

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 de 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 consistance¹⁵.

¹⁵ forward checking et look-ahead [Mac77]

Le calcul des tours en PPC

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 solveur 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.

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 :

- o 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 vacances admises pour cet agent ce jour.

Format :

- o [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

Le calcul des tours en PPC

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 :

- o [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 :

- o 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.

Format simple :

- o 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 :

- o [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 :

- o L'acteur concerné
- o La période concernée (haute saison/promotions/période estivale, etc.)
- o Le contexte (les délais de notification, etc.)
- o Le passé (questions d'équité)
- o 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.

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 :

- v_m la valeur numérique de la vacation du matin,
- v_e celle du soir,
- v_n celle de la nuit et
- v_r la valeur du repos hebdomadaire

Typiquement, les valeurs sont $v_r = 0$, $v_m = 1$, $v_e = 2$, $v_n = 3$.

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 $V_n = \text{NUIT}$ 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([N1, ..., Nn], [X(e1, j), ..., X(en, 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]
```

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 = V_n 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 $[sum,1,#=,[Vm,Ve]]$ 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 mêmes 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.

Le calcul des tours en PPC

Sur une journée j , soit N_{jm} les besoins de la vacation matin, N_{je} (les besoins de soir) et N_{jn} (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 XN_{jm} variables parmi les variables x_j prendront la valeur V_m , XN_{je} variables la valeur V_e , et XN_{jn} variables la valeur V_n . Il s'agit de la variante `multi-among` de la contrainte globale `among`.

N_t = 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
 $[0,0,N_t]$, $C1 = [card, \#, [BegST], 1, \#, [1]]$, $C2 = [card, \#, [EndST], 1, \#, [1]]$
- 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
 $[N, N_t, 2]$, $C1 = [card, \#, [BegST], 1, \#, [1]]$, $C2 = [card, \#, [EndST], 1, \#, [1]]$
- 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)
 $[0, N_t/2, 2]$, $C1 = [card, \#, [BegST], 1, \#, [1]]$, $C2 = [card, \#, [EndST], 1, \#, [1]]$
- 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
 $[0,0,N_t]$, $C1 = [card, \#, [BegST], 1, \#, [1]]$, $C2 = [card, \#, [EndST], 1, \#, [1]]$

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 (consécutives) 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

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

La contrainte `among` peut s'utiliser pour interdire la suite de 4 vacances 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]]]`

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.

3.4 LES CONTRAINTES REDONDANTES

Etant donné la généralité des contraintes globales, la quantité de propagation obtenue dans certains cas est insuffisante. 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 parenthèses¹⁶.

```
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 vacances 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 vacances. Ici, nous proposons une contrainte redondante applicable sur toutes les variables pour chaque paire de variables (salarié-jour).

```
Sm = ∑j=1,J Njm
Se = ∑j=1,J Nje
Sn = ∑j=1,J Njn
IJ is I*J,
XSm :: Sm..IJ,
XSe :: Se..IJ,
XSn :: Sn..IJ,
among( [XSm, XSe, XSn], X, Zeros, [[Vm],[Ve],[Vn]], all).
```

Cette contrainte utilisant la variante multiple `among` spécifie que chaque valeur v_m , v_e , v_n doit apparaître au moins s_m , s_e ou s_n 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 v_m , v_e or v_n), cette contrainte échouera immédiatement dès la pose.

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

Le calcul des tours en PPC

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. s_r) 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 vacances qu'un salarié doit effectuer sur une période de temps (ex. vacances de repos) est très difficile à respecter lorsque l'intervalle est grand, car il y a 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 V_r 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 $X_e(k)$ 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 X_i (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.

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 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 créé par la règle, si les affectations sont faites de façon gloutonne et sans retour arrière.

