraie si au moins un code travaillé est dans le domaine
- **Instancier (Valeur)** enlève toutes les valeurs possibles du domaine, sauf Valeur
- **Ne pas travailler** enlève tous les codes travaillés du domaine

tel-00004380, version 1 - 29 Jan 2004

L'algorithme d'affectation

L'algorithme d'affectation s'effectue suivant l'ordre suivant :

- jour par jour : on traite tous les besoins d'une journée avant de passer à une autre journée,
- dans l'ordre chronologique, ce qui facilite énormément le travail des contraintes de transition. Cette démarche est recommandée par les concepteurs de CHIP au cours du projet Gymnaste V1.

Le processus est dirigé par des *besoins restants*. Pour chaque jour de l'horizon, un besoin cyclique (hebdomadaire) a été saisi. Des modifications ponctuelles par besoin au jour le jour peuvent être saisies. On calcule ainsi les besoins restants en enlevant les affectations déjà réalisées sur le planning : cela peut avoir comme origine des affectations manuelles ou des parties d'affectations faites précédemment par le générateur.

Si un besoin restant est alors non satisfait, parmi les candidats disponibles ce jour là (donc non affectés), le système trie suivant un critère d'équilibre choisi par l'utilisateur.

Plusieurs critères sont offerts :

- Equilibre du nombre total d'affectations travaillées
- Equilibre du nombre total d'affectations par code horaire
- Equilibre du nombre total d'affectations par code horaire par jour

Ce critère est néanmoins un critère de qualité et ne préjuge en rien de la faisabilité du problème. Si le besoin non-satisfait est trop grand, il faudra choisir un autre critère d'équilibre.

Dans l'approche **propagation puis affectation**, on vérifie si l'affectation est possible (en fonction de toutes les contraintes en présence, pour le salarié candidat). Au cours de ces tests, l'état du système n'est pas modifié. Si le système ne détecte pas de contradiction, alors l'affectation sera réalisée. Sinon, la prochaine variable sera examinée.

Des algorithmes ont été mis en place pour détecter des conditions limites car si les situations « obligatoires » ne sont pas satisfaites, une contrainte serait violée.

Suite à une affectation ou modification de domaines, le système exploite des algorithmes de filtrage des domaines des variables, pour éliminer les valeurs qui sont en contradiction avec les contraintes.

tel-00004380, version 1 - 29 Jan 2004

Le calcul des tours en PPC

```
For Day = 1 to Number-Days
      Sort(Needs)
      For each Need in Needs
            Remaining = Calculate-Remaining (Need)
            If Remaining > 0 Then
                  Sort(Candidates, Need, Day)
                  For each Candidate in Candidates
                        If Look-Ahead(Candidate, Need, Day) Then
                              Assign(Candidate, Need, Day)
                              Remaining = Remaining – 1
                              If  Remaining  =  0  Then  Exit-Candidate-
                        Loop
                        Else

                        End if
                  Next Candidate
            End if
      Next Need
Next Day
```

Figure 3.36. L'algorithme principal moteur de planification

3.5.5 Réalisation des contraintes

Voici quelques indications sur la réalisation des différentes contraintes dans EQUITIME.

Contraintes d'affectation ponctuelle

La contrainte d'affectation ponctuelle agit directement dans le domaine des variables pour l'ensemble de salariés et pour les jours concernés. Si un seul horaire est proposé, il sera affecté d'office. Si plusieurs horaires sont proposés, alors les valeurs correspondantes seront retenues dans la domaine des variables concernées.

```
Chaque AS doit toujours effectuer les codes [M,S,J] du ? au ?, Sauf pour Agt
Z doit toujours effectuer les codes [M,S] du ? au ?.
```

Contraintes de Disponibilité

La contrainte de disponibilité est interprétée de la même façon que les contraintes d'affectation ponctuelle, sur l'ensemble de salariés et pour les jours de la semaine concernés.

```
AS doit toujours effectuer [?] les jours suivants […] toutes les semaines.
S ne doit jamais effectuer [?] les jours suivants […] toutes les semaines.
```

Contraintes de Transition

L'initialisation des contraintes de transition remplit des tableaux internes au solveur permettant de retrouver lorsqu'on fait une affectation, la/les transitions obligatoires ou interdites, sur des journées consécutives ou non sur une semaine glissante.

```
Pour tous, jamais [N] suivi de [M] du ? au ?.
Pour M. A, toujours Vendredi [S] suivi de Lundi [M] du ? au ?.
```


tel-00004380, version 1 - 29 Jan 2004

Contraintes de Répartition

On stocke les bornes supérieures et inférieures lors de l'initialisation. Lors d'une affectation, on les utilise pour vérifier la consistance.

```
Pour tous, toujours 2 [M] à la suite. Du ? au ?.
Pour tous jamais 6 jours de travail à la suite. Du ? au ?. Sauf pour [M. A]
```

Au moins et au plus : saisir 203 pour au moins 2 et au plus 3 codes. Saisir 104 pour au moins 1 et au plus 4.

Le logiciel EQUITIME offre une option portant sur une semaine civile en plus de l'horizon glissant de journées suites.

Contraintes de Vacations Dues

Les contraintes de vacations dues stockent les valeurs maximum due et minimum due lors de l'initialisation. Pour une semaine, pour un code horaire, et pour un salarié, on retrouve les valeurs dues correspondantes. Ces compteurs sont mis à jour avec chaque affectation et permettent au système de détecter si le seuil maximum est dépassé.

```
A doit faire au moins N et au plus M fois le code horaire [CH] toutes les P
semaines. Du ? au ?.
A doit faire exactement ? fois le code horaire [M] toutes les ? semaines.
```

La négation n'a pas de sens pour cette contrainte.

3.5.6 Priorité entre les contraintes

Les utilisateurs d'EQUITIME ont exprimé le besoin d'expression de priorité entre les contraintes. Par exemple une contrainte de disponibilité (pour un horaire un jour de la semaine) concernant un salarié est considérée plus importante qu'une contrainte d'affectation ponctuelle (applicable sur un intervalle quelconque de temps) ou que le besoin programmé.

Pour réaliser cette priorité dans le cadre de l'algorithme existant, nous avons ajouté une itération supplémentaire par niveau de priorité :

- Contrainte de disponibilité sur des jours précis de la semaine
- Contrainte d'affectation ponctuelle sur un intervalle de temps
- Contrainte de charge

Dans le choix des candidats pour un besoin précis, au cours de chaque itération on ne considère que les candidats faisant l'objet de la contrainte correspondante.

tel-00004380, version 1 - 29 Jan 2004

3.6 L'INTERFACE UTILISATEUR

Nous rapportons dans ce paragraphe l'interface utilisateur des différents versions des logiciels Gymnaste et d'EQUITIME. Ces applications ont été conçues en tant que systèmes interactifs d'aide à la décision. Dans le contexte opérationnel, il y a très souvent des contraintes implicites que les intéressés ne souhaitent pas écrire. Des entreprises fonctionnent grâce aux négociations entre les salariés et l'encadrement de proximité, ce qui nécessite des affectations manuelles. Des plannings ainsi obtenus sont souvent optimaux en ce qui concerne la satisfaction des salariés et cela se traduit par une hausse de la productivité. Des systèmes complexes fonctionnant comme des boîtes noires seraient vouées à l'échec.

3.6.1 Modèles conceptuels principaux d'EQUITIME V3

Tableur dédié à la planification

Le modèle conceptuel principal de notre logiciel de planification est le « *tableur dédié à la planification* ». Le tableau principal au milieu de l'écran affiche les affectations d'horaires par salarié et par jour, que l'utilisateur modifie via la souris. Des informations en ligne se rapportent aux salariés : à gauche le panneau salariés donne les informations statiques telles que le contrat temps, et à droite les données dynamiques de planification comme le nombre total d'heures travaillées, les durées moyennes par semaine ou par mois. Des informations en colonne indiquent le bilan de la planification du jour (manques ou sur-effectifs). Cette disposition correspond à la plupart des plannings muraux, avec une ligne par ressource gérée.

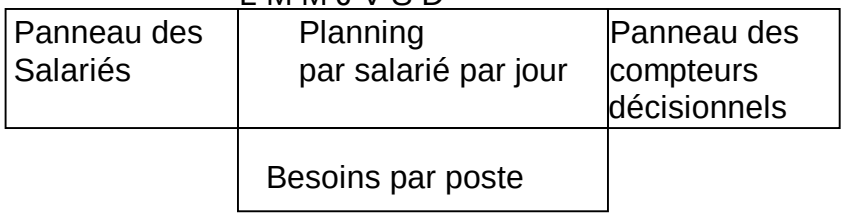


Figure 3.37. Modèle conceptuel en « T » du tableur de planification

Comme un *tableur*, les affichages sont *en temps réel* suite à tout changement d'horaire, donnant l'état des affectations par rapport aux besoins, les compteurs verticaux, et par rapport aux salariés, les compteurs horizontaux.

Nous distinguons d'une part les compteurs décisionnels et d'autre part des compteurs de reporting. Les premiers sont des indications destinées au planificateur pour la prise de micro-décision d'affectation. Mis à jour dynamiquement, ils renseignent le planificateur sur la situation légale, notamment le nombre d'heures de travail ou d'heures supplémentaires, le nombre de congés annuels ou des journées RTT déjà pris. Les compteurs de reporting sont des éléments variables de paie, utiles pour le pilotage opérationnel de l'équipe.

Les informations dynamiques de planning sont données par les compteurs décisionnels sur le panneau de droite. Lorsque les valeurs des compteurs dépassent des seuils paramétrés par l'utilisateur, la cellule est affichée en rouge, en blanc si la valeur égale le

tel-00004380, version 1 - 29 Jan 2004

seuil, ou en vert[17]. Malheureusement, la cellule est parfois invisible et il faut alors faire défiler les compteurs pour pouvoir la voir.

Le panneau des besoins par poste dans la partie inférieure de l'écran affiche l'écart entre le nombre d'affectations sur le planning et le nombre actuellement affecté. Le choix des couleurs (rouge pour danger : sous-effectif, et vert sur-effectif) permet de visualiser les besoins même avec un niveau de zoom élevé où les chiffres ne sont plus visibles.

Le crayon et la gomme

La figure ci-après illustre l'écran principal de la version 3 qui dispose de la métaphore « *le crayon et la gomme* », rendue possible avec la fonction d'annulation.

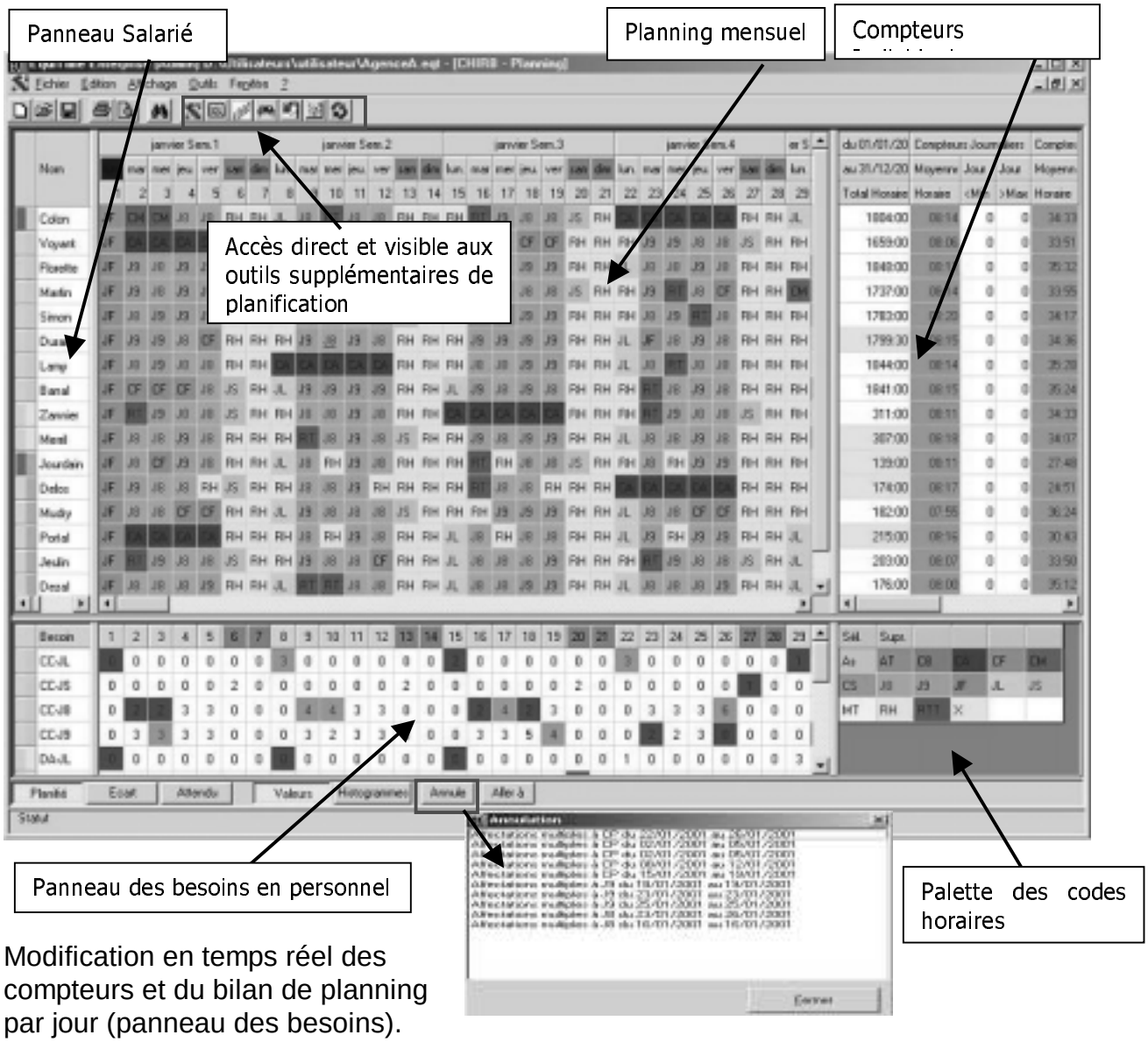


Figure 3.38. EQUITIME Version 3 avec l'écran des annulations

Effectivement, les modifications sont stockées dans une pile dont l'utilisateur peut demander l'annulation dans un ordre quelconque. Cette opération est une facilité qui

[17] L'affichage en couleur suivant les seuils est maintenant une fonction standard dans les tableurs courants.

tel-00004380, version 1 - 29 Jan 2004

restaure l'affectation précédente de la cellule. Comme elle n'est pas transitive, un ordre différent d'annulation donnera un planning résultant différent. Même les affectations proposées par les outils automatiques y sont stockées afin de pouvoir les annuler d'un seul coup.

La liste des codes horaires est stockée dans une palette en bas à droite, facilitant ainsi le choix.

L'élaboration progressive de planning

L'utilisateur contrôle complètement son planning : il peut demander l'aide des générateurs automatiques de planning cyclique ou acyclique. Il peut en effacer une partie, ajouter quelques affectations et relancer le générateur acyclique. A la vue des différents compteurs, il peut compléter et finaliser le planning. A tout instant, l'utilisateur peut annuler les différentes actions qu'il a effectuées.

Ainsi, l'élaboration de planification peut être progressive :

1. Activer les contraintes à long terme, telles que les indisponibilités dues aux congés annuels, formation, etc.
2. Activer les contraintes quotidiennes telles que les congés exceptionnels
3. Pré-affecter les salariés qui ont exprimé des demandes
4. Lancer le Générateur Automatique de Planning Acyclique et évaluer les résultats
5. Modifier manuellement la partie de la solution qui n'est pas satisfaisante
6. Pré-affecter certains salariés et changer quelques contraintes (ou leurs paramètres)
7. Réitérer les étapes 4-5-6-7 jusqu'à ce que toute la solution soit satisfaisante

Les affectations automatiques sont en 'blanc' (Gymnaste)

L'outil permet de combiner des affectations manuelles et automatiques. Afin de mieux distinguer ce que propose le générateur automatique de planning par rapport aux affectations précédentes, les propositions du générateur peuvent être affichées différemment. Le logiciel Gymnaste utilise le texte en couleur blanche ; on peut aussi afficher le texte en italique.

L'utilisateur peut valider les affectations qu'il estime bonnes, en les remettant en noir. Lors du lancement, le moteur effacera les propositions blanches avant de faire de nouvelles affectations (en blanc). La métaphore des affectations en blanc rassure les utilisateurs : toute modification indésirable du moteur peut être reprise par l'utilisateur. Malheureusement, cela limite le nombre de couleurs des codes horaires (les couleurs trop pales ne peuvent plus être utilisées).

Des photo-copies de planning

Un logiciel de planning nécessite une base de données pour gérer les données du personnel et du planning sur un horizon d'une ou plusieurs années. Afin de faciliter l'utilisation d'un tel logiciel par des planificateurs, nous évoquons la métaphore de la feuille de planning.

tel-00004380, version 1 - 29 Jan 2004

Une base de données d'un planning est assimilée à une feuille de planning. On peut modifier un planning, faire une (photo-) copie pour affiner un planning prévisionnel, faire des simulations différentes pour « optimiser » le planning, etc.

3.6.2 Les outils annexes de planning

La version 3 ajoute de nombreuses interfaces nouvelles : les plannings annuels et postés et les vues cumuls et soldes. Ces outils fournissent des informations complémentaires pour aider le planificateur à optimiser son planning.

Communication

Or, tous ces outils sont communicants : dès qu'il y a un changement de planning dans un outil, les autres outils sont informés et se mettent en conformité. Bien entendu, plus il y a d'outils ouverts, plus la charge en machine pour la mise à jour devient lourde. Comme la mise à jour est synchronisée avec l'interaction graphique, l'utilisateur perd la main après chaque modification pendant que les différents outils se remettent à jour.

Planning annuel

Le planning annuel restitue le planning d'un employé au cours de douze mois, débutant le mois demandé sur l'interface indiqué en rouge dans la figure ci-après.

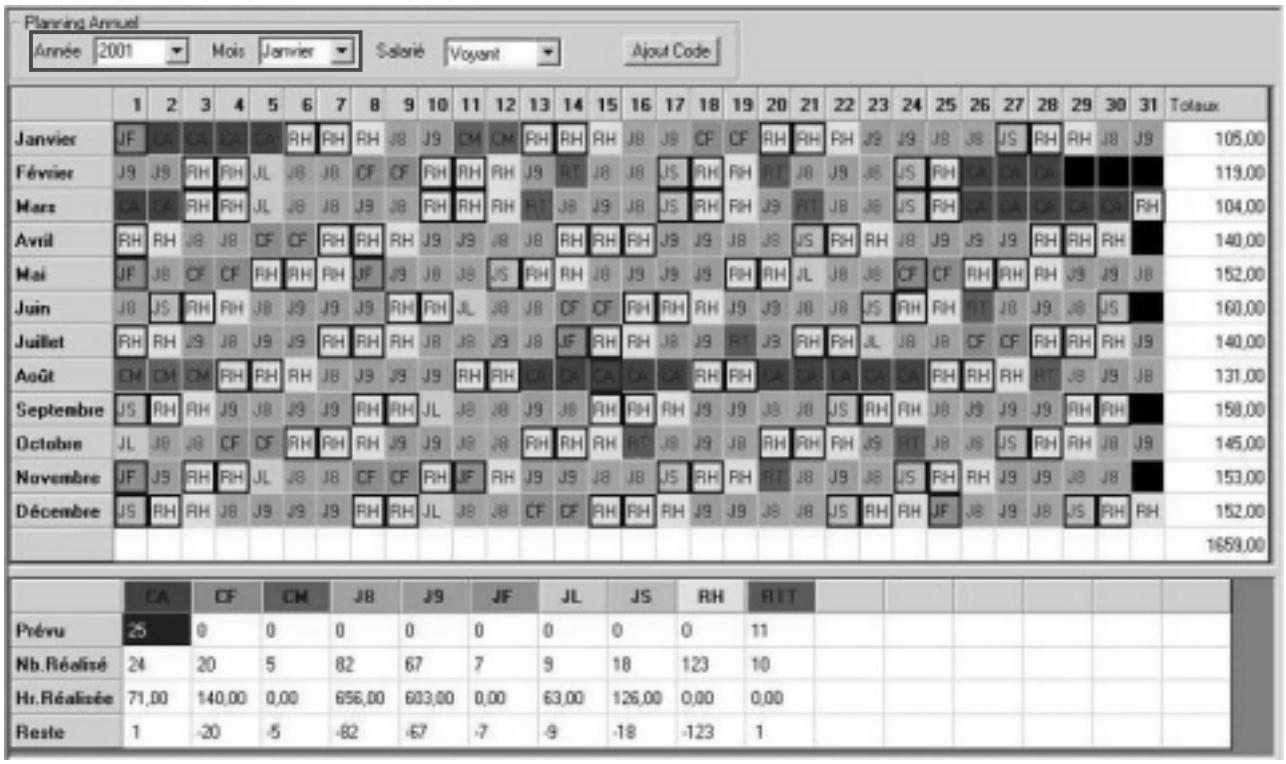


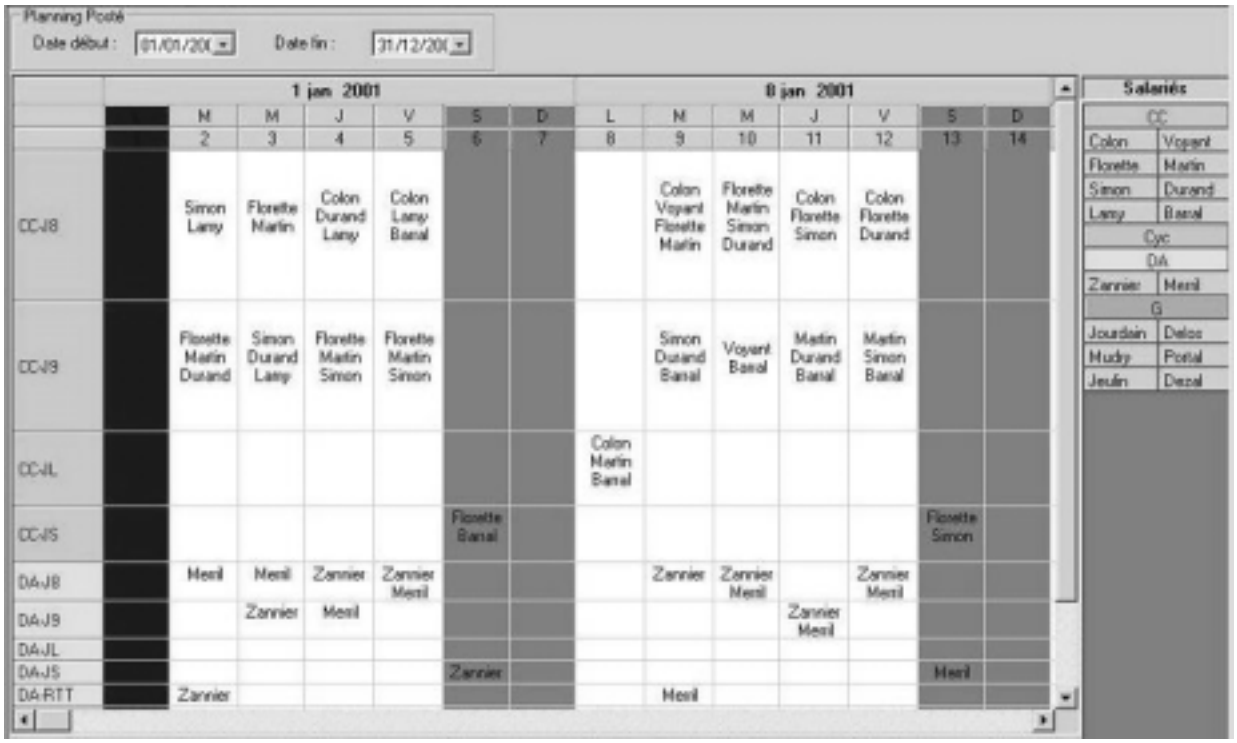
Figure 3.39. EQUITIME Planning Annuel

Sur le planning annuel, chaque ligne représente le planning d'un mois. Les week-ends qui sont grisés sur le planning mensuel, sont entourés en gris ; les jours fériés qui sont en fond bleu sont entourés en bleu. En dernière colonne, on affiche le total des heures travaillées. En dernière ligne sur cette dernière colonne, on obtient le total annuel.

tel-00004380, version 1 - 29 Jan 2004

Le calcul des tours en PPC

Sur la partie inférieure du planning annuel, on dispose de compteurs sur le nombre d'horaires dans ce planning, ainsi que les heures travaillées. Si le nombre prévu de chaque horaire a été saisi, le système affiche le nombre restant.



Poste	Planning	Choix de salariés

Sur tous les plannings, nous avons gardé l'axe horizontal comme l'axe du temps.

Figure 3.40. EQUITIME Planning Posté

Cet outil est intéressant parce qu'il montre le modèle dual des variables de décision, ce que nous avons vu au chapitre 2.

L'outil peut être utilisé pour la visualisation des données ainsi que pour les ajustements fins une fois le planning établi. Elle permet à l'utilisateur de faire des améliorations locales à partir d'une solution partielle ou rapprochée, éventuellement au terme d'une négociation inter personnelle.

Exemple : dans la figure ci-dessus, il manque une personne en CC-J9 le 10 janvier. On pourrait faire venir M.Durand en J9 au lieu du J8 initialement prévu, avec une heure de présence en plus du J8.

tel-00004380, version 1 - 29 Jan 2004

Vues Soldes et Cumuls

Les deux vues Soldes et Cumuls permettent d'évaluer le planning en cours et de contrôler l'équité.

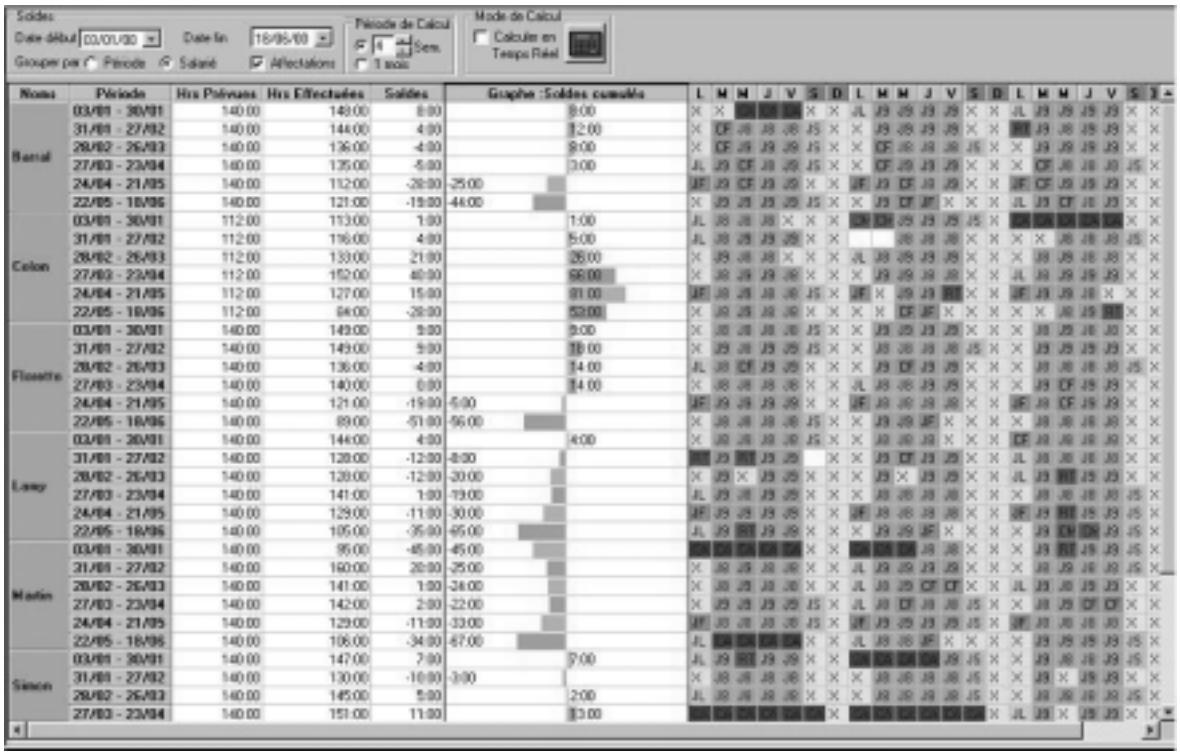


Figure 3.41. EQUITIME Vue Soldes

La vue Cumuls permet de contrôler le nombre d'affectations par salarié par code horaire et suivant le jour de la semaine ou suivant les jours fériés.

Figure 3.42. EQUITIME Vue Cumuls et son paramétrage

tel-00004380, version 1 - 29 Jan 2004

3.6.3 EQUITIME V4 : interface utilisateur

Cette version du produit n'est pas encore disponible aujourd'hui. On ne peut qu'évoquer au passage les améliorations apportées par cette version.

L'ergonomie

Nous avons décidé qu'objectivement une meilleure ergonomie s'obtient en réduisant le nombre de clicks de la souris pour accéder à une fonction ou en réduisant la distance parcourue par la souris pour les opérations les plus courantes. Pour parvenir au second objectif, nous proposons un système où des fenêtres détachables peuvent être déposées à un endroit propice. Ainsi au cours de l'affectation manuelle dans le choix d'un code horaire, cela permet de réduire les allers et retours de la souris entre la palette et du planning.

