



HAL
open science

Emploi du temps en milieu hospitalier : heuristiques stochastiques d'aide à l'élaboration

Claudie Jacquier-Duret

► **To cite this version:**

Claudie Jacquier-Duret. Emploi du temps en milieu hospitalier : heuristiques stochastiques d'aide à l'élaboration. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1987. Français. NNT : . tel-00325680

HAL Id: tel-00325680

<https://theses.hal.science/tel-00325680>

Submitted on 30 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

**l'Université Scientifique, Technologique
et Médicale de Grenoble**

pour obtenir le grade de
Docteur de 3ème cycle
"Recherche Opérationnelle"

par

Claudie JACQUIER-DURET

**EMPLOI DU TEMPS EN MILIEU HOSPITALIER:
HEURISTIQUES STOCHASTIQUES D'AIDE A L'ELABORATION**

Thèse soutenue le 25 septembre 1987 devant la commission d'examen

Monsieur C. BENZAKEN	Président
M. le Professeur GUIGNIER	Examineurs
Messieurs J. FONLUPT	
C. LACOTE	
J.P. UHRY	

Je tiens à remercier:

Monsieur Claude BENZAKEN de l'honneur qu'il me fait de bien vouloir présider le jury de cette thèse.

Jean-Pierre UHRY qui a toujours été disponible (même pendant ses jours de repos !!), pour me conseiller, me guider et m'encourager tout au long de ce travail.

Monsieur le Professeur GUIGNIER, Chef du service de Réanimation Médicale du Centre Hospitalier de La Tronche, ainsi que toute son équipe, qui a toujours répondu avec patience et gentillesse, à nos diverses sollicitations.

Jean FONLUPT et Christophe LACOTE qui ont bien voulu faire partie du jury.

Monsieur IGLESIAS et toute l'équipe de Reprographie pour la promptitude et le soin qu'ils ont apportés à la reproduction de cette thèse.

Je profite de l'occasion pour exprimer toute ma sympathie à tous les membres de l'équipe de Recherche Opérationnelle pour l'ambiance de travail qu'ils m'ont créée.

Claudie JACQUIER

INTRODUCTION

Le problème d'établissement de plannings de travail, bien qu'il ne soit pas spécifique au milieu hospitalier, est l'un des principaux problèmes d'organisation rencontrés par tout service de soins.

Il s'agit de déterminer pour un personnel donné et sur une période fixe, les jours et horaires de travail et les jours de repos.

Ces affectations doivent obéir à un certain nombre de contraintes, contraintes liées le plus souvent au type de services requis, tout en respectant autant que possible les souhaits de ce personnel.

L'étude qui constitue l'objet de cette thèse a été effectuée sur le cas précis d'un service de Réanimation de l'Hôpital des Sablons de GRENOBLE. Le but recherché était de satisfaire à la fois un hôpital ou plus précisément une unité fonctionnelle de cet hôpital, devant produire une quantité minimum de soins par jour, et l'équipe des infirmières de cette unité, exprimant de son côté ses propres désirs.

Dans ce type de problèmes à résoudre, les intérêts des deux parties entrent généralement en conflit lorsque le nombre minimum de personnes devant travailler pour produire ces soins, ne peut être atteint qu'en contrariant les souhaits de certains membres du personnel.

Ce conflit se traduit en général par l'installation de "modèles de travail" ou "emplois du temps" considérés indésirables par les personnes devant les effectuer.

L'élaboration des "modèles de travail" ou "emplois du temps" comprend alors une phase délicate et longue de négociations avec le personnel concerné.

D'un point de vue pratique, la mise au point et, par la suite, le suivi de ces plannings de travail demandent aux hôpitaux la mobilisation d'une certaine partie de leur personnel. Dans ces conditions, ceux-ci ont été amenés à réfléchir dans quelle mesure l'informatique pouvait contribuer à faciliter la gestion du service.

Devant la complexité et la multiplicité des contraintes à satisfaire, l'idée d'un modèle mathématique "parfait" devait être abandonnée au profit d'un logiciel plus souple et plus ergonomique par son interactivité.

Les différences de fonctionnement des multiples services hospitaliers demandeurs, et le besoin d'autonomie de plus en plus ressenti devant la lourdeur d'une programmation sur site central, nous firent envisager la solution "micro-ordinateur". La facilité d'accès, la flexibilité et la souplesse d'utilisation apparaissaient comme des atouts majeurs pour son utilisation. Le service de réanimation disposant d'un APPLE IIe, la première étape fut donc la réalisation d'un logiciel, utilisant au mieux les fonctionnalités de cette machine, logiciel facilement adaptable aux différents services susceptibles d'être intéressés.

La micro-informatique n'ayant pas acquis avant les années 83-85 une image très sérieuse dans les milieux professionnels, nous ne pouvions partir d'aucune expérience et d'aucun travaux effectués sur micro-ordinateurs, mais l'étude des outils et des modèles existant sur site central nous facilita l'analyse du problème et l'expression des besoins non familiers à des personnes étrangères au milieu hospitalier.

Un effort particulier fut apporté à l'ergonomie et à la convivialité du logiciel destiné à des utilisateurs "candides" devant tout outil et langage informatique. La reproduction graphique sur écran des feuilles d'emploi du temps habituelles et l'utilisation de couleurs ont été mises en place de manière à faciliter le dialogue avec toute personne

effectuant une mise au point ou le suivi de l'emploi du temps. Pour les mêmes raisons, la représentation du travail de chaque infirmière sur des fiches récapitulatives, avec le lien direct de ces fiches et des emplois du temps a été élaboré sur les bases du fichier tenu manuellement à l'époque.

Après la réalisation de ce logiciel, l'élaboration des emplois du temps pouvait être effectuée de la même manière que manuellement, le gain de temps étant surtout appréciable dans la mise au point et le suivi de l'activité des infirmières (avec des mises à jour automatiques des fiches infirmières).

La nécessité d'une méthode automatique de remplissage de ces plannings de travail semblait alors intéressante afin de donner à la personne chargée de leur élaboration une proposition initiale "assez bonne". L'application de méthodes ou d'heuristiques stochastiques (classiques ou assez simples à mettre en oeuvre) nous donna des résultats assez encourageants n'ayant pas pour prétention d'être d'aussi bonne qualité que ceux obtenus manuellement, mais d'être uniquement une bonne base de discussion et de mise au point.

Dans le premier chapitre, on présente différents systèmes informatiques réalisés sur site central dans le but de répondre à des problèmes similaires. Partant d'approches classiques de modélisation du problème, celles-ci nous prouvent la difficulté d'intégrer dans un modèle mathématique souvent "rigide", les contraintes de préférences du personnel.

Dans le deuxième chapitre, nous exposons le cas précis du service de Réanimation de Grenoble dont la demande fut à l'origine de ce travail. Le principal objectif dans l'élaboration d'un logiciel sur micro-ordinateur était un souci d'ergonomie et de respect des procédures de travail actuelles. Une visualisation graphique utilisant la couleur devait améliorer la convivialité et l'interactivité du programme, permettant à la fois des modifications "en temps réel" et facilitant un suivi journalier de l'activité des infirmières.

La contrainte matérielle du développement sur APPLE IIe, contrainte technique donnée par le service de Réanimation de Grenoble, nous intéressait techniquement. L'étude a débuté par une enquête sur le travail effectif, étude qui a permis la spécification d'un prototype interactif, puis une validation par le service de cet outil.

Dans le troisième chapitre on présente une heuristique stochastique pour élaborer une proposition initiale de planning 'satisfaisante' afin de faciliter le travail de départ des personnes chargées de cette élaboration. Cette heuristique peut être adaptée à une vaste classe de problèmes combinatoires. Elle est due à Kirkpatrick [KI] et est issue d'une analogie qui existe entre les problèmes d'optimisation combinatoire et le procédé expérimental de recuit, utilisé en thermodynamique pour trouver les états stables.

Après une description rapide de la méthode de Kirkpatrick, on explique l'adaptation assez simple de cette méthode à notre problème et les résultats de nos expérimentations numériques. Nous présentons également dans ce chapitre une autre heuristique stochastique inspirée des principes de la génétique, heuristique qui présente sur la précédente l'avantage d'une parallélisation plus facile, ce qui permettrait de réduire énormément le temps de calcul, souvent long avec ce type de méthodes.

En dernière partie, on donne une bibliographie concernant le problème d'élaboration d'emplois du temps sur outil informatique, bibliographie complétée par les études déjà effectuées sur les heuristiques stochastiques sur différents problèmes combinatoires.

CHAPITRE I: ANALYSE DES OUTILS ET MODELES EXISTANTS

Les seules approches de systèmes informatiques avoisinant notre problème avaient donc été effectuées sur site central.

En effet dans la littérature, nous trouvons deux manières d'aborder le problème.

I-1 L'approche traditionnelle (le plus souvent entièrement manuelle):

On divise le temps en périodes de longueur fixe selon certains critères. A chaque nouvelle période de planning on recommence à zéro les affectations et on essaie, pas à pas d'établir un planning respectant les contraintes et les demandes du personnel.

AHUJA et SHEPPARD [AH-SH] employaient les facilités d'un terminal interactif pour aider le "superviseur" à sélectionner les modèles de travail fournissant le recouvrement désiré dans chaque équipe.

L'atout majeur de cette solution est sa **flexibilité**. Les heuristiques interactives offrent l'avantage de permettre à la personne effectuant les plannings de mettre au point ceux-ci au fur et à mesure qu'ils évoluent.

Cependant, le temps demandé pour l'élaboration des plannings sur chaque période, le travail minutieux et long que cela sous-entend, l'instabilité et le manque d'optimalité des modèles de travail du personnel, rendent cette solution très critiquable.

De plus aucune évaluation qualitative n'étant effectuée par infirmière, la notion d'égalité des avantages n'apparaît pas, ce qui laisse grande ouverte la porte au favoritisme.

I-2- L'approche par ordonnancement cyclique:

Cette solution fut celle proposée par M.WARNER [WA], par HOWEL [HO], FRANCES [FR], MAIER-ROTHER et WOLFE [MA-WO], et SMITH [SM] dans les années 70.

On établit dans un premier temps de manière très soignée un horaire "de bonne qualité" pour le personnel, sur une période fixe. Ce modèle est ensuite répété de périodes en périodes.

Cette méthode rejoint celle étudiée par TREMOLIERES [TR] sur la génération automatique de plannings de roulements de "quarts" (terminologie que l'on attache aux situations dans lesquelles une disponibilité de 24h sur 24 est requise). Dans cette dernière approche, l'aspect opérationnel du problème est abordé: compte tenu d'un certain nombre de contraintes, comment générer automatiquement des plannings cycliques acceptables, afin de déterminer ceux qui offrent le plus de "fiabilité" pour pallier certaines contraintes occasionnelles comme les congés, la formation, les absences. Dans cette méthode, le personnel a ainsi l'avantage de pouvoir **prévoir** à l'avance des **jours de repos hebdomadaires**, et n'est pas soumis aux risques d'inégalité des avantages, contrairement à la méthode précédente: les séquences de travail non appréciées sont réparties de manière identique sur le personnel à l'aide de rotations.

Cependant cette solution manque totalement de flexibilité: il est très difficile de s'adapter facilement à la moindre demande exceptionnelle. Ce manque de flexibilité a causé très vite l'abandon de cette méthode.

On rencontre des difficultés à établir des plannings pour du personnel très spécialisé. De plus les individus sont obligés de négocier des échanges entre eux lorsqu'ils ne peuvent, pour une raison personnelle, effectuer le service sur lequel ils sont affectés.

Sur ces deux idées se sont alors greffées d'autres méthodes, ayant pour but de restreindre les inconvénients de celles-ci.

Ce sont ces méthodes que nous allons étudier maintenant.

I-3 Méthode de D. Michael WARNER:

M. WARNER a étudié un système visant à améliorer les élaborations de plannings hospitaliers en incorporant dans un modèle existant de BALINIFY et BLACKBURN [BA-BL] les préférences du personnel.

Le système proposé essaie d'incorporer la flexibilité de la méthode traditionnelle en maintenant la stabilité de la méthode des ordonnancements cycliques.

Dans un premier temps, un ensemble de "règles de décision" et de contraintes est déterminé par l'infirmière chef ou le chef du personnel: nombre minimum d'infirmières devant être présentes, nombre de week-ends travaillés, règles de rotation entre services de jour et services de nuit, appréciation qualitative de certaines séquences de travail... Puis, pour chaque roulement spécifique, des restrictions et des souhaits particuliers à la période considérée, sont introduits comme paramètres à la deuxième phase qui a pour but de réduire le nombre d'emplois du temps "potentiels" à son strict minimum.

La qualité du planning est évaluée par les infirmières elles-mêmes lors de la deuxième phase et l'on peut faire apparaître une notion d'égalité dans les différentes affectations. L'algorithme permet alors, par échanges de "blocs", une amélioration de l'évaluation qualitative du planning.

Le programme de M. WARNER est une modification du programme de BALINTFY et BLACKBURN [BA-BL], modification permettant de prendre en compte les préférences du personnel.

Détaillons les deux phases préalables à l'algorithme:

I-3-a Règles et contraintes sur chaque "roulement".

Les règles de conduites suivantes sont déterminées initialement, puis mémorisées:

a- Spécification du nombre minimum de personnes devant être présentes dans les différentes tranches horaires d'un jour.

On spécifie aussi le nombre maximum de personnes supplémentaires autorisées.

b- Week-end : Par souci d'égalité on introduit un roulement cyclique permettant de répartir les week-ends travaillés.

Chaque infirmière pourra donc prévoir les week-ends "bloqués".

c- Rotation : Chaque infirmière sera classée en "infirmière permanente" ou "infirmière mobile" selon qu' elle peut ou non changer d'horaires de travail (différentes tranches horaires de service).

On pourra ainsi spécifier les jours de rotation en essayant de prendre en compte la fatigue occasionnée par ces rotations. Une règle de rotation est mise en oeuvre sur les infirmières mobiles.

d- Chaque infirmière évalue selon ses propres goûts, les avantages et les désagréments de certains types de modèles de travail.

Par exemple on peut juger certaines séquences de jours travaillés plus ou moins bonnes, ou réprouver de travailler un jour isolé (entre deux repos)...

Pour maintenir une certaine égalité, on permet à chaque infirmière de répartir une même quantité de points de pénalité sur des modèles de séquences de plannings (exemple: un jour seul - six jours d'affilée - six jours d'affilée avant un week-end libre...).

I-3-b Phase 2 : Introduction des restrictions sur un roulement particulier.

Toute demande spéciale est introduite (jour de congé ou changement dans la politique des week-ends).

Les demandes de congé sont entrées de trois manières:

- L'infirmière responsable accorde ou refuse la requête à la vue de celle-ci.
- L'infirmière évalue les requêtes en terme de poids: le poids traduit l'insistance affectée à la requête.

Associé à une requête, il permet par opposition, d'atténuer les "évaluations qualitatives négatives" du

personnel sur certaines séquences de travail et permet ainsi d'intervenir dans la formulation mathématique.

L'affectation de ces poids est mémorisée afin de maintenir l'égalité des avantages du personnel.

- Chaque infirmière affecte elle-même différents poids à ses requêtes. Chaque mois, elle peut répartir une certaine quantité de points sur ce type de requêtes.

Elle peut aussi cumuler ses points pour une utilisation ultérieure.

Cette méthode abolit les notions de favoritisme.

Rotations

Un souhait de modification de la règle de rotation peut être spécifié sur la feuille de planning.

Le nombre de rotations qu'une infirmière doit effectuer est déterminé par des bornes inférieures et supérieures.

Dans un même temps, le programme essaie d'égaliser le nombre de jours de rotations de chaque infirmière et mémorise celui-ci.

I-3-c LE MODELE MATHEMATIQUE

Pour chaque infirmière i , on engendre un ensemble d'emplois du temps potentiels E_i .

Une solution est l'affectation à chaque infirmière d'un et un seul élément de son ensemble E_i .

Pour des raisons de taille du problème, de capacité mémoire et temps de calcul de l'ordinateur, la période initiale de planni-

fication doit être réduite de moitié ou partagée en trois petites périodes. (Wagner propose des périodes de 14 jours). Ces sous-périodes sont traitées indépendamment.

Cette réduction réduit considérablement le nombre de solutions possibles pour chaque infirmière. L'effet de sous-optimalité est négligeable devant le gain de temps.

Cette période de 14 jours ne peut cependant être réduite davantage sans poser de sérieux problèmes pour les rotations.

Les variables:

Soit $i = 1, 2, 3, \dots, L$ le personnel à répartir

Soit J_i le nombre de solutions possibles pour l'infirmière i

Soit j l'indice de l'ensemble des possibilités pour chaque infirmière

($j=1, 2, 3, \dots, J_k$ pour l'infirmière k)

Soit A_{ij} le vecteur de longueur égale au nombre de tranches horaires multiplié par 14 jours représentant le jème modèle de travail possible pour l'infirmière i ,

avec A_{ij} ayant un 1 dans les positions (jours) où l'infirmière i est de service
ayant un 0 où elle ne l'est pas.

Par exemple:

Dim Jeu

| |

$A_{11} = \{11100111010000/00000000001110/00000000000000\}$

 matin après-midi nuit

A_{11} correspond à l'infirmière 1 travaillant suivant le modèle:

DIM Lun Mar Mer Jeu Ven Sam DIM Lun Mar Mer Jeu Ven Sam

M M M X X M M M X M S S S X

avec M = service matin

S = service après-midi

N = service nuit

X = ne travaille pas

Les variables du problème sont:

$X_{ij} = 1$ si l'infirmière i travaille selon A_{ij}

$= 0$ sinon

Sans restrictions sur le nombre des possibilités de modèles de travail pour une infirmière i , l'ensemble J_i serait inexploitable car trop volumineux.

Heureusement,

- * Le nombre de jours travaillés par semaine est généralement constant.
- * Certaines infirmières peuvent avoir un modèle de travail déjà fixé (dans la phase 2).
- * Les week-ends de repos sont généralement fixés par les règles de la phase 1.

- * Toutes les infirmières ne sont pas soumises aux rotations et celles qui le sont, le sont pour un nombre fixé de jours.
- * La longueur des séquences de jours travaillés n'est pas illimitée.
- * Plusieurs jours sont fixés à l'avance pour chaque infirmière à cause de certaines requêtes accordées, certains engagements...

Les auteurs affirment que ces restrictions, ainsi que d'autres non énumérées ici, ramènent le nombre de modèles de travail potentiels à une moyenne de 10 à 20 par infirmière.

Dans la pratique, on génère pour chaque infirmière 600 possibilités, parmi lesquelles les 50 "meilleures" sont utilisées dans le modèle mathématique.

Les contraintes:

Si NbMin est le nombre minimum d'infirmières requis chaque jour, la contrainte que l'on peut appeler "contrainte de recouvrement" se formule ainsi:

$$\sum_{i=1}^L \sum_{j=1}^{J_i} X_{ij} A_{ij} \geq \text{NbMin}$$

avec
$$\sum_{j=1}^{J_i} X_{ij} = 1 \quad i=1,2,3,\dots,L$$

$$X_{ij} = 0 \text{ ou } 1 \quad \begin{array}{l} i=1,2,3,\dots,L \\ j=1,2,3,\dots,J_i \end{array}$$

La fonction objective :

Si C_{ij} est la valeur relative du modèle de travail j pour l'infirmière i , l'objectif est de maximiser

$$\sum_{i=1}^L \sum_{j=1}^{J_i} C_{ij} X_{ij}$$

Les C_{ij} sont calculés à partir de

- l'évaluation qualitative des infirmières sur les modèles de travail "réalisables".
- l'évaluation du respect des requêtes de poids importants.
- l'égalisation du nombre de rotations.

On déduit l'aversion de l'infirmière pour certains modèles (en terme de poids), à la satisfaction qu'elle éprouve à avoir certaines séquences dans ceux-ci. (ex: jours de repos...)

La prise en compte des désirs des infirmières entre en jeu dans le calcul des C_{ij} .

Ce coefficient est la combinaison

- des sentiments de l'infirmière i devant l'ensemble de l'emploi du temps j (évaluation habituelle de cette séquence déterminée en phase 1).
- des spécificités de cet emploi du temps par rapport à ses désirs de jours de congé pendant cette période (requêtes particulières énoncées en phase 2).
- du calcul de répartition des rotations (égalisation).

Par exemple, sur la séquence "XXDDDDDDDXDXDD" où le premier jour est un dimanche:

Si une infirmière évalue à -3 points de pénalité le fait de travailler un jour "isolé" et que le fait de travailler 7 jours de suite (dans la même séquence); vaut -9 points de pénalité, ce modèle de travail sera évalué à $-3-9 = -12$.

Si d'un autre côté, l'évaluation en terme de poids d'avoir le premier dimanche et lundi de repos est de 18, la séquence vaudra $-3-9+18 = 6$.

Le C_{ij} sera de plus réajusté selon le nombre de rotations déjà effectuées.

Nous ne détaillerons pas plus cette évaluation des sentiments du personnel, partie la plus délicate de la formulation, qui fut d'ailleurs reprise ultérieurement [RO-1].

I-3-d HEURISTIQUE DE RESOLUTION

La solution est une modification de l'algorithme proposé par BALINTFY et BLACKBURN [BA-BL]. Les modifications sont effectuées de façon à tirer avantage de la structure en (0,1) des A_{ij} .

La première étape de l'algorithme consiste en la recherche d'une solution réalisable.

Dans 20 à 30 % des cas, les restrictions apportées par les infirmières n'amènent pas à une solution réalisable.

Les "problèmes", (contraintes violées, en général insuffisance de personnel) sont alors déplacés jusqu'aux jours susceptibles de les supporter plus facilement, par modifications interactives des spécifications entrées en phase 2, (jours spécifiés en données au programme).

Dans la deuxième étape, on essaye d'améliorer la fonction objective de la formulation mathématique, afin de prendre en compte au mieux les désirs du personnel qui n'apparaissent pas jusque là dans la fonction objective.

Pour cela, on utilise la méthode des "blocs pivotants".

Un bloc représente un modèle de travail pour une infirmière.

Certains "blocs" sont échangés avec d'autres appartenant à l'ensemble des possibilités, cet échange n'étant effectué que lorsqu'il ne détériore pas la "réalisabilité" de la solution, et qu'il améliore l'évaluation de la fonction objective. Cette méthode est très analogue à celle connue en Recherche Opérationnelle sous le nom de 2-optimisation pour le problème de "Voyageur de Commerce".

Comme dans l'algorithme de BALINTFY et BLACKBURN, le programme teste l'échange simultané de 1, 2 ou 3 "blocs".

On remarque dans la pratique que les bonnes solutions sont obtenues en utilisant des échanges de 2 blocs au maximum.

Cette méthode consiste donc à générer au départ, un très grand nombre de modèles de travail potentiels pour chaque infirmière, puis de ramener ce nombre à une dimension raisonnable. Le problème revient alors à choisir, à l'aide d'un système d'énumération, un ensemble de configurations A_{ij} qui respectent les contraintes les plus importantes: couverture des tranches de service,....

Une fois cette solution trouvée, une procédure d'échange de "blocs" permet une amélioration de l'évaluation qualitative du planning.

Cet algorithme a été implanté en 75-78 sur un IBM 360 et demandait 20 à 40s d'exécution pour établir un planning d'un service de 25 infirmières.

Cette approche fut l'une des deux premières, avec celle de ROTSCHEIM [RO] à ne pas être une méthode cyclique de planification, méthode caractérisée par sa rigidité par rapport aux demandes du personnel.

Cependant, la souplesse de ce système est considérablement alourdie par la restriction nécessaire des possibilités de chaque infirmière. Si ces restrictions sont trop importantes, la réalisabilité d'une solution devient fortement compromise...