3.5.2 Les choix généraux du système

Affectation puis propagation

Un tel système peut être réalisé de la façon suivante :

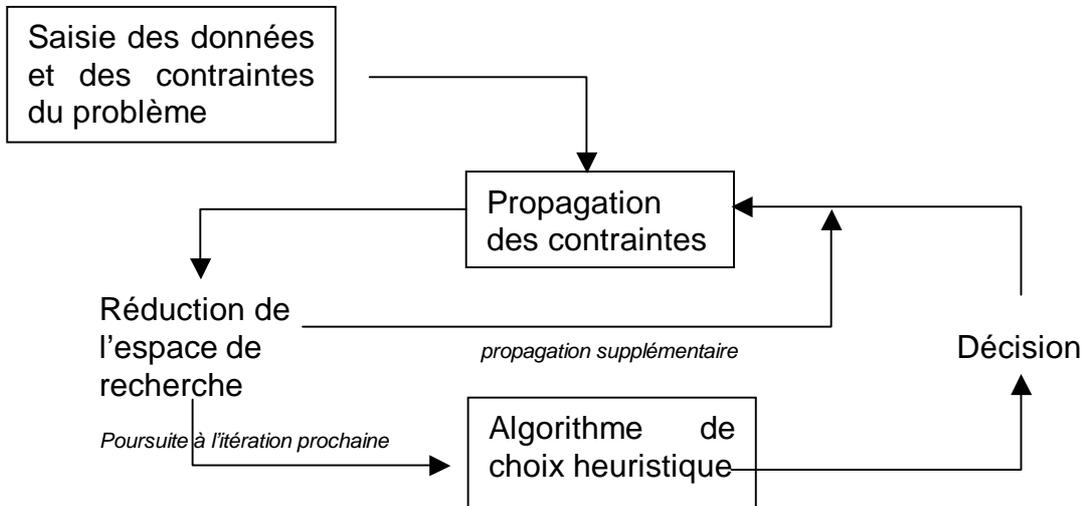


Figure 3.33. Le schéma général du moteur de planification

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.

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, certains salariés ne seront pas affectés. 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.