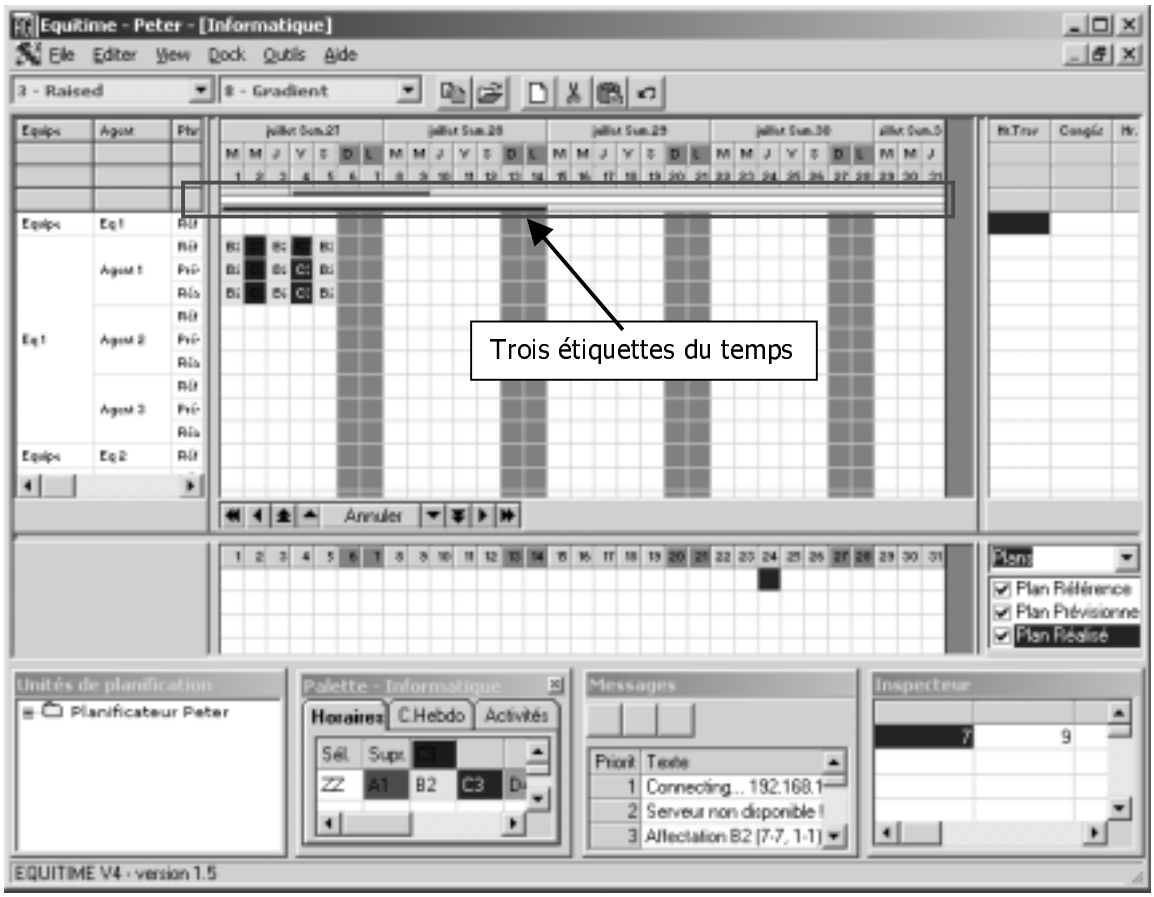


Figure 3.43. EQUITIME Version 4 : vue globale

La métaphore des cadrans

L'utilisateur peut visualiser plusieurs plannings : chaque planning dispose d'informations qui lui sont spécifiques, disposées sur les panneaux de la version 3. Dans la version 4, nous proposons des cadrans. Ces cadrans servent à centraliser l'attention de l'utilisateur sur des dimensions supplémentaires :

- *Cadran Navigation* : Ce cadran permet de naviguer parmi les organisations organiques de l'entreprise et de retrouver les plannings correspondants.
- *Cadran Message* : Ce cadran permet d'afficher des messages d'alertes, disposés dans une liste consultable à tout instant. En version 3, si la cellule affichant une donnée dans le

tel-00004380, version 1 - 29 Jan 2004

panneau compteurs passe au rouge, il y a danger que l'utilisateur ne le prenne pas en compte si la cellule n'est pas visible. Ce cadran résout ce problème.

- *Cadran Inspecteur* : Ce cadran fait un zoom sur un ensemble d'affectations sélectionnés dans le planning. Disponible en permanence sur l'écran, il livre des renseignements précieux pour le planificateur. Normalement invisible, l'inspecteur de la version 3 doit être activé par l'utilisateur avec le bouton droite de la souris. Si un message d'alerte est sélectionné, ce cadran affichera les détails correspondants.

Tout comme la palette, ces trois cadrans sont détachables pour être disposés à volonté n'importe où sur l'écran, au dessus des différents plannings. On peut aussi les fermer si l'on ne souhaite pas les consulter, ou si l'on veut réserver tout l'écran pour le ou les plannings.

Chaque jour peut être qualifié d'un nombre illimité d'étiquettes, souligné en rouge dans la Figure 3.43. Cette figure montre aussi l'affichage des plans référence, prévisionnel et réalisé, choisi par l'utilisateur dans le panneau en bas à gauche.

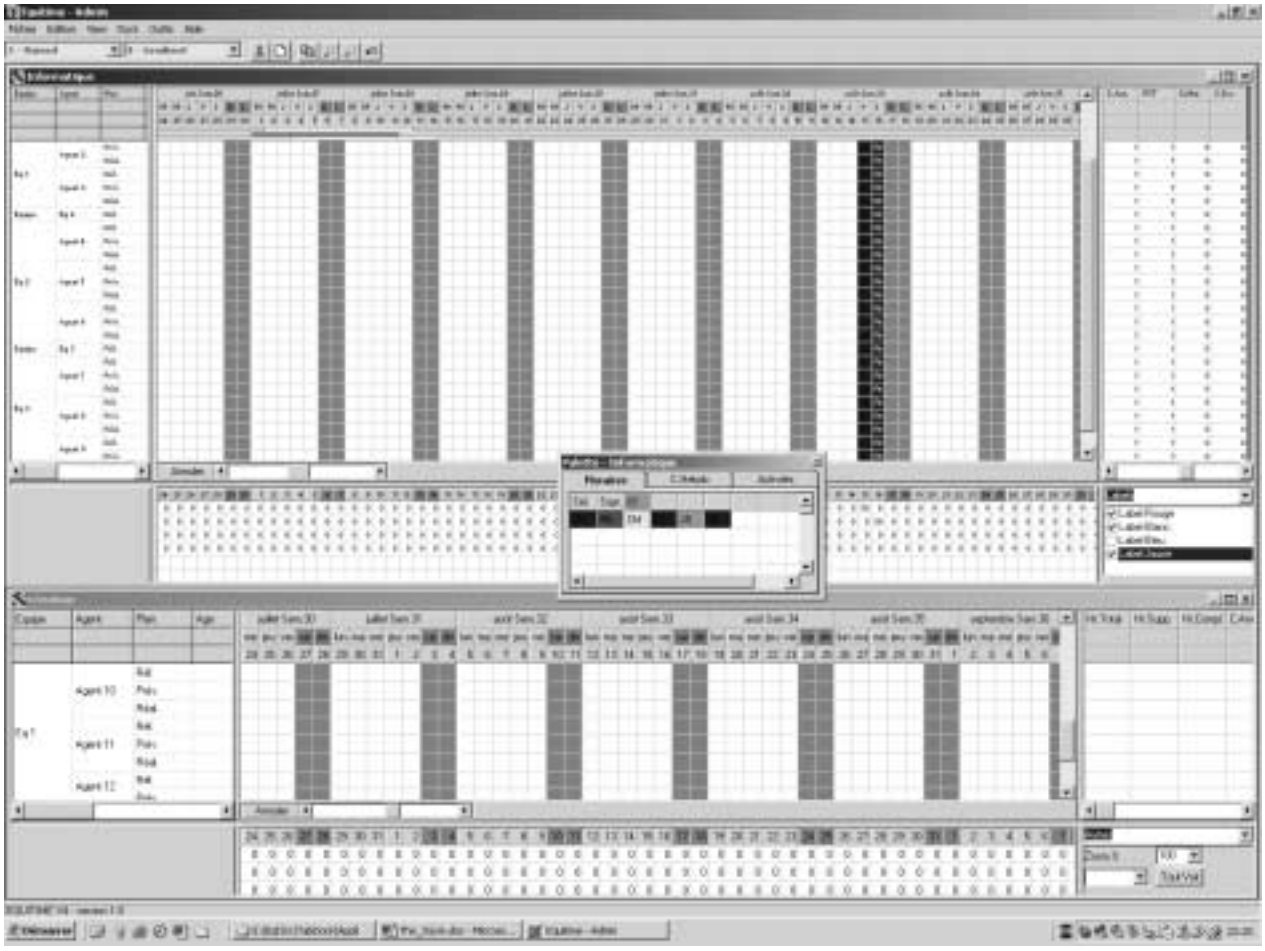


Figure 3.44. EQUITIME Version 4 avec un seul cadran ouvert

Tous les cadrans sauf la palette ont été fermés dans la Figure 3.44. Ainsi l'utilisateur peut consacrer le plus de place sur l'écran aux plannings.

tel-00004380, version 1 - 29 Jan 2004

3.7 LES LIMITES DES APPROCHES PRECEDENTES

L'expérience acquise dans l'exploitation pratique des logiciels Gymnaste et d'EQUITIME, nous permet d'analyser les limites de ces approches. Elles portent sur les modèles et les méthodes.

3.7.1 Les restrictions du modèle

Comme le même modèle est utilisé dans les deux logiciels, ces remarques s'appliquent aux deux.

a. Chaque agent a une compétence unique :

Ce modèle relativement simple ne peut traiter le cas des multiples compétences que dans l'ordre de priorité décroissante. Bien entendu, cela ne convient pas dans de nombreuses situations de planning.

- Cette restriction sera levée dans les modèles présentés au chapitre 5.

b. Le modèle des contraintes peut engendrer des conflits :

Une même personne peut être concernée par plusieurs contraintes éventuellement contradictoires sur certains jours. Dans la pratique, les planificateurs ne doivent pas définir de telles contraintes éventuellement contradictoires.

- Un système de vérification a été envisagé pour détecter des éventuels recouvrements dans le temps et sur les agents, entre deux contraintes de la même catégorie.

c. Le modèle des étiquettes :

En remplaçant les absences de type repos hebdomadaire, repos compensatoire, congé annuel ou congé exceptionnel, une seule étiquette repos suffirait : ce modèle permettra de gagner en complexité [Heu96]. Cependant, elle ne permet pas de modéliser les absences dont la durée compte dans le total des heures payées (par ex. formation). Ce dernier peut ainsi dépasser le total des heures par semaine.

- Il faudra deux codes qui ne font aucune contribution aux charges de la journée : le repos hebdomadaire qui n'a aucune durée de travail et le travail hors charges qui est considéré comme du travail. Ce dernier doit être pris en compte dans les contraintes de vacations dues ou de répartition (dans « tous » codes travaillés).

  D'une manière générale, le niveau journalier d'abstraction permet de réduire la complexité des calculs mais il peut engendrer des plannings illégaux, par exemple au niveau des heures hebdomadaires ou mensuelles. Le modèle ne tient compte que des affectations suivant les besoins du planning. Les autres affectations qui comptent dans le temps de travail sont ignorées, donc le temps total hebdomadaire peut dépasser la limite légale.

tel-00004380, version 1 - 29 Jan 2004

- Il faut que le générateur de planning prenne en charge le nombre d'heures hebdomadaires, par exemple dans le cadre d'un modèle multi-niveaux (MLM) décrit au chapitre 5.

d. Vérification des contraintes :

Les contraintes ne sont actives que pendant la génération automatique de plannings. Suite aux affectations manuelles, Il faudra lancer systématiquement une vérification des contraintes et au cas échéant, produire des alertes. Cette fonction ne présente aucune difficulté théorique, mais au niveau pratique, ces tests prennent du temps à s'exécuter.

- Dans la présentation de la nouvelle version, nous avons réservé le cadran Messages pour recevoir des messages provenant d'un serveur qui lancera des tests régulièrement, sans consommer les ressources matérielles des postes clients.

  Pour aller plus loin, à la manière des correcteurs d'orthographe des logiciels traitements de texte, lorsqu'une conséquence due à une contrainte est imminente elle doit être proposée automatiquement, par exemple les transitions obligatoires. Bien entendu, il faut pouvoir annuler ces propagations automatiques.

e. Le respect de l'équité :

Les contraintes de vacation due ont été conçues pour que le planning respecte l'équité. Ainsi l'utilisateur doit compter les affectations de chaque salarié sur un horizon passé et ré-paramétrer ces contraintes : c'est très pénible à faire manuellement.

- EQUITIME v3 implémente l'équité de façon plus simple, en tant que règle heuristique d'affectation. Nous utilisons dans l'algorithme d'énumération une heuristique cherchant à équilibrer l'un des trois paramètres :
  - Nombre total des heures réalisées par chaque salarié
  - Nombre total de chaque vacation réalisée par chaque salarié
  - Nombre total de chaque vacation réalisée par jour de semaine par chaque salarié

  Suivant l'heuristique choisie par l'utilisateur, les bonnes solutions sont proposées en priorité. Attention ces heuristiques peuvent conduire à des conflits résultant en besoins non-satisfaits.

3.7.2 La recherche de solutions par contraintes globales

a. Les contraintes de cardinalité

Ces contraintes provoquent beaucoup de retours en arrière. Le respect simultané des sommes en ligne (pour un agent) et en colonne (pour un jour) est très difficile à satisfaire en CHIP car les contraintes globales `among` et `séquence` sont appliquées indépendamment sur les mêmes variables, et elles s'ignorent.

tel-00004380, version 1 - 29 Jan 2004

- Une méthode encore plus globale doit être utilisée, portant à la fois sur des lignes et des colonnes. Elle pourrait détecter qu'il n'y a plus de solutions et ainsi éviter le temps pour le prouver, telle que celle proposée par [CGL95] présentée au chapitre 2 du mémoire.

  Au chapitre 5, nous présenterons des conditions nécessaires dans le contexte des modèles multi-niveaux, ainsi que des méthodes heuristiques pour générer un planning.

b. L'équité des affectations :

Les contraintes de *vacation due* (qui s'appliquent sur de nombreux jours) sont très difficiles à satisfaire, puisque l'énumération des solutions se fait jour par jour.

c. Problème pratique d'application

Il s'avère que les contraintes globales ne sont pas applicables en pratique sur tout l'horizon de planification. Ainsi en présence des pré-affectations telles que des congés annuels sur plusieurs semaines les contraintes de répartition des vacations de nuit ne doivent pas être appliquées. Comme les pré-affectations doivent prévaloir sur les contraintes globales, il convient alors de ne pas les appliquer.

d. Les contraintes préférentielles

Les contraintes de type *si possible* et *si possible pas*, peuvent entraîner des affectations qui empêchent par la suite la satisfaction des contraintes obligatoires. Leur traitement par point de choix est loin d'être performant. La recherche locale (par exemple échange 2-opt) pour respecter au mieux les préférences, serait une bien meilleure approche.

3.7.3 La génération de solutions

En général, EQUITIME résout des plannings dans des situations réelles, avec suffisamment de ressources en réserve. Dès que les marges en ressources sont insuffisantes, le moteur donne des résultats peu satisfaisants : en certains jours, des demandes de personnel ne sont pas satisfaites. Une analyse de cette situation suit :

L'approche propagation puis affectation

A priori intéressante parce qu'elle évite le retour arrière « profonde », cette approche s'avère pénalisante car on ne peut vérifier que les effets premiers d'une contrainte et non les effets cumulés de plusieurs contraintes.

- Afin d'effectuer les tests en cascade : les contraintes peuvent s'influencer mutuellement. Donc il faut pouvoir faire des retours en arrière, ou un traitement des échecs (ou l'absence définitive de solution faisable). Nombreuses méthodes de recherche locale décrites au Chapitre 2 du mémoire ont été proposées pour remplacer le retour-arrière systématique. Effectivement, le retour-arrière peut être

tel-00004380, version 1 - 29 Jan 2004

simulé par la désaffectation d'une variable. Il faut éviter le bouclage, éventuellement avec une liste tabou.

- A la place du retour-arrière systématique, on pourrait faire une suite heuristique d'échanges d'affectations pour absorber le manque d'effectif. Plusieurs méthodes d'échanges seront présentées au chapitre 5 pour les différents niveaux.

Equilibrage

Les propositions de [MGS96] plaident pour l'emploi des règles d'affectation parce qu'elles permettent d'obtenir des plannings vérifiant des conditions globales impliquant une bonne partie des variables de décision.

Dans un contexte sous contraintes, certaines règles ne peuvent pas s'appliquer au moment de l'affectation. Il faut implanter des méthodes supplémentaires pour s'assurer que les règles sont bien respectées. Une idée est de faire des échanges entre affectations, tout en tenant compte des contraintes de transition.

Affectation de repos

EQUITIME adopte la méthode du système Gymnaste où les jours de repos sont affectés en même temps que les besoins. Lorsque les besoins du jour sont satisfaits, le personnel restant sera affecté au repos. Cependant, des règles récentes portant sur le repos ont été rendues légales ; en France dans le domaine de la santé par exemple, chaque salarié a droit à 4 jours de repos sur deux semaines, dont au moins deux consécutifs et dont l'un est un dimanche. Ces besoins en jours de repos ne sont pas pris en compte par les systèmes Gymnaste, par conséquent EQUITIME.

La satisfaction des besoins en repos doit être traitée de façon spécifique car on n'exprime pas de besoin en nombre de repos par jour. Par conséquent, on ne l'affecte pas comme les autres codes horaires. Il faut le traiter avant tout besoin d'horaires.

3.7.4 Conclusions

Il est certain que les contraintes globales constituent une approche originale dans la résolution de problèmes de grandes tailles.

Cependant, les contraintes globales `among` et `séquence` de CHIP ne sont pas suffisantes pour résoudre le problème de calcul de tours car elles s'ignorent. D'une part, elles sont appliquées indépendamment sur les mêmes variables, et elles n'effectuent pas de raisonnement ensemble. D'autre part, les pré-affectations les empêchent d'être appliquées sur tout un horizon de planning.

Dans le cas de contraintes contradictoires, les utilisateurs souhaitent choisir les priorités associées à chaque contrainte et pilotent la stratégie de relaxation des contraintes.

tel-00004380, version 1 - 29 Jan 2004

4 LE DEROULEMENT DES CYCLES AUTOUR DES PRE-AFFECTATIONS

Résumé :
Si la littérature scientifique sur la création de cycles abonde, il a eu peu d'intérêt dans le déroulement des cycles. Dans la majorité des situations, les cycles sont déroulés ou appliqués systématiquement.

Dans ce chapitre, nous décrivons notre travail de déroulement des cycles autour des pré-affectations comme des congés annuels. D'une part, il y a une volonté d'adhérer à un cycle, ce qui est mathématiquement précis. D'autre part, il y a le souhait de rendre le planning aussi humain que possible. Il était impératif de relaxer des contraintes de cycle, de déformer le déroulement du cycle suivant les départs programmés en congés annuels.

Nous proposons un modèle capable de couvrir une grande diversité de cycles de travail. Certains de ces cycles trouvent leur origine dans les centres pénitentiaires des pays européens voisins.

4.1 GENERALITES

Dans ce chapitre, nous étudions un aspect opérationnel de la planification cyclique. Réalisé pour le compte du Ministère de la Justice en 1999-2000, le projet vise la création automatique de planning basé sur des cycles de travail approuvés par l'administration pénitentiaire. L'objectif est de générer un planning annuel pour une ou plusieurs équipes d'agents de sécurité suivant un ensemble de paramètres prédéfinis.

Historiquement, les cycles journaliers ont été mis en œuvre depuis plusieurs décennies parce qu'ils sont souples : le cycle peut être rallongé ou raccourci facilement. Comme ils ne donnent que peu de repos le week-end, ils sont socialement inadaptés. Pour corriger ce défaut, les cycles hebdomadaires ont été créés, en prenant le cycle journalier de base et adaptant certains jours afin de fournir le personnel demandé avec un rythme de travail adéquat. Un acquis social est de fournir un repos hebdomadaire un week-end sur deux.

La base de cycles réglementaires contient 14 cycles, fruit de l'expérience sur le territoire français et fruit de collaboration avec des instances européennes. L'application visée doit dérouler ces cycles.

Nous présentons ici quelques éléments pour décrire une situation d'une grande complexité. Pour pourvoir un poste de travail en permanence 24H sur 24, sept jours sur sept et sur toute l'année, il faut en général 6 agents à plein temps. A un jour donné, 3 personnes travaillent le matin, soir et nuit, 1 en repos après travail de nuit, 1 en repos hebdomadaire et 1 en congés annuels. En France, on dispose de 30 jours de congés annuels légaux et 21 jours fériés, qui seront compensés par du repos. Donc une équipe de 6 agents aura 6*51=306 jours d'absence à tour de rôle. Il en reste 365-306 = 59 jours où tous les agents sont présents et il y potentiellement trop de ressources par rapport aux besoins constants.

Les équipes de travail peuvent être organisées différemment. Suivant le cycle, on peut intégrer une ou plusieurs équipes volantes qui permettent de renforcer les effectifs

tel-00004380, version 1 - 29 Jan 2004

durant la journée, ce qui permet d'avoir une différence entre les effectifs du jour et de la nuit.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
R_1	A_L	A_L	A_L	Sp	Sp	Sp	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4
R_2	W_2	W_3	W_4	A_L	A_L	A_L	Sp	Sp	Sp	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1
R_3	W_3	W_4	W_1	W_2	W_3	W_4	A_L	A_L	A_L	Sp	Sp	Sp	W_1	W_2	W_3	W_4	W_1	W_2
R_4	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	A_L	A_L	A_L	Sp	Sp	Sp	W_1	W_2	W_3
R_5	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	A_L	A_L	A_L	Sp	Sp	Sp
R_6	Sp	Sp	Sp	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	A_L	A_L	A_L

R_i	Sp	Sp	Sp	Sp	$\mathbf{S_1}$	$\mathbf{S_2}$	$\mathbf{S_3}$	A_L	A_L	A_L	…	A_L	A_L	S_1	S_2	S_3	S_4	S_5

Figure 4.45. Intégration d'une équipe volante dans un cycle de 4 semaines ou un cycle de 5 jours

On peut envisager d'organiser les congés : *par équipe entière ou individuellement.* Dans le premier cas, toute l'équipe s'absente en même temps et peut être gérée comme une ressource unique. Dans le deuxième cas, un nombre adéquat d'agents au sein de chaque équipe prend ses congés. Suivant l'exemple de 6 agents pour 1 poste en permanence décrit ci-dessus, 1 agent parmi 6 prendra ses congés.

Les contraintes que nous énumérons au paragraphe 4.2 sont difficiles à satisfaire, et il nous paraît nécessaire d'utiliser une méthode de recherche arborescente pour aboutir à des plannings cohérents.

Le problème avec le déroulement systématique est certaines contraintes dures doivent être respectées. Par exemple la contrainte liée aux congés annuels : les dates étant prédéterminées d'avance, les vacations précédent le départ doivent être du repos hebdomadaire. Trouver des plannings satisfaisants est un problème complexe

- *Cycles Journaliers* : possibilité de rallonger/raccourcir le cycle
- *Cycles Hebdomadaires* : Glissement des congés, c'est à dire les avancer ou retarder d'une ou deux semaines au maximum, sauf pour les congés d'été.

tel-00004380, version 1 - 29 Jan 2004

Le déroulement des cycles

4.2 LA DEFINITION DU PROBLEME

Le problème du déroulement de cycles est défini par l'ensemble des contraintes présentées dans ce paragraphe. Nous supposerons que les cycles sont légaux, nous n'exposons que les contraintes de déroulement.

4.2.1 Les contraintes de base

Définition : Une vacation S_i est un ensemble d'heures de travail légal.

4.2.1.1 La contrainte de cycle

Définition : Un cycle de travail journalier est défini par une séquence de N codes de travail, où N n'étant pas un multiple de 7. Un code de travail D_i est une vacation S_i. On suppose que le cycle est légal au niveau du nombre d'heures travaillées, et on ne fait que dérouler les cycles. En conséquence, on peut ignorer les horaires précises de chaque vacation[18], mais on fait l'hypothèse que le repos hebdomadaire est compris dans le cycle. Ici, la durée d'un code ou l'intervalle est un jour.

Définition : Un cycle de travail hebdomadaire est défini par une séquence de N codes de travail, chaque code W_i étant une séquence de vacations S_i pour chaque jour de la semaine. L'intervalle est la semaine.

D_1	S_1
D_2	S_2
D_3	S_3
D_4	S_4
D_5	S_5

	L	M	M	J	V	S	D
W_1	S_1	S_1	S_5	S_5	S_3	S_3	S_3
W_2	S_5	S_5	S_1	S_1	S_5	S_5	S_5
W_3	S_3	S_3	S_5	S_5	S_1	S_1	S_1
W_4	S_5	S_5	S_3	S_3	S_5	S_5	S_5

Figure 4.46. Exemple d'un cycle de 5 jours et un cycle de 4 semaines

Définition : Dans les deux cas, N est la période du cycle. Au bout de cet intervalle de temps, un salarié retrouve son planning de départ. Pour simplifier la description, $Code_i$ dénote D_i dans le cas du cycle journalier ou W_i dans le cas du cycle hebdomadaire.

Une ressource est affectée à un $Code_i$ puis au $Code_{i+1}$ à l'intervalle suivant. A la fin du cycle, on revient sur $Code_1$. Pendant un intervalle quelconque, au moins une ressource doit être affecté à chaque code du cycle. L'utilisation de tels cycles sans des congés annuels est illustrée ci-dessous :

[18] Ex. S_1=matin, S_2=après midi, S_3=nuit, S_4=repos nuit et S_5= repos hebdomadaire

tel-00004380, version 1 - 29 Jan 2004

intervalle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R_1	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5
R_2	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4
R_3	D_3	D_4	D_5	D_1	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1
R_4	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2
R_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1

Figure 4.47. Application directe d'un cycle journalier de 5 jours pour 5 ressources. On obtient un planning hebdomadaire analogue en remplaçant les codes D_i par W_i : l'intervalle est une semaine.

Contrainte de Cycle de période N : Chaque ressource est affectée successivement aux Code $_i$ puis Code $_{i+1}$, et puis Code $_N$ au Code$_1$. Code $_i$ est soit un code journalier D_i dans le cas des cycles journaliers, soit W_i pour les cycles hebdomadaires, N étant la période du cycle dans les deux cas.

4.2.1.2 La contrainte de charge

Chaque code du cycle est affecté à au moins une ressource au cours de chaque intervalle. Au niveau des charges, la somme des présences de tous les codes doivent satisfaire les besoins de l'intervalle. Plus précisément, pour les cycles journaliers, la somme des présences des vacations D_i doit couvrir les besoins de la journée. Pour les cycliques hebdomadaires, la somme des codes W_i (donc les présences de chaque jour de la semaine) doit couvrir les besoins de la semaine.

Contrainte de charge : Pendant chaque intervalle de l'horizon, chaque code du cycle est affecté à au moins une ressource. Si deux ou plusieurs ressources sont affectées à un code, alors il y a surplus d'effectifs.

Conséquence : pour éviter des surplus, on applique un cycle de période N toujours pour N ressources.

4.2.1.3 La contrainte de repos journalier

Les salariés doivent avoir suffisamment de repos entre deux jours de travail. Par exemple, après une vacation de nuit se terminant vers 7 ou 8H du matin le lendemain, les salariés doivent avoir une journée de repos. Dans les cycles hebdomadaires, on suppose que cette contrainte est déjà respectée pour les vacations au sein de chaque code W_i. Lors du déroulement, on doit s'assurer que cette contrainte est respectée pour les vacations du dernier jour d'un W_i et le premier jour du code suivant W_{i+1}.

Contraintes de repos journalier : Certains codes du cycle doivent suivre des codes spécifiques afin d'assurer suffisamment de repos journalier.

De la même façon pour les cycles hebdomadaires, certains codes D_i doivent suivre des codes spécifiques.

tel-00004380, version 1 - 29 Jan 2004

Le déroulement des cycles

4.2.1.4 La contrainte de congé annuel

Les congés annuels seront considérés fixes, car ils sont attribués sur plusieurs années afin de permettre un départ en plein été pour chaque salarié. Or, dans le secteur tertiaire, les congés commencent le lundi et les salariés bénéficient du week-end précédent. La règle équivalente dans le travail continu est la suivante :

> *Contrainte de congé annuel : Le jour précédent les congés annuels doit être le repos hebdomadaire. Inversement, le jour suivant les congés ne doit pas être le repos hebdomadaire.*

Cette contrainte peut être assurée par une contrainte de suite entre les vacations. Dans le cas des cycles hebdomadaires, les codes W_i avec des repos hebdomadaires en fin de semaine doivent précéder les congés. Les W_i avec des week-ends travaillés doivent suivre les semaines de congés.

Dans notre cas, la règle étant de fournir un week-end de repos sur deux, et seuls les codes pairs, par ex. W_{2i}, offrent du repos en week-end. En conséquence, il faut que les codes pairs précèdent les congés, cf. Figure 4.49.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
R_1	S_p	C_A	C_A	C_A	C_A	C_A	C_A	C_A	C_A	C_A	C_A	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1
R_2	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	C_A	C_A	C_A	C_A	C_A	C_A	C_A	C_A	C_A	C_A	??
R_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5
R_4	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4
R_5	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3
R_6	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2	S_3	S_4	S_5	S_1	S_2

Figure 4.48. Application d'un cycle à 5 jours sur 21 jours avec un congé de 10 jours

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R_1	Sp	C_A	C_A	C_A	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4
R_2	W_1	W_2	W_3	W_4	C_A	C_A	C_A	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1
R_3	W_2	W_3	W_4	W_1	W_2	W_3	W_4	C_A	C_A	C_A	W_1	W_2	W_3	W_4	W_1	W_2
R_4	W_3	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	C_A	C_A	C_A	W_1	W_2	W_3
R_5	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	W_1	W_2	W_3	W_4	C_A	C_A	C_A

Figure 4.49. Application d'un cycle à 4 semaines sur 16 semaines avec un congé de 3 semaines

4.2.2 Les contraintes supplémentaires

- Contraintes lorsque aucun agent n'est en congé

Avec une équipe dimensionnée pour une personne en congé, pendant les jours où personne n'est en congé, il y a une surcapacité par rapport aux besoins fixes. Le principe est de redistribuer le temps équitablement parmi tous les salariés.

- Les cycles journaliers de N jours peuvent être rallongés à N+1 jours. Ce jour supplémentaire peut prendre des valeurs spéciales et variables.

tel-00004380, version 1 - 29 Jan 2004

- Une ressource effectuant les cycles hebdomadaires doit être affectée au code « Volant » ou repos. Comme les intervalles sont tous d'une semaine, on n'a pas à créer de nouveaux cycles.