Dans ce type d'approches, il semble aussi prudent de marquer un certain scepticisme devant la méthode des "notes" données par les infirmières. Celle-ci se transforme le plus souvent en un "jeu" dont les règles sont très obscures...

I-4 Méthode de HOLMES E. MILLER:

Holmes E. Miller présenta en 1975 une méthode quelque peu différente. Le problème de planification des infirmières est abordé comme un problème de création d'une configuration de modèles de travail, spécifiant le nombre et les identités des infirmières travaillant chaque jour du planning.

Pour trouver une solution, E. Miller étudia les procédures qui tentaient de résoudre des problèmes similaires ou se rapprochant:

- Le travail de WOLFE et YOUNG [WO-YO] qui construisirent des modèles mathématiques minimisant le coût d'affectation d'infirmières de différentes catégories à différentes tâches.
- Les approches par la méthode cyclique de HOWELL [HO] et de MAIER-ROTHE et WOLFE [MA-WO].
- La méthode WARNER présentée ci-dessus.

Cette méthode assure l'existence d'une solution réalisable en séparant les contraintes en deux classes: celle des contraintes "obligatoires", et celles dont la violation n'entraîne qu'une pénalité qui intervient alors dans la fonction objective. Les souhaits des infirmières sont introduits par le biais de contraintes non obligatoires. Cette formulation permet d'intégrer la contrainte de recouvrement de manière beaucoup plus souple et de mieux opposer la pénalité qu'entraîne la violation de celle-ci, à celle engendrée par la contrariété des désirs du personnel. L'algorithme proposé donne, par explorations successives, une solution proche de la solution optimale.

I-4-a LE MODELE MATHEMATIQUE PROPOSE:

Le modèle proposé répartit les jours de travail et de repos sur toutes les infirmières d'un service, sur une période de longueur fixe (2 - 4 - 6 semaines), en prenant en compte les exigences hospitalières et certaines contraintes des employés.

Il est possible, à cause du nombre important de contraintes pouvant entrer en compte, qu'aucune solution réalisable n'existe. Pour cette raison, on est amené à séparer ces contraintes en deux classes:

- Les contraintes de réalisabilité, c'est à dire celles qui définissent les ensembles de "solutions possibles". Ces contraintes sont "obligatoires".
- Les contraintes non obligatoires dont la violation n'entraîne qu'une pénalité qui apparaîtra dans la fonction objective.

En général P_i est l'ensemble des modèles qui:

- satisfont une requête particulière d'infirmière.
- satisfont dans la mesure du possible les contraintes que nous aimerions voir respecter, une fois les requêtes particulières connues.

Cette méthode laisse une assez grande souplesse dans la mesure où des contraintes autres peuvent être ajoutées (ex: un week-end sur deux libre etc...).

B - Deuxième classe de contraintes: contraintes non obligatoires.

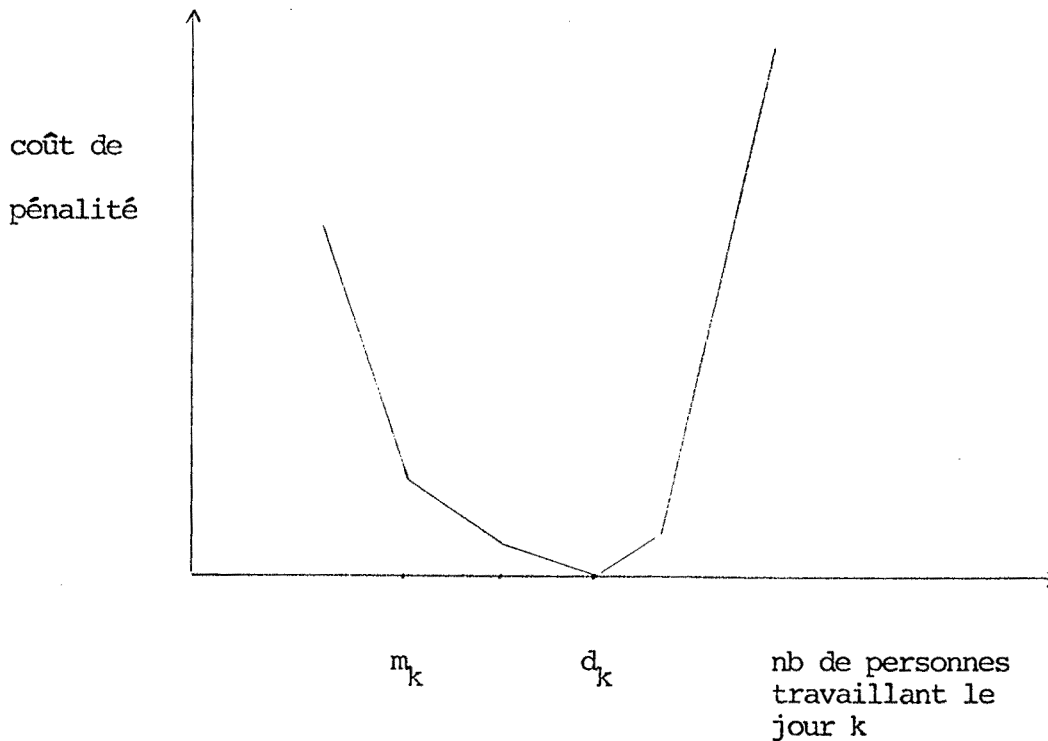
Chaque modèle de travail généré dans le P_i peut violer des contraintes non impératives, mais on affecte alors à ce modèle une pénalité.

Ces contraintes peuvent être:

- pas de séquences de jours travaillés de S_i jours, $S_i < \mu$
- pas de séquences de jours travaillés plus courtes que T_i jours, $T_i > \mu$
- pas de jour de congé isolé (pas de modèle comprenant "101").
- pas plus de k_i séquences "101" consécutives.
- Q_i week-ends libres dans chaque période.
- pas plus de w_i week-ends consécutifs travaillés dans une période.
- pas de modèles comprenant 4 jours de congés successifs.
- pas de modèles contenant des week-ends mixtes (samedi travaillé - dimanche repos ou réciproquement).

Ces contraintes sont évaluées sur une période fixe à planifier, mais elles tiennent aussi compte de la période antérieure déjà écoulée.

En plus de ces contraintes "non obligatoires", nous avons aussi des contraintes de recouvrement de service. Si d_k est le nombre d'infirmières au travail souhaité le jour k et m_k le nombre minimum requis, on définit une pénalité de recouvrement de la forme:



On pénalise ainsi fortement les écarts négatifs.

Fonction objective:

La fonction objective a pour valeur la somme des deux classes de pénalité: celles provenant de la violation des contraintes "non obligatoires" et celles provenant de l'écart entre le taux réel et le taux désiré d'infirmières au travail, par jour.

*** Pénalité de recouvrement:**

La somme des pénalités journalières est effectuée pour le nombre de jours de la période.

* Pénalités autres:

Pour chaque infirmière $i = 1, 2, \dots, I$, le coût pour un modèle particulier x_i mesure:

- 1) la pénalité de ce modèle par rapport aux contraintes non obligatoires violées.
- 2) l'appréciation de ces violations par rapport aux préférences de l'infirmière i et aux autres contraintes non obligatoires non violées.
- 3) comment cette pénalité peut être modifiée par rapport à l'"historique" de l'infirmière. Si l'infirmière a eu des modèles de travail "déplaisants" ou pauvres, le coût de pénalité du modèle de travail étudié aura intérêt à être assez élevé pour être évité lors de l'appréciation du programme et réciproquement.

Aussi nous définissons

$g_{in}(x^i)$ la pénalité de violation de la contrainte n de l'ensemble des contraintes non obligatoires pour le modèle x^i .

c_{in} le poids que l'infirmière i donne à la violation d'une contrainte n .
(coefficient d'aversion).

A_i un taux d'équilibre, (égalité) par rapport à l'historique plus ou moins bon de l'infirmière.

Le coût de pénalité pour cette partie sera, pour une infirmière i et un modèle x^i

$$A_i \sum_n b_{in} g_{in}(x^i) \quad \text{pour les } n \text{ contraintes non obligatoires}$$

et l'on sommerá ces coûts pour toutes les infirmières i .

FORMULATION DE LA FONCTION OBJECTIVE

On choisit un paramètre a appartenant à $(0,1)$ de façon à ce que la pénalité de recouvrement et la pénalité de violation de contraintes s'équilibrent.

Le problème revient à définir les modèles de travail,

$$x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,14}) \text{ (si la période est de 14 jours)}$$

de façon à minimiser:

$$a * (\text{pénalité de recouvrement}) + (1-a) * (\text{pénalité des contraintes non obligatoires})$$

I-4-b Description de l'algorithme:

L'algorithme proposé trouve une solution proche de la solution optimale.

Il démarre avec une configuration de modèles de travail, un pour chaque infirmière. En fixant tous les modèles de travail à l'exception de celui de l'infirmière i , il génère P_i .

Si en construisant P_i , on trouve un modèle qui entraîne un coût de pénalité total inférieur au précédent, on mémorise ce modèle à la place de l'ancien.

Le processus est repris en changeant l'infirmière i et se termine lorsqu'aucune configuration meilleure n'est trouvée pendant un nombre de tests consécutifs égal au nombre d'infirmières.

La façon de définir les contraintes "obligatoires" entraîne que chaque P_i contient au moins un modèle de travail possible.

Pour arriver à une solution initiale, on doit sélectionner un modèle de travail dans chaque P_i d'une manière appropriée. (par exemple sélectionner le modèle de travail avec le plus petit coût d'insatisfaction).

La convergence de l'algorithme est assurée car c'est une heuristique améliorante sur un nombre fini de solutions:

$$\overline{\prod_{i=1}^{\text{nb inf.}} || P_i ||} \quad \text{avec} \quad || P_i || = \text{Card}(P_i)$$

Afin d'obtenir de meilleurs résultats (plus de rapidité), P_i doit être réduit le plus possible, soit par des requêtes spéciales, soit par un ensemble de contraintes obligatoires plus affinées.

I-4-c RESULTATS:

Une comparaison a été effectuée sur un exemple simple avec un algorithme de "Branch and Bound" qui permettait d'obtenir la solution optimale.

Les tests effectués démontraient que cet algorithme générait des plannings "presque aussi bons" en un temps beaucoup plus restreint.

Par exemple, sur un exemple de 4 infirmières avec une moyenne de 20 modèles dans les P_i , une solution avec une fonction objective de 12,3 est trouvée en 0,367 sec de temps CPU sur CDC 6400 alors que le coût de la solution initiale était de 239,45.

La solution optimale, de coût 7,55 était précédemment trouvée en 30 sec avec la même solution initiale, et en 10,509 sec si la borne supérieure du Branch and Bound est la solution de coût 12,3.

Cet algorithme peut être étendu pour prendre en compte les rotations dans le service et les différentes tranches de services en redéfinissant les P_i d'une manière appropriée.

Par exemple si l'on considère les rotations

- on apprécie l'algorithme pour pourvoir les services du soir et de nuit en premier.
- Si la contrainte de recouvrement implique des rotations pour réduire la pénalité correspondante et que le service de jour possède des infirmières susceptibles d'effectuer des rotations, on pourra choisir des infirmières disponibles et les faire changer de type de service.

Le modèle de rotation choisi pour celles-ci doit être conforme avec les contraintes de rotation et doit réduire la pénalité de recouvrement de la configuration totale.

- On planifie alors le service de jour en considérant ces rotations comme des conditions fixées.

On peut aussi aménager cet algorithme pour les employés à mi-temps.

Cette méthode présente donc un modèle mathématique formulant le problème de constitution de plannings comme un problème de sélection d'un ensemble de modèles de travail minimisant à la fois les coûts "d'aversion" du personnel et les coûts et désagréments entraînés par un personnel trop ou trop peu nombreux.

Ce modèle est relativement souple dans le sens où il peut accepter tout type et n'importe quel nombre de découpages.

La manière d'intégrer la contrainte de recouvrement semble beaucoup plus souple que dans la méthode WARNER, car elle autorise, vis à vis de la demande des écarts négatifs par rapport à la moyenne désirée.

Les deux pénalités, celle engendrée par la contrariété des désirs et celle de non recouvrement optimal sont mieux opposées dans ce modèle.

D'autre part, une solution "acceptable" est trouvée sans beaucoup de difficultés grâce à la séparation en deux niveaux de contraintes

- les "non violables".
- les violables avec pénalité.

Cette approche nous montre bien la complexité d'un modèle voulant intégrer les désirs du personnel sous forme de pénalités. La traduction de l'insatisfaction de l'infirmière et de ses préférences par des simples coûts de pénalité, est d'ailleurs un problème délicat, et son intégration dans un modèle mathématique est souvent trop rigide.

I-5 Méthode Douglas SMITH et A. WIGGINS:

En 1977, Douglas Smith et A. Wiggins généralisèrent le problème et présentèrent un système informatique établissant un planning d'affectation de personnel sur la base d'un mois. Cette méthode donne la possibilité de reproduire les procédures utilisées dans les méthodes manuelles par les personnes expérimentées.

L'objectif n'est plus d'aboutir directement à l'élaboration d'un planning complet, mais de proposer dans un premier temps un planning qui sera une aide à la mise en place d'actions "remèdes" pour les points critiques obtenus par anticipation.

Ce système fut conçu afin d'utiliser une capacité ordinateur modeste, d'incorporer les préférences individuelles du personnel pour certains jours de congé et certains types de service, d'incorporer les demandes spéciales, les vacances, et les jours récupérateurs et de considérer les situations spéciales dans une équipe, comme la capacité qu'a une personne expérimentée de compenser l'absence d'un personnel plus qualifié.

Smith et Wiggins retinrent le modèle proposé par WARNER [WA] et précédemment exposé, avec son approche d'optimisation sur une période d'un mois.

Résumons ce modèle de programme en nombre entiers:

$$\text{Maximiser} \quad \sum_i \sum_j C_{ij} X_{ij} + \sum_k \sum_d g(S_{kd})$$

sous les contraintes

$$\sum_i \sum_j A_{ij} X_{ij} - S_k = \text{Min}_{kd} \quad \begin{matrix} k=1,2,3 \\ d=1,\dots,14 \end{matrix}$$

$$X_{ij} = (0 \text{ ou } 1)$$

avec

$$X_{ij} = \begin{cases} 1 & \text{si l'infirmière } i \text{ est affectée au modèle } j \\ 0 & \text{sinon} \end{cases}$$

A_{ij} est une colonne de 0 et de 1 correspondant à l'emploi du temps j de l'infirmière i .

Min_{kd} nombre minimum d'infirmières nécessaire dans l'équipe effectuant le type de service k du jour d

S_{kd} surplus d'infirmières sur le service k du jour d .

$g(S_{kd})$ valeur affectée à l'écart par rapport au nombre minimum d'infirmières nécessaires.

$k = 1,2,3$ si trois équipes (tranches horaires) différentes.

$d = 1,2,\dots,14$ car la période de planning est coupée en deux fois 14 jours.

Le coeur du problème est assez bien représenté dans cette formulation. Les A_{ij} peuvent être générés pour satisfaire les contraintes générales et pour refléter les demandes individuelles pour certains jours de congé et certaines affectations à des services particuliers.

Cependant, dans ce modèle, nous n'arrivons pas à traduire certains choix opérés en général par des personnes expérimentées constituant les plannings manuellement: des facteurs ne sont pas traduisibles. Par exemple la faculté d'une élève infirmière de supporter exceptionnellement la charge d'une infirmière ne peut être pris en compte.

De plus des "entorses" aux règles générales sont fréquemment faites pour répondre à une requête spéciale ou devant des circonstances particulières pour le service.

Lorsque nous essayons d'élargir la formulation pour prendre en compte ces considérations, l'énoncé mathématique du problème devient incommode. On atteint aussi rapidement les limites de capacité de l'ordinateur (IBM 170/125), car le problème en nombre entier demanderait des ressources supérieures à celles disponibles dans la plupart des hôpitaux.

I-5-a L'approche adoptée:

Peut-être que la caractéristique la plus significative de la constitution des plannings est la tendance à débiter avec un **ensemble de spécifications trop serré**, puis de relâcher certaines contraintes lorsqu'il devient évident qu'elles ne peuvent pas toutes être respectées.

Il semble cependant impossible de définir une hiérarchie simple ou un ensemble de priorités pour effectuer une relaxation automatique des contraintes.

Cette vision du problème des contraintes trop serrées, ainsi que la difficulté de prendre en compte dans un programme mathématique des contraintes plus compliquées poussèrent Smith et Wiggins à adopter une approche heuristique utilisant des listes de modules de traitement et une structure de données comme le suggérait MACK et SMITH [MA-SM].

La fonction de l'heuristique est de suppléer aux jugements des personnes établissant les plannings en leur permettant de se concentrer sur l'analyse des points critiques où la relaxation des contraintes peut avoir causé un déséquilibre.

Avantages de la structure de données et des listes de traitements:

Ces techniques sont en général conçues pour faciliter le travail de l'individu qui doit transformer un algorithme donné en un programme opérationnel. Elles permettent:

- Un très haut niveau de modularité. Les étapes du programme étant décomposées en combinaisons de quelques opérations fondamentales sur les structures de données.
- Une modification facile de l'algorithme pour considérer d'autres facteurs et contraintes.
- Une génération facile d'édicions, avec un nombre minimum de transformations demandées dans la préparation des sorties.
- La possibilité de reproduire les procédures utilisées par les personnes expérimentées dans les méthodes manuelles.

Plutôt que de considérer le problème comme un problème d'énumération des modèles de travail possibles pour chaque infirmière, puis de sélectionner la meilleure combinaison dans les ensembles de solutions possibles, les modèles individuels sont générés dynamiquement, semaine par semaine.

Des **essais** sont effectués pour accommoder les préférences individuelles pour chaque affectation, avec les limites des contraintes "courantes".

Si toutes les contraintes ne peuvent pas être satisfaites par les affectations possibles, une contrainte est relâchée et la recherche est répétée pour trouver l'affectation la plus adéquate à l'ensemble réduit des contraintes.

Pour considérer le désagrément de chaque infirmière devant son modèle de travail personnel, et pour pouvoir prendre en compte l'éventuelle aversion plus tard, chaque infirmière dispose de 10 points d'aversion à répartir entre les trois différentes tranches horaires.

Un indice d'aversion au type de service est remis à jour par rapport à la nouvelle affectation.

Chaque semaine, l'infirmière possédant l'indice le plus haut est affectée en priorité au service de son choix. Un indice identique est tenu et mis à jour pour les choix des jours de repos.

I-5-b L'algorithme:

L'algorithme opère en mode "batch" selon trois phases:

1- La production d'un état sommaire de répartition du personnel pour la semaine sur chaque tranche horaire (équipe), une fois que les vacances ou jours de congés sont déterminés.

Ceci est effectué pour encourager la redistribution du "surplus" de personnel aux services qui ont, par expérience des difficultés de recouvrement, avant de générer les plannings détaillés.

- 2- Génération de plannings d'essais indiquant le manque ou surplus de personnel par jour et identifiant le cas où les contraintes sont violées.
- 3- Ajustement manuel sur les plannings d'essais et constitution du planning final.

Un mécanisme est mis en place pour lister les individus capables de remplacer d'autres personnes d'une catégorie différente.

Par exemple, si une infirmière particulière est capable de se substituer à un superviseur, cette infirmière et le superviseur doivent être spécifiés sur une liste spéciale d'équivalence.

Différentes options donnent une puissance considérable à l'utilisateur pour déterminer le planning final.

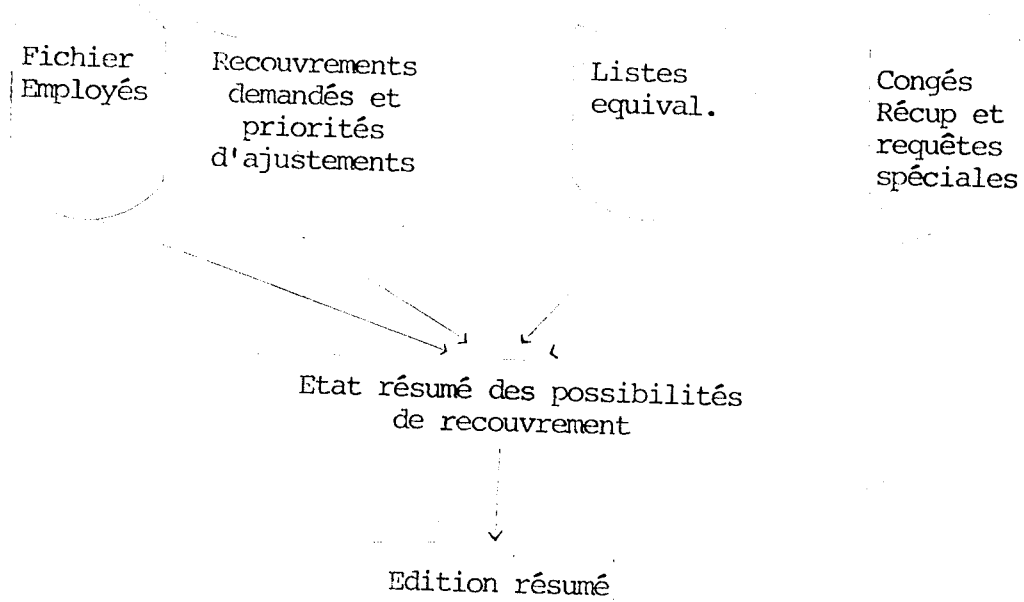
Modules de l'algorithme:

Chaque module effectue une opération simple sur les éléments de la structure des données, le plus souvent en reproduisant une méthode de la solution manuelle.

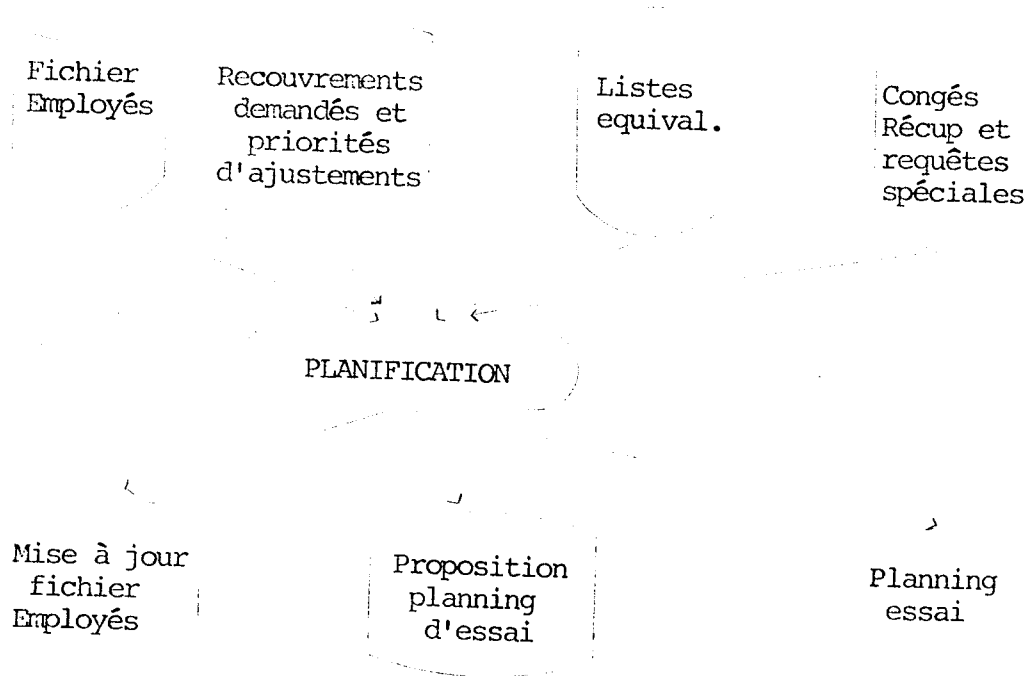
Une procédure d'évaluation est toutefois incorporée à l'algorithme, les modules correspondant évaluent ainsi les conséquences d'un acte avant de l'effectuer de manière irrémédiable.