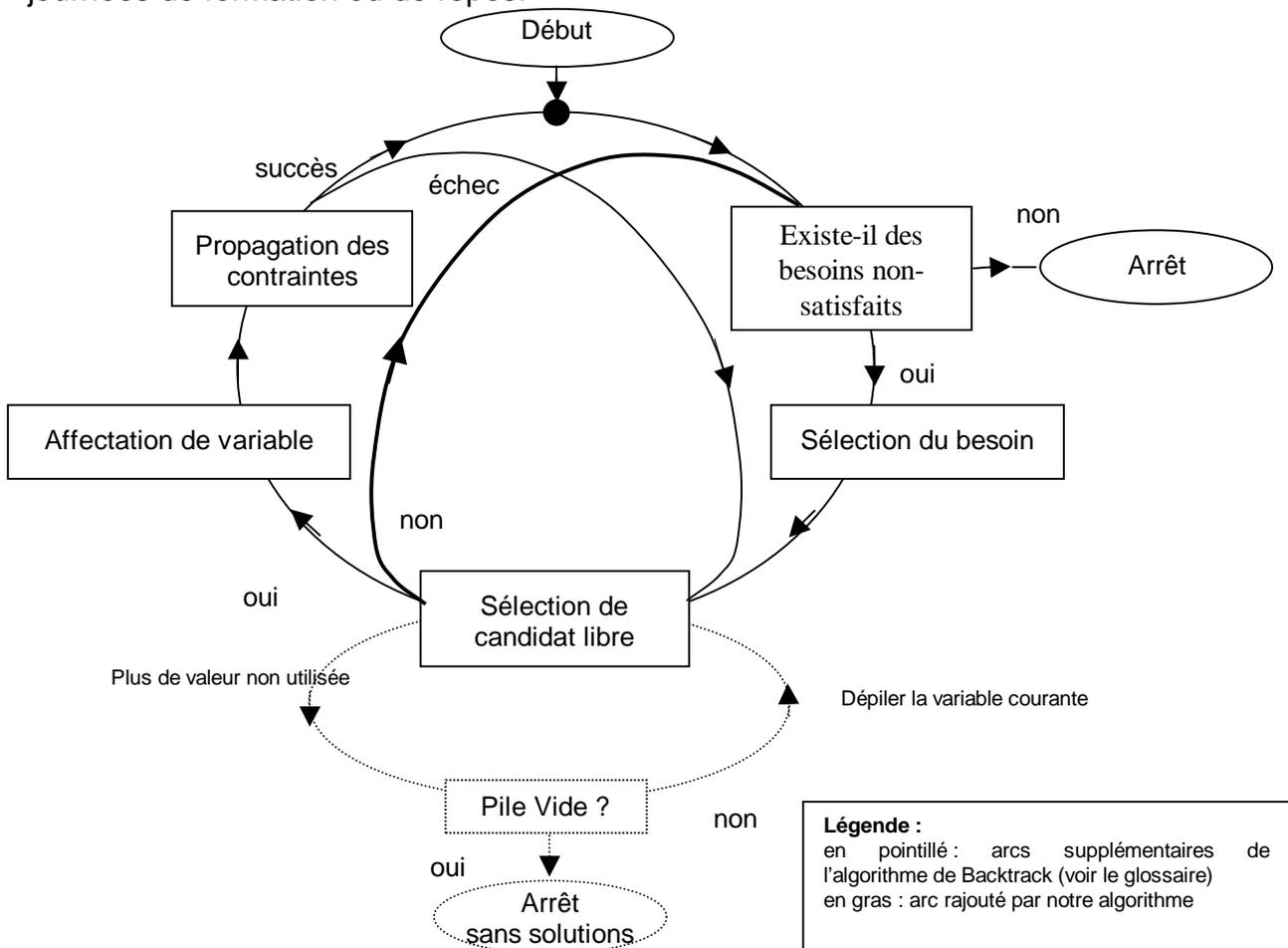


Figure 3.34. Le schéma algorithmique d'EQUITIME

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

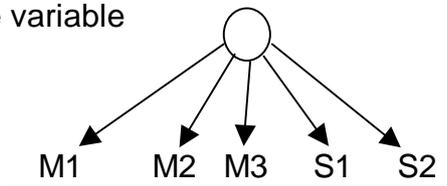


Figure 3.35. Nœud disjonctif dans un arbre de recherche : exemple d'une variable ayant comme domaine {M, S} et les besoins du jour sont 3 matins et 2 soirs

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

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.

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 le 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 ?.

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 :

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

3.6 L'INTERFACE UTILISATEUR

Nous rapportons dans ce paragraphe l'interface utilisateur des différentes 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.

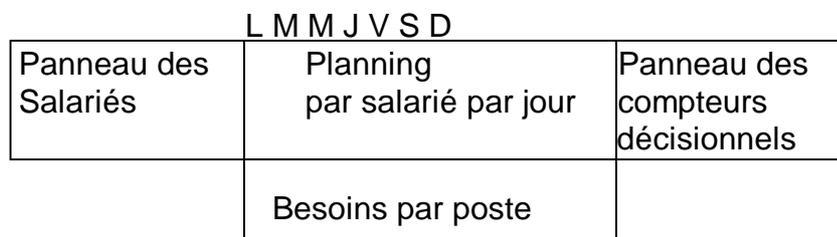


Figure 3.37. Modèle conceptuel en « T » du tableur de planification

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

seuil, ou en vert¹⁷. 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.