- Contrainte du travail « essentiel »

Avant le départ en congés, on demande aux agents effectuant un cycle journalier de faire les trois premiers codes qui constituent l'élément essentiel du cycle journalier car les autres codes sont du repos :

R_i	Vo	Vo	Vo	Vo	**S_1**	**S_2**	**S_3**	C_A	C_A	C_A	…	C_A	C_A	S_1	S_2	S_3	S_4	S_5

Figure 4.50. Travail essentiel dans un cycle de 5 jours

4.2.3 La relaxation de contraintes

Afin de produire des solutions même si les contraintes spécifiées sont en contradiction, nous devons les relaxer.

- Relaxation via le code « Volant »

Suivant le cycle utilisé, le code « volant » peut suivre et succéder tous les codes du cycle.

- Relaxation des contraintes de cycle journalier

Dans les contraintes journalières, afin de permettre aux ressources de prendre leurs congés après un repos hebdomadaire, il est admissible que le cycle soit déformé localement : le cycle peut être rallongé ou raccourci. Par exemple:

S_1-S_2-S_3-S_4-S_5. → S_1-S_2-S_3-S_5
S_1-S_2-S_3-S_4-S_5. → S_1-S_2-S_3-S_4-S_4-S_5.
S_1-S_2-S_3-S_4-S_5. → S_1-S_2-S_3-S_4-S_4-S_4-S_5.

Dans la Figure 4.48, il y a un problème le jour 22 lorsque la ressource R_2 serait normalement affectée au code S_1 au retour de congés. Cependant la ressource R_1 est aussi affectée à S_1 suivant le cycle. La solution consiste à relaxer la contrainte de cycle, donnant deux adaptations locales S_1-S_2-S_3-S_5 et S_1-S_2-S_3-S_4-S_4-S_5. Nous remarquons que la contrainte de charge est respectée. La contrainte de repos journalier doit être prise en compte dans la définition des possibilités de rallonge ou de raccourci.

- Relaxation des contraintes de congés

Dans les cycles hebdomadaires, les contraintes de congés provoquent des problèmes semblables lorsque les congés durent un nombre pair de semaines. Il n'est pas possible de relaxer la contrainte de cycle qui garantit un week-end de repos sur deux. La solution admise est d'avancer ou de reculer les congés 1 ou 2 semaines, exception faite pour les congés d'été qui sont définitivement fixes.

tel-00004380, version 1 - 29 Jan 2004

Le déroulement des cycles

4.3 LE MODELE ET SON IMPLANTATION

Nous présentons d'abord notre modèle permettant d'unifier les cycles journaliers et hebdomadaires. Dans la suite du document, on ne traite plus que des cycles hebdomadaires. Nous utiliserons le terme cycles hebdomadaires dans un sens plus général : l'horizon du cycle peut comprendre un nombre de jours > 2 et pas forcément un multiple de 7.

On présente une contrainte fondamentale : les contraintes de transition dénotée par X→Y. Cette contrainte spécifie que la valeur Y peut suivre la valeur X sur deux intervalles consécutifs pour la même ressource. On montre comment réécrire les différentes contraintes du problème en terme de ces contraintes.

Enfin, nous montrons comment implanter ces transitions utilisant la contrainte globale séquence. Dans les listings qui suivent, le code hebdomadaire W_i aura la valeur i+1, le code W_0 dénote les congés (valeur=0) et W_{VO} dénote "Volant" avec la valeur 1.

4.3.1 La construction d'un cycle hebdomadaire

Exemple : A partir d'un cycle à 5 jours (S_1, S_2, S_3, S_4 et S_5), construisons un cycle hebdomadaire à 4 jours. Nous utilisons les codes S_1, S_2, S_3, S_4 pour le code W_1. Si l'affectation du dernier jour de la rangée i est S_k. alors sur la rangée i+1, la première affectation doit être S_{k+1} (ou S_1, si k = N). La figure 4.51 montre le cycle résultant où le cycle journalier est déroulé 4 fois.

	1	2	3	4
W_1	S_1	S_2	S_3	S_4
W_2	S_5	S_1	S_2	S_3
W_3	S_4	S_5	S_1	S_2
W_4	S_3	S_4	S_5	S_1
W_5	S_2	S_3	S_4	S_5

Figure 4.51. Transformer un cycle à 5-jours en un cycle hebdomadaire de 4 jours. La contrainte de charge est toujours respectée chaque jour.

A partir d'un même cycle journalier, cette méthode peut aussi générer des cycles à 5 ressources à 3, 5 or 6 jours. Pour simplifier la description des contraintes, nous limitons nos exemples aux codes W_1 … W_5 de semaine de 4 jours dans cette note. L'application génère ces codes au vol et les relie ensemble avec les contraintes appropriées.

Théorème : Soit un cycle journalier de N jours et soit K un entier positif strictement inférieur à N. Construisons un cycle hebdomadaire sur K jours et pour N ressources de la manière suivante :

- Le code W_1 est constitué des K premiers codes du cycle
- Soit S_i la dernière vacation du code W_i. Le code W_{i+1} est obtenu avec les K codes $S_{((i+1) \bmod N)+1}$, …, $S_{((i+K) \bmod N)+1}$

Le cycle hebdomadaire ainsi obtenu est constitué de codes W_i qui respectent la contrainte de cycle : $W_i \rightarrow W_{i+1}$ (pour i=1, …, N-1) et $W_N \rightarrow W_1$.

tel-00004380, version 1 - 29 Jan 2004

Cette méthode d'agrégation exige la décomposition de l'horizon de planification N_{total} intervalles en un nombre d'intervalles de la bonne taille. Comme la construction de cycle ignore les congés pendant la semaine, tous les intervalles y compris les congés, doivent tomber en un nombre entier d'intervalles de la taille correspondante. Soit NB_P le nombre d'intervalles de longueur P, la condition est

$$N_{total} = NB_{N-1} * (N-1) + NB_{N+1} * (N+1) + NB_{N+2} * (N+2)$$

On rappelle que N est le nombre d'intervalles du cycle. La condition exprime que la somme des durées en cycle de longueur N-1, N+1 et N+2 couvre bien l'horizon N_{total}.

4.3.2 Implantation de la contrainte de cycle

Les contraintes de cycles hebdomadaires sont créées en vérifiant que les successions des vacations journalières du dernier jour d'un code hebdomadaire et le premier jour du code hebdomadaire suivant. Les codes journaliers au sein d'un code hebdomadaire doivent respecter les contraintes de succession. Dans la semaine de 4 jours, la contrainte de cycle est représentée par la suite de transitions $W_1 \rightarrow W_2$, $W_2 \rightarrow W_3$, $W_3 \rightarrow W_4$, $W_4 \rightarrow W_5$ et $W_5 \rightarrow W_1$. Les patterns sont :

```
Pats = [[sum,1,#=,[2]],[sum,1,#\=,[3]],      % W1 → W2
        [sum,1,#=,[3]],[sum,1,#\=,[4]],      % W2 → W3
        [sum,1,#=,[4]],[sum,1,#\=,[5]],      % W3 → W4
        [sum,1,#=,[5]],[sum,1,#\=,[6]],      % W4 → W5
        [sum,1,#=,[6]],[sum,1,#\=,[2]]]      % W5 → W1

Listing 1. Patterns pour les contraintes de cycle dans un cycle de période 5
```

Dans le cas des cycles journaliers, lorsque plusieurs cycles hebdomadaires de différents horizons sont utilisés, les contraintes de cycle doivent être créées pour chaque cycle.

4.3.3 Implantation des contraintes de charge

La contrainte globale `among` est utilisée directement pour réaliser la contrainte de charge pour tout intervalle j. Elle prend en compte toutes les demandes simultanément et agit sur les variables de toutes les ressources 1 à R :

```
among([N1,…, NN], [V1,j, …, V R ,j ], Zeros R, [[S1],…,[SN]], all)

Listing 2. Contrainte de charge
```

Rappel : N_i est le nombre demandé des valeurs S_i, l'une des valeurs possibles des variables V, `Zeros R` est une liste de R zéros demandés par la contrainte. La contrainte exige qu'il y ait exactement le nombre N_i de variables $V_{1,j}$, …, $V_{R,j}$. prend la valeur S_i, pour tout i simultanément.

4.3.4 Implantation des contraintes de repos journalier

Dans l' exemple de la figure 4.51, S_3 (nuit) doit être suivi par S_4 (repos); donc il faut ajouter la transition $W_2 \rightarrow W_3$ parce que W_2 finit en S_3 et seul W_3 démarre avec S_4 parmi les codes semaines. Cette transition doit être ajoutée aux cas où des cycles supplémentaires sont crées. Voir Listing 4.

tel-00004380, version 1 - 29 Jan 2004

4.3.5 Implantation des contraintes de congés annuels

Dans la semaine de 4 jours, si S_5 est le jour de repos hebdomadaire alors la transition journalière $S_5 \rightarrow 0$ est réécrit en transition hebdomadaire $W_5 \rightarrow W_0$ puisque S_5 est placé à la fin de la semaine W_5. Si $S_4 \rightarrow 0$ est permise, alors la transition $W_1 \rightarrow W_0$ est aussi permise. Le premier jour après les congés ne peut être un repos. Cela veut dire que la transition $W_0 \rightarrow W_5$ n'est pas permise. Il suffit d'exclure cette transition car seules les transitions permises sont énumérées. Après les congés, certaines vacations journalières sont hautement préférables, notamment S_1. Par conséquent, on admet la transition $W_0 \rightarrow W_1$ comme W_1 débute avec S_1 dans l'exemple.

```
Pats = [
  [sum,1,#=,[0]],[sum,1,#\=,[0,2]],     % W0 → W0 and W1
  [sum,1,#=,[2]],[sum,1,#\=,[3]],       % W1 → W2
  [sum,1,#=,[3]],[sum,1,#\=,[4]],       % W2 → W3
  [sum,1,#=,[4]],[sum,1,#\=,[5]],       % W3 → W4
  [sum,1,#=,[5]],[sum,1,#\=,[6]],       % W4 → W5
  [sum,1,#=,[6]],[sum,1,#\=,[0,2]]      % W5 → W0 and W1
]
```

Listing 3. Patterns pour implanter les contraintes de cycle et de conges annuels

4.3.6 Implantation des contraintes supplémentaires

- Contraintes lorsqu'aucun agent n'est en congé

Dans le cas des cycles journaliers, un cycle de N jours est transformé en cycle de (N+1) jours. Le code supplémentaire est dénoté N+1, ex. S_6 dans notre exemple. On peut calculer les variants de ce cycle étendu sur des horizons de différentes tailles, à l'instar d'un cycle de N jours.

	1	2	3	4	5
W_6	S_1	S_2	S_3	S_4	S_5
W_7	S_6	S_1	S_2	S_3	S_4
W_8	S_5	S_6	S_1	S_2	S_3
W_9	S_4	S_5	S_6	S_1	S_2
W_{10}	S_3	S_4	S_5	S_6	S_1
W_{11}	S_2	S_3	S_4	S_5	S_6

Figure 4.52. Rallonger un cycle de 5 jours en un cycle de 6 jours (W_6- W_{11})

Ici, les contraintes de cycle sont $W_6 \rightarrow W_7$, … , $W_{11} \rightarrow W_6$. Afin de permettre l'utilisation d'un cycle de N jours suivis par un cycle de N+1 jours par des variables successives, on doit vérifier la contrainte de repos journaliers pour le dernier jour du premier cycle et le premier jour du second cycle. Par exemple si la transition $S_4 \rightarrow S_5$ est admise, il faut ajouter les transitions suivantes: $W_1 \rightarrow W_8$ (car W_1 finit par S_4 et W_8 débute par S_5), qu'on complète avec $W_2 \rightarrow W_9$, $W_3 \rightarrow W_{10}$, $W_4 \rightarrow W_{11}$, $W_5 \rightarrow W_7$, et $W_0 \rightarrow W_6$.

tel-00004380, version 1 - 29 Jan 2004

Inversement, pour permettre le cycle de N+1 jours à suivre le cycle de N jours, il faut ajouter $W_6 \rightarrow W_1$, $W_7 \rightarrow W_2$, $W_8 \rightarrow W_3$, $W_9 \rightarrow W_4$, $W_{10} \rightarrow W_5$, et $W_{11} \rightarrow W_0$. Les patterns sont résumés dans le Listing 4. Cette méthode peut aussi permettre l'utilisation de différents cycles suivant les périodes différentes, par exemple été et hiver.

```
Pats = [       %N-day cycle
  [sum,1,#=,[2]], [sum,1,#\=,[1,3,9]],  %W1 → WSP,W2,W8
  [sum,1,#=,[3]], [sum,1,#\=,[1,4,10]], %W2 → WSP,W3,W9
  [sum,1,#=,[4]], [sum,1,#\=,[1,5,11]], %W3 → WSP,W4,W10
  [sum,1,#=,[5]], [sum,1,#\=,[1,6,12]],       %W4 → WSP,W5,W11
  [sum,1,#=,[6]], [sum,1,#\=,[0,1,2,8]]       %W5 → W0,WSP,W1,W7
 %(N+1)-day cycle
  [sum,1,#=,[7]], [sum,1,#\=,[1,2,8]],  %W6 → WSP,W1,W7
  [sum,1,#=,[8]], [sum,1,#\=,[1,3,9]],  %W7 → WSP,W2,W8
  [sum,1,#=,[9]], [sum,1,#\=,[1,4,10]], %W8 → WSP,W3,W9
  [sum,1,#=,[10]],[sum,1,#\=,[1,5,11]], %W9 → WSP,W4,W10
  [sum,1,#=,[11]],[sum,1,#\=,[1,6,12]], %W10 → WSP,W5,W11
  [sum,1,#=,[12]],[sum,1,#\=,[0,1,7]]   %W11 → W0,WSP,W1
 ]
 Listing 4. Patterns pour le cycle de (N+1) jours
```

- Contrainte du travail essentiel

Dans les cycles journaliers avec volant avant les congés, afin d'intégrer les vacations essentielles (S_1-S_2-S_3) avant les congés, nous avons créé un cycle spécifique de 3 jours, décrit ci-après, et affecté avant le départ en congés.

	1	2	3
W_{12}	S_1	S_2	S_3
W_{13}	S_4	S_5	S_1
W_{14}	S_2	S_3	S_4
W_{15}	S_5	S_1	S_2
W_{16}	S_3	S_4	S_5

Figure 4.53. Cycle hebdomadaire sur 3 jours, déduit du cycle journalier de 5 jours

Ces nouvelles vacations (W_{12} etc.) doivent être connectées aux autres cycles tout en respectant le repos journalier : $W_{12} \rightarrow W_0$, $W_1 \rightarrow W_{13}$ (car $S_4 \rightarrow S_5$), $W_2 \rightarrow W_{14}$, etc.

4.3.7 La relaxation des contraintes

- Relaxation via le code « Volant »

Il suffit d'ajouter la valeur du code volant dans le domaine des variables et introduire les patterns suivants :

```
Pats =                                  %WEEKLY CYCLES
     [[sum,1,#=,[0]], [sum, 1, #\=,[0,1,6]],
      [sum,1,#=,[1]], [sum, 1, #\=,[0,1,2,…,6]],% CodeVolant suit tous les codes
        ...  ]
Pats = [                                %DAILY CYCLES
 [sum,1,#=,[0]], [sum, 1, #\=,[0,1,7]], %W0 →W0,WV0,W6
 [sum,1,#=,[1]],[sum,1,#\=,[0,1,2, …, 12]], % Code volant suit tous les codes

Listing 5. Relaxation de contraintes avec le code « Volant »
```


tel-00004380, version 1 - 29 Jan 2004

Le déroulement des cycles

- Relaxation des contraintes de cycle journalier

Afin de déformer le cycle journalier, il suffit d'autoriser des contraintes de transitions supplémentaires, permettant de rallonger et de raccourcir le cycle. Considérons le cas de la transition autorisée $S_3 \rightarrow S_5$. Au niveau de la semaine, en plus de la transition habituelle $W_2 \rightarrow W_3$, nous autorisons les transitions $W_2 \rightarrow W_2$ et $W_2 \rightarrow W_8$. Le système cherchera sur toutes les transitions autorisées pour trouver l'ensemble des valeurs satisfaisant ces contraintes.

Le pattern correspondant dans `Pats` est `[[sum,1,#=,[3]],[sum,1,#\=,[1,3,4,9]]]`. De la même façon, pour rallonger le cycle il faut autoriser la transition $S_4 \rightarrow S_4$, ce qui nous permet de créer les relaxations au niveau hebdomadaire.

Ainsi la déformation du cycle peut intervenir entre deux intervalles. Plus il y a d'intervalles, plus le système s'assouplit. Si l'horizon de planification est découpé en intervalles d'un jour, la souplesse est maximale, mais les résultats peuvent devenir très aléatoires et les utilisateurs ne reconnaîtront plus le cycle.

- Relaxation des contraintes de congés

Dans les cycles hebdomadaires, lorsqu'une ressource i est définitivement affectée aux congés la semaine j, alors la variable $V_{i,j}$ est affectée à la valeur 0. Quand un congé peut être affecté à la semaine j, et potentiellement avancé ou retardé d'un intervalle, alors le domaine des trois variables candidates j-1, j, j+1 contiendront la valeur 0, en plus des autres valeurs possibles. La contrainte suivante est nécessaire afin d'affecter le bon nombre de semaines de congés; elle spécifie que le nombre de variables avec la valeur 0 parmi les variables candidates est exactement N_o:

```
among([No], [Vi,j-1, V(i, j), Vi,j+1], Zeros3, [[0]], all)
```

Listing 6. Contrainte sur le nombre total des congés

Quand les congés durent les trois semaines j, j+1 et j+2, et lorsqu'ils peuvent être avancés ou retardés d'un intervalle, l'intervalle j est définitivement en congés. Malheureusement dans le cas des cycles journaliers, comme la période avant congé pourrait avoir une taille différente des périodes après les congés, cette méthode ne peut pas être appliquée.

Remarque : cette technique distinguant la partie dure et la partie « floue » est très utilisée dans la PPC.

tel-00004380, version 1 - 29 Jan 2004

4.4 LA RECHERCHE DE SOLUTIONS

4.4.1 La génération de solutions

Nous utilisons une méthode très simple de labelling d'un intervalle j à la fois et de la gauche vers la droite. Pour chaque intervalle j, les variables correspondant à toutes les ressources sont énumérées afin de satisfaire les contraintes de charge et la contrainte de cycle. Les variables de l'intervalle sont sélectionnées par l'heuristique suivante :

- Choix de la variable avec le domaine le plus restreint
- Choix des valeurs dans l'ordre croissant

La stratégie de la gauche vers la droite facilite le traitement de nombreuses contraintes de transition générées à la volée. Il y a retour en arrière sur des intervalles différents pour satisfaire la contrainte de cycle.

4.4.2 Les conditions nécessaires de congés annuels

Une première condition nécessaire que doive satisfaire la constitution d'une équipe hebdomadaire est que chaque ressource puisse prendre tous ses congés (soit 7 semaines en tout). Dans une équipe de N+1 ressources, compte tenu des 52 semaines par an, amputées de 3 semaines pour l'organisation manuelle pour les fêtes de fin d'année, il faut que

$(N+1) * 7 \leq 49$, soit $N \leq 6$

Dans des équipes où au plus deux ressources peuvent être en congés simultanément, la condition est $N \leq 12$.

Dans le traitement des *cycles hebdomadaires*, certains jeux de données n'admettent pas de planning. J'ai mis au point un modèle hebdomadaire pour analyser ce phénomène. Ce modèle ne contient que deux codes W_1 = I pour les semaines impaires et W_2 = P pour les semaines paires. La condition nécessaire mais insuffisante pour que les besoins soient respectés dans ce modèle est qu'il y ait chaque semaine, un même nombre de W_1 et W_2.

R_1	V_0	C_A	C_A	C_A	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1
R_2	W_1	W_2	W_1	W_2	C_A	C_A	C_A	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2
R_3	W_2	W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A	C_A	W_1	W_2	W_1	W_2	W_1	W_2	W_1
R_4	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A	C_A	W_1	W_2	W_1	W_2
R_5	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	C_A	C_A	C_A	V_0

Figure 4.54. Modèle Pair/Impair de N=4 pour 3 semaines de congés annuels

Figure 4.54 est un tel planning pour N=4, pour un nombre **impair** de semaines de conges annuels. On suppose que les congés sont pris selon un ordre chronologique (sinon, il suffit de changer l'ordre des lignes). La contrainte des congés annuels (week-end en repos avant départ en congés) est interprétée par une semaine paire avant la semaine CA. Lorsqu'il n'y a pas de CA, une équipe est désignée Volante à tour de rôle.

Dans le cas d'un nombre pair de congés annuels, comme le montre la Figure 4.55, lorsque les ressources sont en congés à tour de rôle de façon continue, les besoins par

tel-00004380, version 1 - 29 Jan 2004

semaine ne peuvent pas être respectés, comme il n'y a que des semaines paires ou impaires.

La solution consiste à insérer des semaines sans congés (avec une ressource en volant) comme le montre la Figure 4.56. On obtient ainsi un nombre égal de semaines paires et impaires. Bien entendu, cela exige qu'on puisse insérer deux semaines sans congés. Pour l'équipe N=4, avec cinq ressources et 35 semaines de congés, une organisation en 2+3+2 semaines exige l'insertion de 4 semaines sans congés. Or pour une équipe N=6, il y a sept ressources qui mobilisent 49 semaines de congés par an. En conséquence, on ne peut qu'offrir des congés en nombre impair de semaines, par exemple 1+3+3 semaines.

R_1	C_A	C_A	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1
R_2	W_1	W_2	C_A	C_A	W_1	W_2	W_1	W_2	W_1	W_2	W_1
R_3	W_1	W_2	W_1	W_2	C_A	C_A	W_1	W_2	W_1	W_2	W_1
R_4	W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A	W_1	W_2	W_1
R_5	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A	W_1

W_2	C_A	C_A	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2
W_2	W_1	W_2	C_A	C_A	Vo	W_1	W_2	W_1	W_2	W_1	W_2	W_1
W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A	W_1	W_2	W_1	W_2	W_1
W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A	Vo	W_1	W_2
Vo	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	W_1	W_2	C_A	C_A

Modèle Pair/impair pour 2 semaines de congés annuels

Figure 4.55. Nombre insuffisant de codes pairs et impairs par semaine

Figure 4.56. Planning avec des semaines sans congés, donnant 2 W_1 et 2 W_2 par semaine

Or une fois résolue, toute solution du modèle pair/impair représente N/2 solutions (N étant pair) dans le modèle de base. Effectivement, chaque code W_1 peut représenter un départ de cycle. On pourra ainsi choisir la meilleure solution.

4.4.3 Le traitement des ruptures

Comme toutes applications pratiques, le système doit fournir des solutions même si les contraintes sont contradictoires, notamment quand les dates des congés sont en conflit avec le déroulement mathématique des cycles. Dans de telles situations, la contrainte de charge n'est plus postée localement et les ressources sont affectées à la valeur "volante" via la relaxation de contraintes.

Ces semaines n'arrivent qu'isolement car il n'y ait pas de contraintes de cycle sur la valeur volante... Les ressources sont affectées de façon heuristique en deux étapes. D'abord au niveau de la semaine, un système spécial basé sur des règles est utilisé pour spécifier les jours où chaque ressource doit travailler, en considérant les affectations des week-ends. Ensuite, au niveau journalier, l'application applique les codes travaillés du cycle afin de respecter les contraintes de charge.

tel-00004380, version 1 - 29 Jan 2004

4.5 NOS RESULTATS ET CONCLUSIONS

4.5.1 Résultats théoriques et une justification de la PPC

Nous avons proposé un modèle des cycles permettant d'unifier les cycles journaliers et les cycles hebdomadaires. Il agrége les données sur l'axe du temps : l'intervalle de temps peut ainsi représenter plusieurs jours voire des semaines. A l'intérieur d'un intervalle, le cycle est déroulé de façon parfaite sans relaxation. Cette technique facilitera le traitement d'horizon d'une année, voire plus. Il serait capable de traiter des horizons de 3 ans, et de prendre en compte des contraintes nouvelles telles que la suivie de carrière, la formation continue, etc.

Nous avons mis en évidence la contrainte transition qui permet de modéliser les contraintes du problème du déroulement des cycles hebdomadaires, ainsi que la relaxation. Ces mêmes contraintes peuvent être utilisées pour intégrer des cycles différents applicables aux différentes périodes de l'année.

Nous avons montré comment mettre en œuvre les contraintes globales dynamiquement, à partir des données du problème, y compris la relaxation. Dans le déroulement des cycles journaliers, la PPC a été nécessaire parce qu'il faut *déformer* progressivement le cycle afin de permettre aux agents de partir en congés suite à un repos hebdomadaire.

De même, le déroulement des cycles hebdomadaires avait besoin de la PPC parce qu'on ne sait pas a priori comment déplacer les congés annuels autour du cycle.

4.5.2 Une application complète et indépendante

Les concepts de la planification cyclique basée sur des cycles de travail sont implantés dans l'application MOSAR. Divers aspects systèmes ont été implantés, par ex. la saisie des paramètres de résolution, l'édition des cycles et l'affichage du planning résultant ainsi que les statistiques pour chaque ressource. Le système de résolution est bâti avec les contraintes globales **among** et **sequence** du système CHIP V5. Ce dernier s'avère très utile pour l'implantation des méthodes de relaxation des contraintes. L'application a été livrée avec 14 cycles journaliers et hebdomadaires qui peuvent être déroulés suivant les paramètres standards. Elle est exploitée sur plusieurs sites en France depuis début 1999 afin de produire des plannings annuels pour jusqu'à 150 personnes à la fois. Le solver a été capable de produire des plannings pour des cycles complexes où l'énumération est limitée aux cinq premiers jours, le reste du problème étant résolu par propagation de contraintes sur 20 secondes sur un Pentium I @ 200Mhz.

Notre expérience montre que le déroulement de cycles de travail est un problème complexe. Tout en respectant les contraintes de charge, les cycles doivent être adaptés autour des absences prévues de type congés annuels ou de formation et donnant des repos hebdomadaires avant chaque départ en congés. La relaxation des contraintes est nécessaire pour obtenir des plannings satisfaisants.

L'un des avantages majeurs d'un outil de déroulement cyclique dans l'organisation du travail est la souplesse obtenue par la combinaison de plusieurs cycles adaptés à

tel-00004380, version 1 - 29 Jan 2004

chaque type de besoin en charge. Les cycles ont été conçus pour donner un bon équilibre entre le travail et le repos. Il faut en disposer d'un nombre suffisant afin de répondre à nombreuses situations telles que les salariés administratifs, les gardiens opérationnels ou les officiers. Notre expérience montre que les cycles hebdomadaires peuvent être déroulés parfaitement (la seule relaxation étant les dates de congés) alors que les cycles hebdomadaires exigent la relaxation du cycle, rompant ainsi l'équilibre du cycle.

Le travail cyclique permet de partager toutes les activités de l'établissement sur l'ensemble du personnel et encourage une certaine paix sociale. Les cycles de travail sont un progrès social dans l'organisation du travail en continu qui encourage l'esprit d'équipe, en partageant toutes les activités, même celles qui sont les moins appréciées telles que les vacations de nuit, le week-end ou les jours fériés. Les cycles permettent aux salariés de mieux planifier leur vie sociale.

4.5.3 Conclusion

Ce travail m'a permis de comprendre la problématique de la planification cyclique, déchirée entre le souhait d'un rythme de travail équilibré et le souhait de respecter des impératifs humains. Il a fallu analyser, trancher, mettre en évidence des mécanismes de relaxation, et réaliser les programmes permettant de les mettre en œuvre.

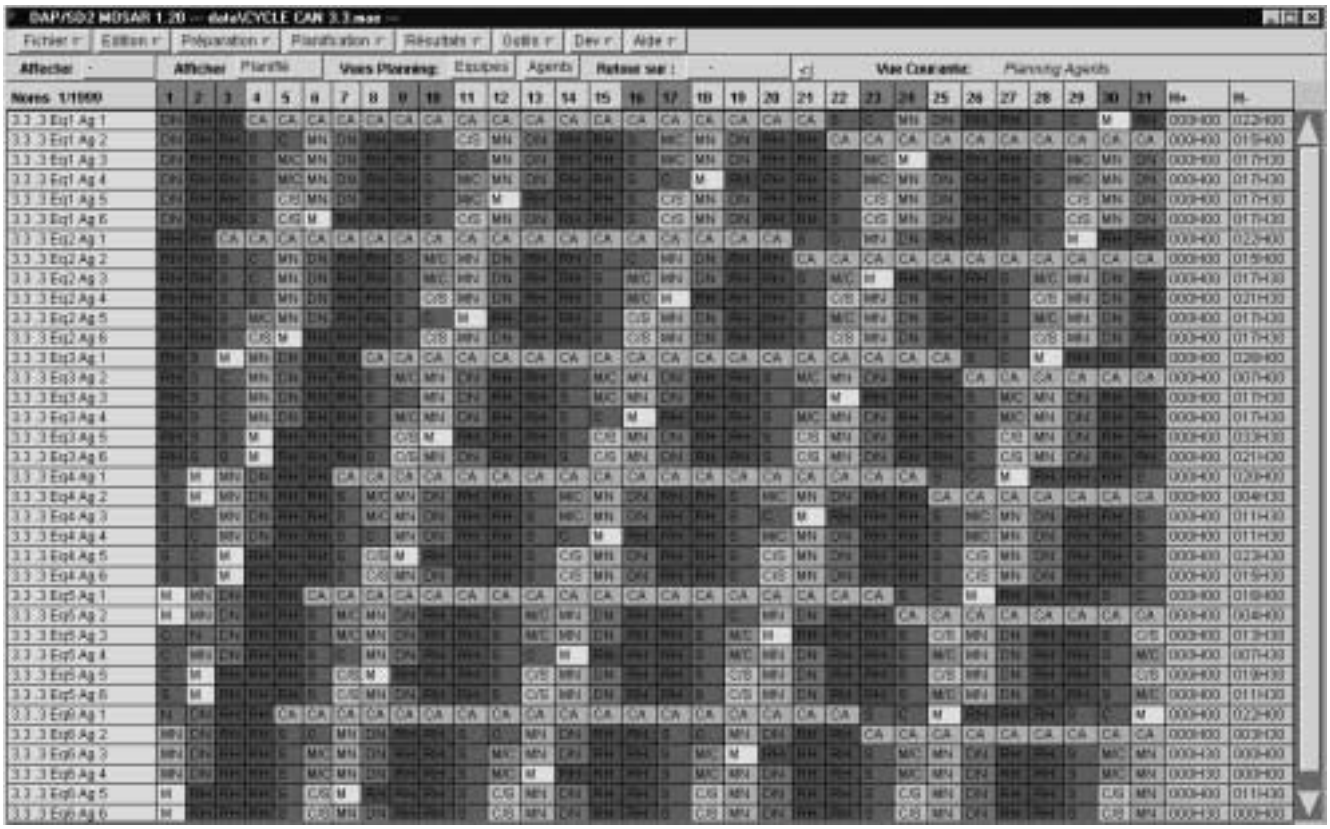
Plus généralement, j'ai pu constater que les mêmes contraintes peuvent se représenter sur différents niveaux d'agrégation (par ex. la contrainte de repos journalier dans un cycle hebdomadaire). De ce travail est née l'idée que les différents niveaux d'agrégation peuvent coexister et même cohabiter au sein d'un même système de planification. Le modèle à multiples niveaux sera présenté au chapitre suivant.

tel-00004380, version 1 - 29 Jan 2004

CYCLE CAN3.3 – Organisation des congés individuellement

Le Cycle CAN3.3 est un cycle journalier de 6 jours. Il est défini par la suite :
MN[19]+DN+RH+RH+S+(X)

X est une journée variable ayant pour moitié C/S et l'autre moitié M/C. Si le nombre de gardes est impair, ce dernier est affecté à C.