L'algorithme final:

PHASE 1 (optionnelle):



PHASE 2 :



PHASE 3 :

Planning
d'essai
proposé

Fichier employés
mis à jour

Modifications
dans planning

MODIFICATIONS

Planning modifié
+ messages

La plus grande demande au niveau capacité ordinateur est effectuée lors de la phase 2, lorsque l'algorithme de remplissage du planning est employé.

L'exécution de la phase 2 (compilée par FORTRAN G) demande environ 120 KBytes de mémoire centrale et une seconde de temps CPU sur un IBM 370/168 pour une unité fonctionnelle de 35 employés).

On peut aussi exécuter ce programme en différé ou en partition secondaire.

Ce système a été implanté à l'Hôpital Juif de St Louis.

Entre 5 et 12 % des personnes semblent être habituellement "modifiées" dans la troisième phase. Les ajustements et modifications manuelles semblent prendre 20 à 110 mn pour une UF de 35 personnes.

Cette dernière approche relève bien la difficulté à traduire l'expérience des personnes constituant les plannings dans une formulation mathématique.

Les contraintes "obligatoires" ne peuvent généralement pas être relâchées de manière acceptable, l'exception ne devant pas devenir coutume...

On se rend compte de la nécessité de faire intervenir les personnes chargées de la planification.

Cette approche est donc nouvelle dans sa conception: le but n'est plus de constituer un planning définitif par ordinateur, mais de proposer un planning temporaire, modifiable de manière interactive si possible, facilitant énormément la constitution du planning définitif.

CHAPITRE II: ERGONOMIE ET INTERACTION

Nous devons concevoir et réaliser un outil informatique rendant plus fiables et plus faciles:

- l'élaboration, la mise au point et la modification au fur et à mesure, de trois plannings:
 - * le planning "en cours", d'une durée de vie de quatre semaines. Les principales modifications sur ce planning étant liées à l'activité réalisée de chaque infirmière, activité souvent différente de celle prévue.
 - * le planning "futur proche" représentant le prévisionnel de la prochaine période.
 - * le planning "futur lointain", planning en cours d'élaboration concernant la prévision à long terme.

- le suivi des infirmières à l'aide de fiches à partir du planning en cours (stocké en mémoire).

Les diverses fiches ayant pour source les plannings, il semblait intéressant de développer un module "interface" entre les plannings et les fiches des infirmières.

Cette interface évitait la double saisie (planning-fiches) d'une grande partie de l'activité de l'infirmière.

L'outil à développer devait s'intégrer dans un service possédant ses traditions, ses habitudes, et être utilisé par plusieurs personnes de ce service (suivi de l'activité effectué quotidiennement).

L'interactivité et la convivialité d'un "programme Micro" semblaient des atouts majeurs, permettant à la fois des modifications "en temps réel" et un suivi journalier de l'activité des infirmières.

Un effort particulier dans l'ergonomie du logiciel a été effectué. Une enquête préalable sur le travail effectif du service nous donna une vision précise des besoins exprimés ainsi que les spécifications détaillées pour l'élaboration d'un outil interactif, prototype réutilisable sur tout service.

La validation de l'outil par les opérateurs devait être effectuée à la suite de ce travail. Seule une présentation à différents responsables de services médicaux a pu être effectuée pour des raisons personnelles de changement de situation: quittant la région, il ne m'était plus possible d'assurer le suivi minimum à la mise en route du logiciel.

II-1 CONTRAINTES MATERIELLES:

L'essai d'élaboration de planning de travail sur micro-ordinateur était toute nouvelle et ne pouvait pas, dans sa réalisation être comparée aux approches précédentes. Les ressources dont nous disposions ne nous permettaient pas d'effectuer une approche aussi coûteuse en temps de calcul et en capacité mémoire que celles présentées dans les méthodes précédentes.

Le service disposait d'un micro-ordinateur "APPLE IIe" avec deux lecteurs de disquettes.

L'emploi de la carte graphique "CHAT MAUVE", avec l'utilisation des couleurs, semblait intéressante pour la visualisation d'un planning à l'écran, car elle nous permettait d'établir un écran planning très proche de la feuille. Ce sujet retint donc dans un premier temps notre intérêt par son côté technique: arriver à élaborer un logiciel graphique couleur sur cette gamme de matériel.

Le logiciel a donc été développé en PASCAL UCSD et en ASSEMBLEUR sur un "APPLE IIe" muni d'une carte "CHAT MAUVE". Cette configuration est maintenant disponible en standard sur un "APPLE IIc" ou un "APPLE IIGS".

La carte graphique permet de plus

- la visualisation du texte en 80 colonnes.
- une extension mémoire (utilisation du Pascal 128K)
- une visualisation couleur permettant d'affecter une couleur à chaque information à porter sur le planning -Amélioration ergonomique importante- .
- une extension des possibilités graphiques de l'"APPLE IIe".

II-2 ORGANISATION DU SERVICE:

La nécessité première d'un service de soins étant de fournir un service continu (ou de maintenir les services essentiels), l'organisation du travail du service de réanimation a été conservée. Elle permet à des équipes successives d'occuper le même poste de travail, ce que l'on nomme organisation d'une rotation de "postés" en continu. Cette organisation se fonde, en milieu hospitalier sur la prise en compte de contraintes de nature sociale(taux d'absentéisme, obligation d'équilibrages négociés pour le roulement des équipes, règlement intérieur..) ou juridique (durée légale de travail, aménagement du temps de travail). L'équilibre entre ces contraintes est en général le résultat de négociations effectuées par des partenaires sociaux classiques: direction, syndicats,...

C'est pourquoi l'introduction de l'informatique, bien que laissant au départ envisager une plus grande souplesse dans l'affectation du personnel et les horaires de travail, n'a pas remis en cause une organisation

tenant compte de l'extrême diversité des équilibres négociés entre ces contraintes.

Le problème annexe d'une organisation entre services, pour une meilleure répartition de la force de travail, bien qu'intéressant au niveau de la recherche, n'a pas été abordé dans ce travail.

Le service de réanimation concerné dispose d'une quarantaine d'infirmières réparties entre trois différentes tranches horaires de travail:

- * Service MATIN : 7h - 15h
- * Service SOIR : 13h30 - 21h30
- * Service NUIT : 21h15 - 7h15

Les plannings sont élaborés sur une période de 4 semaines appelée "roulement".

Chaque infirmière remplit un formulaire où elle exprime ses désirs concernant le type de service auquel elle veut être affectée et les dates de congé souhaitées pour une période (congé annuel, récupération,...).

Règles de fonctionnement:

Certaines règles sont établies à l'intérieur du service ou de l'hôpital:

- 1) Une infirmière en congé ne peut pas être rappelée.
- 2) Un jour férié est récupérable s'il n'est pas dans une période de congés exceptionnels.
- 3) Les congés annuels ne peuvent pas être récupérés au cours de l'année suivante contrairement aux jours fériés qui peuvent l'être.
- 4) Une infirmière doit faire, 75 nuits par an en cinq tranches de 15 nuits successives. Ces nuits sont effectuées par permutations: une infirmière de jour permute avec une infirmière venant d'effectuer sa quinzaine de nuits.

5) Un congé de deux semaines pris hors période d'été donne droit à deux jours de congés supplémentaires.

Méthode cyclique de base:

Une permanence de 24h sur 24 doit être assurée dans le service, ceci 365 jours par an, ce qui implique aussi un nombre minimum d'infirmières présentes.

Pour ces raisons, les congés hebdomadaires ou jours de repos, ne coïncident pas obligatoirement avec la fin normale de la semaine.

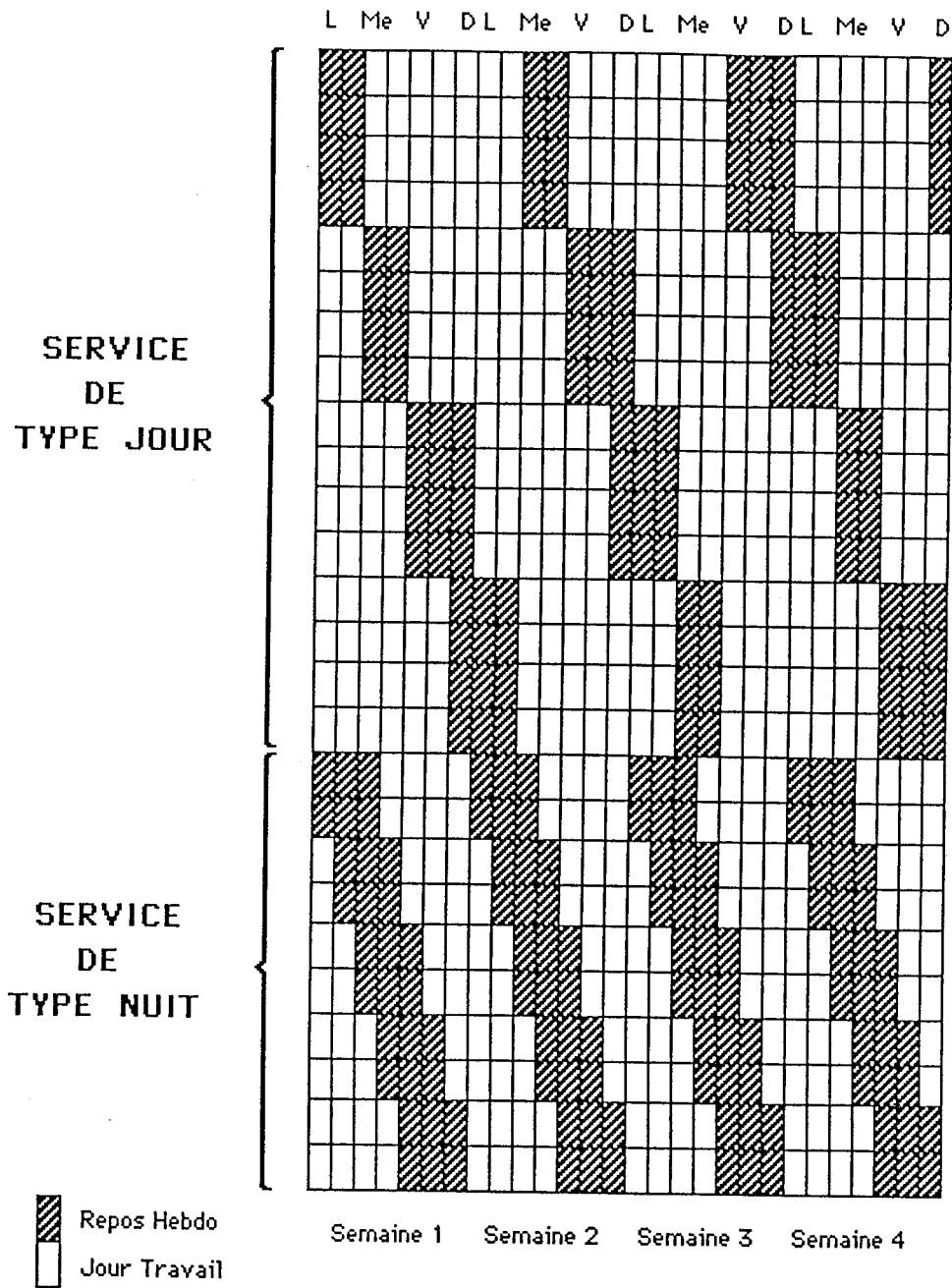
Afin de faciliter la couverture régulière des besoins, un service type de "jour", cyclique sur 4 semaines a été défini:

	L	Ma	Me	J	V	S	D	
Semaine 1	0	0	1	1	1	1	1	
Semaine 2	1	1	0	0	1	1	1	0= Repos
Semaine 3	1	1	1	1	0	0	0	1= Travail
Semaine 4	1	1	1	1	1	1	0	

Pour le service de nuit, la durée du travail étant 10 heures, le service type est de 4 jours de travail et 3 jours de repos sur une semaine, indépendamment de la place du week-end.

Un roulement est une succession cyclique des quatres configurations ci-dessus.

Pour chaque tranche de 4 semaines, le service de réanimation produit un planning vierge avec implantation des jours de repos et des différents services matin, soir, nuit.



Le problème revient alors à affecter chaque infirmière à une ligne du planning. Le choix de la ligne est l'un des premiers risques de conflits, car les désirs ou droits des infirmières peuvent s'opposer aux besoins du service.

II-3 ENQUETE SUR LE TRAVAIL EFFECTIF DU PERSONNEL CHARGE DE L'ORGANISATION DU SERVICE:

Les responsables administratives essayent, à partir du formulaire rempli par chaque infirmière, de constituer le planning idéal des activités de toutes les infirmières sur une période.

Ce planning doit, respecter la **contrainte de recouvrement**:

- 6 infirmières minimum pour les services de jour
- 9 pour celui de nuit

L'opération de mise en place du planning peut être défaite et refaite jusqu'à ce que le planning ainsi constitué réponde aux besoins du service.

En parallèle, une autre personne s'occupe de la mise au point, au jour le jour du planning courant et de la mise à jour des différentes fiches infirmières. Ces fiches sont destinées à la gestion du personnel (heures supplémentaires, absence...)

Documents produits:

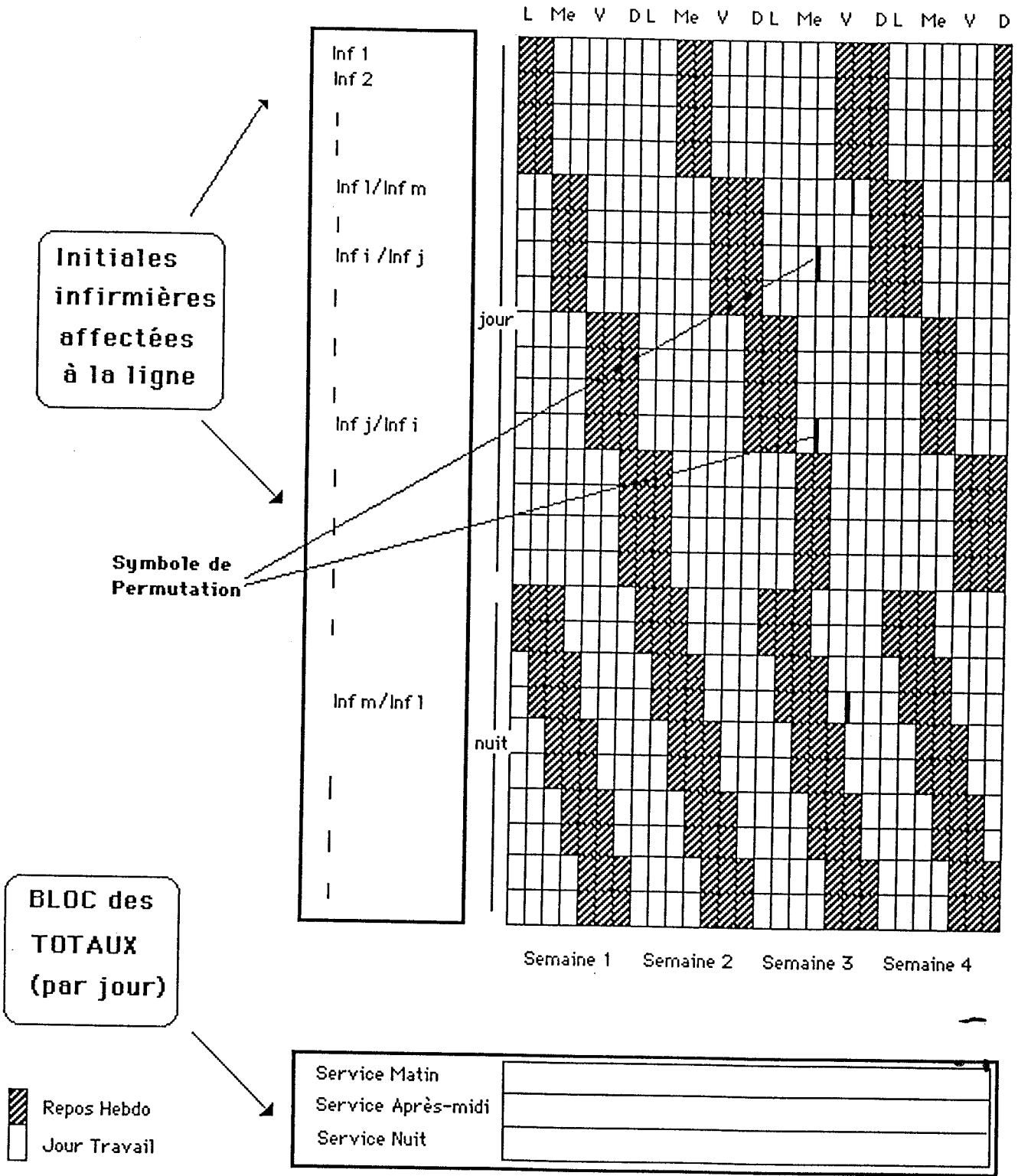
Le bilan des activités des infirmières figure sur un planning ayant pour support une grande feuille.

Un planning se présente sous forme d'un tableau à deux entrées, résumant l'activité du service sur la période de quatre semaines civiles(voir annexe 1).

On distingue deux parties distinctes sur chaque feuille de planning:

- Planning JOUR qui regroupe l'ensemble des infirmières affectées aux services Matin et Soir.
- Planning NUIT correspondant à l'activité des infirmières affectées au service de nuit.

Sur la feuille support ces deux types de plannings sont représentés l'un à la suite de l'autre.



En bas de chaque planning, une ou deux lignes "Total" indiquent le nombre total d'infirmières en activité par jour et par service.

L'objectif est d'amener ce total à être voisin du nombre souhaité d'infirmières par service.

En dessous de 5 infirmières il est nécessaire de déroger au règlement du service et d'appeler des infirmières qui sont en congé.

Distinguons les différents états possibles pour une infirmière, avec l'abréviation donnée par la suite:

En activité:

Service Matin	:	Matin
Service Soir	:	Soir
Service Nuit	:	Nuit

En repos:

Congé annuel	:	Congé
Repos hebdomadaire	:	Hebdo
Congé exceptionnel prévu (mariage,...)	:	Exept
Congé exceptionnel imprévu (maladie..)	:	Abs
Récupération	:	Récup
39ème Heure	:	X39 ^e h

a) Congé annuel:

Période inférieure ou égale à 27 jours soumise à la direction pour accord.

b) 39ème Heure:

Au bout de 39 jours travaillés l'infirmière a droit à un jour de congé (car elle travaille 8 h par jour, 1 heure de trop par semaine). Cette journée est généralement accordée dans la semaine suivante.

c) Récupération:

Les jours fériés seront généralement récupérés ultérieurement. Les dates de récupérations sont fixées en fonction du nombre total d'infirmières en service.

- d) Congés exceptionnels prévus:
mariage, maternité ...
- e) Congés exceptionnels non prévus:
maladie, décès ou maladie d'un proche ...
- f) Repos hebdomadaires:
définis par le planning type sur 28 jours.

Ces différents états sont représentés sur la feuille planning par différents codes et couleurs.

Comme nous l'avons vu précédemment trois plannings existent simultanément:

- **Planning en cours:**

Durée de vie de 4 semaines et mis à jour quotidiennement.

- **Planning "futur proche":**

Planning prévisionnel de la prochaine période.

- **Planning "futur lointain":**

Permet la prévision à long terme.

Sur les deux plannings "futur" on note les arrêts prévisibles (congé exceptionnels prévus, récup...).

Les congés exceptionnels imprévisibles perturbent fréquemment la bonne marche du service. Dans ce cas, les responsables sont amenés à mettre à jour de nouveau le planning et à redistribuer le travail sur d'autres infirmières.

Suivi du personnel:

La responsable administrative du service doit constamment se soucier de la tenue de différentes fiches:

1) fiche des congés:

Cette fiche regroupe tous les congés pris par l'infirmière et note les récupérations (jours fériés) auxquelles elle a droit. Les congés exceptionnels sont aussi mémorisés.

2) fiche "heures supplémentaires":

Les dimanches travaillés et les heures supplémentaires effectuées sont mémorisés sur une fiche récapitulative. Les renseignements de cette fiche, ainsi que ceux de la fiche congé, sont nécessaires chaque mois pour le programme de paie.

3) fiche récapitulative sur le service:

Cette fiche sert à évaluer les besoins en personnel du service. Le nombre de nuits effectuées par chaque infirmière est aussi noté.

Ce suivi des infirmières est effectué de manière manuelle et demande beaucoup de manipulations de fiches.

II-4 REALISATION D'UN OUTIL INTERACTIF:

II-4-a UTILISATION DE MENUS

Un menu est une liste d'articles dont on dispose à un instant donné et qui permet d'actionner les commandes accessibles à cet instant. C'est la base de tout programme se voulant ergonomique et interactif puisqu'il permet le dialogue avec l'utilisateur.

Les commandes sont réparties de manière arborescente sur plusieurs niveaux. A un niveau donné de cet arbre de commandes, un menu contiendra un ensemble de commandes.

GESTIONNAIRE DE MENUS:

Dans le but de généraliser le problème à résoudre et de pouvoir l'étendre facilement à d'autres services de soins pouvant fonctionner différemment, la première étape de notre travail fut l'élaboration d'un gestionnaire de menus, générateur d'un programme Pascal, squelette de notre outil interactif.

Copiant le principe (et point fort) des langages de 4ème génération, une indépendance "données - programme" (commandes du menu - gestionnaire) semblait indispensable.

Le gestionnaire de menus va générer un programme de parcours de l'arborescence des commandes.

A chaque appel d'une commande du menu courant, il permet l'exécution d'une procédure correspondant à cette commande, puis continue l'exploration de cet arbre, soit en accédant à un niveau plus bas, ou plus élevé, soit en restant au même niveau.

Cette procédure de parcours d'arbre est indépendante de l'action effectuée par la commande. Il suffit de connaître, pour chaque commande, le "chemin à suivre" après l'exécution de cette commande.