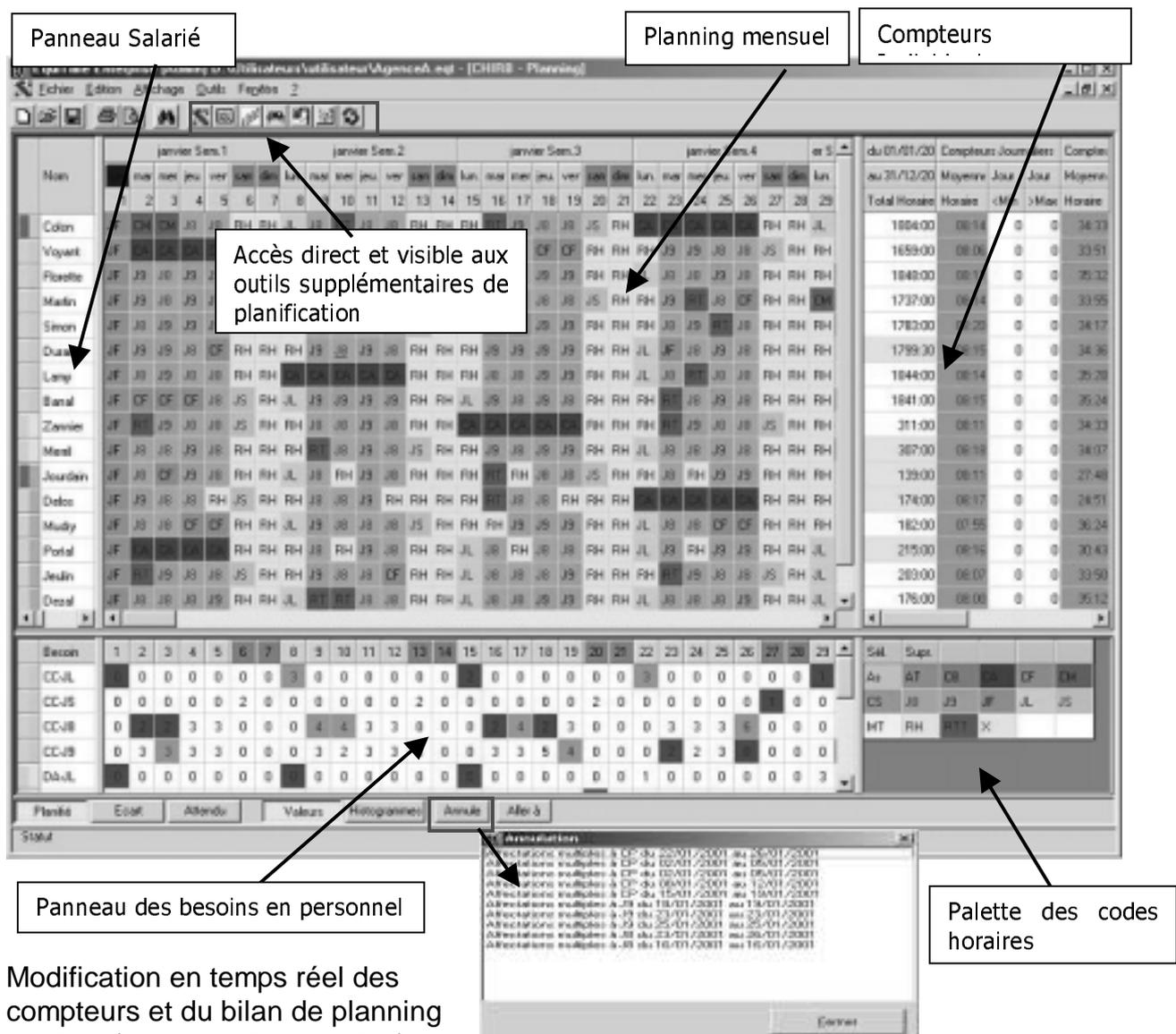


Figure 3.38. EQUITIME Version 3 avec l'écran des annulations

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 tableaux courants.

Le calcul des tours en PPC

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. Répéter 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.