[19] MN = Matin entre 7 H et 15H, et Nuit entre 23H et 7H
DN = Descente de Nuit (repos compensatoire après le travail de nuit)
M = Matin
RH = Repos Hebdomadaire
S = Soir (entre 15H et 23H)
C = Coupure (Journée)
M/C et C/S sont des vacations Matin et Soir incluant Coupure (l'heure de repas du midi)

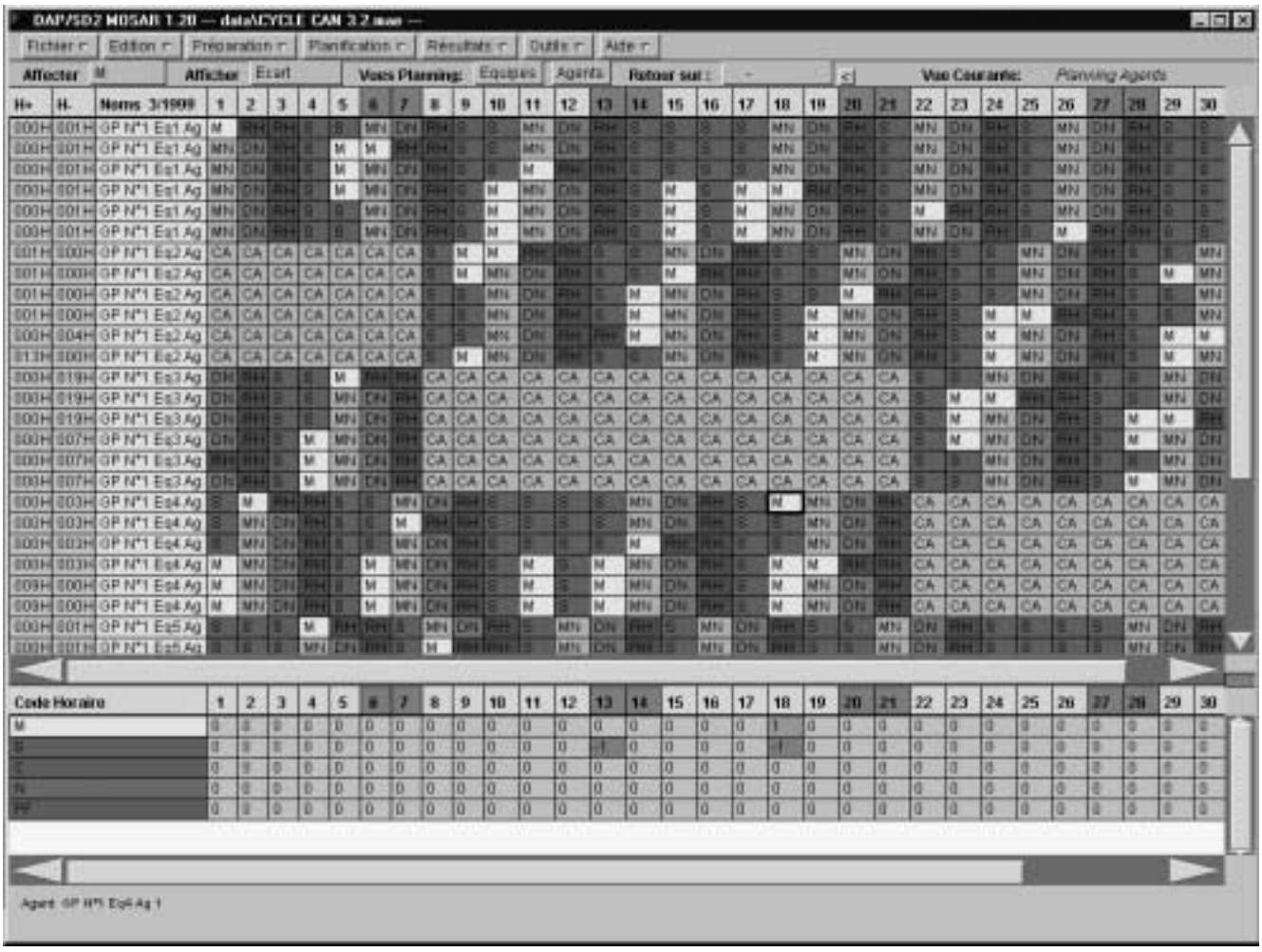
tel-00004380, version 1 - 29 Jan 2004

Le déroulement des cycles

CYCLE CAN3.2 – Organisation des congés par équipe

Le Cycle CAN3.2 est un cycle journalier de 6 jours. Il est défini par la suite :
MN+DN+RH+S+(X)

Pour l'exécution du code horaire X, une moitié de l'équipe effectue le Matin (M) et l'autre moitié le Soir (S). Ces deux codes ont le même nombre d'heures travaillées.

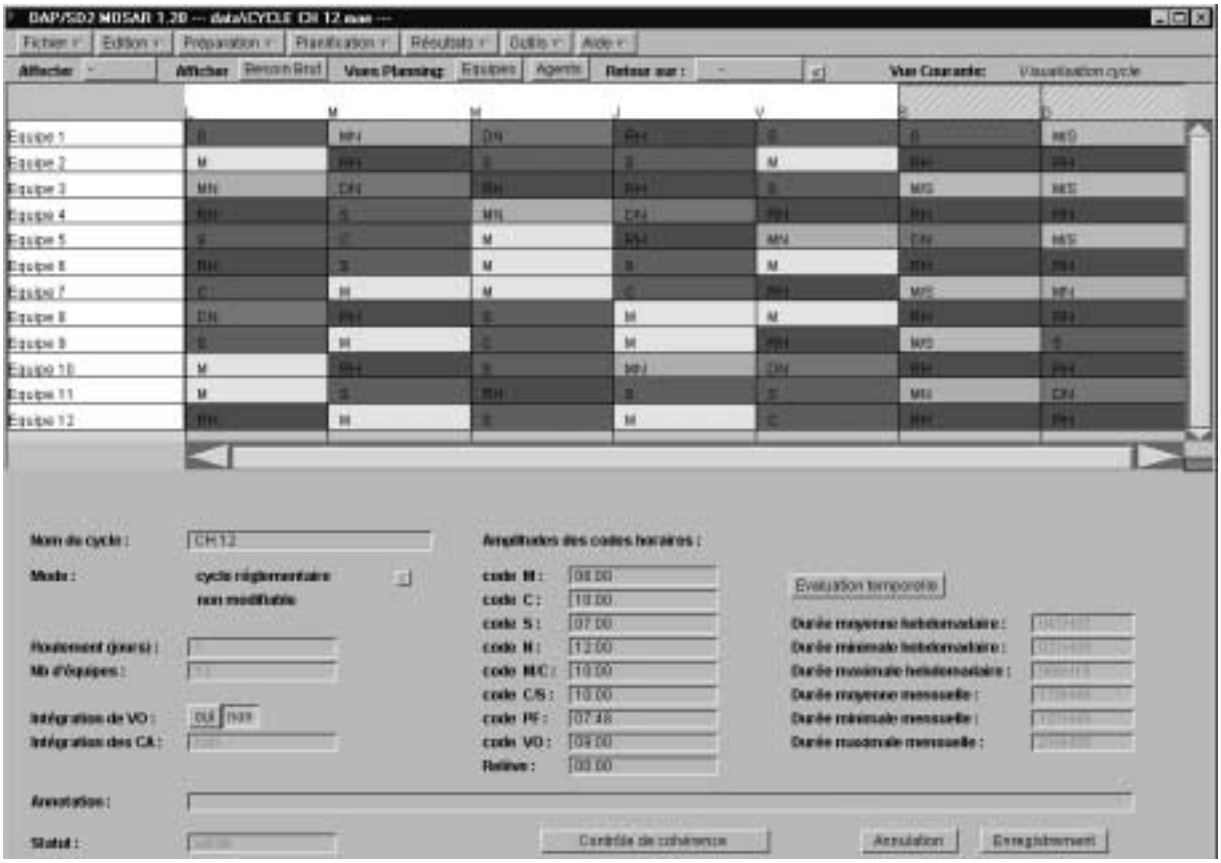


Afin de partir en congés annuels CA après le RH, on attribuerait des suites S+S, S+S+S et S+S+S+S. Les besoins journaliers sont près toujours respectés.

tel-00004380, version 1 - 29 Jan 2004

Cycle CH12

Définition du cycle sur 12 semaines :



Appliqué à 14 ressources, il y aura au plus deux en congés à un moment donné. Il n'y a 2 semaines volantes (ou marge) sur 50 semaines, comme le montre la solution trame ci-dessous.

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

Le déroulement des cycles

Déroulement du CH12 sur 14 ressources

CH12 : Solution Trame pour 50 semaines

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Eq01	0	0	5	6	7	8	9	10	11	12	1	2	3	4	0	0	0	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	1	2	3	4	5	6	7	8	9	10	11	12	1
Eq02	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	0	0	0	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	1	2	3	4	5	6	7	8	9	10	11
Eq03	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	1	2	3	4	5	6	7	8	9
Eq04	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	1	2	3	4	5	6	7
Eq05	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	1	2	3	4	5
Eq06	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	1	2	3
Eq07	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	0	0	**Vo**
Eq08	**Vo**	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	0	0	0	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6	7	8	9	10
Eq09	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	0	0	0	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6	7	8
Eq10	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6
Eq11	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4
Eq12	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2
Eq13	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12
Eq14	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6	7	8	9	10	0	0

Solution trame pour 52 semaines

Semaines	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
Equipe 1	0	0	5	6	7	8	9	10	11	12	1	2	3	4	vo	0	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	vo	0	0	5	6	7	8	9	10	11	12	1	2	3	4	vo
Equipe 2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3
Equipe 3	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1
Equipe 4	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11
Equipe 5	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9
Equipe 6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7
Equipe 7	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	5
Equipe 8	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6	7	8	9	10
Equipe 9	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	0	0	0	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6	7	8
Equipe 10	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4	5	6
Equipe 11	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	0	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2	3	4
Equipe 12	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12	1	2
Equipe 13	6	7	8	9	10	11	12	1	2	3	4	0	0	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	0	0	0	11	12	1	2	3	4	5	6	7	8	9	10	0	0	11	12
Equipe 14	4	5	6	7	8	9	10	11	12	1	2	3	4	0	0	vo	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	0	0	0	vo	11	12	1	2	3	4	5	6	7	8	9	10	0	0

En gris : les codes hebdomadaires de congés. Les chiffres 1 à 12 représentent des codes hebdomadaires. Vo est le code volant

CH12

DAP/SD2 MOSAR 1.20 --- data\CYCLE CH 12.mae ---

Fichier ▽ | Edition ▽ | Préparation ▽ | Planification ▽ | Résultats ▽ | Outils ▽ | Aide ▽

Affecter S | Afficher Ecart | Vues Planning: Equipes | Agents | Retour sur : - | Vue Courante: *Planning Agent*

H+	H-	Noms 2/1999	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
000H	014H	TEST18 Eq1 Ag	M	S	RH	S	S	MN	DN	RH	M	S	M	C	RH	RH	S	MN	DN	RH	S	S	M/S	M	RH	S	S	M	RH	RH
000H	006H	TEST18 Eq10 A	M	RH	S	MN	DN	RH	RH	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	S	S	RH	S	S	MN	DN
000H	023H	TEST18 Eq11 A	DN	RH	S	M	M	RH	RH	S	M	C	M	RH	M/S	S	M	RH	S	MN	DN	RH	RH	CA	CA	CA	CA	CA	CA	CA
000H	001H	TEST18 Eq12 A	C	M	M	C	RH	M/S	MN	DN	RH	S	M	M	RH	RH	S	M	C	M	RH	M/S	S	M	RH	S	MN	DN	RH	RH
000H	000H	TEST18 Eq13 A	RH	S	M	S	M	RH	RH	C	M	M	C	RH	M/S	MN	DN	RH	C	C	C	RH	RH	C	C	RH	RH	C	C	RH
000H	001H	TEST18 Eq14 A	S	C	M	RH	MN	DN	M/S	RH	S	M	S	M	RH	RH	C	M	M	C	RH	M/S	MN	DN	RH	S	M	M	RH	RH
000H	010H	TEST18 Eq15 A	S	M	C	M	RH	M/S	S	M	RH	S	MN	DN	RH	RH	M	S	RH	S	S	MN	DN	RH	M	S	M	C	RH	RH
000H	002H	TEST18 Eq2 Ag	C	RH	C	C	M	RH	RH	M	S	RH	S	S	MN	DN	RH	M	S	M	C	RH	RH	S	MN	DN	RH	S	S	M/S
000H	008H	TEST18 Eq3 Ag	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	S	RH	S	M	M	RH	RH	S	M	C	M	RH	M/S	S
000H	024H	TEST18 Eq4 Ag	RH	S	MN	DN	RH	RH	RH	C	C	RH	RH	C	RH	RH	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA	CA
000H	010H	TEST18 Eq5 Ag	MN	DN	RH	RH	S	M/S	M/S	RH	S	MN	DN	RH	RH	RH	S	C	M	RH	MN	DN	M/S	RH	S	M	S	M	RH	RH
000H	014H	TEST18 Eq6 Ag	S	MN	DN	RH	S	S	M/S	M	RH	S	S	M	RH	RH	MN	DN	RH	RH	S	M/S	M/S	RH	S	MN	DN	RH	RH	RH
000H	010H	TEST18 Eq7 Ag	RH	M	S	M	C	RH	RH	S	MN	DN	RH	S	S	M/S	M	RH	S	S	M	RH	RH	MN	DN	RH	RH	S	M/S	M/S
000H	010H	TEST18 Eq8 Ag	M	RH	S	S	M	RH	RH	MN	DN	RH	RH	S	M/S	M/S	RH	S	MN	DN	RH	RH	RH	S	C	M	RH	MN	DN	M/S
019H	000H	TEST18 Eq9 Ag	CA	CA	CA	CA	CA	CA	CA	S	C	M	RH	MN	DN	M/S	RH	S	M	S	M	RH	RH	C	M	M	C	RH	M/S	MN

Ce cycle peut aussi se dérouler sur 15 ressources, donc avec au plus trois ressources en congés simultanément. Cela offre plus de marge ou de semaines volantes.

tel-00004380, version 1 - 29 Jan 2004

5 LES MODELES A MULTIPLES NIVEAUX D'AGREGATION

Résumé :
La législation du temps de travail appliquée en France depuis janvier 2002 dans toutes les entreprises impose des durées maximales de travail et durées minimales de repos au niveau de la journée, la semaine, le mois et l'année. Cela rend très complexe la planification des horaires des salariés.

Après une initiation à la législation actuelle et la présentation des différents modèles de base, ce chapitre présente le modèle à multiples niveaux d'agrégation dénotés **MMN**, dans le cas des qualifications multiples. Disposant des variables et les contraintes à chaque niveau, le système permet de faire des déductions inter-niveaux pour obtenir les plannings légaux à tous les niveaux.

Nous donnons les conditions nécessaires pour guider la recherche de solutions ainsi que les règles de recherche locale pour résoudre des conflits de ressources.

5.1 LA LEGISLATION EN MATIERE DE DUREES DE TRAVAIL ET REPOS

Tableau 5.57. Variables et contraintes aux différents niveaux

	Modèle Journalier	Modèle Mensuel	Modèle Annuel
Variables	X(employé, intervalle)= qualification ou repos, affecté à l'employé à un intervalle	V(employé, jour) = vacation affectée à l'employé par jour, ex. matin, nuit, repos	Y(employé, qualification, semaine) = nombre d'heures travaillées par l'employé pendant la semaine sur cette qualification
Période	10 – 15 minutes	1 jour	1 semaine
Horizon	1 à 2 jours	28 à 42 jours	14 à 16 mois
Contraintes sur la durée du travail			
	4 – 10 heures par jour 12 heures pour les nuits	20 – 39 Heures par semaine, 8 – 20 jours de travail par mois, 35H par cycle de N semaines	Max. 1600 H / an ou en moyenne 35 H / sem. sur 1 an. Max. 38 H / sem. an moyenne sur 12 semaines
Contraintes sur la durée du repos			
Pause longue	Repas (45 – 60 minutes), après 3 – 6 H de travail	2 jours consécutifs de repos, après 3 à 6 jours de travail	Congés annuels (2 ou 3 semaines)
Pause courte	Pause (10 – 15 minutes), après max. 2 heures	1 jour de repos, après plus de 2 jours de travail	Semaine de repos

Définition: Une étiquette est un ensemble d'heures de travail, définie lors de l'organisation du travail de l'établissement. Les étiquettes sont disjointes. Par exemple :

- Matin : 8H à 16H
- Soir :16H à minuit
- Nuit : minuit à 8H

L'étiquette est la variable au niveau mensuel, décrite au paragraphe suivant. Elle tient compte du repos minimal entre deux vacations. Elle peut être un jour de repos, ce qui veut dire que le solveur journalier pour ce jour n'est pas nécessaire.

tel-00004380, version 1 - 29 Jan 2004

5.2 LES MODELES DE BASE

Nous préciserons dans ce paragraphe, les différents modèles de base qui vont intervenir dans les modèles multi-niveaux.

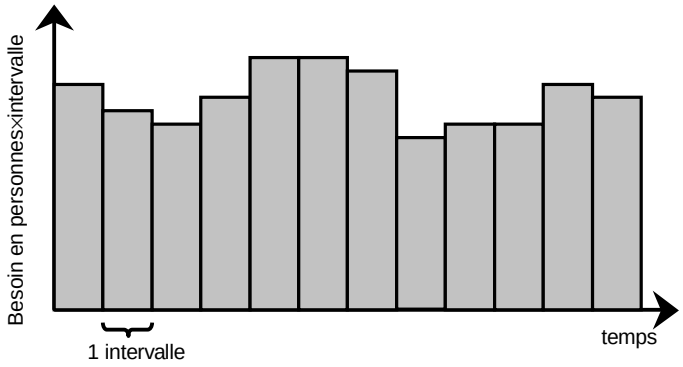


Figure 5.58. Généralités sur les modèles

Les différents modèles sont définis en fonction des intervalles de taille différente. Afin de pas confondre les indices de variables dans le MMN, nous utiliserons le terme *intervalle* pour l'horizon journalier, le terme *jour* pour l'horizon mensuel et le terme *semaine* pour l'horizon annuel.

Il ne s'agit pas pour le législateur de limiter les activités des entreprises avec la loi sur les 35h, mais de définir un cadre légal qui permet de moduler le travail des salariés en fonction des activités fluctuantes des entreprises.

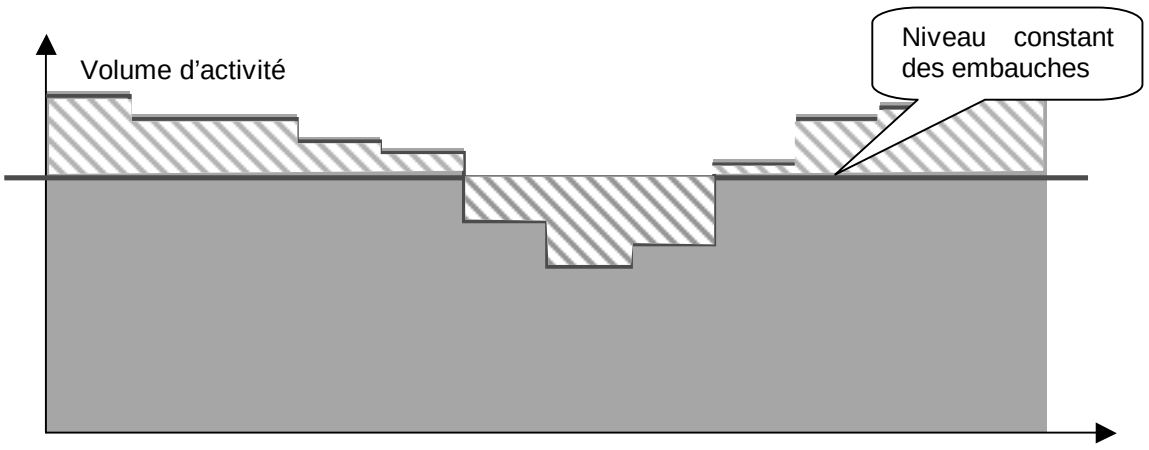


Figure 5.59. Annualisation

En parallèle avec la mise en place des 35 heures moyennes hebdomadaires, la législation française autorise la modulation du temps de travail. Les accords de réduction du temps de travail ne se résument pas seulement à la restriction à 35 heures de travail par semaine, ce chiffre étant une moyenne hebdomadaire sur l'année. Ainsi, afin d'absorber des fluctuations de charges en fonction de la saison, l'employeur va pouvoir définir des horaires variables par semaine et les adapter aux fluctuations de l'activité sur l'année. Cette prévision permet d'avoir un niveau constant de personnel et d'éviter des

tel-00004380, version 1 - 29 Jan 2004

embauches précaires (contrats à durées déterminées ou intérimaires) qui coûtent cher à l'entreprise en termes de formation et coût horaire.

En absence de planning prévisionnel à la fin de l'année, d'une part des individus ayant des compétences rares peuvent atteindre facilement leur limite de temps travaillé annuel, ce qui exige le paiement de bonus. D'autre part, des individus n'ayant pas assez travaillé seront payés en totalité, d'où une perte sèche pour l'entreprise.

5.2.1 Le modèle journalier

La variable de décision à ce niveau est l'affectation d'un employé pendant l'intervalle i à travailler sur la compétence s ou au repos. La taille de l'intervalle est typiquement le quart d'heure. La contrainte de charge journalière W_J par qualification et par intervalle est respectée lorsque :

$$X(e, i) \subseteq \textbf{Qualifications} \cup \{\textbf{Repos}\} \quad (30)$$

$$\forall i \in \textbf{Intervalles}, \forall q \in \textbf{Qualifications}, W_J(i, q) \leq |\{ X(e, i) = q, \forall e \in \textbf{Employés} \}| \quad (31)$$

Où **Intervalles**, **Employés, Qualifications** sont des ensembles d'objets donnés, |S| dénote la cardinalité de l'ensemble S.

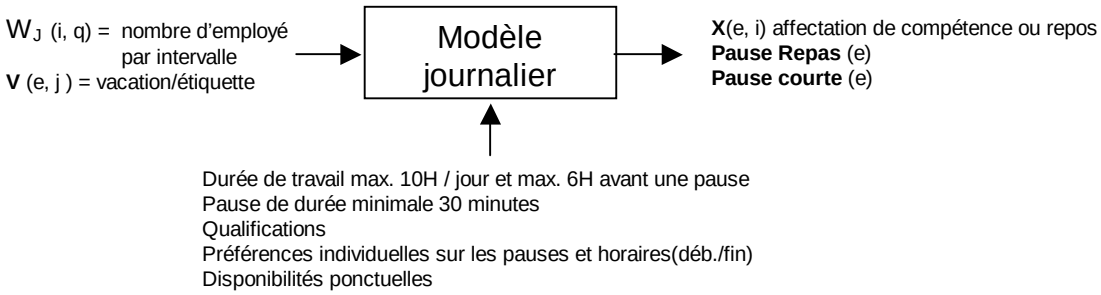


Figure 5.60. Le modèle journalier

Les besoins par intervalle et par qualification en nombre d'employés W_J sont des données d'entrée obtenue par des calculs statistiques vus au chapitre 2. Si la vacation ou l'étiquette des employés est connue, le solveur journalier redéfinira les horaires exacts de l'employé pour chaque jour afin de couvrir les besoins horaires par compétence ce jour là. Il prendra également en compte la durée maximale de travail, les préférences individuelles sur les horaires et les repas.

Le solveur vérifiera aussi les durées de travail hebdomadaires ou la moyenne des heures hebdomadaires sur 12 semaines.

Pour passer au niveau d'agrégation supérieur, on agrége les besoins journaliers. Soit :

- La définition horaire des vacations, sous la forme de A(v, i) une matrice booléenne qui valent 1 si la vacation v couvre l'intervalle i, 0 sinon. Pour le modèle journalier, à la place des vacations réelles, on utilise **en entrée** les étiquettes qui ne se chevauchent pas. Le modèle journalier pourra modifier les heures des étiquettes des employés spécifiques pour absorber la charge journalière.
- La durée constante d'un intervalle i = D_J en heures.

tel-00004380, version 1 - 29 Jan 2004

- Le nombre d'heures par employé équivalent temps-plein ETP = $H_{ETP.}$

Alors le besoin en qualification par vacation W_{JM} (v, q) en ETP est donné par :

$$\forall v \in \textbf{Vacations}, \forall q \in \textbf{Qualifications}, \; W_{JM}(v,q) = \frac{\sum_{i=1}^{nb\_intervalles} W_J(i,q) * A(v,i)}{H_{ETP}} * D_J \qquad (32)$$

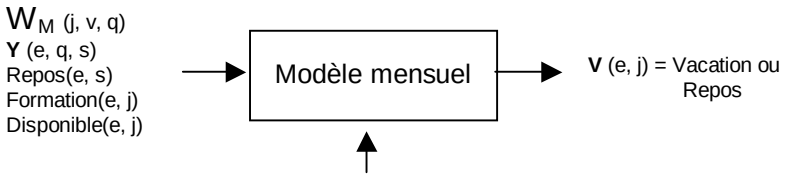
Si la charge journalière est la même tous les jours, $\forall$ v, $\forall$ q, W_{JM} (v, q) = W_M (j, v, q), le besoins au niveau mensuel. Sinon, il faut faire le calcul d'agrégation par vacation pour chaque jour j, en prenant les besoins correspondants par intervalle.

5.2.2 Le modèle mensuel

Au niveau mensuel, les variables prennent leur valeur dans l'ensemble des vacations. Les besoins mensuels W_M (j, v, q) sont exprimés en termes de nombre d'employés par vacation v et par qualification q, pour le jour j. Les employés sont affectés aux vacations en fonction de leurs disponibilités et qualifications.

$$V(e, j) \subseteq \textbf{Vacations} \qquad (33)$$

$$\forall j \in \textbf{Jours}, \forall q \in \textbf{Qualifications}, \qquad (34)$$
$$W_M(j, v, q) \leq |\{ V(e, j) = v, \forall e \in \textbf{Employés} \text{ et Qualifié } (e, q) \}|$$



Durée de travail hebdomadaire max. 48H, max. 6 jours / semaine
Durée de repos entre 2 vacations: min. 11H, min. 24H +11H si week-end inclus
Qualifications et Préférences pour jour de repos et week-end
Type de vacations: tôt/tard, court/long, avec/sans coupure
Repos Hebdomadaire non-isolé, week-end entier (Sam+Dim)
Indisponibilités ponctuelles, Nb. min de jours travaillés successifs

Figure 5.61. Le modèle mensuel

A ce niveau d'agrégation, l'équité des affectations sur un horizon assez large (de l'ordre de trois mois) doit être prise en compte (ex. le nombre de nuits ou travail de week-end).

A partir des besoins W_M (j, v, q), on obtient les besoins W_A (s, q) en sommant les vacations des jours dans la semaine concernée.

5.2.3 Le modèle annuel

Au niveau annuel, la variable est le nombre d'heures travaillées de l'employé e, à la semaine s et à la compétence q. Les employés sont affectés afin de satisfaire des besoins annuels W_A donnés par semaine et par qualification.

tel-00004380, version 1 - 29 Jan 2004

$$Y(e, q, s) \subseteq \mathbf{N} \quad (35)$$

$$\forall s \in \mathbf{Semaines}, \forall q \in \mathbf{Qualifications}, W_A(s, q) \leq \sum_{e \in \mathbf{Employés}} Y(e, q, s) \quad (36)$$

La variable Y est souvent utilisée dans les modèles de management dans le secteur de la distribution.

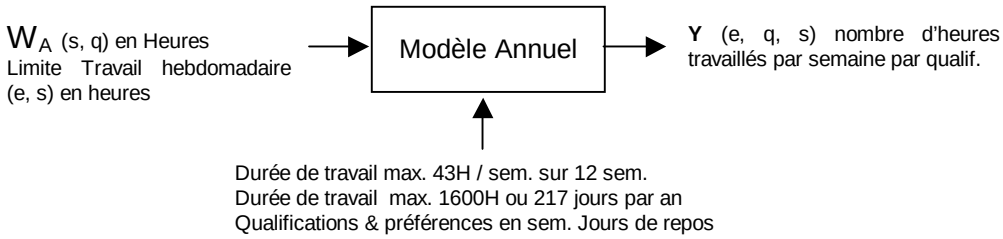


Figure 5.62. Le modèle annuel

5.2.4 Des variantes

Au niveau annuel, la législation française stipule un maximum de 1600 heures de travail cumulées, ou un forfait de 217 jours (7,37 H par jour ou 36,68 H par semaine) pour la catégorie d'encadrement du personnel. Nous traiterons séparément cette catégorie car le modèle journalier ne pourra pas s'appliquer.