La méthode adoptée ici pour préciser ce chemin est la suivante:

- après l'exécution de la "procédure commande", on descend systématiquement au niveau plus bas s'il existe.
- aux niveaux terminaux deux solutions:
 - * remontée automatique, au niveau précédent
 - * remontée par une commande du menu

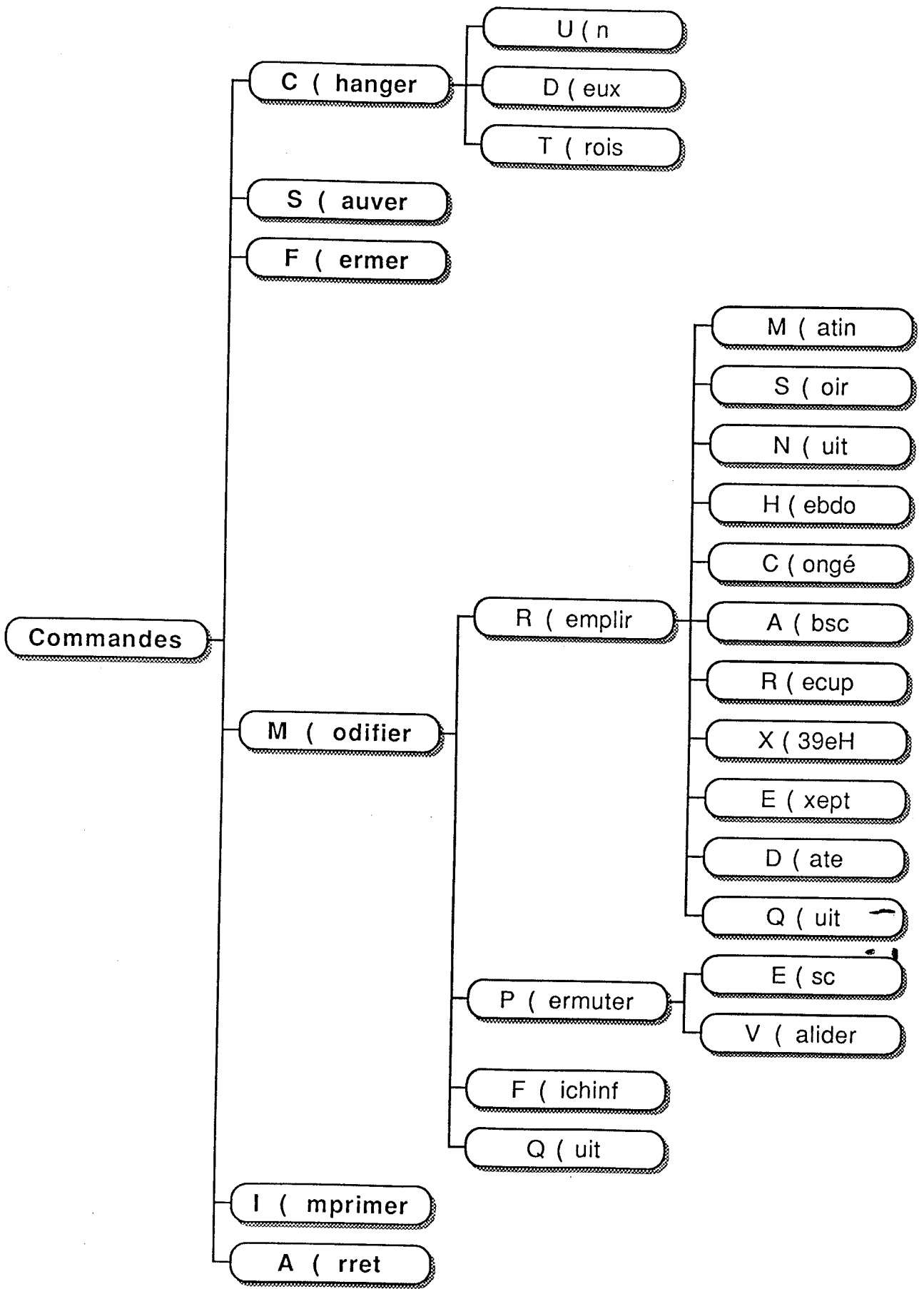
Cette remontée est "pilotée" par une variable logique (booléen) affectée à la valeur désirée dans la procédure commande.

Nous aurions pu introduire cette affectation dans le fichier des données des commandes, mais cette solution nous a paru plus souple.

Si on affecte au booléen la valeur "vrai", on remonte au niveau précédent, sinon on reste au même niveau.

La remontée automatique est permise même si une commande de remontée existe au niveau étudié.

L'exemple de l'arborescence des menus dans notre cas précis est la suivante:



Pour le cas des plannings, il est intéressant d'associer à chaque commande une couleur pour la convivialité du logiciel.

Un code de la feuille planning correspondra à une couleur, et la commande affectant cette couleur à une case du tableau sera présentée dans le menu sur fond de même couleur.

Une commande du menu sera sélectionnée par son initiale. A tout niveau, le système est en attente

- soit du choix d'une commande
- soit d'une commande "système"

Les commandes "système" correspondent ici aux déplacements à l'aide des flèches et à la touche ESCAPE. On pourrait améliorer le gestionnaire en ne les permettant qu'à certains niveaux.

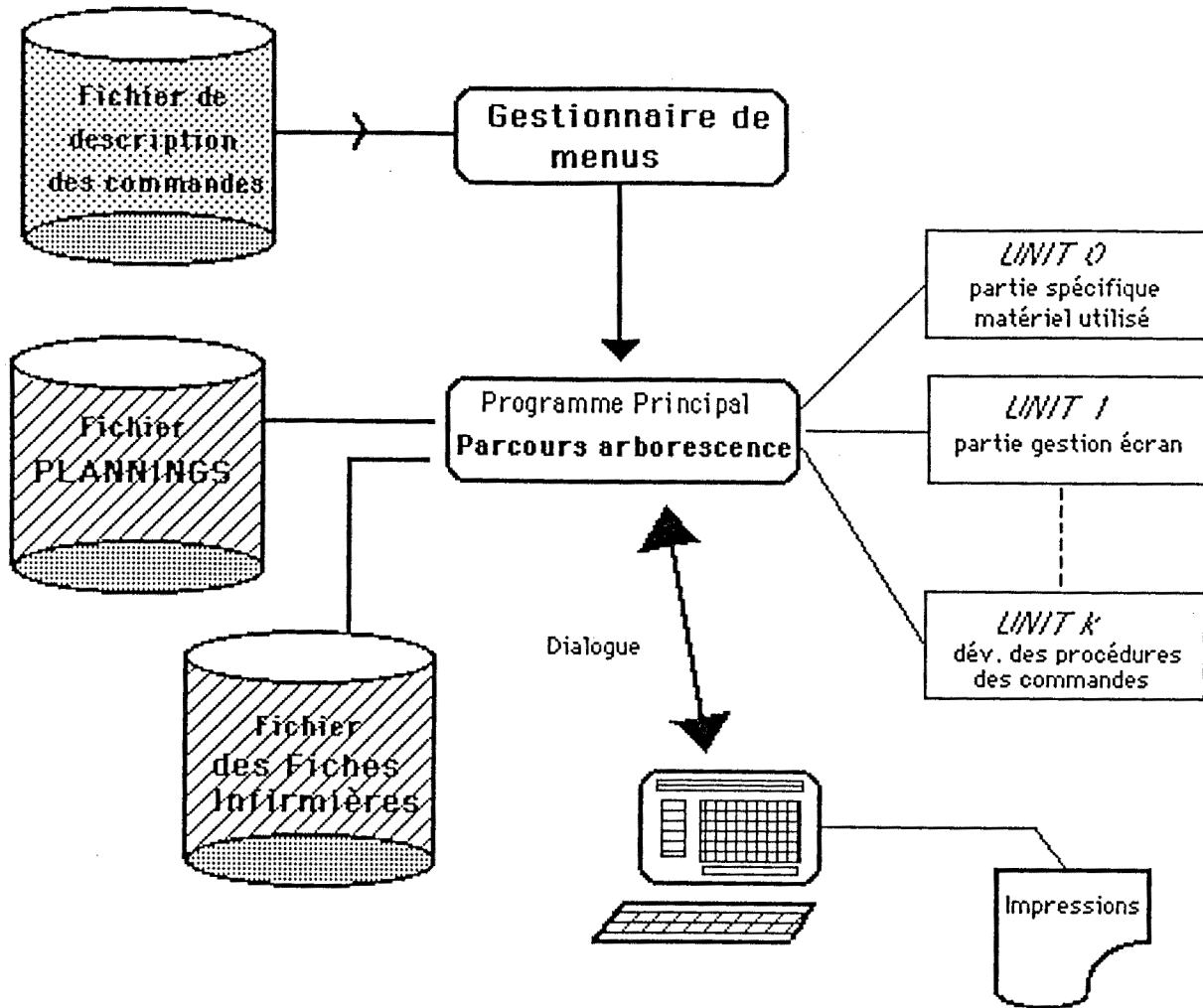
L'atout majeur du gestionnaire de menus présenté ici est de permettre des modifications rapides et une extension immédiate par ajouts de nouvelles commandes (voir annexe 2).

Ce programme a été conçu de la façon la plus modulaire possible afin de permettre une reprise facile sur un autre micro-ordinateur.

Il est évident que les "procédures commandes" devront être développées spécifiquement.

Afin de rendre cette partie modulable, il est possible d'effectuer une édition de lien entre le programme généré et des unités ou bibliothèques où se trouveront ces développements.

Le gestionnaire de menu va donc générer le programme principal qui ne subira plus aucune modification.



Les développements spécifiques seront effectués dans des unités dont on aura précisé le nom à la génération du programme.

Un programme généré est visible en annexe 2.

Pour des raisons de place, nous avons limité le nombre de commandes à 50. Cette limite n'est cependant pas obligatoire.

Ce gestionnaire nous permet donc de modifier à tout instant l'intitulé ou la couleur affecté à une commande ou d'en ajouter ou supprimer une si nécessaire, sans avoir à reprendre le programme principal.

Le développement de la procédure correspondant à une commande supplémentaire reste toutefois à effectuer dans un module séparé!

II-4-b VISUALISATION ET MODIFICATIONS DES PLANNINGS:

Afin de faciliter la lecture et la modification des trois plannings, une représentation graphique en couleur de ceux-ci est disponible sur simple appel par numéro.

La carte graphique "Chat Mauve" met à notre disposition une résolution graphique de 280 * 192 points.

Pour des raisons de convivialité, lecture plus rapide et moins fatigante, nous n'avons fait apparaître que la moitié de la largeur de la feuille à l'écran.

Ceci nous a permis de prendre une largeur double pour les cases représentant les jours et de pouvoir généraliser facilement sur des périodes autres que de 28 jours (largeur des cases paramétrable et longueur de la période indépendante de la visualisation).

La structure des fichiers utilisés est donnée en annexe 4.

De la même manière toutes les lignes du planning, à raison d'une ligne par infirmière, ne peuvent être visualisées simultanément.

En tenant compte de notre présentation, l'affichage de l'affectation de 14 infirmières pendant 14 jours est possible en un seul écran.

Chaque infirmière est identifiée dans le planning par un indice. Le fichier infirmière, que nous décrirons plus tard, a comme clé d'accès les initiales caractéristiques de l'infirmière.

Ces initiales sont d'une utilisation beaucoup plus simple pour la personne constituant le planning.

Une correspondance indice-initiales a été effectuée et mémorisée au lancement du programme.

Voici un exemple d'image écran

Chaque jour du planning est représenté par un rectangle (14 * 7 points). On affecte une couleur à ce rectangle lorsque l'on affecte un état (congé annuel, service matin, service soir,...).

Utilisation des touches curseurs pour se déplacer dans le tableau

On se déplace sur la case à affecter à l'aide des flèches horizontales et verticales. Le curseur est représenté sur le planning par une flèche orientée vers le haut qui s'imprime en transparence sur la case colorée. Le symbole de permutation peut être sur-imprimé sur le début d'une case. Il indique une permutation service de nuit, service de jour.

Afin de pouvoir visualiser rapidement le reste de la feuille planning, (autres jours et autres infirmières), nous avons dû développer des programmes en assembleur de "scrolling" (déplacement) horizontal et vertical. La programmation en Pascal de ces opérations demandait des temps d'attente trop longs et inacceptables.

En déplaçant le curseur vers la droite alors qu'il se trouve sur la dernière case visible à droite, on déplace l'ensemble des jours visibles vers la gauche en perdant la visibilité du premier jour, et on accède instantanément au jour suivant: c'est le scrolling horizontal vers la droite (déplacement de la fenêtre sur la feuille planning).

De la même manière on peut se déplacer vers la gauche, le haut et le bas. Un exemple de ces programmes de scrolling est disponible en annexe 5.

La mise à jour des totaux par type de service est aussi simultanée: l'affectation d'un état à une case remet à jour automatiquement les totaux journaliers. Nous permettons ainsi à l'utilisateur de voir très simplement le résultat d'une modification et ses conséquences vis à vis de la contrainte de recouvrement.

Un complément à l'énoncé de ces quelques principes fondamentaux d'utilisation, est présenté sommairement en annexe 6. L'ensemble des commandes disponibles en suivant l'arborescence décrite précédemment est proposé.

Une sortie imprimante du planning est exécutable sur demande. Le service ne disposant pas d'une imprimante couleur, celles-ci sont remplacées par des codes. Un exemple est proposé en annexe 7.

II-4-c GESTION DE LA FICHE INFIRMIERE:

L'utilisateur a accès un menu principal. Une description de la structure des fichiers est disponible en annexe 8.

- A- CREATION D'UN ENREGISTREMENT INFIRMIER(E)
- B- VISUALISATION D'UN ENREGISTREMENT INFIRMIER(E)
ET MODIFICATIONS.
- C- SUPPRESSION D'UN ENREGISTREMENT INFIRMIER(E)
- D- FIN

Sans nous occuper des relations Planning - Fiches que nous détaillerons plus tard, expliquons les fonctionnalités de ce menu.

A- CREATION D'UNE INFIRMIERE:

L'infirmière va être identifiée par trois lettres.

On propose un écran de saisie:

- 1- Initiale
- 2- Nom
- 3- Prénom
- 4- Récup.préc. (si l'infirmière vient d'un autre service)
- 5- Début de service
- 6- Fin de service
- 7- Type de service (servira à la première affectation à une ligne du service)

Comme pour chaque grille de saisie une modification est possible avant de valider cette grille. Les numéros servent à obtenir directement un positionnement sur l'élément à modifier.

Une vérification est effectuée à chaque niveau.

- sur les initiales pour vérifier la non existence car c'est la clé d'accès.
- sur le nom, prénom selon les caractères permis dans ces champs.
- sur les dates pour vérifier la syntaxe "date"
- sur le type de service pour vérifier ce type.

Si ce contrôle s'avère négatif, une ligne de détail sera imprimée sous le champ non correct.

Une fois les modifications éventuelles effectuées, on propose un retour au menu général ou une sortie du programme. La création est effective.

B- VISUALISATION OU MODIFICATION :

La modification est permise à tous les niveaux de la fiche pour laisser toute la souplesse possible aux responsables administratives.

On accède par cette commande à tous les renseignements mémorisés pour chaque infirmière.

L'accès à l'enregistrement se fait par la clé (les trois initiales).

On peut visualiser ensuite trois parties de la fiche qui correspondent à trois pages écrans distinctes:

1) La fiche d'état civil de l'infirmière qui est la fiche saisie lors de la création.

On peut indiquer ici la date de sortie de l'infirmière si celle-ci quitte le service.

2) La table des récupérations

L'ensemble des dates pouvant être "récupérables" est affiché.

A côté de chaque date figure, soit une date de récupération, soit la même date (si l'infirmière n'avait pas le droit à cette récupération), soit une date vierge (la date n'est pas encore récupérable ou non encore récupérée).

On visualise aussi le code d'affectation.

Cette table est remplie,

- à l'aide du fichier des dates récupérables pour les dates donnant droit à une éventuelle récupération

- à l'aide du planning individuel de l'infirmière qui traduit son activité.

Un champ supplémentaire est visible: le nombre de récupération de l'année précédente restant à prendre .

Les récupérations sont en effet reportables sur les années suivantes.

Ce nombre sera décrémenté en priorité si l'infirmière prend des récupérations.

Dans cette table, les renseignements relatifs à l'infirmière sont tous modifiables. Une idée de la page écran est donnée dans l'édition de la fiche infirmière (Partie "dates des jours récupérés" en Annexe 9).

3) Le planning individuel:

La visualisation de l'activité annuelle est intéressante et ne sert en général qu'à vérifier une contestation.

Afin de pouvoir visualiser l'activité sur un seul écran et de l'éditer sur une seule feuille, nous avons défini des abréviations des codes état.

-	: congé hebdomadaire	: ---
N	: service Nuit	: Nui
M	: service Matin	: Mat
S	: service Soir	: Apm
H	: 39e heure	: H39
A	: congé except non prév	: Abs
R	: récupération	: Rec
C	: congé annuel	: Cog
E	: congé except prév	: Exp

Toutes ces abréviations sont paramétrables. Elles se trouvent dans un fichier (Fcode)

Un fichier (Fmois) est de la même manière utilisé pour les libellés et abréviations de mois.

Le planning individuel par infirmière est donc la mémorisation sur l'année civile de l'activité de l'infirmière. Des renseignements supplémentaires sont donnés:

- le nombre de nuits déjà effectuées et restant à effectuer.
- le nombre de dimanches travaillés.
- le nombre de congés annuels pris et restant.
- le nombre de congés exceptionnels pris.

C'est la feuille de base de renseignements nécessaires pour la gestion du personnel (paie,...)

Une idée de la page écran est donnée en annexe 9 (Partie "planning infirmière" de la fiche).

La modification de ce planning est aussi possible en donnant la date du jour à modifier.

Le nombre de nuits effectuées, le nombre de dimanches travaillés, le nombre de congés annuels pris, le nombre de congés exceptionnels sont mis à jour automatiquement sur la base du planning individuel.

Les autres champs, nombre de nuits à effectuer, nombre de congés annuels restant à prendre, sont modifiables directement.

C- SUPPRESSION D'UN ENREGISTREMENT INFIRMIER(E)

Afin de pouvoir récupérer la place mémoire lors du départ d'une infirmière, la possibilité de supprimer un enregistrement est donnée à l'utilisateur.

Les initiales de l'infirmière à supprimer sont demandées et une confirmation est nécessaire.

D- FIN

Permet de sortir du logiciel en proposant un menu d'impression.

Menu des différentes impressions :

- 1- Impression d'un enregistrement infirmier(e).
- 2- Impressions des enregistrements infirmier(e)s.
- 3- Impression de la liste des infirmier(e)s.
- 4- Impression de la liste des récupérations.
- 5- Sortie.

- 1- Impression d'un enregistrement infirmier(e).

L'édition est visible en annexe 4. On édite la fiche infirmière complète pouvant servir de support au programme de paie.

- 2- Impressions des enregistrements infirmier(e)s.

Impression de toutes les fiches infirmières (par exemple pour archive à la fin d'une année).

- 3- Impression de la liste des infirmier(e)s.

- 4- Impression de la liste des récupérations.

Cette impression sort, pour chaque date récupérable, la liste des infirmières ayant récupéré avec la date de récupération correspondante et la liste des infirmières n'ayant pas droit à cette récupération.

Respectant la même philosophie que précédemment, ce programme est constitué de modules distincts permettant une reprise plus facile. La gestion des écrans et le contrôle des saisies sont regroupés dans une même unité.

La carte graphique n'est pas utilisée ici, si ce n'est pour son extension 80 colonnes.

II-4-d INTERFACE FICHE INFIRMIER(E) - PLANNING

Nous avons vu que la clé d'accès au fichier infirmière est la chaîne des trois initiales alors que la clé d'accès du planning est l'indice de l'infirmière (pour des raisons de place mémoire).

Un fichier interface est constitué pour relier ces deux notions.

Structure de l'enregistrement:

- un champ initiales de 3 caractères
- un indice (numéro séquentiel de création de l'infirmière)
- la ligne de départ sur le planning 1
- le type de service d'affectation au départ (planning 1)

Ce fichier va servir de base pour remplir la partie "Initiales" du planning écran et pour déterminer le nombre de lignes de ce planning.

Lors de la création d'une infirmière dans le fichier infirmière, une ligne va être automatiquement générée dans le planning, avec pour type de service celui affecté à la nouvelle infirmière.

Ce fichier va donc être mis à jour par la partie gestion de fiches infirmières, puis, lors d'une fermeture de planning 1, il sera remis à jour par la partie gestion de planning.

C'est la clé de l'interface entre le planning et la fiche infirmière. Afin de minimiser les temps d'accès au fichier infirmière, on mémorise, par l'intermédiaire de ce fichier, un ensemble de 30 infirmières "courantes".

Un infirmière est une infirmière "courante" si:

- son emploi du temps est visualisé sur l'écran planning (ses initiales apparaissent donc dans la partie "initiales")
- son emploi du temps vient d'être parcouru à l'aide du curseur.

Cette limite à 30 est due à la place mémoire disponible.

Le temps d'attente pour une consultation de la fiche ou pour des vérifications d'affectations est ainsi considérablement amélioré.

Lors de la fermeture du planning en cours, après écoulement de la première période, l'ensemble des fiches infirmières est mis à jour définitivement, alors qu'avant cette fermeture, la mise à jour n'était que provisoire (calculée à partir du planning l et de l'historique de l'infirmière).

L'aspect modulaire, donc facilement modifiable et l'indépendance, dans la gestion du planning, des commandes, par rapport au programme de parcours de l'arborescence, rendent ce logiciel facilement adaptable à d'autres unités de soins.

La possibilité de modifier à tout moment aussi bien le planning que les fiches infirmières, augmentent considérablement la souplesse d'utilisation.

On pourrait aussi introduire dans les vérifications d'affectation des contraintes plus fines, comme l'interdiction d'affecter une infirmière consécutivement à un service nuit, puis à un service jour.

L'expérience des personnes constituant ces plannings, et le fait que d'une manière exceptionnelle ces contraintes puissent être violées nous a fait abandonner cette option qui alourdirait le système.

Ces contraintes ne devront cependant pas être omises lors du remplissage initial du planning.

Ce logiciel permet donc l'élaboration semi-manuelle du planning, puisque la responsable administrative va constituer de toute pièce le planning d'affectation sur une période.

Ceci ne satisfait pas au second objectif visé:

"Comment élaborer un planning "provisoire" répondant le plus possible à la fois aux besoins et contraintes du service, et aux désirs des infirmières?"

III-1 FORMULATION DU PROBLEME

On dispose de n infirmières, numérotées de 1 à n et constituant l'ensemble I des infirmières affectées aux lignes considérées. Sachant que $J=\{1,2,..,28\}$ représente l'ensemble des jours constituant un roulement (le premier jour correspondant à un lundi), le problème consiste donc à répartir, pour chaque infirmière, un nombre $r(i)$ de jours de repos, nombre limité par ses droits, tout en respectant les contraintes de recouvrement et les désirs des infirmières.

Si l'on nomme $y(i,j)$ la variable d'état positionnée à 1 si l'infirmière i bénéficie d'un jour de repos le jour j , à 0 sinon.

Si l'on nomme $C(i,j)$ la pénalité (positive ou négative) affectée par l'infirmière i si le jour j est un jour de repos.

Si $nbInf(j)$ est le nombre minimum d'infirmières exigé le jour j , le problème peut se formuler comme un programme linéaire en variables binaires:

$$\begin{aligned} & \sum_{j=1}^{28} y(i,j) \leq r(i) && \text{pour } i \in I \\ & \sum_{i \in I} y(i,j) \leq |I| - nbInf(j) && \text{pour } 1 \leq j \leq 28 \\ & && y(i,j) \in \{0,1\} \\ & \sum_{j=1}^{28} \sum_{i \in I} (y(i,j) * C(i,j)) = W \text{ (min)} \end{aligned}$$

Afin d'affiner cette formulation, nous pouvons introduire un nombre minimum d'infirmières désirées $nbDesir(j)$ par jour j et une pénalité attribuée si le nombre réel d'infirmières par jour est inférieur à ce nombre désiré.

La fonction objective devient alors:

$$\sum_{j=1}^{28} \sum_{i \in I} (y(i,j) * C(i,j)) + \sum_{j=1}^{28} p(j) * CP(j) = W \text{ (min)}$$

avec la contrainte supplémentaire

$$\text{nbDesir}(j) - \left(|I| - \sum_{i \in I} y(i,j) \right) \leq p(j) \quad p(j) \geq 0$$

CP(j) représentant le coût unitaire de pénalité d'écart (peut-être par exemple la moyenne des coûts d'insatisfaction)

Dans cette formulation nous n'avons pas traduit le fait que nous voulions affecter des jours de repos dans une grille de roulement déjà constituée. Ceci s'introduit simplement en spécifiant des variables:

$y(i,j) = 1$ si l'infirmière i est en repos hebdomadaire le jour j
avec $C(i,j)$ ayant une valeur infinie si l'infirmière i est en repos hebdomadaire le jour j .