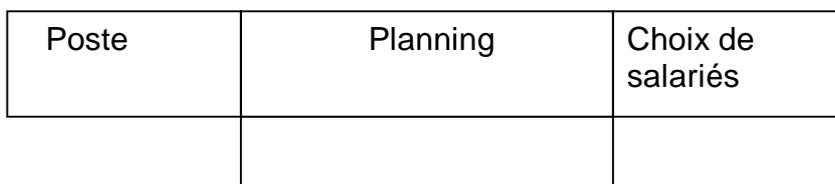
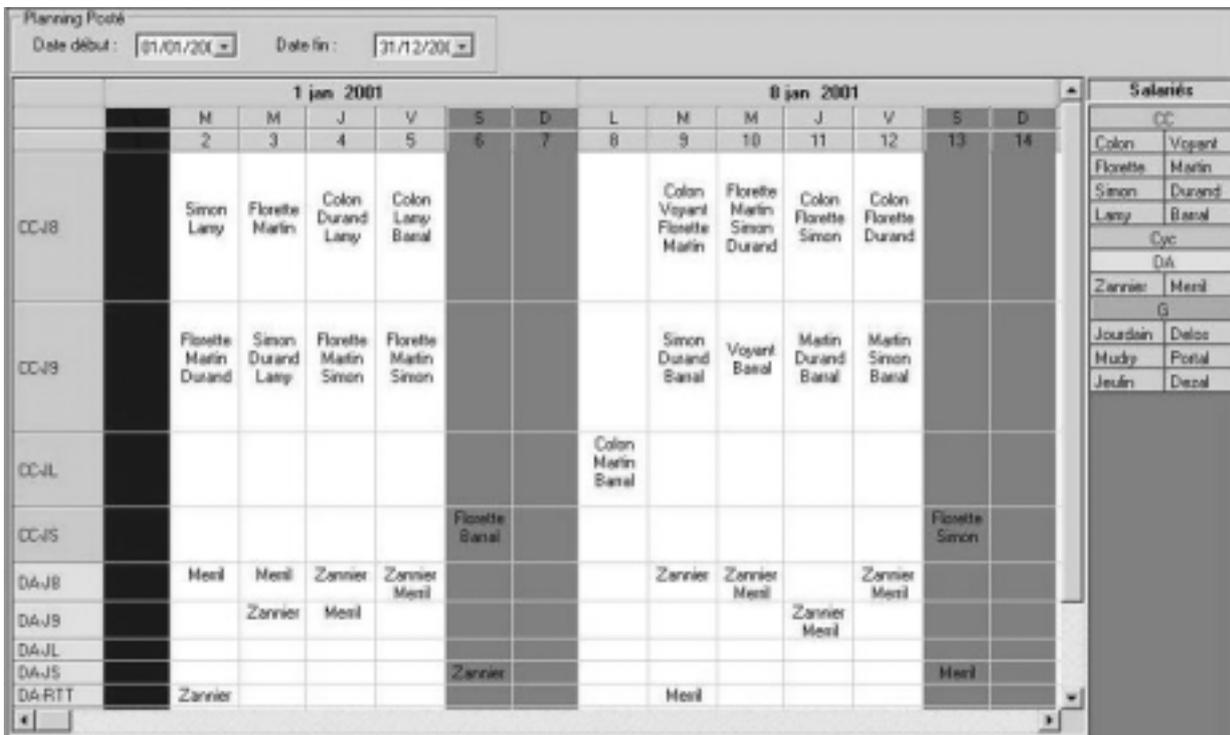
L'utilisateur peut valider les affectations qu'il estime bonnes, en les remettant en noir. Lors du lancement, le moteur effacera les propositions blanches avant de faire de nouvelles affectations (en blanc). La métaphore des affectations en blanc rassure les utilisateurs : toute modification indésirable 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.

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.



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.