Dans certaines industries, la modulation annuelle se définit par la spécification du nombre des semaines à 30h, 35,5h ou 40 h. Dans notre modèle annuel, cela se traduit par une restriction du domaine des variables Y qui ne peuvent plus prendre une valeur entière quelconque entre les bornes. Cette discrétisation exigerait une recherche dans l'espace des solutions qui peut devenir combinatoire.

tel-00004380, version 1 - 29 Jan 2004

5.3 LE MODELE A MULTIPLES NIVEAUX

5.3.1 Le schéma général fonctionnel

Afin de produire des plannings légaux, nous proposons d'utiliser trois niveaux dans un solveur à multiples niveaux. Le cadre du CSP fournit la *colle* entre les solveurs individuels. Les décisions prises à un niveau peuvent avoir des conséquences qui se propagent sur d'autres niveaux. Une contrainte peut être réalisée au sein des méthodes de mise à jour dans les langages de programmation orientés objet.

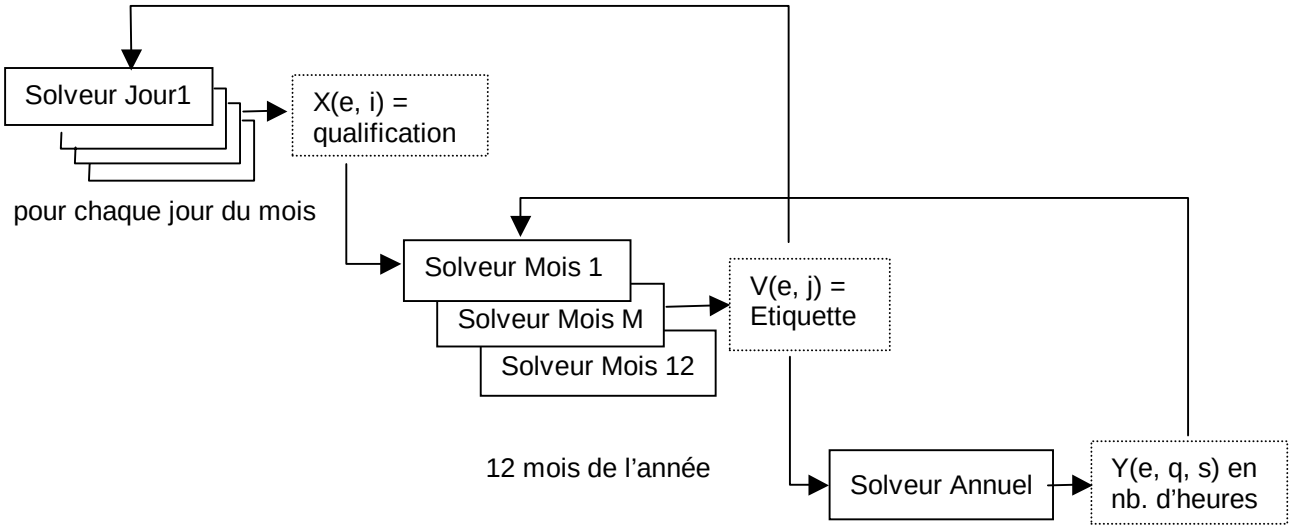


Figure 5.63. Produire un planning mensuel

Comme le montre la Figure 5.63, pour produire un planning mensuel utilisant des contraintes au niveau journalier, mensuel et annuel, sont nécessaires 31 modèles journaliers, 12 modèles mensuels et 1 modèle annuel. Les résultats d'un niveaux N servent de contrôle au niveau N-1 et d'entrée au niveau N+1.

Cette figure ne donne qu'une indication sur la démarche à suivre pour résoudre le MMN. Effectivement, on peut démarrer avec le solveur annuel, proposer les affectations mensuelles puis générer les horaires.

5.3.2 La propagation inter - niveaux

Exemple : Considérons un vendeur fast-food dont **Qualifications** = { Caisse, Livraison, Cuisine, Comptabilité}. Les cinq premiers employés e_1, …, e_5 sont capables de toutes les Qualifications sauf Comptabilité. Supposons que l'employé e_5 est affecté au travail en Cuisine pendant l'intervalle i :

$$X(e_5, i) = \text{Cuisine} \quad (37)$$

Au niveau mensuel, l'ensemble des vacations possibles pour ce jour est immédiatement restreint aux vacations qui contiennent l' intervalle i.

$$V(e_5, j) = \{ v \in \text{Vacations} \mid A_{pv} = 1\} \quad (38)$$

tel-00004380, version 1 - 29 Jan 2004

Il peut y avoir des déductions secondaires si les jours successifs travaillés atteignent la limite de travail pour la semaine. Il y aura des déductions au niveau annuel si le total des affectations travaillées de la semaine dépasse 38 heures, cf. la contrainte appliquée sur un horizon glissant de 12 semaines.

Inversement, lorsque l'ensemble des vacations possibles pour un jour j et pour l'employé e est modifié, on peut déduire que les périodes non-travaillées communes aux vacations possibles restantes doivent être affectées au repos.

$$\forall v \in V(e, j), \text{ soit } P = \{p \mid p \in \text{Périodes}, A_{pv} = 0\}, \text{ alors } X(e, p) = \text{Repos}, \forall p \in P \quad (39)$$

Effectivement, au cours des périodes P, si toutes les vacations possibles pour l'employé pour ce jour (au modèle mensuel) ne demandent à travailler, alors l'employé est au repos dans le modèle journalier.

Les décisions de haut niveau d'agrégation (mensuel ou annuel) peuvent avoir un grand nombre de conséquences ou décisions aux niveaux inférieurs (tels qu'hebdomadaire ou journalier).

D'autres exemples de propagation inter – niveaux sont :

- Après avoir travaillé 35 heures sur plusieurs jours (consécutifs ou non) et si aucune des vacations compatibles de l'employé n'a une durée de 4 heures (39-35), alors le reste de la semaine doit être affecté au repos. Sinon, la limite de 39 Heures par semaine ne peut pas être respectée.
- Lorsque la vacation standard du jour d a été définie, la contrainte de 6 jours consécutifs travaillés peut être utilisée pour déduire que les jours précédents ou successifs doivent être au repos..
- Lorsque la vacation de l'employé e est affectée au repos, toutes les variables périodes de la journée X peuvent être affectées au repos.
- Le nombre de semaines consécutives à plus de 38 heures travaillées ne doit pas dépasser le seuil légal. Dans le cas où une semaine risque de durer plus de 38H et fait dépasser le nombre de semaines consécutives, l'ensemble de vacations des jours restants de la semaine peut être restreint à celles ayant un petit nombre d'heures de travail.

5.3.3 Résolution du MMN

Du point de vue de la résolution, le MMN est constitué de plusieurs MSN, avec ses propres variables, domaines et contraintes. Il y a des contraintes supplémentaires qui lient les variables des différents MSN, comme l'indiquent les exemples de la propagation inter-niveaux du paragraphe précédent.

Algorithme de recherche

Nous proposons une méthode de recherche pour résoudre le MMN, en construisant itérativement la solution. On peut exploiter les structures et possibilités offertes par la navigation à travers les différents modèles. La sélection de variables requiert plus d'attention car les variables sont de niveau d'agrégation différente. Les variables aux

tel-00004380, version 1 - 29 Jan 2004

niveaux supérieurs sont prioritaires parce qu'elles correspondent à un grand nombre de variables aux niveaux inférieurs.

L'algorithme le plus simple pour résoudre le MMN serait un *seul* mécanisme de retour arrière systématique, semblable à la Programmation en Logique avec la satisfaction des contraintes. Le cadre théorique d'un tel système a été décrit dans [Hen89]. Le solveur appliquerait le "fail-first principle"[20] successivement à travers les différents niveaux. Le solveur annuel pourrait choisir la semaine la plus *contraignante*, par exemple celle où les besoins sont très élevés par rapport aux disponibilités. Une affectation d'heures hebdomadaires pour un employé à la compétence la plus rare pourra déclencher le solveur mensuel, qui à son tour, choisira le jour le plus contraint et en déduira des vacations. Finalement le solveur journalier du jour pourra choisir la période la plus contrainte. Le solveur journalier pourrait poursuivre et instancier les périodes voisines de l'employé déjà présent aux compétences avec les plus grands besoins.

L'ensemble des vacations du jour serait progressivement réduit et enfin la variable V(employé, jour) pourrait s'instancier. A son tour, cela pourra déclencher d'autres contraintes au niveau mensuel sur les jours voisins.

Une autre approche consiste à utiliser un algorithme de retour arrière pour chaque solveur, dans son propre processus, voire exécuter sur des processus ou des machines indépendantes. Cela est en dehors du cadre de notre étude d'un progiciel de planification.

Heuristiques

L'espace de recherche du MMN est de taille astronomique puisque l'horizon de planification est très large. Afin de produire des *bons* plannings, le MMN peut tenir compte des *préférences* de l'*employeur ou de l'employé* qui ne pourraient pas tenir dans l'horizon limité du MSN.

Une *préférence de l'employeur* est typiquement de produire des plannings qui sont équilibrés par rapport aux affectations non-populaires, telles que les affectations de « jour férié », les affectations « tôt le matin suivant un jour de repos » (tel que le week-end, le jour férié ou le congé annuel) ou les affectations « tard le soir précédent un jour de repos ». Bien sur, ces préférences doivent céder si une contrainte légale n'est pas respectée sur une autre partie du planning partiel.

Une autre préférence du management serait d'encourager des relations entre tuteur-étudiant (en les faisant travailler ensemble sur certaines tâches) ou d'avoir un personnel expérimenté à certaines périodes de la journée.

En fait, le grand horizon du modèle annuel permettra au MMN de modéliser des phénomènes qui changent lentement, tels que les besoins liés aux saisons et les non – disponibilités liées au changement de postes.

Les préférences de l'employé au niveau journalier incluent :

- Employés à mi-temps qui préfèrent travailler le soir, d'autres le matin

[20] Principe de « l'échec préalable ».Voir le glossaire et la référence [HE80]

tel-00004380, version 1 - 29 Jan 2004

Les modèles à multiples niveaux

- Employés qui préfèrent des journées longues (et moins de jours par semaine) ; d'autres préfèrent des journées courtes pour des activités annexes, ex. sportives
- Employés qui préfèrent travailler de façon continue, d'autres préfèrent des pauses longues
- Employés qui préfèrent des heures fixes tous les jours.

De même on peut considérer des préférences portant sur la pause

- Heures souples ou fixes
- Durée de pause d'une demi-heure, d'une ou deux heures

Aux niveaux hebdomadaire et mensuel, les préférences individuelles peuvent être exprimées pour du travail ou du repos sur différents jours de la semaine. Au niveau annuel, on pourra exprimer des préférences pour des semaines de repos. Puisque le MMN peut modeler un horizon très grand, il peut tenir compte de beaucoup de contraintes ou préférences de différents types.

Lorsque l'horizon du planning est suffisamment long, on peut s'appuyer sur l'historique afin d'assurer l'équité entre personnes au niveau des horaires désagréables telles que :

- Les affectations en jour férié, week-end ou suivant des périodes (vacances scolaires, etc.)
- Les affectations tôt le matin le lendemain d'un jour d'absence (par exemple jour férié, week-end ou congé)
- Les affectations tard le soir la veille d'une absence
- Une semaine haute précédent une semaine d'absence par exemple congé annuel

Algorithme de recherche locale

Afin d'éviter une recherche systématique trop coûteuse en temps machine, nous adapterons l'algorithme à une recherche locale. Disposant toujours d'un état constitué de l'ensemble des variables de décision, l'algorithme simule le retour arrière en cas de conflit de ressources. En fonction du conflit courant, l'algorithme exécute une série d'opérations d'affectations à défaire et à faire.

Chaque série d'opérations, formalisée en règle, est applicable sous des conditions différentes. Les règles sont souvent sujettes à des choix : une méthode gloutonne sera appliquée, en attendant des raffinements ultérieurs. Plusieurs règles seront expliquées au paragraphe 5.5.

tel-00004380, version 1 - 29 Jan 2004

5.3.4 Comparaison du MSN et MMN

Instanciation partielle des modèles

La Figure 5.63 illustre les différents modèles à instancier dans une instance typique du problème à résoudre. Afin de prendre en compte les affectations passées, l'horizon annuel doit couvrir le passé, mais doit couvrir l'année en cours, afin de ne pas dépasser les heures annuelles. En conséquence, il est nécessaire de couvrir 16 – 18 mois de planning. Cependant, il n'est pas nécessaire d'instancier tous les modèles et pour toutes les périodes de 15 minutes. Nous proposons l'*instanciation partielle des modèles*, i.e. instancier le bon niveau de détails.

Dans un système de planification avec un horizon du mois à 15 minutes près, l'espace de recherche est énorme. Avec les structures offertes par le MMN, il est possible d'exprimer les préférences des employés. Dans le MSN offrant le même horizon avec le même degré de détails, il n'y a pas assez de structures pour exprimer les préférences. En conséquence, les heuristiques basées sur le MMN peuvent prendre ces expressions en compte, les utiliser pour guider la recherche dans des espaces préférables. Quand une préférence entraîne une conséquence sanctionnée par une contrainte, la préférence doit être ignorée.

Considérons l'ordonnancement d'un établissement sur 1 mois, avec des contraintes aux niveaux journaliers, hebdomadaires, mensuels et annuels. Dans un MSN monolithique, il faut 365 jours * 144 = 52560 périodes de 10 minutes. Chaque période correspondrait à une variable par employé. Avec 4 valeurs (représentant des qualifications ou du repos) possibles par variable, la complexité théorique est $4^{52560} \sim 10^{31500}$.

Avec le MMN, on aura créé un modèle annuel, 12 modèles mensuels et 31 modèles journaliers pour couvrir un mois de planning. Chaque modèle journalier contient 144 variables par période de 10 minutes. On décompte 31*144 = 4464 variables journalières chacune avec 4 valeurs qui représentent des qualifications ou du repos. Les 12 modèles mensuels possèdent 12*31 = 372 variables avec par exemple 5 valeurs représentant des vacations travaillées ou non. Les 52 variables du modèle annuel ont des valeurs entières, entre 0 et 48. La complexité théorique du MMN est de $4^{4464} * 5^{372} * 52^{49} \sim 10^{2687} * 10^{260} * 10^{84.}$, soit au total 10^{3031}, d'où un gain en complexité théorique de 10^{28469} pour chaque employé.

L'approche opportuniste

Les contraintes des modèles à multiples niveaux ont essentiellement un comportement non-séquentiel ; leurs conséquences changent de niveau de façon opportuniste. Cela permet une résolution en collaboration entre plusieurs processus. Les déductions faites aux niveaux supérieurs (annuel ou mensuel) peuvent se traduire par un grand nombre de déductions aux niveaux inférieurs (hebdomadaire ou journalier).

Le MMN peut aussi bénéficier des méthodes courantes traitant deux niveaux (cf. [Par98]) où une solution journalière peut être réutilisée pour traiter un autre jour avec des besoins similaires. Cela est applicable lorsque chaque jour peut être considéré comme un sous-problème indépendant.

tel-00004380, version 1 - 29 Jan 2004

5.4 CONDITIONS NECESSAIRES EN MULTIPLES QUALIFICATIONS

Le trait commun des modèles aux niveaux simples présentés au paragraphe 5.2 est la multiple qualification des salariés. Nous présentons dans ce paragraphe des conditions redondantes qui peuvent détecter l'inconsistance dans un planning partiel, basées sur la législation sur les durées de travail et du repos. Ces conditions sont inspirées des travaux de dimensionnement (voir 2.9.4).

Intuitivement, ces conditions sont des *sommes en colonne* du planning sur l'ensemble des employés, pour un intervalle spécifique. Des conditions nécessaires peuvent être obtenues avec des *sommes en ligne* sur l'ensemble des intervalles pour un employé donné.

Par rapport aux travaux de [CGL95] où nous faisons une analogie entre le code horaire et la qualification, les contraintes suivantes sont absentes de notre problématique :

- Par jour j, le nombre de salariés affectés au code horaire V a une borne supérieure, alors que nous ne connaissons que la borne inférieure donnée par la charge à couvrir
- Par code horaire V, le total des affectations par salarié doit être compris entre des bornes. Dans notre problème, nous n'avons que le nombre d'heures travaillées.

Ainsi, notre problème est moins contraignant et les conditions proposées ci-après sont moins fortes que celles de [CGL95]. Elles peuvent être appliquées à chaque niveau (journalier, mensuel et annuel), mais nous le détaillerons uniquement pour le niveau journalier.

Par commodité, on utilisera les variables en PLNE :

$X(e, i, q) = 1$ si l'employé e est affecté à la qualification q au cours de l'intervalle i (**40**)

Mais ces conditions peuvent être implantées en PPC ou en PLNE. Nos hypothèses de travail sont :

- Une personne possède une ou plusieurs qualifications q. Elle ne peut travailler sur la qualification q que si elle est qualifiée.
- Il n'y a aucun délai notable lorsqu'une personne change d'affectation et travaille sur une autre qualification.

5.4.1 Conditions nécessaires par intervalle

Une première condition nécessaire pour tout intervalle i, est que la somme des disponibilités de tous les salariés soit supérieure ou égale au besoin total sur toutes les qualifications. Elle est inspirée du problème où il n'existe qu'une seule qualification.

$$\forall i \in \textbf{Intervalle} \mid \sum_{q \in \textbf{Qualifications}} W(i, q) \leq \sum_{q \in \textbf{Qualifications}} \sum_{e \in \textbf{Employés}} X(e, i, q) \quad (\textbf{41})$$

Dans le contexte de multiples qualifications, il n'y a pas de transfert entre deux qualifications distinctes. Ainsi pour tout intervalle i et pour chaque qualification q, on peut retrouver la même condition nécessaire : la somme des disponibilités des salariés qui ont au moins cette qualification doit être supérieure ou égale au besoin W en cette qualification.

tel-00004380, version 1 - 29 Jan 2004

$$\forall q \in \textbf{Qualifications}, \forall i \in \textbf{Intervalle} \mid W(i, q) \leq \sum_{e \in \textbf{Employés(q)}} X(e, i, q)) \quad (\textbf{42})$$

Ici, on ne considère que l'ensemble des employés qualifiées en q.

Cette condition découle de la possibilité que tous ces salariés soient affectés à cette qualification. Bien entendu, elle n'est pas suffisante car un salarié multi-qualifié peut être affecté à d'autres qualifications.

5.4.2 Conditions nécessaires supplémentaires par intervalle

Elucidons le cas de deux compétences q_1 et q_2. Les conditions sont pour l'ensemble des qualifications et pour chaque qualification :

$$\forall i \in \textbf{Intervalle} \mid W(i, q_1) + W(i, q_2) \leq \sum_{e \in \textbf{Employés}} X(e, i, q_1) + X(e, i, q_2) \quad (\textbf{43})$$

$$\forall i \in \textbf{Intervalle} \mid W(i, q_1) \leq \sum_{e \in \textbf{Employés}} X(e, i, q_1) \quad (\textbf{44})$$

$$\forall i \in \textbf{Intervalle} \mid W(i, q_2) \leq \sum_{e \in \textbf{Employés}} X(e, i, q_2) \quad (\textbf{45})$$

Prenons un petit exemple numérique avec trois employés et deux qualifications :

	1	2	3	4	5	6	7	8	9	10	11	12
e_1 (q_1)	q_1	q_1	q_1	q_1	repas	q_1	q_1	q_1				
e_2 (q_1, q_2)					q_1	q_2		repas	q_1	q_1	q_1	q_1
e_3 (q_2)			q_2	q_2	q_2	repas	q_2	q_2	q_2	q_2		
$W(q_1)$	1	1	1	1	1	1	1	1	1	1	1	1
$W(q_2)$			1	1	1	1	1	1	1	1		

Figure 5.64. Exemple d'un planning avec 3 employés et 2 qualifications

Les conditions précédentes sont satisfaites pour tout intervalle i et toute qualification q_i.

Nous proposons une expression permettant de généraliser ces conditions

$$\forall i \in \textbf{Intervalle}, \forall Q \text{ sous-ensemble de l'ensemble des } \textbf{Qualifications}, \\ \mid \sum_{q \in \textbf{Q}} W(i, q) \leq \sum_{q \in \textbf{Q}} \sum_{e \in \textbf{Employés}} X(e, i, q)) \quad (\textbf{46})$$

Effectivement pour 2 qualifications, on a Q = { { q_1 }, { q_2 }, { q_1, q_2 } }, donc cette équation remplace les trois équations précédentes pour { q_1 }, { q_2 } et { q_1, q_2 }.

5.4.3 Conditions nécessaires dues au repos

Des conditions semblables s'appliquent aussi à une suite d'intervalles consécutifs sur les lignes du planning. Par exemple, la disponibilité totale de tous les employés est supérieure ou égale à la somme des besoins de toutes les qualifications sur tout l'horizon, en sommant l'équation 42 :

$$\sum_{i \in \textbf{Intervalle}} \sum_{q \in \textbf{Qualifications}} W(i, q) \leq \sum_{i \in \textbf{Intervalle}} \sum_{q \in \textbf{Qualifications}} \sum_{e \in \textbf{Employés}} X(e, i, q) \quad (\textbf{47})$$

Or, tout employé e ne travaille pas tout le temps : il y a des pauses et des repos à chaque niveau. Pour un horizon mensuel, *Heures Totales* vaux 45 h pour un employé en plein temps ou 22,5 h pour un employé à mi-temps. Si l'horizon est supérieur à la journée, il y a une telle contrainte pour chaque suite d'intervalles égale à la journée.

tel-00004380, version 1 - 29 Jan 2004

$$\forall\ e \in \textbf{Employés} \mid \sum_{i \in \textbf{Intervalle}} \sum_{q \in \textbf{Qualifications(e)}} X(e, i, q) \leq \textit{Heures Totales}\ (e) \qquad (\textbf{48})$$

Au niveau journalier, la contrainte des heures consécutives maximales soit **HCM**, peut être formulée comme une contrainte somme sur (HCM+1) intervalles consécutifs.

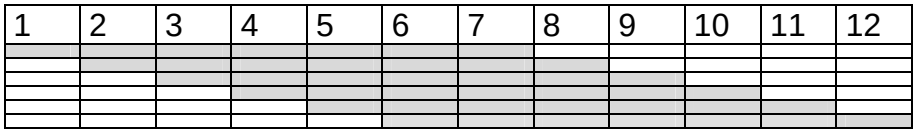


Figure 5.65. Condition de repos applicable sur 7 périodes (avec N=12 et HCM=6)

Sur un horizon contenant N intervalles, il faut considérer toute suite de (HCM+1) intervalles consécutifs, comme le montre la figure ci-dessus. Au total, le nombre de telles suites est = N – (HCM+1) + 1 = N – HCM. Pour chacune de ces suites, on peut appliquer (48), comme le montre l'équation suivante :

Soit j l'index de chaque ligne de la figure précédente représentant une suite d'intervalles qui commence à i=j et termine en i= j + HCM. Attention, la base des indices est 1.

$$\forall\ e \in \textbf{Employés}\ \forall\ j = 1, \ldots, \text{N-HCM} \mid \sum_{i = j, \ldots, j+\text{HCM}} \sum_{q \in \textbf{Qualifications}} X(e, i, q) \leq \text{HCM} \qquad (\textbf{49})$$

On peut conjuguer cette limitation avec les besoins (48) pour obtenir une condition encore plus contraignante : Partant de (48) valable pour chaque intervalle, on somme sur les intervalles

$$\forall \text{ partition Q de l'ensemble des } \textbf{Qualifications}, \forall\ j = 1, \ldots, \text{N-HCM} \quad (\textbf{50})$$
$$\mid \sum_{i = j, \ldots, j+\text{HCM}} \sum_{q \in \textbf{Q}} W(i, q) \leq \sum_{i = j, \ldots, j+\text{HCM}} \sum_{q \in \textbf{Q}} \sum_{e \in \textbf{Employés}} X(e, i, q)$$

En inversant les sommes et nombre d'employés (q) étant le nombre d'employés avec la qualification q.

$$\sum_{i = j, \ldots, j+\text{HCM}} \sum_{q \in \textbf{Q}} W(i, q) \leq \sum_{q \in \textbf{Q}} \text{HCM} \qquad (\textbf{51})$$
$$\sum_{i = j, \ldots, j+\text{HCM}} \sum_{q \in \textbf{Q}} W(i, q) \leq \text{HCM} * \text{Nombre d'employés (q)}$$

5.4.4 Application des conditions

Ces conditions sont vérifiables pour chaque i où il y a une modification de domaine des variables X de l'intervalle i. La mise à zéro (ou repos) pour l'employé à un intervalle i, intervenant lorsque e a atteint ses heures légales est un signal **fort**, il faudrait vérifier la condition (48) pour l'ensemble des qualifications q de l'employé.

Le nombre total de tests est potentiellement très grand. Pour réduire les calculs, il faudra stocker les besoins par intervalle suivant toute partition des qualifications.

tel-00004380, version 1 - 29 Jan 2004

5.5 METHODES HEURISTIQUES A UN NIVEAU

Nous présentons ici nos travaux utilisant une méthode heuristique appliquée à un niveau. L'objectif de cette méthode est de résoudre un besoin indirectement, lorsqu'il n'y a pas de candidat avec la qualification requise et satisfaisant les contraintes applicables.

Le principe est de faire une suite d'échange d'affectations localement afin de dégager une ressource qualifiée. Il s'agit d'une sorte de retour arrière non-chronologique, dirigé par les besoins et les contraintes du problème. Cette méthode est liée très fortement aux contraintes à satisfaire.

L'algorithme général de cette méthode est le suivant :

- Choisir le besoin *critique* à traiter : Les meilleures heuristiques seront utilisées pour traiter le problème dans un ordre entraînant le moindre échec possible.
- Résoudre les manques et traiter les éventuels problèmes
- Répéter jusqu'à l'épuisement les besoins à satisfaire.

Bien entendu, ces méthodes heuristiques font l'abstraction des propagations intra et inter niveaux.

5.5.1 Le modèle annuel

Nous allons détailler chaque étape de calculs dans la suite, pour le cas du niveau annuel où l'intervalle est la semaine. Au cours de cet intervalle, on affecte à un employé un nombre d'heures à chacune des qualifications qu'il possède.

Nous avons décidé de traiter une semaine complète à la fois avant de passer à la suivante, afin de simplifier le traitement des contraintes de charge par semaine.

1. Déterminer la semaine *critique*
2. Attribuer les heures minimales à chaque employé
3. Résoudre les manques par compétence
4. Traiter les problèmes éventuels
5. Répéter 1. jusqu'à résoudre tout le problème.

Au niveau annuel, la contrainte Semaine Consécutive Haute relève de la réglementation : sur tout intervalle de 12 semaines, la moyenne hebdomadaire ne doit pas dépasser un seuil, p. ex. 38 Heures.

Etape 1. Déterminer la semaine critique

Il est certain que l'ordre de traitement des semaines jouera un rôle primordial dans la performance de l'algorithme. Il faut traiter en premier les semaines dites *critiques* où la *marge de manœuvre* est la moindre : elles s'estiment par rapport à une différence entre la disponibilité du personnel et les besoins.

- Calculer le besoin global de la semaine, sur toutes les compétences.

tel-00004380, version 1 - 29 Jan 2004

- Calculer la disponibilité du personnel. Or il y a deux façons de la calculer :
  La disponibilité minimale est la somme des bornes inférieures de disponibilité des salariés.
  La disponibilité maximale est la somme des bornes supérieures de disponibilité des salariés.

- Distances : Différence entre le besoin global et disponibilité minimale ou maximale

$$\text{Distance}_1 = \sum_{e=1}^{nb\_agents} B_{\sup}(e,s) \quad - \quad \sum_{c=1}^{nb\_qualifications} Besoins(s,q) \tag{52}$$

$$\text{Distance}_2 = - \sum_{c=1}^{nb\_qualifications} Besoins(s,q) \quad + \quad \sum_{e=1}^{nb\_agents} B_{\inf}(e,s) \tag{53}$$

Après essais nous avons finalement opté pour une distance composée des deux premières :

$$\text{Distance}_3 = \sum_{e=1}^{nb\_agents} B_{\sup}(e,s) + \sum_{e=1}^{nb\_agents} B_{\inf}(e,s) \quad - 2\times \sum_{c=1}^{nb\_qualifications} Besoins(s,q) \tag{54}$$

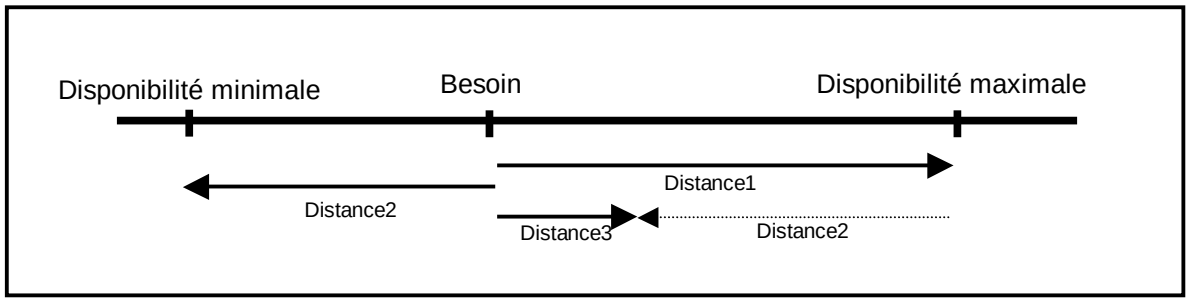


Figure 5.66. Distances pour évaluer les semaines dans le modèle annuel

On sélectionne comme semaine critique celle pour laquelle la distance est minimale. Bien entendu, on peut toujours se tromper dans l'ordre de traitement des semaines.

Etape 2. Attribuer les heures minimales à chaque employé

La résolution d'une semaine commence par l'attribution pour chaque employé du nombre minimal des heures qu'il doit réaliser pour cette semaine. Une fois de plus on traite les employés suivant un ordre de priorité. On prend d'abord l'ordre croissant du nombre des compétences ; en cas d'égalité, on prend celui pour qui la valeur

$$\text{No. d'heures annuelles} - \text{No. d'heures affectées} - \text{Min. Heures Effectuables}$$

est maximum.