Afin de traduire le désir des infirmières d'avoir des jours de repos accolés, nous pouvons encore améliorer cette formulation en introduisant un coefficient de cohésion modulant la pénalité si le jour de repos implanté est contigu à une autre zone de repos.

$$\begin{aligned} C(i,j) &= \text{coût d'insatisfaction si } y(i,j)=1 \text{ et } y(i,j-1)=y(i,j+1)=0 \\ &= \text{coût d'insatisfaction} * \text{coefficient de cohésion si } y(i,j)=1 \text{ et} \\ &\quad (y(i,j-1)=1 \text{ ou } y(i,j+1)=1) \end{aligned}$$

Le problème d'implantation de jours de repos se formule donc comme un programme linéaire en variables bivalentes, avec un nombre très grand de variables, ce qui rend complexe la résolution directe.

On pourrait améliorer la fonction objective, en cherchant une définition plus adéquate du coefficient de cohésion: par exemple en interdisant certaines séquences de travail comme la séquence "un jour travaillé - un jour repos - un jour travaillé", ou en favorisant certaines autres séquences. Il serait possible d'introduire aussi l'historique de l'infirmière (évaluation qualitative des plannings précédents) dans cette fonction objective. Ces modifications, intéressantes à développer,

sont toutefois assez complexes à prendre en compte, c'est pourquoi nous avons gardé la formulation exposée précédemment.

III-2 LA METHODE DE RECUIT SIMULE

La méthode exhaustive pour déterminer l'optimum consisterait à générer tous les modèles de travail satisfaisant les contraintes de recouvrement sur une période de 28 jours. Cette méthode n'est pas envisageable pour un problème de cette taille.

Le laboratoire ayant déjà expérimenté le méthode de recuit simulé sur des problèmes de placements et d'ordonnancement, il nous sembla intéressant d'observer les résultats de celle-ci sur le problème des emplois du temps.

L'optimisation combinatoire dispose d'outils bien plus efficaces, mais qui malheureusement ne possèdent pas en général la qualité de robustesse souhaitée. De plus la notion d'optimalité n'a guère de sens dans un tel modèle où l'on veut simplement produire de très bonnes solutions.

III-2-a LES METHODES DE MONTE-CARLO:

L'idée première de ces méthodes est très simple: il suffit de générer aléatoirement une grande quantité de solutions, de calculer pour chacune d'elles la valeur de la fonction objective et de garder la meilleure.

La génération aléatoire d'une solution sera obtenue de la façon suivante:

Itérer

Choisir aléatoirement un jour dans les 28 jours du roulement;

Choisir aléatoirement une infirmière;

Si cette infirmière n'est pas en repos et si elle peut encore bénéficier d'un jour sans contrarier le nombre minimum d'infirmières requis, alors noter ce jour comme un jour de repos;

Calculer la valeur de la fonction objective et mémoriser celle-ci si elle est la meilleure déjà calculée;

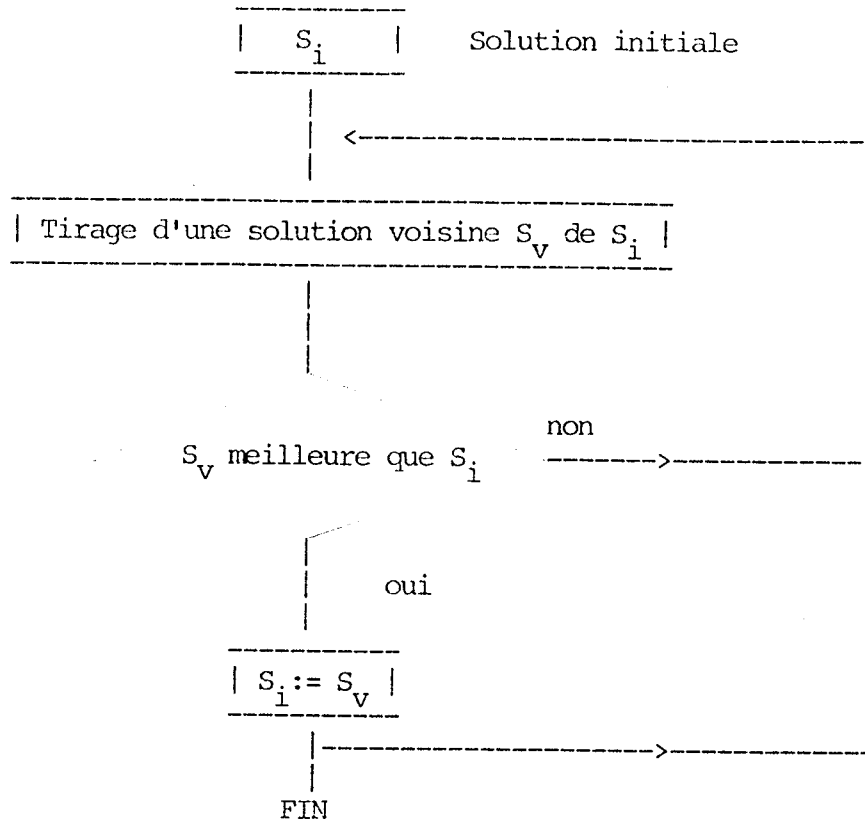
Fin itération

L'itération sera effectuée soit jusqu'à ce que le nombre d'infirmières minimum requis chaque jour soit atteint, soit jusqu'à ce qu'un nombre d'affectations de jours de repos soit atteint, soit pendant un nombre d'itérations fixé à l'avance.

III-2-b UNE HEURISTIQUE D'AMELIORATIONS ITERATIVES:

Une autre méthode classique est l'heuristique gloutonne d'améliorations itératives.

Après avoir défini la notion de voisinage d'une solution, on part d'une solution initiale S_i , on tire aléatoirement une solution voisine S_v de cette solution et on remplace S_i par S_v si la valeur de la fonction objective est améliorée pour S_v . On itère cette opération jusqu'à obtenir une solution "stable".



Dans notre cas précis le voisinage est défini de la manière suivante:

Tirer au sort un jour et une infirmière

Si cette infirmière n'est pas en repos hebdomadaire

Si cette infirmière est en repos ($y(i,j)=1$) lui enlever

Si cette infirmière n'est pas en repos, lui accorder

Les itérations s'effectueront jusqu'à ce que l'on ait obtenu une solution "stable" (n'ayant pas évolué pendant un nombre d'itérations fixé à l'avance). Cet algorithme est de type exploration locale et descente, il est connu aussi, en mécanique statistique, sous le nom d'algorithme de TREMPE.

III-2-c LA METHODE DE KIRKPATRICK:

Plus récemment, Kirkpatrick [KI] proposait une méthode dite de "Recuit simulé" pour résoudre des problèmes d'optimisation complexes.

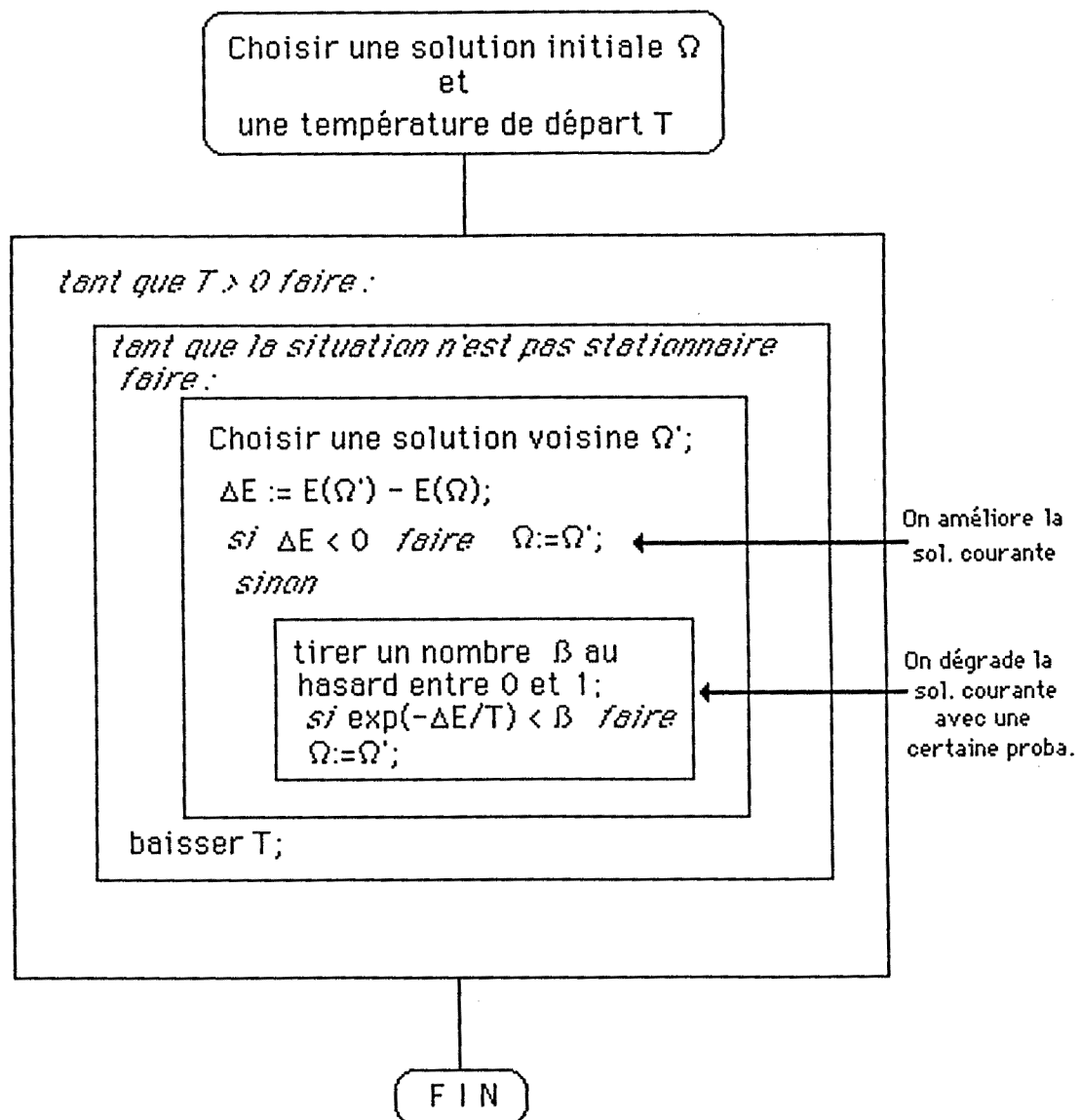
Cette méthode est fondée sur l'étude, en mécanique statistique, du comportement de la matière. Celle-ci est considérée comme un système d'atomes qui se déplacent aléatoirement suivant différentes lois internes. A cause du nombre d'atomes, (10^{23} par cm^3), seul le comportement moyen des déplacements des atomes autour de leur position d'équilibre peut être observé lors d'un équilibre thermique. Les lois de la thermodynamique stipulent que si E_i désigne l'énergie interne d'une configuration i , elle sera observée, à une température donnée T , avec une probabilité proportionnelle à $\text{Exp}(-(E_i - E_0)/kT)$ où E_0 est l'énergie minimum et où k est la constante de Boltzmann.

A $T=0$, le système doit donc normalement se trouver dans un des états d'énergie minimum.

Dans beaucoup de cas, le refroidissement d'un matériau ne conduit pas à un état d'énergie minimum, mais à des états métastables où subsistent des défauts. La méthode de recuit est un procédé expérimental pour éliminer les défauts cristallins. Il permet de trouver l'état minimum du système en partant d'un état liquide de celui-ci obtenu par chauffe.

La baisse de la température s'effectue pas à pas chaque fois que le système a trouvé son état d'équilibre. On évitera ainsi l'apparition de structure cristalline non régulière ou de verre. On doit donc rester très longtemps dans les températures voisines du point de gel. C'est l'analogie de cette méthode et des méthodes itératives d'optimisation combinatoire qui donna à Kirkpatrick l'idée d'essayer une méthode heuristique simulant ce procédé (implanté par Métropolis [ME]).

La température étant le paramètre le plus important dans ce procédé, puisqu'il permet de sortir des "puits locaux" avec une probabilité de $p = \text{Exp}(-\Delta E/kt)$, Kirkpatrick et al. introduisirent dans l'algorithme glouton un paramètre équivalent. Ce procédé se résume ainsi



Monte-Carlo équivaut à un recuit à température infinie. La méthode d'amélioration itérative ou de Trempe présentée auparavant est obtenue avec $T=0$ dès le départ.

L'absence de paramètre jouant le rôle de la température dans les méthodes classiques d'optimisation combinatoire explique que celles-ci donnent le premier minimum local rencontré lors de leur parcours. Une première amélioration de ces méthodes fut proposée par Lin et Kerningham avec des algorithmes d'exploration locale. L'idée en était très simple: à partir d'une solution réalisable du problème considéré, on définit un "voisinage" $V(s)$ de cette solution, et l'on cherche s'il existe s' , appartenant à $V(s)$ tel que $E(s') < E(s)$. Si tel est le cas, on remplace s par s' et on recommence. Lorsque l'on ne peut plus progresser (optimum local), on élargit le voisinage considéré. Ces algorithmes permettent donc de franchir des "petits cols". L'algorithme de recuit a lui, l'avantage d'augmenter la taille des cols que l'on franchit et d'atteindre plus facilement la vallée la plus basse.

Comme pour toute heuristique, l'optimalité de la solution n'est pas garantie.

III-2-d EXPERIMENTATION DE LA METHODE DU RECUIT:

A- Solution initiale:

Afin de pouvoir comparer les différentes méthodes, un cas particulier a été observé. le roulement de 28 jours étant conservé, le problème a été étudié pour 12 infirmières affectées à une même tranche horaire. Les valeurs numériques choisies pour les coûts n'influent pas sur l'efficacité des algorithmes. La solution de départ est la grille de roulement déjà constituée des repos hebdomadaires. Aucune affectation de jours de repos particuliers n'est déjà effectuée.

B- Configurations voisines:

La configuration sera un planning induisant une solution réalisable.

La relation de voisinage à établir semble assez simple: elle consiste à prendre la même configuration et à affecter un jour choisi au hasard pour une infirmière choisie également au hasard, soit à 1 (repos) s'il est à 0, soit à 0 s'il est à 1.

Certaines conditions sont toutefois à respecter pour ne pas violer les contraintes:

- si la configuration comporte un 1 au jour j pour l'infirmière i, on ne peut lui donner la valeur 0 que s'il ne s'agit pas d'un repos hebdomadaire.
- si la configuration comporte un 0 au jour j pour l'infirmière i, on ne peut lui donner la valeur 1 que si l'infirmière i a encore à son crédit un jour de repos et si le nombre d'infirmières minimum requis est encore satisfait après cette affectation.

C- Schéma de refroidissement:

Dans le procédé expérimental, les physiciens font décroître la température par paliers, mais la stratégie de décroissance n'est pas clairement définie.

Afin d'évaluer le nombre de configurations explorées, on prend ici une décroissance géométrique en n paliers.

La température de départ T_0 est calculée selon le critère suivant: comme elle doit correspondre à l'état liquide du système, elle doit être celle qui permet une détérioration la plus forte du système avec une grande probabilité.

Donc $P_0 = -\frac{\Delta E_{\text{Max}}}{k T_0}$ proche de 1

T_{n-1} étant la température d'arrêt au $n^{\text{ième}}$ palier, le choix de décroissance donne:

$$T_i = a T_{i-1} = a^i T_0 \quad (a < 1)$$

A la température d'arrêt, on accepte avec une probabilité faible une petite détérioration du système, ce qui revient à définir la notion de gel du système:

$$\text{Si } \Delta E_{\text{Min}} = 1 \quad P_{n-1} = -\Delta E_{\text{Min}} / k T_{n-1}$$

$$1/k = T_{n-1} \log(P_{n-1})$$

$$T_0 = E_{T_0} * -\Delta E_{\text{Max}} * 1/k \log(P_0)$$

$$a = \text{Exp} (\log (T_0/T_{n-1})/n)$$

Une configuration voisine C_v de la configuration actuelle C_a sera donc acceptée à la température T_i avec la probabilité:

$$P = 1 \quad \text{si} \quad E_{Cv} - E_{Ca} < 0$$

$$= \text{Exp} ((E_{Cv} - E_{Ca}) * \log(P_{n-1}) / a^i) \quad \text{sinon}$$

Dans nos tests nous avons pris

$$P_0 = 0,9 \quad P_{n-1} = 0,01 \quad n = 100 \quad -\Delta E_{\text{Max}} = 10 \quad -\Delta E_{\text{Min}} = 1$$

D- Critère d'équilibre:

Le critère choisi est celui de Kirkpatrick: à chaque température, on reste jusqu'à ce qu'un nombre N_a des configurations acceptées soit atteint ou bien qu'un nombre N_t d'itérations soit dépassé. Les nombres N_a et N_t sont choisis expérimentalement selon les données du problème. N_a permet de ne pas perdre de temps à une température assez haute. Ce paramètre ne jouant plus aucun rôle à partir d'un certain seuil, il faut alors choisir N_t de façon à ce que le système trouve alors à chaque température son état d'équilibre. La combinaison de ces deux paramètres permet d'obtenir un algorithme deux à trois fois plus efficace que si l'on garde uniquement le paramètre N_t .

Afin de comparer cette méthode avec des méthodes plus classiques, l'expérimentation, sur ce même exemple de la méthode de Monte-Carlo et de la méthode de Trempe a été effectuée.

La solution initiale pour la méthode de trempe, a été générée aléatoirement selon le principe de Monte-Carlo pendant un nombre fixé d'itérations.

E- Résultats:

Les courbes des résultats sont exposées en annexe 10.

1- Monte-Carlo:

Critère d'arrêt: Nombre d'itérations fixé (Nt) ou crédit de jour à allouer pour chaque infirmière nul, ou nombre minimum d'infirmières atteint chaque jour.

Nt=1000

Nb essais = 110 Energie Minimum=-188 Energie moyenne=-120

Nt=2000

Nb essais = 110 Energie Minimum=-193 Energie moyenne=-125

2- Trempe:

Solution initiale: MIC + descente

Nb essais = 100 Energie Minimum=-214,4 Energie moyenne=-170

3- Kirkpatrick:

T=438 Nombre de températures = 100

Nt=1500 Na=50 Energie Minimum=-310 Temps=2h10

Nt=1500 Na=100 Energie Minimum=-313 Temps=2h30

Nt=2000 Na=100 Energie Minimum=-319 Temps=3h00

Nt=1500 Na=200 Energie Minimum=-319 Temps=3h10

La solution KIRKPATRICK donne donc des résultats beaucoup plus concluants que les méthodes classiques, en un temps à peu près équivalent. La difficulté de la méthode réside en la détermination du nombre d'itérations et du nombre d'améliorations à retenir. L'expérimentation permet de donner une valeur intéressante à ces deux critères d'arrêt de chaque étape.

Le schéma de refroidissement classique par décroissance géométrique peut aussi être revu. Plus récemment, des chercheurs hollandais, AARTS E.H.L. et VAN LAARHOVEN [AA] proposèrent un autre schéma de refroidissement: le nombre d'itérations à chaque température reste constant, mais la baisse de la température varie selon des indicateurs statistiques. L'expérimentation avec ce second schéma serait intéressante à étudier pour compléter cette étude.

On peut, dans le même ordre d'idée, remettre en question le choix du critère de rejet classique implanté dans cette méthode. Si l'on accepte la possibilité de rejeter une solution améliorante, le remplacement de la solution courante C_a par une solution voisine C_v serait spécifié de la manière suivante:

Tirer un nombre b au hasard entre 0 et 1

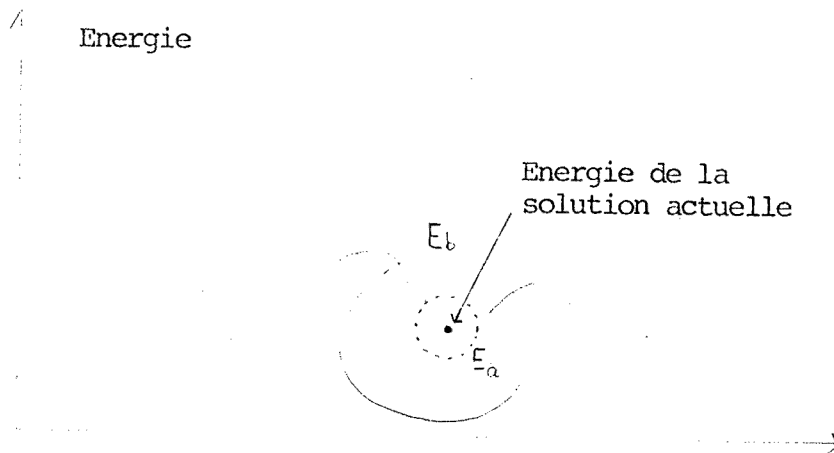
Si ($\Delta E < 0$ et $\frac{\exp(\Delta E/kT)}{\exp(-\Delta E/kT) + \exp(\Delta E/kT)} < b$) alors $C_a := C_v$;

Si ($\Delta E \geq 0$ et $\frac{\exp(-\Delta E/kT)}{\exp(-\Delta E/kT) + \exp(\Delta E/kT)} < b$) alors $C_a := C_v$;

Cette dernière possibilité n'a pas été expérimentée dans ce travail, les résultats obtenus de manière "classique" étant satisfaisant comparativement aux méthodes de Monte-Carlo et de Trempe.

Il s'agit cependant d'être prudents lors de l'élimination de la méthode de Trempe.

Dans ce cas précis, les résultats sont incontestablement plus mauvais que ceux obtenus par la méthode de Kirkpatrick. Il faut cependant rappeler que le principe fondamental de cette méthode est la notion de voisinage. Si le "terrain" exploré est représenté par le schéma suivant, on peut définir le voisinage de trois manières:



- 1) Toute solution est voisine de la solution actuelle:
C'est la méthode de Monte-Carlo: le résultat est mauvais, quel que soit le nombre d'itérations de la méthode.
- 2) Si le voisinage V_a est trop restreint, ce qui est peut-être le cas de notre exemple puisque l'on a pris le voisinage le plus simple possible (changement d'une position dans la configuration), la méthode de Trempe ne pourra pas "franchir les cols" aisément. Ceci explique en partie le fait que la méthode de Kirkpatrick donne alors des résultats nettement meilleurs.
- 3) Si le voisinage V_b est choisi avec plus de soins, la méthode de Trempe peut donner des résultats très bons. On peut imaginer, par exemple des permutations de colonnes ou de blocs de la matrice pour définir des solutions voisines plus complexes susceptibles d'améliorer la valeur de la fonction objective. Cette possibilité, plus difficile à mettre en oeuvre, n'a pas été testée ici.

III-3 LA METHODE HASARD ET NECESSITE "HN":

Le recuit simulé demande toutefois des temps de calcul importants sur micro-ordinateur. L'idée d'une parallélisation de ce processus de recuit semble donc intéressante, d'autant que l'on voit apparaître de nombreuses machines à processeurs parallèles à des coûts raisonnables. La première méthode de parallélisation consiste à faire évoluer des processeurs indépendamment. La division du travail à effectuer, pour une distribution à ces différents processeurs semble assez naturelle: on fait évoluer, à une température T des parties distinctes d'une même solution sur les différents processeurs.