Vues Soldes et Cumuls

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

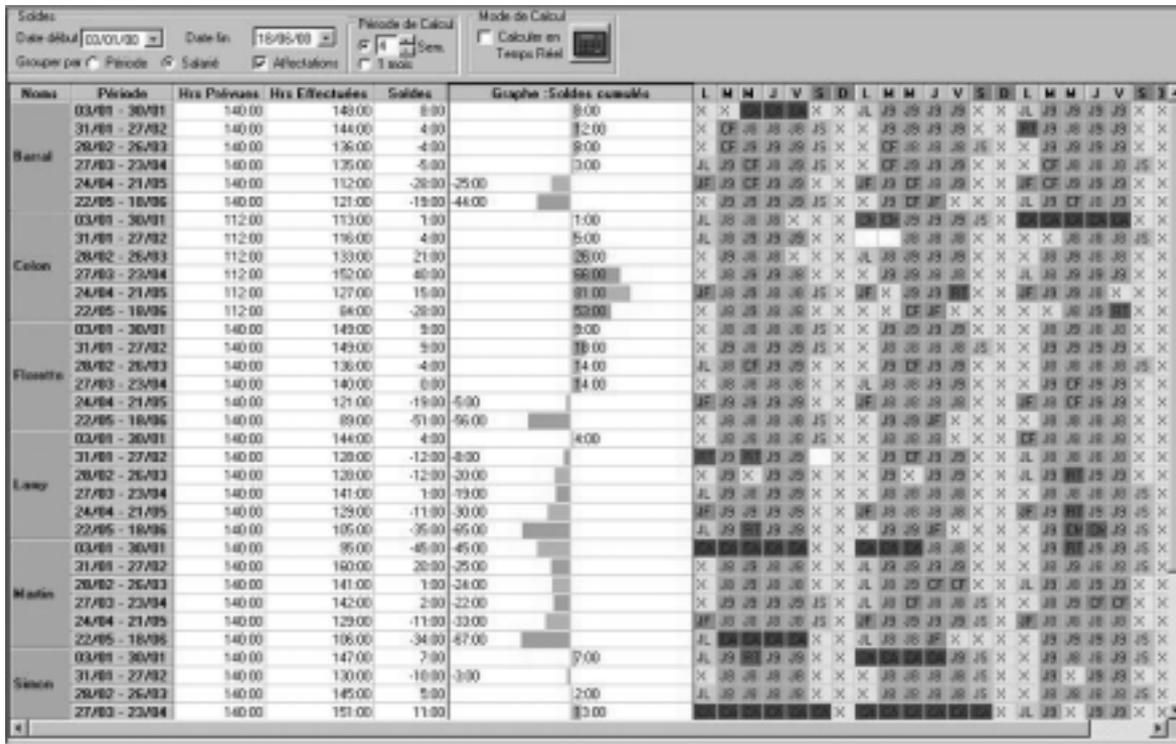


Figure 3.41. EQUITEME Vue Soldes

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.

Nom / Cumul	Tot Travail		Lun JL		Mer JB		Tot X		Mer RTT		Tot CA		
	Nb	Hrs	Nb	Hrs	Graph: Mer JB: Nb	Hrs	Nb	Hrs	Nb	Hrs	Nb	Hrs	
Colon	86	711.00	7	56.00	7	56.00	96.00	65	0.00	0	0.00	16	0.00
Voyant	75	612.00	6	48.00	17	136.00	69	0.00	3	0.00	11	7.00	
Florette	106	855.00	4	32.00	12	96.00	62	0.00	1	0.00	5	0.00	
Martin	105	777.00	8	64.00	7	56.00	52	0.00	1	0.00	12	0.00	
Simon	37	732.00	5	40.00	17	136.00	59	0.00	2	0.00	16	112.00	
Lamy	105	863.00	7	56.00	9	72.00	50	0.00	7	0.00	0	0.00	
Barnal	113	838.00	7	56.00	7	56.00	58	0.00	0	0.00	3	21.00	
Zannier	119	918.00	13	104.00	11	88.00	48	0.00	6	0.00	0	0.00	
Merril	103	843.00	8	64.00	12	96.00	57	0.00	9	0.00	0	0.00	

Figure 3.42. EQUITEME Vue Cumuls et son paramétrage

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 clics 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.