Un employé est dit « effectuable » s'il n'a pas encore atteint sa borne supérieure pour la semaine, si ses heures annuelles permettent de satisfaire les bornes inférieures des

tel-00004380, version 1 - 29 Jan 2004

semaines non encore traitées, et s'il ne transgresse pas la contrainte *Semaines Haute Consécutives*.

- A un employé mono-compétent, on affecte le seuil minimum d'heures à effectuer d'un seul trait.

- Pour un employé à multiples compétences, il faut choisir la compétence *prioritaire* suivant la différence entre le besoin et la disponibilité par compétence.
  Comme on ne sait pas a priori, à quelle compétence affecter ces heures minimales, on affecte une heure à la fois jusqu'à l'épuisement des heures minimales de l'agent, à la *compétence prioritaire agent*, définit comme suit :

  La compétence prioritaire pour une semaine et pour un employé est celle pour laquelle la valeur *Besoin Restant* / *Nb. Agents Compétents* est maximale. En cas d'égalité, on prend la compétence dont la différence *Disponibilité Restante – Besoin Restant* est minimale. Ces critères ont été testés et donnent des bons résultats sur la plupart des jeux.

Etape 3. Résoudre les manques de compétence

Une fois que les heures minimales ont été attribuées, on s'attaque aux manques des compétences. La résolution des manques se fait heure par heure, en choisissant la compétence et l'employé concernés. La fonction *compétence prioritaire semaine* retourne la compétence dont les besoins ne sont pas satisfaits et dont la différence *Disponibilité Restante – Besoin Restant* est minimale. L'employé dispose des heures restantes par rapport au seuil B_{sup} (e, s). On prend celui dont la valeur *No. d'Heures annuelles – No. d'heures affectées – Min. Heures Effectuables* est maximum.

A la fin de cette étape, soit tous les besoins ont été comblés (dans ce cas on passe à la résolution de la semaine suivante), soit certaines compétences ont encore des manques car aucun employé n'est disponible directement pour les combler. On parle alors du traitement d'éventuels problèmes.

Etape 4. Traiter les problèmes éventuels

Le traitement d'éventuels problèmes dont l'origine peut être :

- Il y des employés disponibles en cette semaine qui sont incompétents :
  le traitement consiste en déplaçant des heures afin d'en dégager suffisamment pour résoudre le problème
- Il n'y a plus d'employés disponibles : il faut faire un retour arrière.

Exemple Besoin de n heures en qualification q° en semaine s :

Situation 1 : Il reste des employés affectables pour la semaine s, mais aucun ne possède la qualification q°. De plus, il existe un employé e_1, possédant la qualification q° (il n'est donc a priori plus affectable) et ayant une affectation à une qualification q_1, également possédée par un des employés encore affectables, appelons-le e_2. Il est alors

tel-00004380, version 1 - 29 Jan 2004

possible de libérer e_1, en redonnant certaines de ses heures en q_1 à e_2 et enfin de combler le manque en q° en réaffectant e_1.

Employé	Qualification	Intervalle i		Intervalle j	Total
e_1	q°	+n heures			
	q_1	-n heures			
	Total	Pas de changement			Pas de changement
e_2	q_1	+n heures			
	total	Ajouter n heures			Ajouter n heures
Total	q°	Ajouter n heures			
	q_1	Pas de changement			Pas de changement

Figure 5.67. Résolution d'un besoin en q° au niveau annuel dans la situation où e_1 qualifié n'a plus d'heures, e_2 a encore des heures mais non – qualifié et une qualification commune q_1

Etape 1 : on allège e_1 de 4 heures pour la qualification q_1 et on redonne ces heures à e_2.
Etape 2 : e_1 dispose de 4 heures et peut donc combler le manque en q°.

Remarque : Il s'agit bien sûr ici d'un exemple simple, il est parfois nécessaire de « passer » par plusieurs employés « encore affectables » (type e_2) ou par plusieurs agents « type e_1 » pour combler un manque. Si ici le manque en q_1 avait été de plus de 6 heures, on aurait été limité au niveau de la borne supérieure de e_2, il aurait fallu un second agent pouvant encore être affecté. De la même façon, e_1 ne pouvait suffire que pour un manque inférieur à 10 heures (= la valeur de son affectation en q_2)

Situation 2 : Contrairement au cas précédent, il se peut qu'aucun des employés encore affectables n'aient de qualification commune avec un des employés possédant la qualification déficitaire q°. Il est alors parfois possible de passer par un 3ème employé e_3, intermédiaire, ainsi que par une 3ème qualification q_2.

tel-00004380, version 1 - 29 Jan 2004

Employé	Qualification	Intervalle i		Intervalle j	Total
e_1	q°	+n heures			
	q_2	-n heures			
	Total	Pas de changement			Pas de changement
e_2	q_1	+n heures			
	Total	Ajouter n heures			Ajouter n heures
e_3	q_1	+n heures			
	q_2	-n heures			
	Total	Pas de changement			Pas de changement
Total	q°	Ajouter n heures			
	q_1	Pas de changement			Pas de changement
	q_2	Pas de changement			Pas de changement

Figure 5.68. Résolution d'un besoin q° au niveau annuel dans la situation où e_1 qualifié n'a plus d'heures, e_2 a des heures mais non – qualifié en q° et sans qualification commune entre e_1 et e_2

Etape 1 : on allège e_3 de n heures pour la compétence q_1 et on redonne ces heures à e_2.
Etape 2 : e_3 a maintenant n heures disponibles et peut prendre en charge n heures en q_2 qui auparavant étaient assurées par e_1.
Etape 3 : e_1 dispose maintenant de n heures et il peut couvrir le manque en q°.

Même remarque que pour la solution 1, il peut y avoir plusieurs employés de type e_1, e_2 ou e_3. Chacun de ces employés peut contribuer une ou plusieurs heures.

Situation 3 : Lorsque la permutation d'heures entre les employés au cours d'une même semaine ne suffira pas à résorber les manques rencontrés, il faudra trouver des disponibilités sur une autre semaine déjà traitée. L'employé e_2 dans la situation 2 dispose d'heures sur son contrat, mais est déjà à leur maximum pour la semaine s. L'employé e_1 ne disposait plus d'heures sur son contrat.

Pour traiter cette situation, nous exposons la méthode heuristique dans le cadre du modèle journalier au paragraphe 5.5.3.

5.5.2 Le modèle journalier : un exemple de traitement

Au niveau journalier, on peut rencontrer des situations analogues. On rappelle que l'intervalle est de l'ordre des 15-30 minutes sur un horizon d'un jour. On affecte à un salarié au cours d'un intervalle, une qualification.

Prenons le cas où les besoins en trois qualifications sont définis sur 24 intervalles.

```
Besoins par intervalle 0..23
      0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
Q=0 : 0, 1, 2, 3, 3, 4, 6, 7, 7, 6, 6, 3, 3, 4, 4, 5, 6, 6, 4, 3, 1, 1, 1, 1
Q=1 : 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 2, 2, 1, 2, 1, 1, 1, 2, 2, 3, 3, 2, 2, 1
Q=2 : 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 2, 2, 2, 1, 2, 1, 1, 1, 2, 2, 3, 2, 2, 1
```

tel-00004380, version 1 - 29 Jan 2004

On dispose de 17 agents, travaillant un nombre maximal d'heures indiqué

```
Salarié {0,1,2}   Heures Max 12, Compétence(0)
Salarié {3,4,5,6} Heures Max 12, Compétence(1)
Salarié {7,8}     Heures Max 12, Compétence(2)
Salarié {9,10,11} Heures Max 12, Compétence(0, 1)
Salarié {12}      Heures Max 12, Compétence(1, 2)
Salarié {13,14}   Heures Max 12, Compétence(0, 2)
Salarié {15,16}   Heures Max 12, Compétence(0, 1, 2)
```

Pour chaque colonne du tableau suivant, on lit l'affectation d'un salarié à une qualification (q=0, q=1 ou q=2). En fin de ligne on totalise le nombre d'intervalles travaillées (limité à 12 correspondant aux maxima ci-dessus) et l'amplitude en nombre d'intervalles (limité à 16 maximum, dans cet exemple).

```
            0 1 2 3  4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
Salarié 00: - - - -  - 0 0 0 0 0 0 - - 0 0 0 0 0 0 - - - - -  12 Ampl=14
Salarié 01: - - - -  - - 0 0 0 0 0 - - 0 0 0 0 0 0 0 - - - -  12 Ampl=14
Salarié 02: - - - -  0 0 0 0 0 0 0 0 0 - - 0 0 0 - - - - - -  12 Ampl=14
Salarié 03: - - - -  - 1 1 1 1 1 1 1 - 1 - - - 1 1 1 1 - - -  12 Ampl=16
Salarié 04: - - - -  - - - 1 1 1 - - - - 1 1 1 1 1 1 1 1 1 -  12 Ampl=16
Salarié 05: - - 1 1  1 1 1 1 1 1 1 1 1 1 - - - - - - - - - -  12 Ampl=12
Salarié 06: - - - -  - - - - - - - - - - - - - - - 1 1 1 1 1  05 Ampl=05
Salarié 07: - - - -  - - - 3 3 3 3 3 3 - 3 - - - 3 3 3 3 3 -  12 Ampl=16
Salarié 08: - - - -  - 3 3 3 3 3 - - - - 3 3 3 3 3 3 3 - - -  12 Ampl=16
Salarié 09: - - - 0  - - 0 0 0 0 0 - - 0 0 0 0 0 0 - - - - -  12 Ampl=16
Salarié 10: - - - -  0 0 0 0 0 0 0 0 0 0 - - - 0 0 - 0 - - - -  12 Ampl=16
Salarié 11: - - 0 0  - - - 0 0 - 0 0 0 0 0 0 0 0 - - - - - -  12 Ampl=16
Salarié 12: - - 3 3  3 3 3 3 3 3 3 3 3 3 - - - - - - - - - -  12 Ampl=12
Salarié 13: - - - -  - - - - 0 0 - - - - - - - - 0 0 3 3 3 3  08 Ampl=16
Salarié 14: 3 3 - 3  3 0 0 0 - - - - - - - - - - - - - - - -  07 Ampl=08
Salarié 15: - - - -  - - - - - - - - - - - - - - - - 0 0 0 0  04 Ampl=04
Salarié 16: - - - -  1 1 1 - - - - - - - - - - - - - - - - -  03 Ampl=03
Qualif   0:00010101 01 00000000000000000000000000000000000000 04
Qualif   1:01010101 00 00000000000000000000000000000000000000 04
Qualif   2:00000000 00 00000000000000000000000000000000000000 00
Total Qual:01020202 01 00000000000000000000000000000000000000 00
```

Figure 5.69. Heuristique dans le modèle journalier

L'exemple ci-dessus correspond au besoin d'un salarié avec la qualification q=0 en i=4.

Nous présentons une heuristique simple pour résoudre ce besoin. On trouve un salarié A compétent qui est libre en cet intervalle, et occupé l'intervalle suivant (ex. 00). L'idée est de décaler son planning d'un intervalle : on recherche le dernier intervalle travaillé, i=18 dans l'exemple. On cherche alors un autre salarié B (ex.15) compétent et libre en intervalle i=18. A est affecté à q=0 en i=04. Pour garder son total en heures travaillées, l'affectation de A en i=18 est transformée à B en i=18.

Plus formellement, supposons :

`X(Intervalle, Employé)` la qualification affectée, sinon libre ou repos. La fonction booléenne `CEmpl_OK (Employé, Intervalle)` retourne vraie s'il est possible d'affecter cet employé à l'intervalle. On vérifie s'il est compétent et libre cet instant là et si ses heures de travail et l'amplitude de travail sont légales.

`Amplitude_Last (Employé)` donne le dernier intervalle travaillé de l'employé, ou -1.

tel-00004380, version 1 - 29 Jan 2004

`Assign (Intervalle, Employé, Qualification)` Affecter l'employé à cette qualification au cours de cet intervalle, mise à jour des compteurs.

`UnAssign (Intervalle, Employé, Qualification)` Désaffecter, mise à jour des compteurs.

```
Private Function Solve_More(Intv As Val_Interv, Qual As Val_Qualif) As Boolean
'Ne trouve pas d'agent à l'intervalle Intv, avec la qualification Qual,
'et disposant d'heures de travail et avec une amplitude acceptable

  Dim CEmpl As Cls_Empl, CEmp2 As Cls_Empl  'ce sont des objets
  Dim Empl As Val_Employe                   'il s'agit de son identifiant
  Dim Last As Val_Interv                    'Dernier intervalle travaillé
  Dim Found As Boolean                      'la règle a fonctionné
    For Each CEmpl In Glo_Qualif(Qual)
      Empl = CEmpl.Id
      If X(Empl, Intv) = Val_Libre And X(Empl, Intv + 1) <> Val_Libre Then
      'l'agent est disponible
        Last = Amplitude_Last(Empl)  'Find last hr to reduce Emp1's amplitude
        For Each CEmp2 In Glo_Qualif(Qual)  'Find Emp2 who is competent and
          If CEmpl_OK(CEmp2, Last) Then     'Free, within hrs and amplitude
            Call Assign(Intv, Empl, Qual)
            Call UnAssign(Last, Empl, Qual)
            Call Assign(Last, CEmp2.Id, Qual)
            Found = True
            Exit For
          End If
        Next
      End If
      If Found Then Exit For
      'Else Try other methods
    Next
  Solve_More = Found
End Function
```

Cet exemple ne montre qu'une seule règle pour résoudre un problème d'affectation d'une qualification au cours d'un intervalle. Pour que le système soit efficace, il faut disposer d'un ensemble conséquent de règles.

5.5.3 Le modèle journalier : analyse et méthode

Nous cherchons à systématiser l'implantation de la méthode heuristique appliquée au niveau journalier. Elle est mise en route lorsque nous ne trouvons pas de candidat compétent q°, libre au cours de l'intervalle i, et dans ses heures de travail et amplitude.

Employés libres en intervalle i

Parmi les employés libres en intervalle i et compétent en q°, nous appelons

- *Employés Indisponibles*(i) dont les limites en heures et amplitude sont atteintes. C'est le cas de l'exemple du paragraphe précédent.

Parmi les employés libres en intervalle i et incompétent en q°, nous appelons

- *Employés Disponibles*(i) avec des heures de travail et suffisant en amplitude.

tel-00004380, version 1 - 29 Jan 2004

Les modèles à multiples niveaux

- *Employés Pleins*(i) les employés dont les limites en heures et amplitude sont atteintes.

Plusieurs types d'échange sont proposés dans la suite du paragraphe.

Premier échange :

Pour utiliser $e_1 \in$ groupe 1, supposons qu'il existe un intervalle j différent de i, et un employé e_2 disponible en j : on affecte $X(e_2, j, q°) = 1$, et on fait un échange $X(e_1, i, q°) = 1$, $X(e_1, j, q°) = 0$. Si possible on prend j de telle sorte que l'amplitude de e_1 soit réduite : j est soit le dernier intervalle travaillé soit un intervalle de repos supplémentaire de e_1. Les autres possibilités sont : le premier intervalle travaillé d'une personne, le dernier intervalle, avant une pause d'un intervalle, après une pause d'un intervalle, etc. Dans ces situations, on pourrait libérer cette personne pour résoudre le problème initial. Cette règle de résolution recherche des disponibilités dans le temps. L'avantage est qu'elle n'exige pas de personnel à multiples qualifications.

Employé	Qualification	Intervalle i		Intervalle j	Total
e_1	q°	+n heures		-n heures	
	Total	Ajouter n heures		Supprimer n heures	Pas de Changement
e_2	q°	Pas disponible		+n heures	
	total			Ajouter n heures	Ajouter n heures
Total	q°	Ajouter n heures		Pas de Changement	

Figure 5.70. Règle de résolution dans le temps

Problème : Besoin d'affectation d'un employé à l'intervalle i en qualification q°
Existant : L'employé e_1 a atteint ses limites heures et amplitude
Primitif : Rechercher un employé e_2 compétent en q° dispose de n heures et libre en intervalle j.
Action : $X(e_1, i, q°) = 1$; $X(e_1, j, q°) = 0$ donc pas de modification en heures pour e_1. $X(e_2, j, q°) = 1$, donc ajout de n heures pour e_2

Second échange :

Pour utiliser $e_1 \in$ groupe 2, supposons qu'il existe une personne e_2 déjà affectée qui possède la compétence q° et une compétence q_1 parmi les compétences de e_1. Alors on affecte $X(e_1, i, q_1) = 1$ et on échange $X(e_2, i, q_1) = 0$, $X(e_2, i, q°) = 1$. Cette règle de résolution recherche des disponibilités dans l'espace des employés et ne modifie pas l'amplitude de e_1 et e_2. Elle complémente l'action de la première règle.

tel-00004380, version 1 - 29 Jan 2004

Employé	Qualification	Intervalle i		Intervalle j	Total
e_1	q_1	+n heures			
	Total	Ajouter n heures			Ajouter n heures
e_2	q°	+ n heures			
	q_1	- n heures			
	Total	Pas de Changement			Pas de Changement
Total	q°	Ajouter n heures			
	q_1	Pas de Changement			

Figure 5.71. Règle de résolution parmi les salariés

Problème : Besoin d'affectation d'un employé à l'intervalle i en qualification q°
Existant : L'employé e_1 n'a pas atteint ses limites, mais incompétent en q°
Primitif : Rechercher employé e_2 compétent en q° et q_1 , $X(e_2, i, q_1) = 1$.
Action : $X(e_1, i, q_1) = 1$; $X(e_2, i, q°) = 1$ & $X(e_2, i, q_1) = 0$

Troisième échange :

Pour utiliser $e_1 \in$ groupe 3, libre en i dont les limites en heures et amplitude sont atteintes et incompétentes en q°. Il suffit de combiner les deux actions ci-dessus.

Employé	Qualification	Intervalle i		Intervalle j	Total
e_1	q_1	+n heures		- n heures	
	total	Ajouter n heures		Supprimer n heures	Pas de Changement
e_2	q°	+ n heures			
	q_1	- n heures		+n heures	
	Total	Pas de Changement		Ajouter n heures	Ajouter n heures
Total	q°	Ajouter n heures		Supprimer n heures	
	q_1	Pas de Changement		Pas de Changement	

Figure 5.72. Règle de résolution combinée

Problème : Besoin d'affectation d'un employé à l'intervalle i en qualification q°
Existant : e_1 a atteint ses limites heures et amplitude et est incompétent en q°
Primitif : Rechercher employé e_2 compétent en q° et q_1, $X(e_2, i, q_1) = 1$, dispose de n heures de travail et libre en intervalle j.
Action : $X(e_1, i, q_1) = 1$; $X(e_1, j, q_1) = 0$ donc pas de modification pour e_1.
$X(e_2, i, q°) = 1$; $X(e_2, j, q_1) = 1$

Les échanges proposés dans ce paragraphe s'apparentent à une recherche non déterministe de voisinages autour d'un point dans l'espace des plannings avec l'objectif de faire une affectation à un intervalle i, d'une qualification q°, sans violer les contraintes du problème.

tel-00004380, version 1 - 29 Jan 2004

Aucun employé libre en intervalle i

Cette situation est exclue grâce aux conditions nécessaires présentées en § 5.4.

tel-00004380, version 1 - 29 Jan 2004

5.6 SCHEMAS GENERAUX ALGORITHMIQUES

Ce paragraphe rassemble les différents éléments exposés dans les paragraphes précédents du chapitre 5, au sein des schémas généraux algorithmiques.

5.6.1 Le schéma général algorithmique à un niveau

L'algorithme boucle sur la satisfaction des besoins à couvrir. A chaque itération on dispose d'un état consistant en une affectation partielle de l'ensemble du personnel.

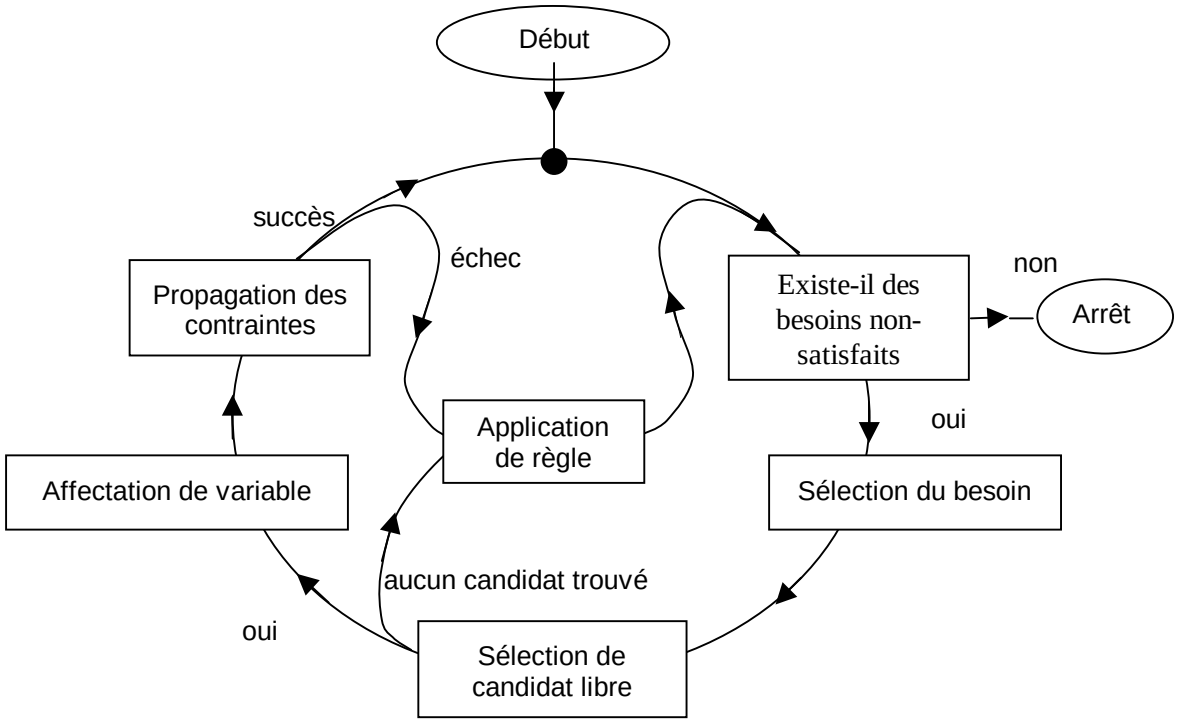


Figure 5.73. Le schéma général de l'algorithme de recherche à un niveau

La résolution du besoin contient un sous-module de choix du besoin *critique*, permettant de rencontrer des problèmes le plus tôt possible. La résolution directe consiste à la recherche d'un candidat qualifié et disponible, dont l'affectation ne transgressera pas les contraintes. Dans le cas contraire, le conflit de ressource est résolu par l'application d'une règle et la suite d'échanges d'affectations correspondantes. Comme les règles n'introduisent pas de transgressions de contraintes, on obtient un état temporaire consistant. Les conditions nécessaires sont vérifiées : si une condition n'est pas respectée, on redemande une autre solution.

Suite à la propagation des effets, l'état temporaire devient l'état courant et le processus réitère. A un niveau d'agrégation, la propagation réussit car les règles conservent les contraintes.

tel-00004380, version 1 - 29 Jan 2004

5.6.2 Le schéma général algorithmique à multiples niveaux

En plus des règles de résolution à un niveau exposées au paragraphe 5.5, nous proposons des règles exploitant les multiples niveaux. On y aura intérêt lorsque les règles à un niveau n'arrivent pas à résoudre un besoin d'affectation.

Par exemple, lorsque les échanges au niveau mensuel ne suffisent pas à dégager une ressource qualifiée, il faudrait passer au niveau annuel pour changer le nombre d'heures hebdomadaires.

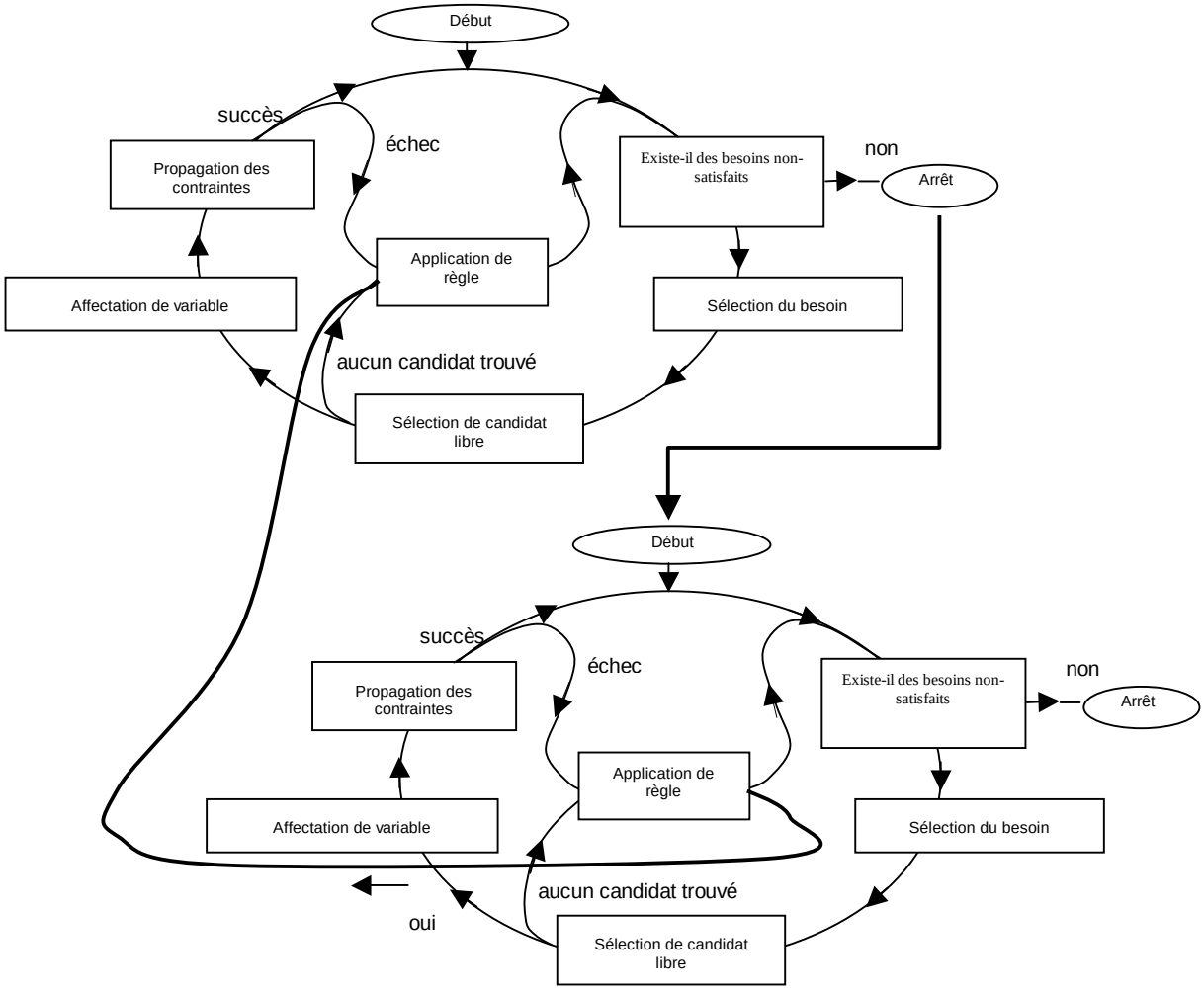


Figure 5.74. Le schéma général de l'algorithme de recherche à deux niveaux

tel-00004380, version 1 - 29 Jan 2004

5.7 CONCLUSIONS

Nous avons présenté un type de modèle intégrant plusieurs niveaux d'agrégation de temps, à la différence des modèles courants avec un seul niveau d'agrégation. Il permet de couvrir un grand horizon (à l'échelle de l'année avec trois niveaux) mais avec un niveau de détail suffisant pour appliquer les contraintes légales. Ces contraintes et préférences sont typiquement inapplicables dans un modèle avec un seul niveau d'agrégation. Les décisions des niveaux supérieurs (annuels et mensuels) peuvent se traduire en nombreuses décisions aux niveaux inférieurs (hebdomadaire ou journalier) et réduisent la complexité globale de l'algorithme de recherche.

Les conditions nécessaires et les méthodes heuristiques proposées dans ce chapitre visent à couvrir les charges définies par intervalle de temps et par qualification. En cas de conflit de ressources, les règles effectuant des échanges d'affectations, sont analogues à une recherche locale.

Notre méthode effectue une recherche locale dans un espace de solutions. A chaque itération, on résout un besoin en affectant une ressource libre. Dans le cas où toutes les ressources sont déjà prises, on effectue un échange d'affectation semblable à la méthode de Lin S. et Kernighan B.W. (1973) pour résoudre le problème du voyageur de commerce.

Avec un peu de recul, notre approche de modèles multiples pourrait remettre en cause la pratique de conception de modèles. La première étape en résolution de problème consiste en l'analyse de la granularité du problème à traiter. Nous montrons que cette étape est peut être erronée, car nous sommes limités par les contraintes applicables au niveau de la granularité choisie. Un problème réel peut cacher d'autres contraintes visibles à d'autres niveaux d'agrégation : elles ne peuvent plus être considérées par la suite de l'étude. Pour résoudre de telles situations, nous proposons l'utilisation du cadre CSP pour coller ensemble les variables et contraintes de chaque niveau. Les déductions d'un niveau peuvent être propagées à d'autres niveaux.

Les travaux autour des modèles à niveaux multiples ont été présentés dans les publications suivantes [CJW02], [CW02], [CFW02].

tel-00004380, version 1 - 29 Jan 2004

TROISIEME PARTIE : CONCLUSIONS

Dans cette partie, nous présentons nos conclusions générales sur l'ensemble des travaux engagés depuis plusieurs années. Les annexes inclurent un glossaire portant à la fois sur les termes du métier de la gestion des ressources humaines et les termes techniques de résolution.