Le traitement d'évolution de la solution est réparti sur les n processeurs de manière à respecter les contraintes de recouvrement et le nombre de jours à affecter pour chaque infirmière: un découpage en "damier" est envisageable, par exemple sur trois processeurs:

Jours du roulement

	P1 processeur1	P3 processeur2	P2 processeur3
Inf.	P2 processeur1	P1 processeur2	P3 processeur3
	P3 processeur1	P2 processeur2	P1 processeur3

Les tirages aléatoires seraient donc effectués par les différents processeurs sur des parties indépendantes de la solution pendant la même phase P_i . Le partage de la solution initiale en parties indépendantes n'est pas trop contraignant lorsque la notion de voisinage est simple. Pour une notion de voisinage plus élaborée, comme les permutations de "blocs colonnes", ce partage pénalise la méthode de

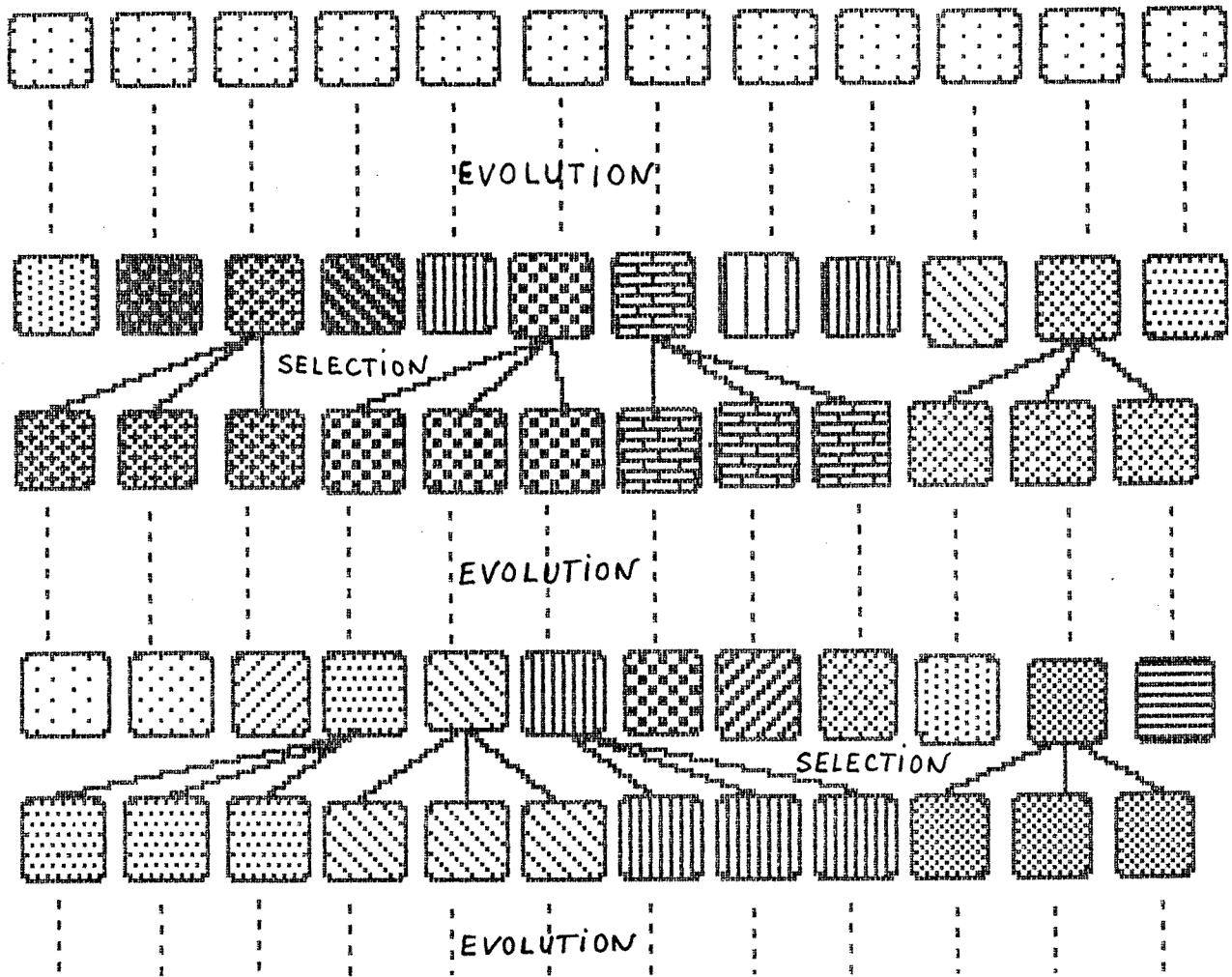
Kirkpatrick en réduisant considérablement le nombre de voisins possibles. La convergence de cette méthode n'est d'ailleurs pas assurée.

La deuxième méthode de parallélisation consiste à faire évoluer tous les processeurs sur la même solution initiale, un superviseur ayant à charge de transmettre toute acceptation du système à chaque processeur. Si l'on regarde alors les différentes étapes de la solution Kirkpatrick, on s'aperçoit que le nombre d'acceptations dans les températures hautes est très important. On se sert d'ailleurs de ce nombre comme critère d'arrêt pour ces températures. Si la parallélisation est mise en place, le "superviseur" doit agir sans cesse pour transmettre toute acceptation, et le gain de temps du procédé de parallélisation devient très contestable. Pour des températures plus basses, où le critère d'arrêt est alors le nombre d'itérations maximum, il est vrai que le nombre d'acceptation de nouvelles solutions étant très réduit, ce procédé devrait être intéressant.

La complexité de la mise en oeuvre de cette méthode nous fit préférer l'étude d'une méthode de parallélisation plus simple à réaliser. S'inspirant d'une idée darwinienne ([MO] "Le Hasard et la Nécessité"), nous fîmes évoluer une même solution initiale sur n processeurs différents (principe de mutation où le hasard intervient).

A une certaine étape de cette "mutation", une sélection est effectuée, et seules les solutions résultant de celle-ci sont laissées en vie et ré-injectées de manière égale sur tous les processeurs.

L'algorithme consiste alors à réitérer ces deux opérations: évolution-sélection.

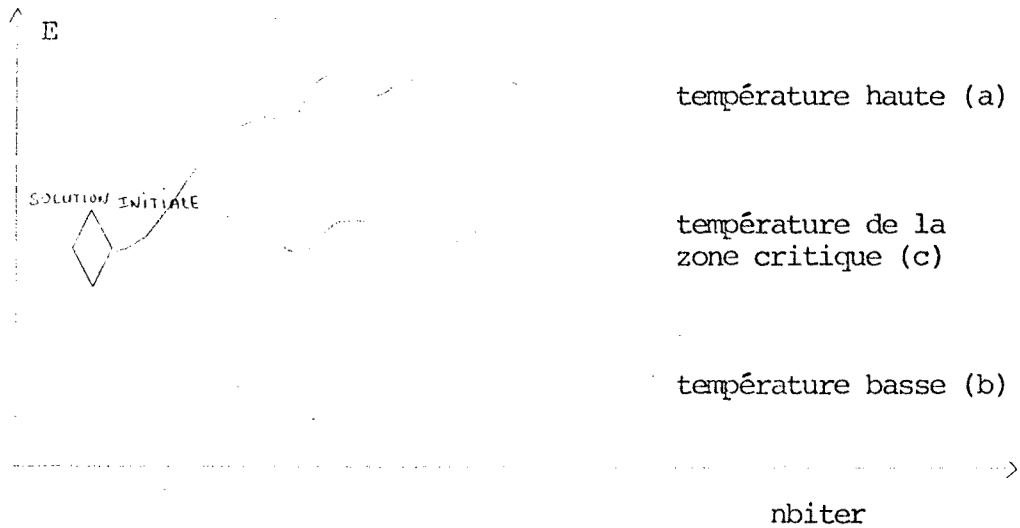


III-3-a Evolution:

L'idée la plus naturelle d'évolution d'une solution est celle retenue dans l'étude précédente pour générer une solution voisine: l'affectation à 1 ou à 0 d'un jour tiré au hasard, pour une infirmière également choisie au hasard.

Afin de permettre un nombre d'itérations plus important, sans trop dégrader le système, ce n'est pas cette évolution simple qui fut choisie. L'étude précédente de la méthode de Kirkpatrick nous permet d'observer l'évolution du système aux différentes températures.

Ce critère de température est le paramètre le plus important du procédé: il permet au système de sortir des "puits locaux", puisqu'à chaque température $T > 0$, il existe une probabilité $P = \exp(-E/kT)$ d'accepter les détériorations de la configuration du système.



- * à des températures hautes (a) le système se dégrade rapidement et l'énergie moyenne d'une solution reste haute. Le désordre total est permis (analogie avec la méthode de Monte-Carlo).
- * à des températures basses, le système se gèle et l'énergie n'évolue plus que très rarement, ce qui se traduirait, dans notre cas, par une évolution nulle.
- * il existe une zone critique de températures (cf annexe 10) où l'introduction du désordre ne dégrade pas le système trop rapidement. On peut dans ce cas augmenter le nombre d'itérations sans crainte d'une détérioration de la solution.

Se servant des résultats obtenus dans la méthode du recuit simulé, le système fut donc placé à une température critique de la méthode de Kirkpatrick . Le critère d'acceptation d'une solution de Kirkpatrick est alors conservé et la solution évolue selon ce critère. Ce n'est donc pas le hasard pur qui joue dans l'évolution des solutions sur chaque processeur.

III-3-b Sélection:

La sélection doit être effectuée de manière à favoriser l'optimisation des solutions (au sens de la fonction objective précédente). Plusieurs critères de sélection sont possibles:

- a- critère le plus naturel: garder la meilleure solution issue de tous les processeurs.
- b- garder les "n" meilleures solutions.
- c- garder les "n" meilleures solutions si elles ne sont pas trop "mauvaises" par rapport à la meilleure.
- d- garder dans les meilleures solutions, les "n" les plus dissimilaires.

Les trois derniers critères ont pour but d'étudier le système en laissant une certaine diversité des "souches", idée développée par R.BRADY [BR].

III-3-c Expérimentation:

Les courbes représentatives des résultats des expérimentations sont disponibles en annexe 10.

Cette méthode a été appliquée en simulant 12 processeurs à partir de la solution initiale retenue pour la méthode de recuit. Les différents critères de sélection ont été testés à deux températures: T=20 et T=30.

Le tableau suivant donne les principaux résultats pour un nombre d'itérations égal à 500 ou 100 améliorations de Kirkpatrick avant chaque sélection.

Energie de la meilleure solution après

n° sél: sél.1 sél.6 sél.11 sél.11 sél.21 sél.26

Critère a:

T=20	-256	-312	-326	-334	-337	-342
T=30	-276	-322	-334	-338	-341	-345

Critère b:

T=20	-260	-307	-326	-340	-340	-342
T=30	-230	-305	-322	-334	-338	-346

Critère c:

T=20	-256	-314	-320	-326	-336	-348
écart=5						
T=20	-244	-308	-320	-326	-334	-342
écart=20						

Critère d:

T=30	-238	-284	-305	-310	-324	-330
------	------	------	------	------	------	------

Les résultats semblent peu différents pour les deux températures: ces deux températures sont dans la zone critique intéressante de Kirkpatrick.

Un seul processeur a été utilisé pour effectuer ces tests. La moyenne de temps de calcul avec sélection et redistribution des différentes solutions retenues est de 7H. Ce qui laisse envisager, si 12 processeurs fonctionnent en parallèle, des résultats obtenus en un peu plus d'une demi-heure.

Le critère le plus simple (a) semble très satisfaisant, la diversité des solutions n'apportant pas une amélioration sensible des résultats. Afin d'optimiser les temps de calcul, des essais ont été effectués avec un nombre d'itérations et d'améliorations restreint avant chaque étape de sélection. Les résultats sont donnés ci-après.

Energie de la meilleure solution après

n° sél: sél.1 sél.6 sél.11 sél.11 sél.21 sél.26

Critère a: Temps de calcul 4h30

Nbiter=100 -58 -220 -284 -311 -339 -340
NbAmelior=50

Critère a: Temps de calcul 3h

Nbiter=20 -60 -206 -254 -282 -292 -302
NbAmelior=20

Nous remarquons que le nombre d'itérations et le nombre d'améliorations peuvent être descendus en dessous de 500 et 100 sans trop détériorer la solution finale. Cette solution semble alors s'améliorer plus lentement.

Efficacité du critère de sélection:

pour évaluer l'efficacité du critère de sélection, nous avons laissé évoluer la solution initiale sur un seul processeur pendant douze fois plus d'étapes que précédemment (12*26*(500 itérations ou 100 améliorations)). L'énergie de la meilleure solution trouvée est de -270.

Nous remarquons que la valeur moyenne de l'énergie d'une solution est de - 230 après quelques étapes et que la solution oscille près de cette moyenne.

En mémorisant à chaque étape la meilleure solution trouvée pendant cette étape et en repartant de celle-ci, l'énergie de la meilleure solution trouvée est de -302, la valeur moyenne étant de -260.

Nous avons donc amélioré, par la parallélisation et la sélection, les résultats obtenus par simple évolution.

Par une méthode de parallélisation simple, nous avons obtenu des résultats améliorant d'environ 10% les solutions trouvées par la méthode plus classique du recuit.

La variété des solutions ne semble pas, dans cet exemple simple apporter une amélioration sensible. Il faut cependant être prudent avant d'éliminer ce critère de sélection, le nombre d'expérimentations effectuées n'ayant pas été assez important pour donner cette conclusion.

La méthode génétique proposée semble donc plus concluante que celle du recuit simulé, il convient toutefois de remarquer qu'une application d'un "recuit" sur la solution initiale est nécessaire pour déterminer la zone critique de la température et donc pour permettre une évolution "moins hasardeuse" dans la phase évolution de cette méthode.

Vu les temps de calcul exigés pour tester le recuit sur un micro-ordinateur, il n'a pas été possible d'effectuer des essais plus complexes sur les deux méthodes préconisées ci-dessus. il serait toutefois intéressant, à l'aide de moyens informatiques plus importants, de tester ces deux méthodes sur des plannings de dimension plus importante.

CONCLUSION

Le problème d'élaboration des emplois du temps est un problème difficile. A cause de la variété des situations, les algorithmes mis en place doivent être avant tout **"robustes"** et facilement adaptables. Dans cette optique, la méthode génétique "H.N." semble intéressante. L'outil doit aussi offrir une grande interactivité Homme-Machine afin de permettre la mise au point finale du planning de travail. Ceci explique qu'il y ait très peu d'outils effectivement disponibles dans les hôpitaux, les écoles et les entreprises à feu continu.

L'origine de notre étude est une initiative locale d'un service hospitalier confronté aux difficultés de l'institution informatique hospitalière. Celle-ci, submergée par des demandes diverses et face aux multiples organisations des différents services, concentre le plus souvent ses efforts sur des problèmes plus généraux: P.M.S.I (Programme de Médicalisation du Système d'Informations) - Budget Global - Gestion de Personnel - Gestion Economique

La non-réponse des services informatiques aux demandes concrètes du corps médical, provoque alors la recherche de solutions extérieures, comme cette demande à notre laboratoire... L'étude algorithmique et la réalisation technique nous offrent alors un sujet très intéressant de recherche qui est de plus appliqué à un cas pratique. Il nous est cependant impossible de mettre en place une solution complète, le fonctionnement des équipes de recherche, n'incluant pas les mêmes prestations "de suivi de logiciel" qu'une société de service. Dans notre cas précis, nous n'avons malheureusement pas pu, pour les raisons énoncées ci-dessus, atteindre la phase exploitation de notre étude.

Le service de Réanimation, profitant du dynamisme de son personnel, a alors élaboré, à l'aide du logiciel intégré "EXCEL" (sur Macintosh), un outil d'aide au suivi des emplois du temps. Cette réalisation démontre bien l'autonomie que procure de nos jours la micro-informatique à des utilisateurs non-informaticiens "passionnés". On ne peut aujourd'hui, que regretter la fragilité et le manque d'évolutions de ces outils auto-produits dans les services, inconvénients le plus souvent dûs à la rigidité et à la complexité des logiciels utilisés pour les concevoir. Ce type de solution exclut aussi l'utilisation d'algorithmes produisant des plannings initiaux d'aide à l'élaboration.

Les résultats encourageants des méthodes stochastiques testées dans ce travail, seraient toutefois à compléter par différents travaux. La convergence théorique de la méthode "H.N.", qui semble la plus concluante dans ce cas précis, reste encore à établir. Il serait aussi intéressant de faire des expérimentations plus réalistes d'implantations effectives en parallèle, et d'observer les variations des résultats obtenus en augmentant le nombre de processeurs.

BIBLIOGRAPHIE

- [AA]- AARTS E.H.L., VAN LAARHOVEN P.J.M. [1985], Statistical cooling:
a general approach to combinatorial optimisation
problems, Philips J.Res. Vol 40 N°4, p 193-226
- [AH-SH]- ANUJA H., SHEPPARD R., Computerized nurse scheduling [1975],
Ind. Engng Vol 7
- [BA-BL]- BALINTFY J.L., BLACKBURN C.R., General purpose multiple
choice programming by truncated block enumeration
[1969], presented at the 36th National Meeting of ORSA
Miami Beach, Florida
- [BR]- BRADY R., Optimization Strategies Gleaned from Biological
Evolution [7/07/1986], IBM Europe Institute,
Stochastich Optimization in Engineering and Biology
Daily Schedule.
- [FR]- FRANCES Sister M.A., Implementing a program of cyclical
scheduling of nursing personnel [1966], Hospitals,
J.Am. Hospital Assoc. Vol 40
- [HO]- HOWELL J.P., Cyclical scheduling of nursing personnel [1966],
Hospitals, J.Am. Hospital Assoc. Vol 40,71

[KI]- KIRKPATRICK S., GELLAT C.D., VECCHI M.P. [1982], Optimization by simulated annealing, R.C. 9355 (#41093), 4/2/82 Computer Science/Engineering Technology. IBM Thomas J.Watson Research Center, Yorktown Heights, New York 10598.

[MA-SM]- MACK H.L., SMITH L.D., An example of the utility of list processing and problem oriented data structure to facilitate the development of computerized scheduling algorithms [1975], Paper presented to the annual Conference of the Canadian Operational Research Society, Waterloo, Canada

[MA-WO]- MAIER-ROTHE C., WOLFE H.B., Cyclical scheduling and allocation of nursing staff [1973], Socio-Econ. Plan. Sci. Vol 17, 471

[ME]- METROPOLIS N., ROSENBLUTH A., ROSENBLUTH M., TELLER A., TELLER E. [1953], Equation of state calculations by fast computing machines. J Chem. Phys. 21 1087-1092

[MI-PI-RA] MILLER H.E., PIERSKALLA W.P., RATH G.J. Nurse scheduling using mathematical programming [1976], Operations Research, Vol 24 N°5

[MO]- MONOD J., Le Hasard et la Nécessité [1970]- Editions du Seuil N°43

- [PR]- PREMTI F., Méthodes stochastiques dans les problèmes de placement. Thèse de troisième cycle. USMG [1983]
- [RO]- ROTSHEIM M., Hospital manpower staff scheduling by mathematical programming [1973], Health Services Research, Vol 8, p 60-66
- [RO-1]- ROGLIN O., Some remarks on scheduling nursing personnel [1979], Ops Res. Vol 23 N°3, p 828-829
- [SM]- SMITH L.D., The application of an interactive algorithm to develop cyclical rotational schedules for nursing personnel [1976], INFOR, Can J. Opl. Res. Information Processing, Vol 14, 53
- [SM-WI]- SMITH L.D., WIGGINS A., A computer-based nurse scheduling system [1977], Computs. and Ops. Rev. Vol 4, p 195-212
- [SM-WI-BI]- SMITH L.D., WIGGINS A., BIRD D., Postimplementation experience with computer-assisted nurse scheduling in a large hospital [1979], INFOR Vol 17 N°4, p 309-321
- [TR]- TREMOLIERES R., Le problème des roulements de quart pour les entreprises à feu continu, RAIRO Recherche Opérationnelle [1976] Vol 10 N°2, p 71-101

[UH]- UHRY J.P., Applications du recuit simulé en Recherche
Opérationnelle. Laboratoire ARTEMIS (Grenoble) [1987]

[WA]- WARNER D.M., Scheduling nursing personnel according to nursing
preference: a mathematical programming approach [1976],
Ops Res., Vol 24 N°5, p 842-856

[WA-PR]- WARNER D.M., PRAWDA J. A mathematical programming model for
scheduling nursing personnel in a hospital [1972],
Management Science Vol 19 N°4

[WO-YO]- WOLFE H., YOUNG J.P., Staffing the nursing unit: Part I and
Part II [1965], Nursing Res. Vol 14, 3-4, p 236-243 et
p 299-303

EMPLOI DU TEMPS EN MILIEU HOSPITALIER :
HEURISTIQUES STOCHASTIQUES D'AIDE A L'ELABORATION

INTRODUCTION	p 5
CHAPITRE I: ANALYSE DES OUTILS ET MODELES EXISTANTS	p 10
I-1 L'approche traditionnelle	p 10
I-2 L'approche par ordonnancement cyclique	p 11
I-3 Méthode de D. Michael WARNER	p 12
I-3-a Règles et contraintes sur chaque "roulement".	p 13
I-3-b Introduction des restrictions sur un roulement particulier. Rotations.	p 14
I-3-c Le modèle mathématique	p 15
I-3-d Heuristique de résolution	p 20
I-4 Méthode de HOLMES E. MILLER	p 22
I-4-a Le modèle mathématique proposé	p 23
A - Première classe de contraintes: les contraintes de réalisabilité.	p 24
B - Deuxième classe de contraintes: contraintes non obligatoires.	p 25
I-4-b Description de l'algorithme	p 28
I-4-c Résultats	p 29
I-5 Méthode Douglas SMITH et A. WIGGINS	p 31
I-5-a L'approche adoptée	p 33
I-5-b L'algorithme	p 35
CHAPITRE II: ERGONOMIE ET INTERACTION	p 40
II-1 Contraintes matérielles	p 41
II-2 Organisation du service	p 42
II-3 Enquête sur le travail effectif du personnel chargé de l'organisation du service	p 46
II-4 Réalisation d'un outil interactif	p 51
II-4-a Utilisation de menus: Gestionnaire de menus.	p 51

II-4-b Visualisation et modifications des plannings.....	p 56
II-4-c Gestion de la fiche infirmière	p 59
II-4-d Interface fiche infirmier(e) - Planning ...	p 65
CHAPITRE III : METHODES STOCHASTIQUES	p 68
III-1 Formulation du problème	p 69
III-2 La méthode du RECUIT SIMPLE	p 71
III-2-a Les méthodes de Monte-Carlo	p 71
III-2-b Une heuristique d'améliorations itératives.	p 72
III-2-c La méthode de Kirkpatrick	p 74
III-2-d Expérimentation de la méthode du recuit....	p 76
III-3 La Méthode HASARD ET NECESSITE	p 82
III-3-a Evolution	p 84
III-3-b Sélection	p 86
III-3-c Expérimentation	p 86
III-4 CONCLUSION	p 90
BIBLIOGRAPHIE	p 92
ANNEXES	
ANNEXE 1 : Feuille de planning existante.	p 98
ANNEXE 2 : Génération du programme de parcours de l'arborescence ...	p 99
ANNEXE 3 : Programme généré par le gestionnaire de menus	p102
ANNEXE 4 : Structure des fichiers contenant les plannings	p108
ANNEXE 5 : Programme de "Scrolling"	p111
ANNEXE 6 : Détail des menus	p114
ANNEXE 7 : Impression de la feuille planning	p119
ANNEXE 8 : Structure des données des Fiches infirmières	p121
ANNEXE 9 : Fiche infirmière et listes produites	p123
ANNEXE 10: Courbes relatives aux diverses expérimentations	p125

"PERSONNEL"	12 13 14 15 16 17 18					19 20 21 22 23 24 25					26 27 28 29 30 31 1					2 3 4 5 6 7 8												
	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D

ANNEXE 2 : Génération du programme de parcours de l'arborescence:

1) Fichier de données des commandes:

Structure d'un enregistrement:

- numéro de niveau dans l'arbre des commandes
- un espace
- numéro de la commande
- un espace
- numéro de la couleur à affecter à la commande
- un espace
- un booléen valant
1 si la couleur correspondant à la commande doit être
stockée
0 sinon
- un caractère alphabétique correspondant à l'initiale
permettant d'exécuter la commande (les majuscules et
minuscules n'ont pas d'importance).
- une chaîne de caractères correspondant au nom de la
commande et servant de nom d'appel à la procédure
commande.
- un retour charriot pour séparer deux enregistrements.