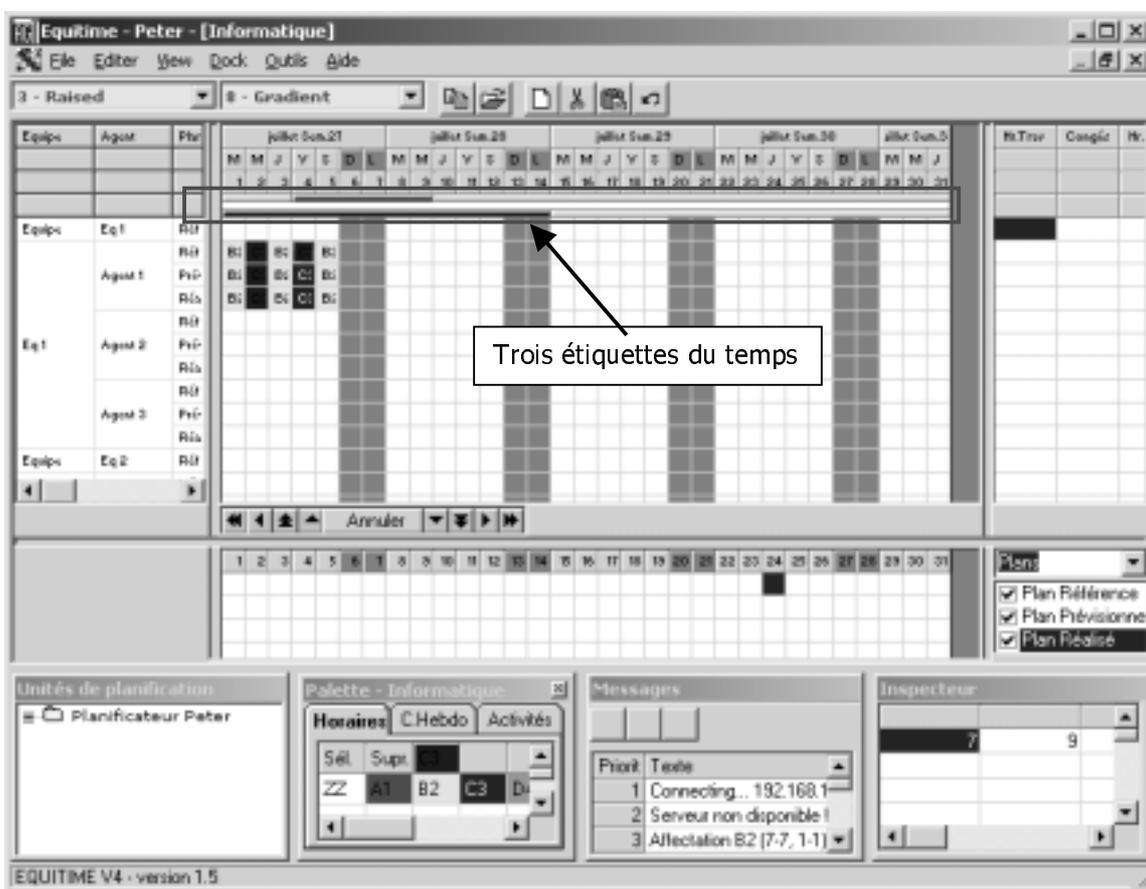


Figure 3.43. EQUITIME Version 4 : vue globale

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

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.