6. Conclusions
    a. Rétrospective
    b. Bilan
    c. Perspectives

7. Annexes
    a. Glossaire
    b. Bibliographie
    c. Index des références par auteur
    d. Liste des publications par auteur

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

6 CONCLUSION

6.1 RETROSPECTIVE

C'est en constatant les limites des méthodes actuelles pour produire des plannings de qualité que nous avons décidé de lancer nos propres initiatives dans la génération automatique pour gérer au mieux les ressources humaines. Le chapitre 1 a mis en évidence les différentes problématiques de la planification du personnel.

A partir du modèle de Dantzig en 1954, différents auteurs ont traité des situations de plus en plus *réalistes* avec des méthodes très variées.

Parmi ces méthodes et travaux, on distingue bien d'une part le monde de la Recherche Opérationnelle, et d'autre part celui de l'Intelligence Artificielle. Née depuis trois siècles[21], la RO a été établie comme une science au cours de la deuxième guerre mondiale afin d'allouer de façon optimale les ressources rares. Elle s'intéresse aux solutions optimales mathématiquement et traite de grandes quantités. Pour cela, l'aîné technologique s'appuie sur des modèles qui ne traitent que les caractéristiques principales du problème et souvent ne distinguent pas les individus. Ce sont les problèmes *bien-définis* : un problème bien-défini peut être combinatoire et intraitable.

La Operational Research Society of Great Britain définit la RO comme suit :
"Operational research is the application of the methods of science to complex problems arising in the direction and management of large systems of man, machines, materials and money in industry, business, government and defence. The distinctive approach is to develop a scientific model of the system, incorporating measurements of factors such as chance and risk, with which to predict and compare the outcomes of alternative decisions, strategies or controls. The purpose is to help management determine its policy and actions scientifically."

Pour parvenir à des résultats optimaux, la RO est amené à simplifier le modèle en réduisant le nombre de variables et de contraintes. Par exemple, les travaux d'ordonnancement de tâches sur une machine sont très nombreux. Ces travaux ne peuvent pas appliqués directement dans la vie pratique.

En tant que cadet, les techniques nouvelles telles que IA tentent de répondre mieux aux besoins de la société contemporaine, avec des modèles plus riches sémantiquement : on tient en compte plus de contraintes et souvent on intègre le niveau de l'individu. Du coup, la notion d'optimalité s'estompe.

Or du fait des préférences individuelles, les problèmes sont souvent *mal-définis :* l'utilisateur ne sait plus reconnaître si une situation donnée est une solution. Dans certains cas on retrouve même des problèmes *malicieux*, leur définition vague, incomplète, contradictoire et changeant souvent.

[21] Par exemple : L. Euler, *"Solution d'une question ingénieuse qui ne paroit soumise à aucune analyse*", Mémoire de l'Académie Royale des Sciences XV, Berlin. pp 310-357 (1759).

tel-00004380, version 1 - 29 Jan 2004

L'état de l'art en planification automatique a été présenté au chapitre 2. Ce survol a couvert de nombreuses méthodes les plus récentes dans la résolution des problèmes d'optimisation combinatoire.

Nos propres travaux de recherche ont mis l'accent sur la Programmation par Contraintes et sur la recherche locale. Réalisée en collaboration avec l'industrie, ces recherches ont un caractère pratique et une finalité précise : celle de l'industrialisation des méthodes scientifiques dans la gestion des ressources humaines à des termes différents. La construction des tours a été le théâtre des recherches présentées aux chapitres 3 et 4.

La partie la plus récente de ces travaux est présentée au chapitre 5. C'est aussi la partie la plus originale et sujet de nombreuses publications.

tel-00004380, version 1 - 29 Jan 2004

6.2 BILAN

Au cours de notre recherche, nous avons appliqué les techniques de la PPC au problème de génération de planning ; tantôt avec des produits du marché, en l'occurrence les contraintes globales de CHIP, tantôt avec nos propres réalisations d'algorithmes de la PPC.

6.2.1 Contributions de ce travail

Cette recherche a abouti à des contributions sur plusieurs plans.

Contributions sur le plan méthodologique

L'approche proposée dans le projet MMN semble être à contre-courant des démarches scientifiques habituelles. A l'école, dès le début de l'analyse d'un problème, on fixerait le cadre de la résolution, i.e. horizon de calcul ainsi que le niveau de détails. Dans le cas de la planification des ressources humaines, nous avons montré que cela fausse déjà la résolution du problème car les contraintes légales existent à plusieurs niveaux (journalier, hebdomadaire, mensuel et annuel). Le planificateur humain doit les prendre en compte simultanément. Agissant à tous les niveaux simultanément, l'énorme combinatoire peut être exploitée pour produire des plannings ayant la préférence des salariés et le grand horizon du système peut être exploité pour générer des plannings équitables.

Nous avons démontré que résoudre le problème de construction de tours en une passe est possible, qu'il y a des énormes combinaisons possibles, quitte à trouver les bonnes relaxations en cas d'impossibilité. Ces espaces de recherche arborescentes peuvent être explorés avec des heuristiques adaptés, mise en évidence dans ce mémoire.

A la différence des travaux [Heu95] et [Par98] qui dépendent des outils PPC du marché, nous nous sommes permis de développer nos propres mécanismes de propagation, afin de mieux exploiter les multiples niveaux, tant la combinatoire est a priori, énorme. Tout le solveur est devenu une sorte de contrainte globale, où toutes les contraintes du système sont traitées ensembles. Les différentes conditions nécessaires que nous avons présentées au chapitre 5 permettent de détecter des inconsistances en amont de l'énumération.

Contributions sur le plan de la recherche

La liste de nos publications est proposée dans la liste des bibliographies par auteur. Au total une dizaine d'articles de recherche publiés lors de conférences nationales et internationales.

En particulier, la méthode de recherche locale que nous avons proposée au chapitre 5, semble très prometteuse. Basée sur des échanges d'affectations pour résoudre des conflits sur les ressources, elle présente une alternative à la recherche à retour-arrière systématique pour résoudre des problèmes de grandes tailles.

tel-00004380, version 1 - 29 Jan 2004

Contributions sur le plan pratique

Ce travail a abouti à la création de *plusieurs* applications opérationnelles. Bâtit sur les bases logicielles de la société COSYTEC, le logiciel Gymnaste a développé la métaphore du « tableur de planification ». Il a été déployé dans plusieurs unités de soins au CHU de Grenoble et dans certains hôpitaux de la région parisienne. Ensuite, j'ai entrepris de séparer le composant règles spécifiques du reste du système, semblable à la démarche des « noyaux » des systèmes experts. Le noyau que j'ai mis en évidence était un logiciel « vertical » pour traiter les problèmes de planning des ressources humaines.

Ce noyau a été réutilisé dans le projet MOSAR dont la partie solveur a été décrit au chapitre 4. Cette application a été utilisée dans huit centres nationaux de l'administration pénitentiaire par des experts d'organisation qui conseillent aux chefs des 190 établissements répartis sur tout le territoire national.

Le produit EQUITIME a rajouté la métaphore de « *le crayon et la gomme* » au « tableur de planification ». Il a été vendu et utilisé dans de nombreux hôpitaux sur la France. Le générateur automatique de planning a été vendu et utilisé chez des clients tels que Fimatex, le groupe Baxter et Nestlé.

6.2.2 Résultats

Les objectifs énoncés au chapitre 1 étaient :

O1. Traitement global : Notre méthode ne découpera pas le processus de planification en étapes de calculs irrévocables. Les différents niveaux d'agrégation temporelle (journalier, mensuel et annuel) doivent travailler simultanément (appliquer et propager les contraintes à leurs niveaux).

Le solveur dans GYMNASTE et EQUITIME génère des plannings en une seule passe. Le modèle MNM fonctionne à tous les niveaux avec les contraintes appropriées.

O2. Tenir compte des caractéristiques individuelles des salariés telles leurs préférences et l'historique de planning. L'objectif est de créer des plannings de grande qualité. Par contre, il ne pourra pas traiter un nombre de salariés semblable. Nous ciblons la centaine de salariés.

Chaque salarié est représenté comme une ressource unique, avec ses propres caractéristiques. C'est le modèle exploité dans GYMNASTE, EQUITIME et MNM.

O3. Notre système inclura une interface permettant aux utilisateurs de proposer des plannings ponctuels.

Nous proposerons des interfaces avancées dans GYMNASTE et EQUITIME, permettant aux non informaticiens de comprendre facilement un planning complexe, et d'y apporter des modifications.

La méthode est constructive et capable de compléter un planning de deux façons : soit incluant des affectations obligatoires de l'utilisateur, soit suite à l'effacement des affectations non désirables. La fonction ANNULER a été très appréciée par nos utilisateurs.

tel-00004380, version 1 - 29 Jan 2004

6.3 PERSPECTIVES

Toute recherche dans un domaine peut et doit ouvrir les champs pour d'autres recherches dans le même domaine ou dans des domaines annexes.

6.3.1 Modèles

Les contraintes pluriannuelles telles que les congés formation (pour changer de compétence), ou le compte épargne temps (CET) ne sont pas encore claire aujourd'hui. Elles pourront être traitées par un niveau supplémentaire.

Ce niveau supplémentaire pourrait aussi traiter la problématique de carrière ou de « turnover » du personnel. L'idée est qu'il faut travailler un temps minimal dans certaines qualifications avant de pouvoir passer à d'autres tâches qui requièrent une qualification supérieure. Le système va devoir traiter des employés qui entrent dans l'établissement et d'autres le quittent.

Nous n'avons pas exploré la possibilité de structurer en plusieurs niveaux l'axe des employés à la manière des niveaux multiples du temps. Ainsi on pourrait créer des ressources représentant une équipe de salariés, voire tout un service. Cela permettrait aux algorithmes de traiter plusieurs salariés en même temps, donc d'augmenter la taille du problème traité. Cependant rien n'est gagné, car les préférences et l'historique des salariés les rendent uniques dans la majorité de situations.

6.3.2 Méthodes

Au cours de nos recherches, une grande partie a été consacrée à l'approche d'élaboration d'algorithmes propres. Cette décision a été influencée par les besoins de la mise sur le marché de logiciels prêts - à - l'emploi. Nous aurions pu analyser les méthodes sous les deux angles ci-dessous.

L'apport des méthodes utilisant des processus ou des machines travaillant en parallèle

L'apport des outils standards du marché

- Utilisation de la PPC pour effectuer le calcul des vacations

  La contrainte globale cumulative est réputée être très performante car elle serait capable de tirer un maximum de déductions à chaque nœud de l'arbre de recherche. Il serait intéressant de voir comment le modèle que nous avons proposé au chapitre 2 se comporte face aux problèmes réels.

- Utilisation de la PLNE pour résoudre le MMN

  Moyennant des simplifications des contraintes à respecter, nous pourrions obtenir un système d'équation illustré par la Figure 6.75, où les lignes représentent des

tel-00004380, version 1 - 29 Jan 2004

ressources et les colonnes représentent des intervalles de temps à différentes échelles. Les rectangles représentent des coefficients non - nuls du système.

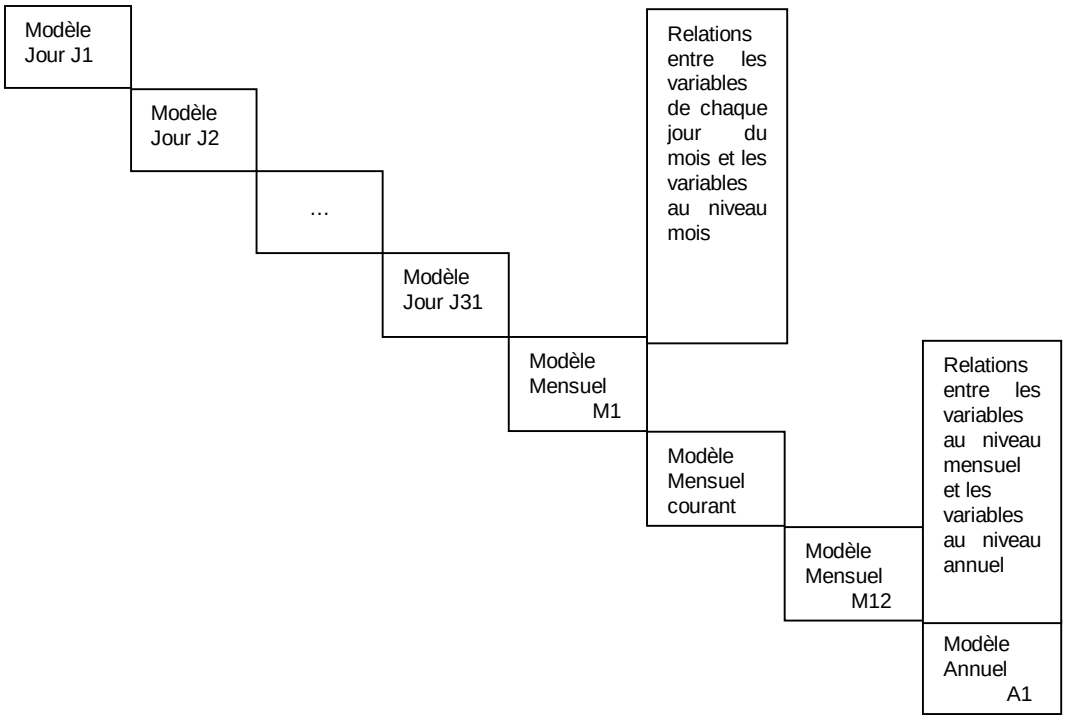


Figure 6.75. Résolution du MMN avec PLNE

tel-00004380, version 1 - 29 Jan 2004

7 ANNEXES

7.1 GLOSSAIRE

1. **Algorithmes de maintenance de consistance :** une méthode de résolution des **Problèmes de Satisfaction des Contraintes**. Diverses sources d'insuffisance de l'algorithme de **Backtrack** a été analysées dans [Mac77] :
   - Si une variable V_i contient des valeurs inconsistantes par rapport à une contrainte un-aire, alors ces valeurs provoqueront des retours arrières systématiques
   - Si une variable V_i contient valeur a incompatible avec toutes les valeurs d'une autre variable V_j, qui est instanciée après V_i, l'algorithme Backtrack essaiera systématiquement les variables V_{j-1}, …, V_{i-1} avant de proposer une autre valeur pour V_i. Cette recherche inutile sera répétée pour toute combinaison de valeurs pour V_1, …, V_{i-1} avec V_i = a.
   - Supposons que V_i = a, V_j = b, et les contraintes un-aire pour V_i et V_j , et la contrainte binaire s'appliquant sur V_i et V_j, soient vérifiées. Il se peut qu'il n'y ait pas de valeur c pour une autre variable V_k, tel que la contrainte un-aire pour V_k et les contraintes binaires s'appliquant sur (V_i et V_k) et (V_j et V_k,) soient vérifiées.

   La solution proposée s'appuie sur la notion de domaine pour chaque variable, l'ensemble de valeurs possibles. Les trois problèmes précédents sont traités par des algorithmes de maintenance de consistance de nœud, d'arc et de chemin, resp.

2. **among** : Une contrainte globale du système CHIP, voir [BC94]. Soit R le nombre total de ressources, N le nombre d'étiquettes distinctes S_1, …., S_N, le jour J, $Zeros_R$ est une liste de R zéros et le paramètre **all** spécifie l'exploitation par cette contrainte de tous les événements de changement de domaine. Il y a plusieurs appels de la contrainte :

   a) among([N_1,…, N_N], [$X_{1,j}$, …, $X_{R,j}$], $Zeros_R$,[[S_1],…,[S_N]], all)

   Cette contrainte spécifie que la variable Nk (pour k = 1,…, N) est le nombre exact d'occurrences de la valeur S_k parmi les variables $X_{r,j}$ représentant les affectations au cours de la période j aux ressources r = 1 à R.

   b) among([Min, Max], [$X_{1,j}$, …, $X_{R,j}$], $Zeros_R$,[V_N], all)

   Min et Max sont deux constantes indiquant les bornes inférieures et supérieures du nombre d'occurrence de la constante V_N, dans la liste de variables X.

   c) among([Min, Max, J], [$X_{R,1}$, …, $X_{R,j}$], $Zeros_J$,[V_N], all)

   Min et Max sont deux constantes indiquant les bornes inférieures et supérieures du nombre d'occurrence de la constante V_N, dans chaque suite de J variables consécutives de la liste X.

3. **Backtrack** : une méthode de résolution des **Problèmes de Satisfaction des Contraintes**.

   a. *Initialiser* : Affecter à chaque variable la valeur non-définie. Les valeurs possibles de chaque variable sont marquées non-utilisées. Initialiser la pile à Vide.
   b. *Sélectionner une variable* : Prendre comme variable courante, une des variables non encore instanciée. Empiler cette variable.

tel-00004380, version 1 - 29 Jan 2004

c. *Sélectionner une valeur* : Sélectionner une valeur non encore utilisée pour la variable courante et la marquer comme utilisée. L'affecter à la variable courante. Si toutes les valeurs sont utilisées, aller à (f).
d. *Vérification* : Vérifier toutes les contraintes dans laquelle figurent cette variable avec les variables déjà instanciées. Si elles sont vérifiées, aller à (b). Sinon aller à (c).
e. *Solution* : Si toutes les variables sont instanciées, alors on a obtenu une solution. Si seule une solution est demandée, alors arrêter avec succès. Sinon aller à (c).
f. *Backtrack* : affecter la valeur non-définie à la variable courante, considérer comme variable courante celle qui est en haut de la pile et dépiler cette dernière. Aller à (c). Si la pile est vide alors arrêter avec échec.

Ce procédé permet d'éviter de générer des valeurs de toutes les variables avant de détecter des transgressions de contraintes avec les variables déjà instanciées, d'où un gain important de performance. Cependant, il souffre d'autres insuffisances liées au mécanisme de retour arrière chronologique. Voir Algorithmes de maintenance de consistance.

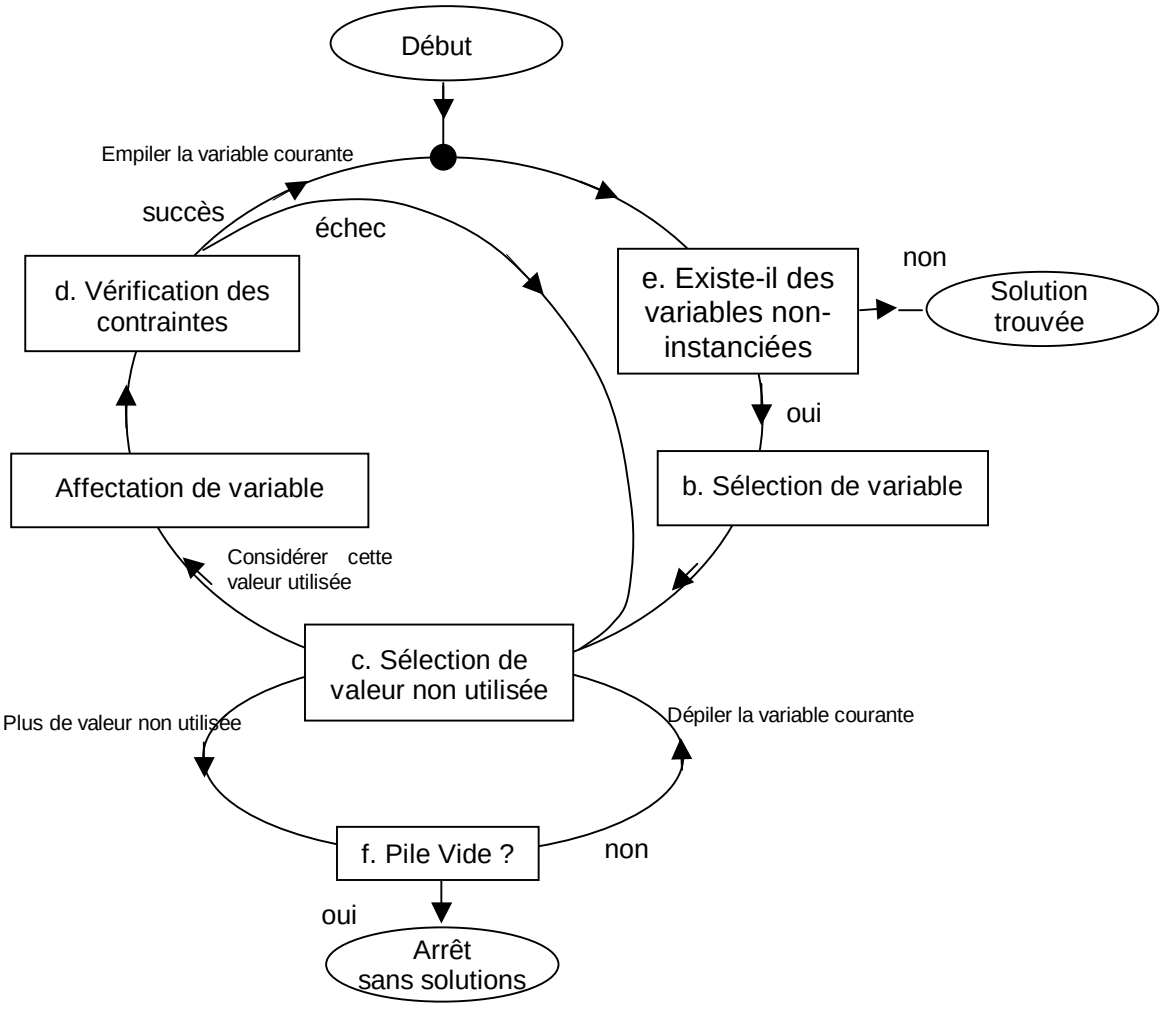


Figure 6.76. Algorithme de Backtrack

A chaque itération dans la boucle supérieure dans le sens de la montre, on sélectionne une variable de décision et explore les diverses valeurs non utilisées. Lorsque aucune transgression de contrainte n'est détectée, on l'empile et passe à l'itération suivante. A chaque itération dans la boucle inférieure dans le sens contraire, on dépile et explore les branches restantes correspondant aux valeurs non utilisées de la variable courante.

tel-00004380, version 1 - 29 Jan 2004

4. **Charge** : un besoin à couvrir par l'ensemble des salariés de l'unité élémentaire de planning.
   Courbe de charge discontinue : les tâches ne dépassent pas 24H en durée.
   Courbe de charge continue : les tâches peuvent dépasser 24H en durée.

5. **Classe de personnel** : ensemble de salariés défini par une durée de travail hebdomadaire et par le type de contrat (temps complet ou partiel)

6. **Code Horaire** : soit un jour d'absence légale (repos hebdomadaire, congés annuels, etc.), soit un ensemble d'horaires de travail pour une journée. Elle ne dit rien sur les tâches que les salariés vont exécuter. Implicitement, un salarié ne peut faire qu'un seul horaire par jour, qu'on désigne par un code préétabli porté sur le planning mensuel. Si les horaires réellement faits différent des horaires prévus, le code devrait être modifié.

7. **Consistance Locale** : Dans un problème de satisfaction de contraintes, avec $Y \subseteq X$. Une instance des variables Y est ***localement consistante*** si et seulement si toutes les contraintes portant sur les variables Y sont satisfaites.

8. **Contrainte** : une relation portant sur des **variables de décision**. Peut être défini par extension (la liste exhaustive de toutes les combinaisons admissibles des valeurs des variables de la contrainte) ou par intension (ex. une expression arithmétique ou logique)

   - **Contrainte dure** : une contrainte dure doit être respectée par tout planning cohérent.
   - **Contrainte souple ou flexible** : contrainte qui n'est pas forcément satisfaite dans un planning ou satisfaite de façon partielle.
   - **Contrainte redondante** : Une contrainte c est redondante par rapport à un ensemble donné de contraintes C lorsqu'elle est une conséquence logique de C.
   - **Contrainte globale** : Une contrainte qui s'applique sur beaucoup de variables, potentiellement dans les milliers. La première contrainte globale était la contrainte **cumulative**. D'autres contraintes sont **among** et **sequence**.

9. **CSP** : voir **Problème de Satisfaction des Contraintes**

10. **cumulative :** Une contrainte globale du système CHIP, voir [AB93]. Soit

N	un nombre non nul de tâches
$[S_1, \ldots, S_n]$	n variables à domaines finis représentant le début des tâches
$[D_1, \ldots, D_n]$	n variables à domaines finis représentant la durée des tâches
$[R_1, \ldots, R_n]$	n variables à domaines finis représentant le nombre de ressources consommées par les tâches
L	le nombre de ressources disponibles et qui doivent être partagées par les tâches

    Si min(V) (resp. max.(V)) est la plus petite (grande) valeur dans le domaine de la variable V, soit a = min($\min(S_1), \ldots, \min(S_n)$), et b = max($\max(S_1), \ldots, \max(S_n)$).

    La contrainte cumulative est respectée lorsque

tel-00004380, version 1 - 29 Jan 2004

$$\forall t \in [a, b] \quad \sum_{\forall j \mid S_j \le t \le S_j + D_j - 1} R_j \quad \le L$$

La contrainte cumulative stipule que, à tout instant t de l'horizon de l'ordonnancement, la somme des ressources consommées par toutes les tâches qui sont actives à cet instant t n'excède pas la limite L.

11. **Cycle de travail** : Le planning des salariés qui suit un cycle est répétitif :about d'un intervalle de temps égale à la période P du cycle, il reprend le même planning.

    - **Cycle de travail hebdomadaire** : est défini par une séquence de P codes hebdomadaires, chaque W_i étant une séquence de **vacations** ou horaires S_i. pour chaque jour de la semaine.
    - **Cycle de travail journalier** : est défini par une séquence de P codes journaliers S_i. où P n'est pas un multiple de 7.

12. **Domaine** d'une variable : l'ensemble fini non vide des valeurs possibles de la **variable**.

13. **Equipe** : groupe de salariés effectuant le même planning au même moment. Voir **unité élémentaire de planning** (UEP).

14. **Etiquette** : (des horaires standards) représente une organisation du travail dit posté. Par exemple : matin, soir et nuit. L'ensemble des étiquettes peut ne pas être disjoints (exemple Jour) afin de couvrir des besoins qui chevauchent deux étiquettes.

15. **Etiquette Repos** : une journée de repos non travaillé. (*Day Off*)

16. **Fail-first Principle** : Ce principe dicte qu'afin de réussir, d'abord essayer là on a le plus de chances d'échec. La motivation est que moins l'arbre de recherche possède de branches à sa racine, moins il y a de nœuds et plus rapide est son parcours. Voir [HE80].

17. **Fenêtre de pause** : intervalle durant lequel une pause peut avoir lieu.

18. **Fenêtre d'heure de début** : intervalle durant lequel les vacations peuvent commencer.

19. **Forward Checking** : voir **Lookahead**.

20. **Générer et Tester** : Chaque combinaison de valeurs pour toutes les variables est générée systématiquement et *ensuite* testée pour vérifier si les contraintes sont vérifiés. La complexité de cette méthodes est $\prod_{i=1}^{n} \left| D(x_i) \right|$.

21. **Grille de travail** (*Roster*) : planning nominatif ou non, avec des **étiquettes**, défini sur un nombre entier de semaines.

22. **Lookahead :** une méthode de résolution des **Problèmes de Satisfaction des Contraintes.** Suite à toute instanciation de variable, les domaines des variables non-

tel-00004380, version 1 - 29 Jan 2004

encore instanciées sont vérifiées par rapport aux contraintes en place. Les valeurs inconsistantes sont enlevées. Si le domaine d'une variable ne contient plus qu'une seule valeur, la variable est affectée à cette valeur. Par contre si le domaine devient vide, le retour arrière est provoquée.

- Forward Checking : pour chaque variable non encore instanciée, cet algorithme assure qu'il y a au moins une valeur consistante par rapport aux valeurs courantes des variables déjà instanciées.

23. **Pause** : l'ensemble de périodes non travaillées au cours de la journée. Exemple : pause repas ou pause café. Elle est soit payée ou pas.

24. **Période** : le plus petit intervalle de temps pour lequel la charge est définie dans un modèle donné. Dans ce document, nous utiliserons le terme **intervalle** au lieu de période, car il y a confusion potentielle avec la période d'un cycle de travail.

25. **Planification d'horaires** : sur un horizon d'un jour à quelques mois, donner l'utilisation des ressources de façon à couvrir un besoin exprimé par une charge de travaille prévisionnelle, tout en respectant des contraintes précises. Elle inclut la planification des repos des salariés.

26. **Planning** : un emploi du temps pour les salariés, suivant des niveaux d'agrégation du temps différents. On distingue
    - **Planning journalier** : horizon d'un jour, la durée de l'intervalle varie entre 10 minutes et 2 heures. Construction de vacations. (*shift scheduling*).
    - **Planning mensuel** :horizon d'un mois, l'intervalle est typiquement un jour. Construction de **tours**. (*tour scheduling*)
    - **Planning annuel** : horizon d'un an, l'intervalle peut être le jour ou la semaine.

    Le planning dit continu ou discontinu :
    - **Planning continu** : les besoins ou les durées des activités durent plus de 24H
    - **Planning discontinu** : dans ce cas, on peut créer le planning jour par jour

    Le planning peut être cyclique ou acyclique :
    - **Planning cyclique** : un planning est cyclique si au bout d'une durée P, tout salarié retrouve son planning initial
    - **Planning acyclique** : un planning est acyclique s'il est différent pour un salarié chaque jour chaque semaine et chaque mois.

    Le planning peut être réalisé de façon nominative, individuel ou anonyme
    - **Planning nominative** : le planning est fait pour une personne donnée
    - **Planning individuel** : sans être nominative, le planning peut être fait pour une personne ayant un certain nombre de caractéristiques individuelles, p.ex. préférence.
    - **Planning anonyme** : le planning réalisé pour des groupes de personnes ayant des caractéristiques communes, ex. qualification.

    Un planning peut aussi être de nature différente, au cours de son cycle de vie
    - **Planning théorique** : suite à une génération de **planning cyclique**
    - **Planning théorique ajusté** : ajustement pour congés, formation, etc.

tel-00004380, version 1 - 29 Jan 2004

- **Planning prévisionnel** : Ce planning est communiqué aux salariés mais il peut y avoir un ou plusieurs révisions.
- **Planning réalisé brut** : les présences sont relevées des pointeuses
- **Planning réalisé net** : les relevées sont corrigées suivant la politique de l'entreprise.
- **Planning validé** : les heures sont validées suivant la démarche de l'entreprise. Ce planning sera payé, avec des bonus pour des modifications effectuées par rapport au planning prévisionnel, sans le délai légal de prévenance.