On peut ainsi "jouer" sur les noms de commandes du menu, et les couleurs affectées à ces commandes.

Pour décrire l'arborescence des commandes, l'ordre d'entrée est très important.

La méthode "descente d'abord" est utilisée (backtracking):

Commencer au niveau le plus haut.

Niveau courant = Niveau 1

Tant qu'il existe une commande non décrite,

S'il existe une commande non encore décrite dans le
niveau courant

Décrire cette commande

Si cette commande a des "descendants"

(un menu de niveau plus important)

Niveau courant = niveau courant + 1

Sinon

Niveau courant = niveau - 1

Fin tant que

Exemple:

Pour notre cas particulier, le fichier est:

1	17	14	OCChanger		
2	18	15	OUn		
2	19	15	ODDeux		
2	20	15	OTTrois		
1	21	14	OSSauver		
1	22	14	OFFermer		
1	23	14	OMModifieur	avec	2: Magenta
2	24	8	ORRemplir		4: Purple
3	2	15	IMMatin		6: Greyl
3	3	8	lSSoier		7: MedBlue
3	1	6	lNNuit		8: LightBlue
3	8	10	lHHebdo		10: Orange
3	7	2	lCCongé		12: Pink

3	5	4	1AAbsc	13:	Yellow
3	6	12	1RRecup	14:	Green
3	4	13	1XX39eH	15:	Aqua
3	9	14	1EEcept	16:	White
3	25	7	1DDate		
3	26	16	0QQuit		
2	27	8	0PPermuter		
3	28	6	0EEsc		
3	29	6	0VValider		
2	30	8	0FFichinf		
2	333	8	0QQuit		
1	34	14	0IImprimer		
1	35	14	0AArret		

ANNEXE 3 : Programme généré par le gestionnaire de menus.

```
Program Genere;  
(*$S+*)  
Uses ChainStuff,PurpleGraPhics,AppleStuff,  
(*$U #5:Unit0.Code*)Unit0,  
(*$U #5:Unit1.Code*)Unit1,  
(*$U #5:Unit2.Code*)Unit2,  
(*$U #5:Unit3.Code*)Unit3,  
(*$U #5:Plan.Code*)Plan;  
  
Procedure P0;Forward;  
Procedure P17;Forward;  
  
Procedure P18;  
Begin  
  Fin:=False;  
  Un;  
  P0;  
End;  
  
Procedure P19;  
Begin  
  Fin:=False;  
  Deux;  
  P0;  
End;  
  
Procedure P20;  
Begin  
  Fin:=False;  
  Trois;  
  P0;  
End;  
  
Procedure P17;  
Begin  
  Fin:=False;  
  Window(0,279,167,191);Paper(CWhite);  
  Fillport;  
  Window(0,279,0,191);  
  MoveAbs(2,183);  
  Ink(CBlack);  
  Paper(CAqua);  
  PrintString('U(n ');  
  Paper(CAqua);  
  PrintString('D(eux ');  
  Paper(CAqua);  
  PrintString('T(rois ');  
  Ens_Comm:=[8,10,11,21,27,85,117,68,100,84,116];  
  Repeat  
    Repeat  
      Lecturech;  
      If Not(Ord(ch) in Ens_Comm) Then Note(12,20);  
    Until Ord(ch) in Ens_Comm;  
    Changer;  
  Case Ord(ch) of  
    85,117:Begin Num_Comm:=18; P18; End;  
    68,100:Begin Num_Comm:=19; P19; End;  
    84,116:Begin Num_Comm:=20; P20; End;  
    8:Depl_Gauche:=True;  
    10:Depl_Bas:=True;  
    11:Depl_Haut:=True;  
    21:Depl_Droite:=True;  
    27:Escape:=True;
```

```
End;  
Until Fin;  
End;  
  
Procedure P21;  
Begin  
  Fin:=False;  
  Sauver;  
End;  
  
Procedure P22;  
Begin  
  Fin:=False;  
  Fermer;  
End;  
  
Procedure P23;Forward;  
  
Procedure P24;Forward;  
  
Procedure P2;  
Begin  
  Fin:=False;  
  Matin;  
  If Fin Then P23;  
End;  
  
Procedure P3;  
Begin  
  Fin:=False;  
  Soir;  
  If Fin Then P23;  
End;  
  
Procedure P1;  
Begin  
  Fin:=False;  
  Nuit;  
  If Fin Then P23;  
End;  
  
Procedure P8;  
Begin  
  Fin:=False;  
  Hebdo;  
  If Fin Then P23;  
End;  
  
Procedure P7;  
Begin  
  Fin:=False;  
  Conge;  
  If Fin Then P23;  
End;  
  
Procedure P5;  
Begin  
  Fin:=False;  
  Absc;  
  If Fin Then P23;  
End;  
  
Procedure P6;  
Begin  
  Fin:=False;  
  Recup;
```

If Fin Then P23;
End;

Procedure P4;
Begin
 Fin:=False;
 X39eH;
If Fin Then P23;
End;

Procedure P9;
Begin
 Fin:=False;
 Exept;
If Fin Then P23;
End;

Procedure P25;
Begin
 Fin:=False;
 Date;
If Fin Then P23;
End;

Procedure P26;
Begin
 Fin:=False;
 Quit;
If Fin Then P23;
End;

Procedure P24;
Begin
 Fin:=False;
 Window(0,279,167,191);Paper(CWhite);
 Fillport;
 Window(0,279,0,191);
 MoveAbs(2,183);
 Ink(CBlack);
 Paper(CAqua);
 PrintString('M(atin ');
 Paper(CLightBlue);
 PrintString('S(oir ');
 Paper(CGrey1);
 PrintString('N(uit ');
 Paper(COrange);
 PrintString('H(ebdo ');
 Paper(CMagenta);
 PrintString('C(onge ');
 Paper(CPurple);
 PrintString('A(bsc ');
 Paper(CPink);
 MoveAbs(2,175);
 PrintString('R(ecup ');
 Paper(CYellow);
 PrintString('X(39eH ');
 Paper(CGreen);
 PrintString('E(xept ');
 Paper(CMedBlue);
 PrintString('D(ate ');
 Paper(CWhite);
 PrintString('Q(uit ');
 Ens_Comm:=[8,10,11,21,27,77,109,83,115,78,110,72,104,67,99,65,97,82,114,
69,101,68,100,81,113];
 Repeat
 Repeat

```
Lecturech;  
If Not(Ord(ch) in Ens_Comm) Then Note(12,20);  
Until Ord(ch) in Ens_Comm;  
Remplin;  
Case Ord(ch) of  
77,109:Begin Num_Comm:=2; P2; End;  
83,115:Begin Num_Comm:=3; P3; End;  
78,110:Begin Num_Comm:=1; P1; End;  
72,104:Begin Num_Comm:=8; P8; End;  
67,99:Begin Num_Comm:=7; P7; End;  
65,97:Begin Num_Comm:=5; P5; End;  
82,114:Begin Num_Comm:=6; P6; End;  
88,120:Begin Num_Comm:=4; P4; End;  
69,101:Begin Num_Comm:=9; P9; End;  
68,100:Begin Num_Comm:=25; P25; End;  
81,113:Begin Num_Comm:=26; P26; End;  
8:Depl_Gauche:=True;  
10:Depl_Bas:=True;  
11:Depl_Haut:=True;  
21:Depl_Droite:=True;  
27:Escape:=True;  
End;  
Until Fin;  
End;  
  
Procedure P27;Forward;  
  
Procedure P28;  
Begin  
Fin:=False;  
Esc;  
If Fin Then P23;  
End;  
  
Procedure P29;  
Begin  
Fin:=False;  
Valider;  
If Fin Then P23;  
End;  
  
Procedure P27;  
Begin  
Fin:=False;  
Window(0,279,167,191);Paper(CWhite);  
Fillport;  
Window(0,279,0,191);  
MoveAbs(2,183);  
Ink(CBlack);  
Paper(CGrey1);  
PrintString('E(sc ^);  
Paper(CGrey1);  
PrintString('V(alider ^);  
Ens_Comm:=[8,10,11,21,27,69,101,86,118];  
Repeat  
Repeat  
Lecturech;  
If Not(Ord(ch) in Ens_Comm) Then Note(12,20);  
Until Ord(ch) in Ens_Comm;  
Permuter;  
Case Ord(ch) of  
69,101:Begin Num_Comm:=28; P28; End;  
86,118:Begin Num_Comm:=29; P29; End;  
8:Depl_Gauche:=True;  
10:Depl_Bas:=True;  
11:Depl_Haut:=True;
```

```

21:Depl_Droite:=True;
27:Escape:=True;
End;
Until Fin;
End;

```

```

Procedure P30;
Begin
  Fin:=False;
  FicheInf;
  P0;
End;

```

```

Procedure P33;
Begin
  Fin:=False;
  Quit;
  P0;
End;

```

```

Procedure P23;
Begin
  Fin:=False;
  Window(0,279,167,191);Paper(CWhite);
  Fillport;
  Window(0,279,0,191);
  MoveAbs(2,183);
  Ink(CBlack);
  Paper(CLightBlue);
  PrintString('R(emplin ');
  Paper(CLightBlue);
  PrintString('P(ermuter ');
  Paper(CLightBlue);
  PrintString('F(icheInf ');
  Paper(CLightBlue);
  PrintString('Q(uit ');
  Ens_Comm:=[8,10,11,21,27,82,114,80,112,70,102,81,113];
  Repeat
    Repeat
      Lecturech;
      If Not(Ord(ch) in Ens_Comm) Then Note(12,20);
    Until Ord(ch) in Ens_Comm;
    Modifier;
  Case Ord(ch) of
    82,114:Begin Num_Comm:=24; P24; End;
    80,112:Begin Num_Comm:=27; P27; End;
    70,102:Begin Num_Comm:=30; P30; End;
    81,113:Begin Num_Comm:=33; P33; End;
    8:Depl_Gauche:=True;
    10:Depl_Bas:=True;
    11:Depl_Haut:=True;
    21:Depl_Droite:=True;
    27:Escape:=True;
  End;
  Until Fin;
End;

```

```

Procedure P34;
Begin
  Fin:=False;
  Imprimer;
End;

```

```

Procedure P35;
Begin
  Fin:=False;

```

```
Arret;
End;

Procedure P0;
Begin
  Ens_Comm:=[];
  Fin:=False;
  Ens_Comm:=[8,10,11,21,27,67,99,83,115,70,102,77,109,73,105,65,97];
  Repeat
  Window(0,279,167,191);Paper(CWhite);
  Fillport;Window(0,279,0,191);
  MoveAbs(2,183);
  Ink(CBlack);
  Paper(CGreen);
  PrintString('C(hanger ^)');
  Paper(CGreen);
  PrintString('S(auver ^)');
  Paper(CGreen);
  PrintString('F(ermer ^)');
  Paper(CGreen);
  PrintString('M(odifier ^)');
  Paper(CGreen);
  MoveAbs(2,175);
  PrintString('I(mprimer ^)');
  Paper(CGreen);
  PrintString('A(rret ^)');
  Repeat
    Lecturech;
    If Not(Ord(ch) in Ens_Comm) Then Note(12,20);
  Until Ord(ch) in Ens_Comm;
  Case Ord(ch) of
    67,99:Begin Num_Comm:=17; P17; End;
    83,115:Begin Num_Comm:=21; P21; End;
    70,102:Begin Num_Comm:=22; P22; End;
    77,109:Begin Num_Comm:=23; P23; End;
    73,105:Begin Num_Comm:=34; P34; End;
    65,97:Begin Num_Comm:=35; P35; End;
    8:Depl_Gauche:=True;
    10:Depl_Bas:=True;
    11:Depl_Haut:=True;
    21:Depl_Droite:=True;
    27:Escape:=True;
  End;
Until Fin;
End;
Begin
  Table_Des_Couleurs[1]:=CGrey1;
  Table_Des_Couleurs[2]:=CAqua;
  Table_Des_Couleurs[3]:=CLightBlue;
  Table_Des_Couleurs[4]:=CYellow;
  Table_Des_Couleurs[5]:=CPurple;
  Table_Des_Couleurs[6]:=CPink;
  Table_Des_Couleurs[7]:=CMagenta;
  Table_Des_Couleurs[8]:=COrange;
  Table_Des_Couleurs[9]:=CGreen;
  Table_Des_Couleurs[16]:=CWhite;
  Debut;
  P0;
  Final;
End.
```

ANNEXE 4 : Structure des fichiers contenant les plannings:

Si l'on nomme "ROULEMENT" une période de 28 jours

Si NbMaxInf représente le nombre maximum d'infirmières dans le service

Si PlanXRoul représente le nombre de jours planifiés à un instant, soit trois roulements ou 84 jours

Si NbGenreService est le nombre de type de service par jour (ici 3)

Deux fichiers vont mémoriser les plannings:

UN FICHER PLANNING

Structure de l'enregistrement (=une ligne du planning):

- * une variable booléenne AFFECTE mémorisant l'affectation d'une infirmière à la ligne
- * le type de service de la ligne (de 1 à NbGenreService)
- * une liste d'infirmières affectées à cette ligne
(on peut sur trois périodes affecter, à cause des permutations, plusieurs infirmières sur une même ligne. Le maximum est de 9)

Pour chaque infirmière on mémorise:

- l'indice de l'infirmière (de 0 à NbMaxInf)
- le début de service: indique la première case du planning à partir de laquelle l'infirmière est affectée à cette ligne (permutation)
- la fin de service: indique la case finale sur la ligne où l'infirmière subit à nouveau une permutation et change de service
- ligne suivante: indique la nouvelle ligne d'affectation de l'infirmière si la fin de service est différente de 0

- * un tableau à une dimension stockant, pour chaque jour l'état de l'infirmière affectée: cet état correspond à une couleur particulière attribuée à travers le gestionnaire de menus.

Si une infirmière est affectée à une seule ligne du tableau, l'emploi du temps qu'elle aura à effectuer sur trois périodes est entièrement mémorisé dans ce dernier tableau.

Ce fichier contient autant d'enregistrements que d'infirmières à affecter (travaillant dans le service).

La clé d'accès dans ce fichier est l'indice de l'infirmière.

UN FICHER TOTAUX

Structure de l'enregistrement:

- * un tableau à deux dimensions TOTAL, mémorisant le total sur les lignes du planning précédent des infirmières affectées par type de service et par jour.
- * une date de début du planning de trois périodes.
- * le nombre d'infirmières à affecter à la date de début de planning.

Ce fichier ne contient qu'un seul enregistrement.

Pour des raisons techniques,

UN FICHER MEMOIRE

Même si, physiquement, pour des raisons de contrôle, les trois plannings sont mémorisés sur un seul fichier, l'utilisateur va travailler sur trois plannings distincts (analogie avec les trois feuilles).

Nous avons donc besoin de mémoriser le planning de travail pour "redémarrer" sur celui-ci après un arrêt de l'appareil. Nous stockons donc le numéro du planning écran lorsque l'utilisateur quitte le logiciel.

ANNEXE 5 : Programme de "Scrolling".

```
.MACRO POP
PLA
STA %1
PLA
STA %1+1
.ENDM
```

```
.MACRO PAP
PLA
STA %1
PLA
.ENDM
```

```
.PROC SCROLH,5
```

```
SCROL(PREMCOL,DERNCOL,HAUTFENET,BASFENET,DEPLCOL)
```

```
RETURN .EQU 0
NBLIG .EQU 2
LIGCD .EQU 3
LIGCA .EQU 4
ADRDEF .EQU 5
ADARRIV .EQU 7
NBCOL .EQU 9
COUNT .EQU 10
COLDEP .EQU 11
COLAR .EQU 12
PRELIG .EQU 13 ;PREMIERE LIGNE DE LA FENETRE SI SCROLLING VERS LE
DERLIG .EQU 14 ;BAS,DERNIERE SINON.
DEPLCOL .EQU 15
COLCD .EQU 16
COLCA .EQU 17
NBCOUR .EQU 18
```

```
POP RETURN
PAP DEPLCOL
PAP DERLIG
PAP PRELIG
PAP COLAR
PAP COLDEP
```

```
LDA 0C018 ;80STORE
PHA
LDA 0C01D ;HIRES
PHA
```

```
STA 0C002
STA 0C004
STA 0C008
STA 0C001
STA 0C057
STA 0C054 ;PAGE2 OFF
```

```

CLO
SEC
LDA DERLIG
SBC PRELIG
STA NBLIG
INC NBLIG
LDX PRELIG
DEX
STX LIGCD

```

```

NEXTLIG LDX LIGCD
LDA HIADR,X
STA ADRDEP+1
LDA LOADR,X
STA ADRDEP
INC LIGCD

```

```

TRANSF LDA #8
STA COUNT

```

```

LOOP3 SEC
LDA COLDEP
SBC COLAR
BMI DROITE
JMP GAUCHE

```

```

DROITE SEC
LDA COLAR
SBC COLDEP
STA NBCOL
SEC
SBC DEPLCOL
STA NBCOUR
INC NBCOUR
LDX COLAR
DEX
STX COLCA
SEC
LDA COLAR
SBC DEPLCOL
STA COLCD
DEC COLCD

```

```

L001 LDY COLCD
LDA @ADRDEP,Y
LDY COLCA
STA @ADRDEP,Y
DEC COLCD
DEC COLCA
DEC NBCOUR
BNE L001

LDA 0C01C
BMI LIV
STA 0C055
JMP DROITE

```

```

LIV JMP LIGSUIV

```

```

GAUCHE SEC
LDA COLDEP
SBC COLAR
STA NBCOL
SEC

```

```
NBC DEPLCOL
STA NBCOUR
INC NBCOUR
LDX COLAR
DEY
STX COLCA
CLC
LDA COLAR
ADC DEPLCOL
STA COLCD
DEC COLCD
```

- 113 -

```
LOOP1 LDY COLCD
      LDA @ADRDEP,Y
      LDY COLCA
      STA @ADRDEP,Y
      INC COLCD
      INC COLCA
      DEC NBCOUR
      BNE LOOP1
```

```
      LDA 0C01C
      BMI LIGSUIV
      STA 0C055
      JMP GAUCHE
```

```
LIGSUIV STA 0C054
```

```
      CLC
      LDA ADRDEP+1
      ADC QUATRE
      STA ADRDEP+1
```

```
      DEC COUNT
      BNE LOP3
      JMP SUI
```

```
LOP3  JMP LOOP3
```

```
SUI   DEC NBLIG
      BEQ FINI
      JMP NEXTLIG
```

```
FINI  PLA
      BMI OLDSTOR
      STA 0C056
```

```
OLDSTOR PLA
      BMI FINOLD
      STA 0C000
```

```
FINOLD LDA RETURN+1
      PHA
      LDA RETURN
      PHA
      RTS
```

```
HIADR .BYTE 20,20,21,21,22,22,23,23
      .BYTE 20,20,21,21,22,22,23,23
      .BYTE 20,20,21,21,22,22,23,23
```

```
LOADR .BYTE 00,80,00,80,00,80,00,80
      .BYTE 28,0A8,28,0A8,28,0A8,28,0A8
      .BYTE 50,0D0,50,0D0,50,0D0,50,0D0
```

```
QUATRE .BYTE 04
```

```
      END
```

ANNEXE 6 : Détail des menus.

MENU PRINCIPAL

Six commandes sont accessibles:

a) C(hanger

Cette commande permet de changer le planning de travail. Un sous-menu est proposé comprenant trois commandes

U(n permet de passer au "Planning en cours"

D(eux permet de passer au "Planning futur proche"

T(rois permet de passer au "Planning futur lointain"

Chacune de ces trois commandes ramène au menu principal

b) S(auver

Permet de sauvegarder les modifications effectuées sur le planning de travail: sécurité en cas de coupure de courant et surtout permet les simulations sur le planning de travail avec retour en arrière possible.

c) F(ermer

Permet d'abandonner le "planning en cours" après mise à jour des fiches infirmières. Le planning 2 devient le 1, le planning 3 devient le 2, génération d'un troisième planning (nouveau "planning futur lointain").

d) M(odifier

Permet l'élaboration et la modification du planning de travail. Commande reprise plus loin.

e) I(mprimer

Permet l'impression d'une feuille de planning. Le service ne disposant pas d'une imprimante couleur, les couleurs sont remplacées par des codes. Cette impression ressemble le plus possible à la feuille précédemment constituée manuellement.

On a ainsi la possibilité d'éditer à tout instant les plannings modifiés pour affichage dans le service.

Un exemple d'édition se trouve en annexe 4.

f) A(rret

Permet la sortie du logiciel en demandant si l'utilisateur veut sauvegarder ses modifications.

Menu MODIFIE

Quatre commandes du sous-menu de la commande M(odifie

a) R(emplir

Donne accès à un autre sous-menu pour affecter les états aux jours. Ce sous-menu sera exposé un peu plus loin.

b) P(ermuter

Demande à l'utilisateur de se positionner à l'aide des flèches sur la ligne de la première infirmière à permuter, de rentrer un retour charriot pour valider, puis sur la seconde et de valider par retour charriot.

Après avoir enregistré les deux infirmières à permuter, propose un menu de deux commandes:

* E(sc pour abandonner sans modifier

* V(alider pour valider cette modification

Ces deux commandes remontent automatiquement au niveau du menu M(odifie.

c) F(ichinf

Permet la consultation de la fiche de l'infirmière sur laquelle est positionné le curseur.

d) Q(uit

Permet de quitter ce niveau et de remonter au niveau du menu principal.

Menu REMPLIR

Onze commandes qui permettent de changer l'affectation de départ de l'infirmière pour une mise à jour, une élaboration ou une modification.

a) M(atin

affecte un service Matin à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

b) S(oir

affecte un service Soir à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

c) N(uit

affecte un service Nuit à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

d) H(ebdo

affecte l'état repos hebdomadaire à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

e) C(onge

affecte, après vérification, l'état congé annuel à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

La vérification des droits de l'infirmière à ce jour de congé s'effectue à partir de la fiche infirmière et des affectations depuis le début du planning 1 jusqu'au jour correspondant au curseur.

Un signal sonore est émis si l'affectation ne devrait pas être permise.

f) A(bsc

affecte l'état congé exceptionnel non prévisible à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

g) R(ecup

affecte l'état récupération d'un jour férié à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

La vérification est effectuée à partir du tableau des dates récupérées dans la fiche infirmière (voir plus loin).

Un signal sonore est émis si l'affectation ne devrait pas être permise.

h) X(39eH

affecte la journée des 39h à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

La vérification des droits de l'infirmière à ce jour s'effectue à partir de la fiche infirmière et des affectations depuis le début du planning 1 jusqu'au jour correspondant au curseur.

Un signal sonore est émis si l'affectation ne doit pas être permise.

i) E(xept

affecte l'état congé exceptionnel prévisible à la case où se trouve positionné le curseur et déplace le curseur d'une position vers la gauche.

j) D(ate

donne la date du jour sur lequel est positionné le curseur.

k) Q(uit

Permet de quitter ce menu et de remonter au niveau du menu M(odifie

Toutes ces "procédures commandes" sont regroupées dans deux modules (un pour la commande Remplir).

Les autres modules contiennent la gestion d'écran, ...

Toute cette première partie permet donc aux responsables du service d'effectuer les mêmes opérations que manuellement, en ayant l'avantage

- * de stocker toute l'information,
- * de rechercher celle relative à l'infirmière de manière beaucoup plus rapide,
- * et surtout de pouvoir éditer la feuille planning à la demande après modifications.

ANNEXE 8 : Structure des données des Fiches infirmières.