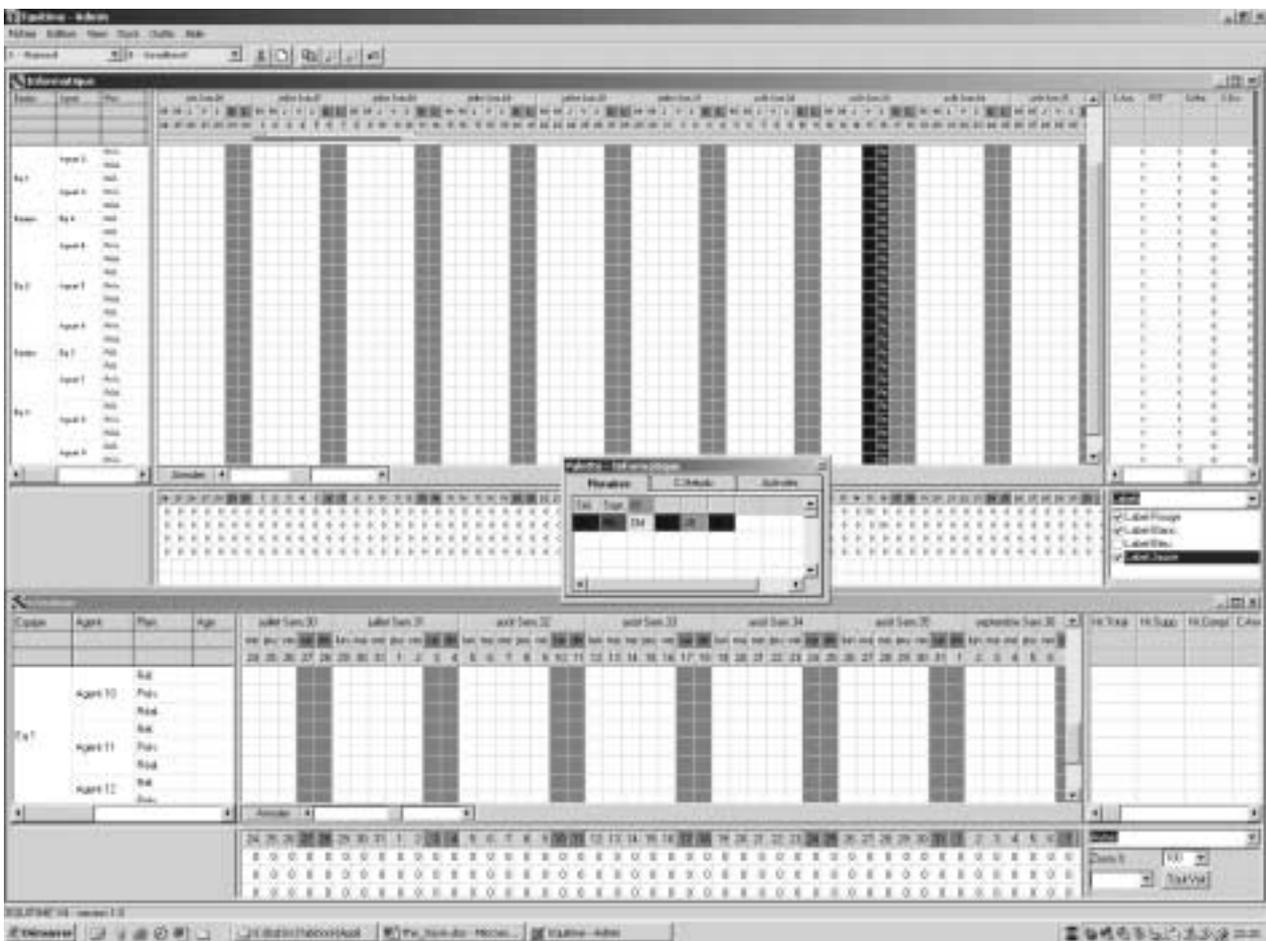


Figure 3.44. EQUITIME Version 4 avec un seul cadran ouvert

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.

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.

- 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.

Le calcul des tours en PPC

- 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 vacances 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

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.