27. **Population** : l'ensemble des salariés susceptibles de couvrir la charge. On distingue la population suivant la classe de personnel et suivant les compétences.

28. **Problème de Satisfaction des Contraintes (PSC) :** est défini par
    - un ensemble de *n* ***variables*** $X = \{x_1, ..., x_n\}$,
    - un ensemble de ***domaines*** courants $D = \{D(x_1), ..., D(x_n)\}$ où $D(x_i)$ est un ensemble fini ou non de ***valeurs*** possibles pour la variable x_i, et
    - un ensemble *C* de *m* ***contraintes*** entre ces variables.

La complexité théorique ou la taille de l'espace de recherche de solutions d'un CSP est égal à $\prod_{i=1}^{n} |D(x_i)|$. Si chaque variable a un domaine comportant *d* valeurs ; la complexité théorique est égale à d^n. Un PSC peut être résolu avec les méthodes suivantes : **Générer et Tester**, **Backtrack**, **Lookahead**.

29. **Problème d'affectation généralisée** (generalized assignment problem) : est défini par
    - Un ensemble de tâches i = 1, …, n
    - Un ensemble d'agents j = 1, …, m dont la capacité disponible maximale est b_j
    - Chaque tâche i doit être affectée à un et un seul agent j : la capacité d'agent j consommée est a_{ij} et le coût est c_{ij}
    - Variables de décision X_{ij} = 1 si la tâche i est affectée à l'agent j, sinon elle vaut 0
    - Nous admettons que : $A_{ij} \leq B_j$ et $\sum_{i=1}^{n} A_{ij} > B_j$
    - Fonction à minimiser : f(x) = $\sum_{j=1}^{m}\sum_{i=1}^{n} C_{ij} X_{ij}$
    - Contrainte de capacité : $\sum_{i=1}^{n} A_{ij} X_{i\ j} \leq B_j$ pour j=1, …, m
    - Contrainte d'unicité de l'affectation : $\sum_{j=1}^{m} X_{ij} \equiv 1$, pour i=1, …, n

30. **Problèmes**
    - **Problèmes bien-définis** (well-defined problems): Le problème et la solution sont clairement expliqués : on reconnaît sans ambiguïté une solution au problème. Un problème bien-défini peut être combinatoire et intraitable, dont la solution exige un temps de calculs et un espace mémoire plus grands que ce qu'on dispose.

tel-00004380, version 1 - 29 Jan 2004

Le glossaire

- **Problèmes mal-définis** (ill-defined problems): Le problème et/ou la solutions ne sont pas clairement expliqués : on ne peut pas reconnaître une solution. Souvent les solutions trouvées peuvent être améliorées et c'est au solveur de décider quand s'arrêter. Des problèmes combinatoires intraitables exigeant des solutions heuristiques ou approchées sont une classe de problèmes mal-définis. On résout un problème mal-défini avec une approche créative.

- **Problèmes malicieux** (Wicked problems) : Ce problème est mal-défini et pire, la définition du problème est vague, incomplète, contradictoire et souvent changeant. Une solution est très difficile à reconnaître. On résout un problème malicieux avec un groupe de travail.

31. **Programmation par contraintes PPC** : méthode de résolution des problèmes de satisfaction des contraintes intégrant la méthode **Backtrack** et des algorithmes de consistance (ou cohérence) des contraintes.

32. **Propagation par contraintes** : voir **Lookahead**.

33. **Programmation Linéaire en Nombres Entiers** PLNE : méthodes de résolution d'un système d'équations linéaires.

34. **sequence** : une contrainte globale du système CHIP, voir [BC94]. Soit R le nombre total de ressources, T étant le nombre total de périodes, $\texttt{Zeros}_R$ est une liste de R zéros et le paramètre **all** spécifie l'exploitation par cette contrainte de tous les événements de changement de domaine.

    $\texttt{sequence([0,0,T],[}X_{i,1}\texttt{, …, }X_{i,T}\texttt{], Zeros}_T\texttt{, Pats, all)}$

    Les variables X représentent les affectations de la ressource i pendant les périodes 1 à T. Pats est la liste de patterns valides sur les variables V. Le premier argument exprime le fait qu'il y ait exactement 0 instance de ces patterns sur les T variables. Par ex. pour contraindre que la valeur W doit être suivie d'une valeur quelconque dans la liste AllowedCodes, le pattern est la liste de 2 listes `[[sum,1,#=,[W]],[sum,1,#\=,AllowedCodes]]`. Le premier sous liste indique que la somme d'une variable est égale à la valeur W et le deuxième spécifie que la somme d'une variable n'est pas égale aux valeurs indiquées.

35. **Qualification** : la capacité d'un salarié à travailler sur une tâche qui requiert des connaissances sanctionnées par un diplôme reconnu.

36. **Séquence de travail** : (*Work stretch*) enchaînement de vacations successives délimité par des repos.

37. **Séquence de repos** : (*Off Stretch*) enchaînement de repos successif délimité par des vacations.

38. **Solveur** : un programme de résolution d'un problème. Dans ce mémoire, il s'agit du problème de planification.

tel-00004380, version 1 - 29 Jan 2004

39. **Système informatique décisionnelle** (Business Intelligence) : Système interprétant des données complexes permettant aux dirigeants d'entreprise de prendre des décisions en connaissance de cause. Les données sont analysées selon plusieurs dimensions (type de produits, régions et saisons par exemple) [source : JDNETsolutions].

40. **Système informatisé d'aide à la décision** (Decision Support System) : Système informatique intégré, conçu spécialement pour la prise de décision, et qui est destiné plus particulièrement aux dirigeants d'entreprise. Le système d'aide à la décision est un des éléments du système d'information de gestion. Il se distingue du système d'information pour dirigeants, dans la mesure où sa fonction première est de fournir non seulement l'information, mais les outils d'analyse nécessaires à la prise de décision. Ainsi, il est habituellement constitué de programmes, d'une ou de plusieurs bases de données, internes ou externes, et d'une base de connaissances. Il fonctionne avec un langage et un programme de modélisation qui permettent aux dirigeants d'étudier différentes hypothèses en matière de planification et d'en évaluer les conséquences [source : JDNETsolutions].

41. **Tour** : enchaînement de vacations et de repos sur un horizon défini, généralement une semaine ou un mois.

42. **Unité élémentaire de planning :** Un groupe de personnes pour qui on établit un planning ; chaque UEP dispose d'un ensemble de besoins par intervalle de temps. Différent de l'**équipe**.

43. **Vacation** : (*Shift*) une journée de travail, défini par des horaires de début et de fin, précisant les pauses. Typiquement utilisée pour désigner une affectation journalière (un salarié, un jour donné). Voir aussi le terme **étiquette**.

    - **Vacation directe** (*straight shift*) : en une seule partie travaillée.
    - **Vacation avec coupure** : (*split shift*) en plusieurs parties travaillées

44. **Variables**
    - **Variable de décision** : pouvant prendre une valeur prise dans le **domaine** de la variable.

Remerciement : Certains termes du glossaire proviennent de la thèse [Par98].

tel-00004380, version 1 - 29 Jan 2004

7.2 BIBLIOGRAPHIE

[AB93] Aggoun A. et Beldiceanu N., *Extending CHIP in order to Solve Complex Scheduling Problems*, J. Mathematical and Computer Modelling, 17 (7): 57-73, 1993.

[Aic99] Aickelin U., *Genetic Algorithms for Multiple Choice Optimisation Problems, Application to Nurse Scheduling,* Thesis at University of Wales Swansea, Sept 1999.

[AR81] Arthur J.L. et Ravidran A. *A multiple objective nurse scheduling model*. AIIE Transactions, 13 (1): 55-60, 1981.

[Ayk96] Aykin T., *Optimal Shift Scheduling with Multiple Break Windows,* Management Science 42 (4): 591-602, 1996.

[Bak74] Baker K.R. *Scheduling a full-time workforce to meet cyclic staffing requirements,* Management Science 20: 1561-1568, 1974.

[Bak76] Baker K.R., *Workforce allocation in Cyclical Scheduling Problems : A Survey.* Operational Research Quarterly. 27 (1): 155-167, 1976.

[BB92] Burgess W.J. et Busby R.E., "Personnel Scheduling" in *Handbook of Industrial Engineering*, 2154-2169, Wiley & Sons, New York, 1992.

[BC85] Burns R.N. et Carter M.W., *Work Force Size and Single Shift Schedules with Variable Demands,* Management Science 31(5) : 599-607, 1985.

[BBC+96] Beldiceanu N., Bourreau E., **Chan P.** et Divreau D., *Solving Resource-constrained Project Scheduling Problems in CHIP*. Proceedings of the Fifth International Workshop on Project Management et Scheduling, PMS'96, Poznan, Poland, 1996.

[BBC+97] Beldiceanu N., Bourreau E., **Chan P.** et Divreau D., *Partial Search Strategy in CHIP*. Proceedings of the Second International Conference on Metaheuristics MIC'97, Sophia-Antipolis, France, 1997.

[BC94] Beldiceanu N. et Contejean E., *Introducing Global Constraints in CHIP*. Journal of Mathematical and Computer Modelling, 20(12): 97-123, 1994.

[BSK+97a] Beldiceanu N., Simonis H., Kay P. et **Chan P.**, *Application Development with the CHIP System*. COSYTEC White paper available at www.cosytec.com, 1997.

[BSK+97b] Beldiceanu N., Simonis H., Kay P. et **Chan P.**, *The CHIP System*. COSYTEC White paper available at www.cosytec.com, 1997.

[BC96] Beasley J.E. et Chu P.C., *A Genetic Algorithm for the set covering problem,* European Journal of Operational Research, 94: 392- 404, 1996.

[Bee66] Stafford Beer, *Decision & Control, The meaning of Operational Research and Management Cybernetics*, Ed. John Wiley ISBN 0471 06210 3, 8th edition 1988.

[BE81] Bar-Yehuda R. et Even S., *A Linear-Time Approximation Algorithm for the Weighted Vertex Cover Problem,* Journal of Algorithms, pp. 198-203, 1981.

tel-00004380, version 1 - 29 Jan 2004

[BJ90] Bechtold S.E. et Jacobs L.W., *Implicit Modelling of Flexible Break Assignments in Optimal Shift Scheduling,* Management Science, 36(11): 1339-1351, 1990.

[BJ93] Brusco M.J. et Jacobs L.W., *A Simulated annealing approach to the cyclic staff-scheduling problem,* Naval Research Logistics, 40: 69-84, 1993.

[BJ96] Bechtold S.E. et Jacobs L.W., *Naval Research Logistics,* 43: 233-249, 1996.

[BJ00] Brusco M.J. et Jacobs L.W., *Optimal Models for Meal-Break and Start-Time Flexibility in Continuous Tour Scheduling* Management Science 46(12) : 1630-1641, 2000.

[BM77] Baker K.R. et Magazine M., *Workforce Scheduling with Cyclic demands and Day-off Constraints,* Management Science, 24 (2): 161-167, 1977.

[BP73] Byrne J.L. et Potts R.B. *Scheduling of toll collectors*. Transportation Sciences, 7: 224-245, 1973.

[BR78] Bartholdi J.J. et Ratliff H.D., *Un-networks with applications to idle time scheduling,* Management Science 24 (8): 850-858: 1978

[BS93] Brusco M.J. et Showalter M.J. *Constraint nurse staffing analysis*. Omega, 21 (2): 175-186, 1993.

[BW90] Balakrishnan N. et Wong R.T., *A network model for the rotating workforce scheduling problem*, Networks 20: 25-42, 1990.

[CFW02] **Chan P**., Fallet V. et Weil G., *Solving large-scale employee scheduling problems with Multiple-Level Models*, The Twelfth International Symposium on Combinatorial Optimisation CO'02, Paris, France. http://www.lip6.fr/CO2002. Book of abstracts pg. 33. 2002.

[CGL95] Caseau Y., Guillo P.Y. et Levenez E., *A Deductive and Object-Oriented Approach to a Complex Scheduling Problem,* Journal of Intelligent Information Systems, 4, pp. 149-166. 1995.

[CGL01] Cezik T., Günlük O. et Luss H. *An integer programming model for the weekly tour scheduling problem*. Naval Research Logistics Vol.48 (7): 607-624 2001.

[Che91] Chew K.L., Cyclic schedule for apron services. Journal of the Operational Research Society, 42: 1061-1069, 1991.

[Cho94] Choueiry B.Y. Abstraction Methods for Resource Allocation. Thèse à l'Ecole Polytechnique Fédérale de Lausanne, soutenue en 1994.

[CHW98] **Chan P**., Heus K. et Weil G. *Nurse Scheduling with Global Constraints in CHIP: GYMNASTE*, Proceedings of the Fourth International Conference on Practical Applications of Constraint Technology PACT'98, London, pp. 157-169,1998.

[Chv79] Chvàtal V., *A Greedy Heuristic for the set covering problem.* Mathematical Operational Research , 4 (3) : 233-235, 1979.

[CJW02] **Chan P.**, Joseph R. et Weil G., Planification d'horaires du personnel à multiples niveaux, Acte du Quatrième Congrès de la société française de

tel-00004380, version 1 - 29 Jan 2004

recherche opérationnelle et d'aide à la décision, ROADEF 2002, pg. 79-80, 2002.

[CL00] Cai X. et Li K.N., *A Genetic Algorithm for scheduling staff of mixed skills under multi-criteria,* European Journal of Operational Research, 125: 359-369, 2000.

[Cos94] Costa, D. A tabu search algorithm for computing an operational timetable. European Journal of Operational Research, 76, 98-110.

[CST00] Chiarandini M., Schaerf A. et Tiozzo F. *Solving Employee Timetabling Problems with Flexible Workload using Tabu Search.* Proceedings of the 3rd Int. Conf. on the Practice and Theory of Automated Timetabling (E. Burke & W. Erben, Eds.) pp. 298-302, 2000.

[CW00] **Chan P**. et Weil G., *Cyclical Staff Scheduling Using Constraint Logic Programming*, Proceedings of the 3rd Int. Conf. on the Practice and Theory of Automated Timetabling (Eds. E. Burke & W. Erben) Lecture Notes in Computer Science 2079, Pg. 159-175, 2000.

[CW02] **Chan P**. et Weil G., *Using Multiple-level models to solve large-scale employee scheduling*, 15 th European Conf. on Artificial Intelligence ECAI 2002, Workshop Modelling and Solving Problems with Constraints, organised by Toby Walsh, pg. 12-24, 2002. http://4c.ucc.ie/web/index.jsp

[Dan54] Dantzig G.B., *A Comment on Edie's Traffic Delays at Toll Booths,* Operational Research, 2 (3): 339-341, 1954.

[Daw76] Dawkins R. *The Selfish Gene*. Oxford University Press, 1976.

[DHS+88] Dincbas M., Van Hentenryck P., Simonis H., Aggoun A., Graf T. et Berthier F. *The Constraint Handling Language CHIP*, Proceedings of International Conference on Fifth Generation Computer Systems FGC'88, pg. 693-702, Tokyo, 1988.

[Eib01] Eiben A.E., *Evolutionary algorithms and constraint satisfaction: Definitions, survey, methodology, and research directions*, In L. Kallel, B. Naudts, and A. Rogers, editors, Theoretical Aspects of Evolutionary Computing, Natural Computing series, pages 13-58. Springer, 2001.

[EB91] Emmons H. et Burns R.N., *Off-Day Scheduling with hierarchical worker categories,* Operations Research, 39 (3), 1991.

[FR95] Feo T.A. et Resende M., *Greedy Randomized Adaptive Search Procedures*, Journal of Global Optimization, 6: 109-133, 1995.

[GK98] Gomes C. et Kautz, H. *Boosting combinatorial search through randomization*. Proceedings of AAAI-98, Madison, WI, pages 281-331, July 1998.

[Glo89] Glover F., *Tabu Search* Part I, ORSA J. Computing 1 (1989), 190-206.

[GL93] Glover F. et Laguna M., *Tabu search*. In Reeves, C. R. (Ed.), Modern Heuristic Techniques for Combinatorial Problems. Scientific Publications, Oxford, 1993

tel-00004380, version 1 - 29 Jan 2004

[GW97] Grossman T. et Wool A., *Computational experience with approximation algorithms for the set covering problem,* European Journal of Operational Research 101 : 81-92, 1997.

[HB76] Henderson W.B. et Berry W.L. Heuristic methods for telephone operator shift scheduling: An experimental analysis. Management Science 22: 1372-1380, 1976

[HE80] Haralick R.M. et Elliot G.L. *Increasing Tree Search Efficiency for Constraint Satisfaction Problems*. Artificial Intelligence 14 :263-313, 1980.

[Hen89] Van Hentenryck P., *Constraint Satisfaction in Logic Programming*, MIT Press, 1989

[Heu96] Heus K., *Gestion des plannings infirmiers: Application des techniques de programmation par contraintes*. Thèse de Docteur de l'Université J. Fourier, 1996.

[HH86] Hall N. G. et Hochbaum D. S. *A fast approximation algorithm for the multi-covering problem*. Discrete Applications of Mathematics, 15:35-40, 1986.

[Hoc82] Hochbaum D.S., *Approximation algorithm for the weighted set covering and vertex cover problems,* SIAM Journal of Computing, 11(3): 256-278, 1982.

[Hol76] Holland J., *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press 1976.

[HW96] Heus K. et Weil G., "*Nurse Scheduling with Constraint Programming*", Proceedings of the Second International Conference Practical Applications of Constraint Technology PACT'96. pp. 115-127, 1996.

[Hun94] Hung R., *Multiple-shift workforce scheduling under the 3-4 workweek with different weekday and weekend labour requirements,* Management Science 40(2) 280-284, 1994.

[JBS94] Jarrah A.I.Z., Bard J.F. et de Silva A.H., *Solving large-scale tour scheduling problems*. Management Science, 40(9) : 1124-1144, 1994.

[JMP98] Jaquet-Lagrèze E., Montaut D. et Partouche A., 1998, *The Shift Scheduling problem : Different formulations et solution methods*. Foundations of Computing and Decision Sciences. Egalement : Cahier du Lamsade No. 146, Juillet 1997, Université Paris-Dauphine.

[JSV 98] Jaumard B., Semet F. et Vovor T., *A generalised linear programming model for nurse scheduling*, European Journal of Operational Research, vol. 107, 1998, p. 1-18.

[Joh74] Johnson D.S., *Approximation Algorithms for Combinatorial Problems*, Journal of Computer and Systems Sciences, 9: 256 – 278 , 1974.

[JM88] Jaquet-Lagrèze E. et Meziani. *Linear Programming and interactivity : a manpower scheduling DSS*. IFORS 88, volume Operational Research pg. 176-189, 1988.

[Kan 01] Kane H., *Etude de l'ajustement de la capacité à la charge pour une gestion quantitative des ressources humaines en production*, Thèse de Doctorat, Productique, Juillet 2001, INSA de Lyon, 179 p.

tel-00004380, version 1 - 29 Jan 2004

[KB 01] Kane H., Baptiste P., *Adjustment of the capacity to the loading rate: a dynamic model to analysis the workforce costs*, Industrial Engineering and Production Management, IEPM 2001, Quebec, in proceedings, vol.1, August 2001, p. 169-177.

[Kei79] Keith E., *Operator Scheduling*. AIIE Transactions, 11: 37-41, 1979.

[Kla73] Klasskin P.M., *Operating to Schedule Operators,* Telephony**,** July 20, 1973.

[Kra96] Kragelund L. V. *Solving a timetable problem using hybrid genetic algorithms*. Software - Practice and Experience, 27(10):1121–1134, 1997.

[KW92] Le Kang et White G. M.. *A logic approach to the resolution of constraints in timetabling*. European Journal of Operational Research, 61:100-112, 1992.

[LNB80] Laporte G., Norbert Y. et Biron J., *Rotating Schedules***,** European Journal of Operational Research, 4: 24-30, 1980.

[Lap99] Laporte G., *The art and science of designing rotating schedules*, Journal of the Operational Research Society, 50: 1011-1017, 1999.

[Lau98] Lau T.L., *Guided Genetic Algorithms*, Thesis at the University of Essex, United Kingdom, 1998.

[LP89] Lévine P. et Pomerol J.C., *Systèmes interactifs d'aide à la décision et systèmes experts,* Editions Hermès, 1989. ISBN 2-86601-188-0.

[Mac77] Mackworth A., Consistency in Networks of Relations, Artificial Intelligence 8: 99-118, 1977.

[MRT53] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A. et Teller E., *Equation of State Calculations by Fast Computing Machines*, Journal of Chemical Physics Vol. 21 : 0187 – 1092. 1953.

[MGS96] Meisels A., Gudes E., and Solotorevsky G., *Employee Timetabling, Constraint Networks and Knowledge-Based Rules: A Mixed Approach*, Proceedings of the 1st Int. Conf. on the Practice and Theory of Automated Timetabling (Burke E. and Ross P., eds.), LNCS, vol. 1153, Springer Verlag, pp. 93-105, 1996.

[MGS97] Meisels A., Gudes E. et Solotorevsky G., *Combining Rules and Constraints for Employee Timetabling*, Intelligent Systems, Vol. 12, No. 6 pp. 419-439, 1997.

[MK01] Meisels A. et Kaplansky E., *Iterative Restart Techniques for solving Employee Timetabling Problems*, Session on Timetabling, EURO2001, Rotterdam, Aug. 2001.

[Nor90] Norman, D.A. *The Design of Everyday Things***.** Ed. Doubleday, NY. 1990.

This book discusses the use of conceptual models and metaphor in the design of objects.

[MN91] Moscato P. et Norman M. G.. A 'mimetic' approach for the travelling salesman problem - implementation of computational ecology for combinatorial optimisation on message-passing systems. In *Proceedings of the International Conference on Parallel Computing and Transputer Application*s. IOS Press (Amsterdam), 1991.

tel-00004380, version 1 - 29 Jan 2004

[Moo76] Moondra S.L., *An LP model for work force scheduling in banks,* Journal of Bank Resources, Vol. 6(4) 299-301, 1976.

[Mus00] Muslija N., Gärtner J. et Slany W., *Efficient Generation of Rotating Workforce Schedules.* Proceedings of the Third International Conference on the Practice And Theory of Automated Timetabling, 2000.

[NB92] Nanda R. et Browner J., *Introduction to Employee Scheduling*. Editor Van Nostrand Reinhold, New York, 1992.

This book is mostly concerned with the problem of designing work cycles which will minimize workforce requirements and fulfill various constraints. The constraints treated here include the required number of days off, the maximum number of consecutive work-days, the number of weekend offs, etc. Heuristics for achieving good solutions are described and some are proven to be optimal. An essential part of the OR approach is the optimality requirement and the price for optimality is the avoidance of either complex constraints, or large sets of simple constraints.

[Nar00] Narasimhan R., *An algorithm for multiple shift scheduling of hierarchical workforce on four-day or three-day workweeks,* INFOR 38(1): 14-32, 2000.

[Par98] Partouche A., Thèse Planification d'horaires de Travail, Méthodologie, modélisation et résolution à l'aide de la PLNE et de la PPC, soutenu en 1998.

[Pom92] Pomerol J.C., *Aide à la décision et IA*, Intelligence Artificielle une Discipline et un Carrefour Inter-disciplinaire, AFIA-92 , pp. 147-149. 1992.

[Rot00] Rottembourg B., *Une Heuristique à base de dualité lagrangienne pour la planification de centres d'appels téléphoniques*, ROADEF2000.

[PSW97] Peleg D., Schechtman G., et Wool A., *Randomized Approximation of Bounded Multicover Problems*, Algorithmica 18(1): 44-66, 1997.

[RR01] Resende M.G.C. et Riberiro C.C., Greedy Randomized Adaptive Search Procedures, State-of-the-Art Handbook in Meta Heuristics, F. Glover and Kochenberger, eds., Kluwer Academic Publishers. Manuscript dated 2001.

[RW84] Rittel H.J. et Webber M.M., *Planning problems are Wicked Problems*, in N. Cross (ed.), Developments in Design Methodology, John Wiley & Sons, New York, pp. 135-144, 1984.

[SC95] Simonis H. et Cornelissens T., Modelling Producer/Consumer Constraints, Proc. Principles and Practice of Constraint Programming, Cassis France, Sept. 1995. and COSYTEC Internal Note : News on Producers and Consumers, 1997.

[Seg74] Segal M., *The Operator-Scheduling problem: a network flow approach,* Operations Research Vol. 22 No. 4. pp. 808-824. 1974.

[Smi01] Smith B. *Reducing Symmetry in a Combinatorial Design Problem*. Proceedings of CP-AI-OR 2001 pp. 351-359. Wye College, UK, 2001.

[SS95] Schaerf A. et Schaerf M. *Local search techniques for high school timetabling.* In Proc. of the 1st Intl. Conf. on the Practice and Theory of Automated Timetabling, pp. 313-323,1995.

tel-00004380, version 1 - 29 Jan 2004

[Tho88] Thompson G.M. *Representing employee requirements in labour tour scheduling*, Omega 21:657-671, 1988.

[Tho93] Thompson G., *Accounting for the multi-period impact of service when determining employee requirements for labour scheduling*, Journal of Operations Management 11:69-287, 1993.

[Tho95] Thompson G., *Improved Implicit Optimal Modelling of the Labour Shift Scheduling problem,* Management Science**,** 41 (4) 595-607, 1995.

[Tré76] Trémolières R., *Le Problème des roulements de quarts pour les entreprises à feu continu*, Revue Française d'Automatique et Recherche Opérationnelle. 10 (2) :1-101. 1976

[Tsa93] Tsang E., *Foundations of Constraint Satisfaction*. Academic Press, 1993. ISBN 0-12-701610-4

[WH95] Weil G. et Heus K. *Eliminating interchangeable values in the nurse scheduling problem formulated as a constraint satisfaction problem*. In Paper presented at the workshop CONSTRAINT'95, 1995.

[WHC+98] Weil G., Heus K., **Chan P.** et François P. : *The Nurse Scheduling Problem: a combinatorial problem, solved by the combination of constraint programming and real user's heuristics*. In Medinfo'98. B. Cesnik et al. (Eds.) IOS Press ; Amsterdam. pp 508-512, 1998.

[WHP+95] Weil G., Heus K., Puget F. et Poujade M. *Solving the nurse scheduling problem using constraint programming*. IEEE Engineering in Medicine and Biology, July-August 1995.

[WHP94] Weil G., Heus K. et Puget F., *Gymnaste: Aide à l'Elaboration des Roulements Infirmiers*. Du Traitement des Absences aux Management Participatif – Eds. J. Demongeot, P. Le Beux et G. Weil, Springer Verlag, Paris, *Informatique et Santé*, 7, pp. 147-160, 1994.

[Wil91] William H.P., *Model Building in Mathematical Programming,* Ed. John Wiley & Sons. Copyright 1991. ISBN 0 471 92581 0 (paperback).

[Wil98] Williamson D.P. *Lecture notes on Approximation Algorithms*. Cours de Recherche Opérationnelle et Ingénierie Industrielle. Cornell University.

[WP72] Warner D.M. et Prawda J. *A mathematical programming model for scheduling nursing personnel in a hospital*. Management Science, 19(4):411-422, 1972

tel-00004380, version 1 - 29 Jan 2004

7.3 INDEX DES REFERENCES PAR AUTEUR

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

7.4 LISTE DES PUBLICATIONS PAR AUTEUR

tel-00004380, version 1 - 29 Jan 2004

tel-00004380, version 1 - 29 Jan 2004

Les publications par auteur

Metropolis N.	[MRT53]
Meziani	[JM88]
Montaut D.	[JMP98]
Moondra S.L.	[Moo76]
Moscato P.	[MN91]
Muslija N.	[Mus00]
Nanda R.	[NB92]
Narasimhan R.	[Nar00]
Norbert Y.	[LNB80]
Norman M.G.	[MN91]
Norman D.A.	[Nor90]
Partouche A.	[JMP98], [Par98]
Peleg D.	[PSW97]
Pomerol J.C.	[LP89], [Pom92]
Potts R.B.	[BP73]
Poujade M.	[WHP+95]
Prawda J.	[WP72]
Puget F.	[WHP+95]
Puget F.	[WHP94]
Ratliff H.D.	[BR78]
Ravidran A.	[AR81]
Resende M.G.C.	[FR95], [RR01]
Riberiro C.C.	[RR01]
Rittel H.J.	[RW84]
Rosenbluth A.	[MRT53]
Rosenbluth M.	[MRT53]
Rottembourg B.	[Rot00]
Schaerf A.	[CST00], [SS95]
Schaerf M.	[SS95]
Schechtman G.	[PSW97]
Segal M.	[Seg74]
Showalter M.J.	[BS93]
Simonis H.	[DHS+88], [SC95], [BSK+97a], [BSK+97b]
Slany W.	[Mus00]
Smith B.	[Smi01]
Solotorevsky G.	[MGS96], [MGS97]
Stafford Beer	[Bee66]
Thompson G.	[Tho88], [Tho93], [Tho95]
Tiozzo F.	[CST00]
Trémolières R.	[Tré76]
Tsang E.	[Tsa93]
Van Hentenryck P.	[DHS+88], [Hen89]
Warner D.M.	[WP72]
Webber M.M.	[RW84]
Weil G.	[CHW98], [CJW02], [CW00], [CFW02], [CW02], [HW96], [WH95], [WHC+98], [WHP+95], [WHP94]
White G. M.	[KW92]
William H.P.	[Wil91]
Williamson D.P.	[Wil98]
Wong R.T.	[BW90]
Wool A.	[GW97], [PSW97]

THÈSE

Présentée par

CHAN Yew Cheong, Peter

Pour obtenir le titre de
DOCTEUR de l'UNIVERSITE
JOSEPH FOURIER – Grenoble 1

Spécialité : Informatique
Formation doctorale : Recherche Opérationnelle

La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes

Date de soutenance : Octobre 2002

Composition du Jury :

Directeur de thèse:	**M. Georges Weil**
Rapporteurs :	**M. Abdelhakim Artiba**
	M. Alain Guinet
Examinateurs:	**M. Jean Charles Pomerol**
	M. Gerd Finké
	M. Eric Jacquet-Lagrèze
	M. Jacques Demongeot

Thèse préparée au sein du Laboratoire TIMC – Institut IMAG

tel-00004380, version 1 - 29 Jan 2004