A-Fichier infirmières: FINF

Structure d'un enregistrement:

- * 3 initiales identifiant l'infirmière.
- * le nom sur 15 caractères maximum
- * le prénom sur 15 caractères maximum
- * une table des récupérations:

Pour chaque jour donnant droit à une récupération, on a :

-la date du jour récupéré ou ../../..

-le code d'affectation

-non affecté si jour récupérable et date du jour >
date du jour férié.

-affecté si jour récupéré

-auto-affecté si jour non récupérable.

-le nombre de jours de récupérations à prendre restant de
l'année précédente.

-le planning individuel annuel

C'est le résumé de l'activité de l'infirmière sur
douze mois, jour par jour.

-la date de début d'activité dans le service.

-la date de fin d'activité dans le service.

Ces deux dernières dates servent à faire apparaître et
disparaître l'infirmière dans le planning écran.

La clé d'accès du fichier est le bloc initiales.

Pour chaque infirmière on doit mémoriser la date du jour férié
récupérable, car l'infirmière n'a pas le droit d'anticiper la
récupération.

Ces dates des jours récupérables sont fixées au début de chaque année civile: le stockage va s'effectuer sur un fichier "Date Récup"

B-Fichier des dates de récupération:

Structure d'un enregistrement:

- * un date comprenant
 - le jour (0..31)
 - le mois (0..12)
 - l'année (0..99)

La création et modification s'effectue à l'aide d'un petit programme de mise à jour qui permet aussi de rentrer dans un fichier de données quelques paramètres.

C-Fichier de données:

Structure des données:

- * le nombre de récupération dans l'année (jours fériés + ponts)
- * le nombre de jours de congé auquel a droit l'infirmière.
- * le nombre de jours travaillés donnant droit à un jour (nb variable avec la réglementation du travail.
- * le nombre de nuits à travailler par an.

Si une infirmière a un régime particulier on pourra modifier ce nombre de nuits dans sa fiche.

ANNEXE 9 : Fiche infirmière et listes produites.

** ENREGISTREMENT INFIRMIER(E) **

INITIALE : DUR

NOM : DURET

PRENOM : CLAUDIE

** PLANNING INFIRMIER(E) **

JANVIER	IC-CCANNMMSS AAEERR -M	FEVRIER	
MARS	IMMSHH E	AVRIL	
MAI		JUIN	
JUILLET	- R	AOUT	
SEPTEMBRE		OCTOBRE	
NOVEMBRE		DECEMBRE	5

I-FAITES : 4 I-PRIS : 3
 NB_NUITS | DIMANCHES FAITS : 0 CONG_ANI
 I-AFAIRE : 71 I-RESTANTS: 24 CONG_EXEPTIONNELS : 3

** DATES DES JOURS RECUPERES **

* RECUPERATIONS ANNEE PRECEDENTE : 0

1_ 1/ 1/84: 1/ 1/84	2_ 22/ 4/84:19/ 1/84	3_ 23/ 4/84:20/ 1/ 01	4_ 30/ 4/84:23/ 7/ 01
5_ 1/ 5/84: 0/ 0/ 01	6_ 8/ 5/84: 0/ 0/ 01	7_ 31/ 5/84: 0/ 0/ 01	8_ 10/ 6/84: 0/ 0/ 01
9_ 11/ 6/84: 0/ 0/ 01	10_ 14/ 7/84: 0/ 0/ 01	11_ 15/ 8/84: 0/ 0/ 01	12_ 1/ 11/84: 0/ 0/ 01
13_ 11/ 11/84: 0/ 0/ 01	14_ 24/ 12/84: 0/ 0/ 01	15_ 25/ 12/84: 0/ 0/ 01	16_ 31/ 12/84: 0/ 0/ 01

** LISTE DES RECUPERATIONS **

- 124 -

DATE : 1/ 1/84		
BRIGARD	SYLVIANE	7/ 4/84
CHEVROLET	D	1/ 1/84
DURET	CLAUDIE	1/ 1/84
ELNACHEF	ANMAR	1/ 1/84
GONZALES	HERVE	1/ 1/84
LEMORDANT	JACQUES	1/ 1/84
TOTO	GH	1/ 1/84
DATE : 22/ 4/84		
CHEVROLET	D	23/ 7/84
DURET	CLAUDIE	6/ 1/84
GONZALES	HERVE	22/ 4/84
LEMORDANT	JACQUES	22/ 4/84
TOTO	GH	22/ 4/84
DATE : 23/ 4/84		
GONZALES	HERVE	23/ 4/84
LEMORDANT	JACQUES	23/ 4/84
TOTO	GH	23/ 4/84
DATE : 30/ 4/84		
GONZALES	HERVE	30/ 4/84
LEMORDANT	JACQUES	30/ 4/84
TOTO	GH	30/ 4/84
DATE : 1/ 5/84		
TOTO	GH	1/ 5/84
DATE : 8/ 5/84		
TOTO	GH	8/ 5/84
DATE : 31/ 5/84		
TOTO	GH	31/ 5/84
DATE : 10/ 6/84		
TOTO	GH	10/ 6/84
DATE : 11/ 6/84		
TOTO	GH	11/ 6/84
DATE : 14/ 7/84		
TOTO	GH	14/ 7/84
DATE : 15/ 8/84		
TOTO	GH	15/ 8/84
DATE : 1/11/84		
TOTO	GH	1/11/84
DATE : 11/11/84		
TOTO	GH	11/11/84
DATE : 24/12/84		
DATE : 25/12/84		
DATE : 31/12/84		

ANNEXE 10: Courbes relatives aux diverses expérimentations

Energie

o Temperature

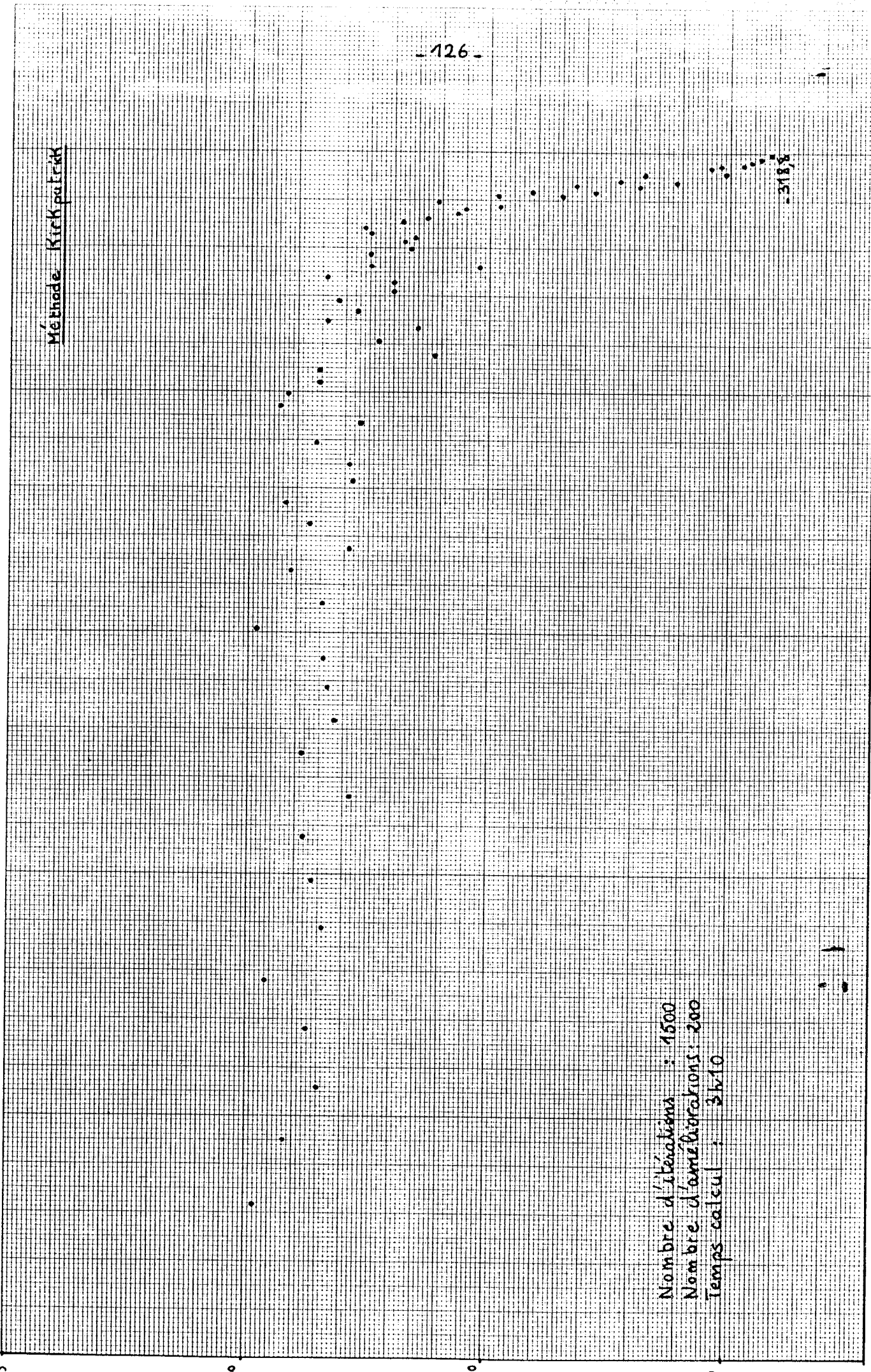
Méthode Kirkpatrick

-126-

-318,8

Nombre d'itérations : 1500
Nombre d'améliorations : 200
Temps calcul : 3h10

500 400 300 200 100



Energie

Méthode Kirkpatrick

-127-

0 Température

100

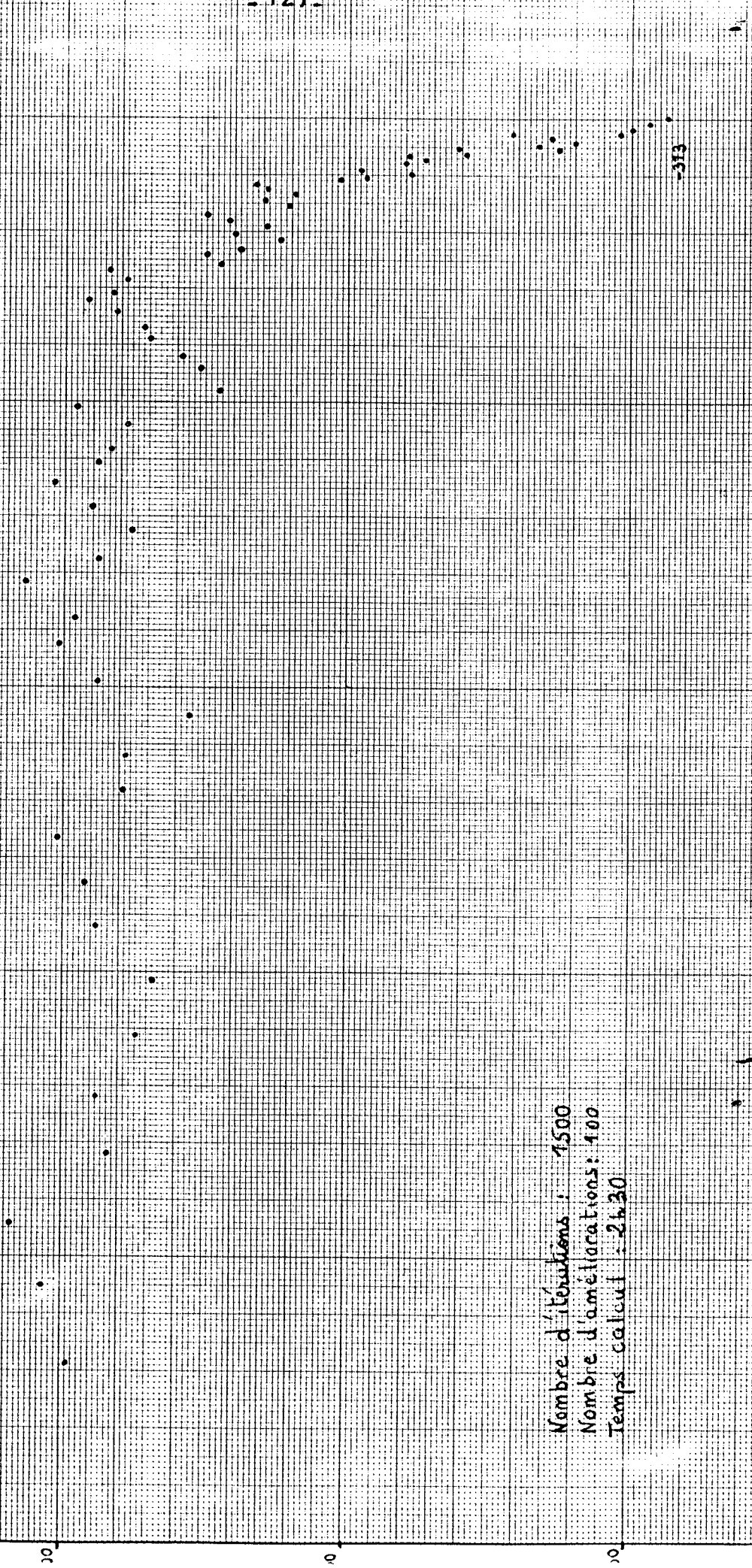
200

300

400

500

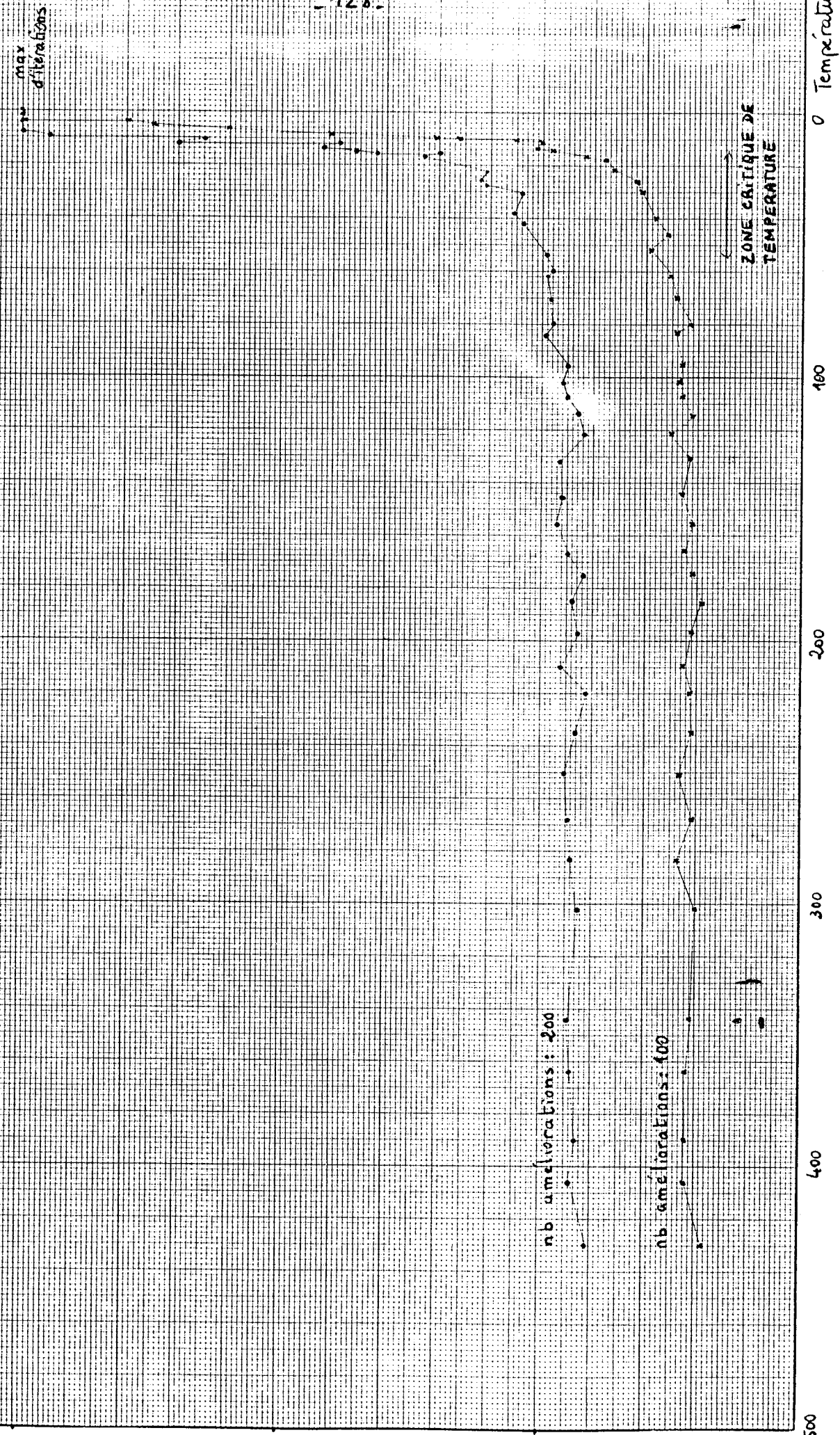
Nombre d'itérations : 1500
Nombre d'améliorations : 100
Temps calcul : 2h30



↑ Nombre d'itérations

Méthode Kikspakich

-128-



500

400

300

200

100

0

Température

Energie

Méthode de Trempe

- 129 -

n° essai

10

10

0

10

20

30

40

50

60

70

80

90

100

110

120

130

140

150

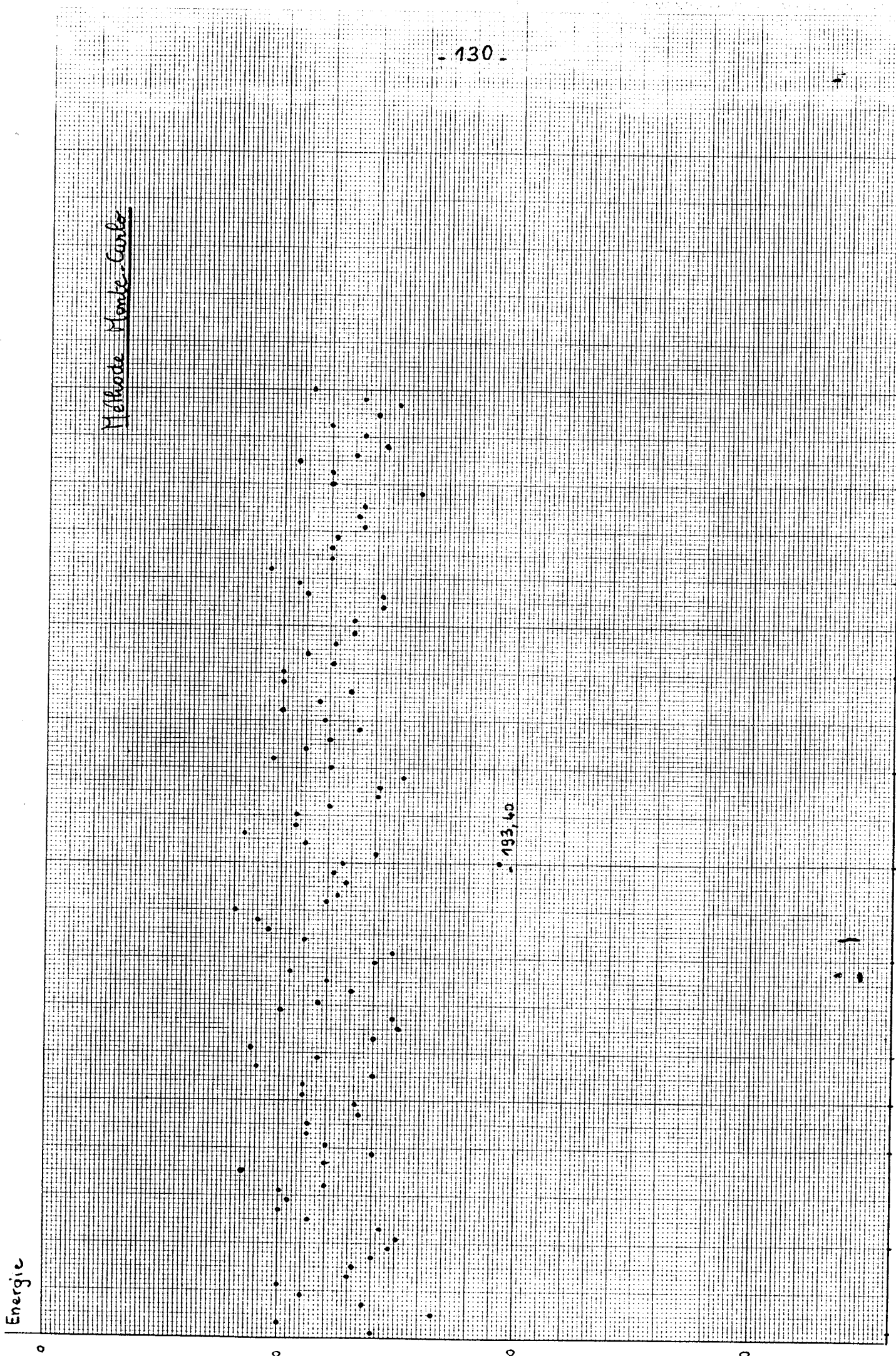
Méthode Monte-Carlo

Energie

n° essai

100
90
80
70
60
50
40
30
25
20
10

193,40



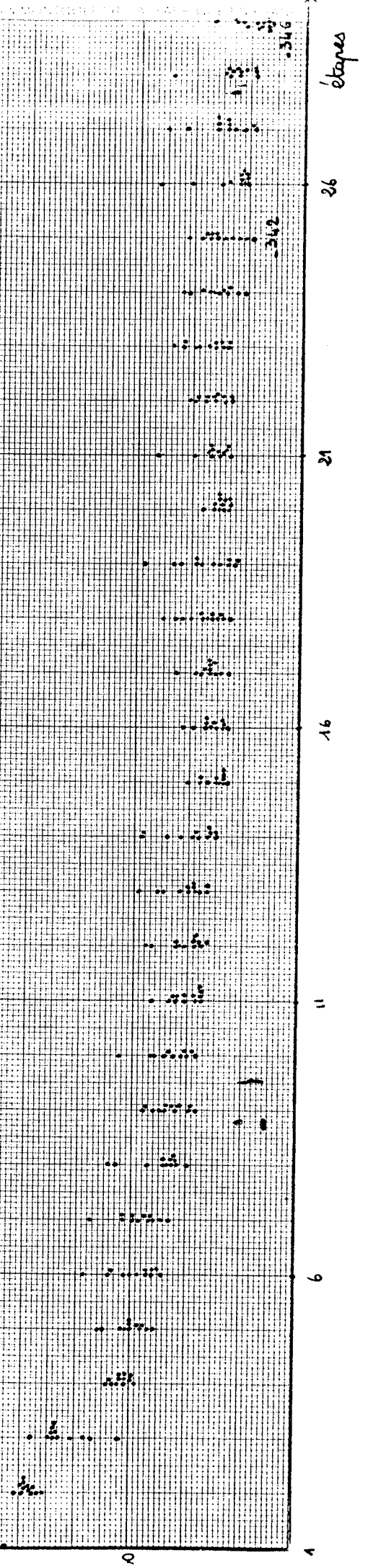
Energie

T=20 critère a : on garde la meilleure solution

nb iter = 500

nb amelior = 100

Méthode HN



Méthode HN

T=20 critère a : on garde la meilleure solution

nb iter = 100

nb amelior = 50

↑ ENERGIE

0

0

0

0

1

6

11

16

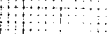
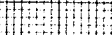
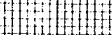
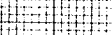
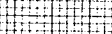
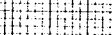
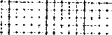
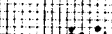
21

26

-341

-328

étapes



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

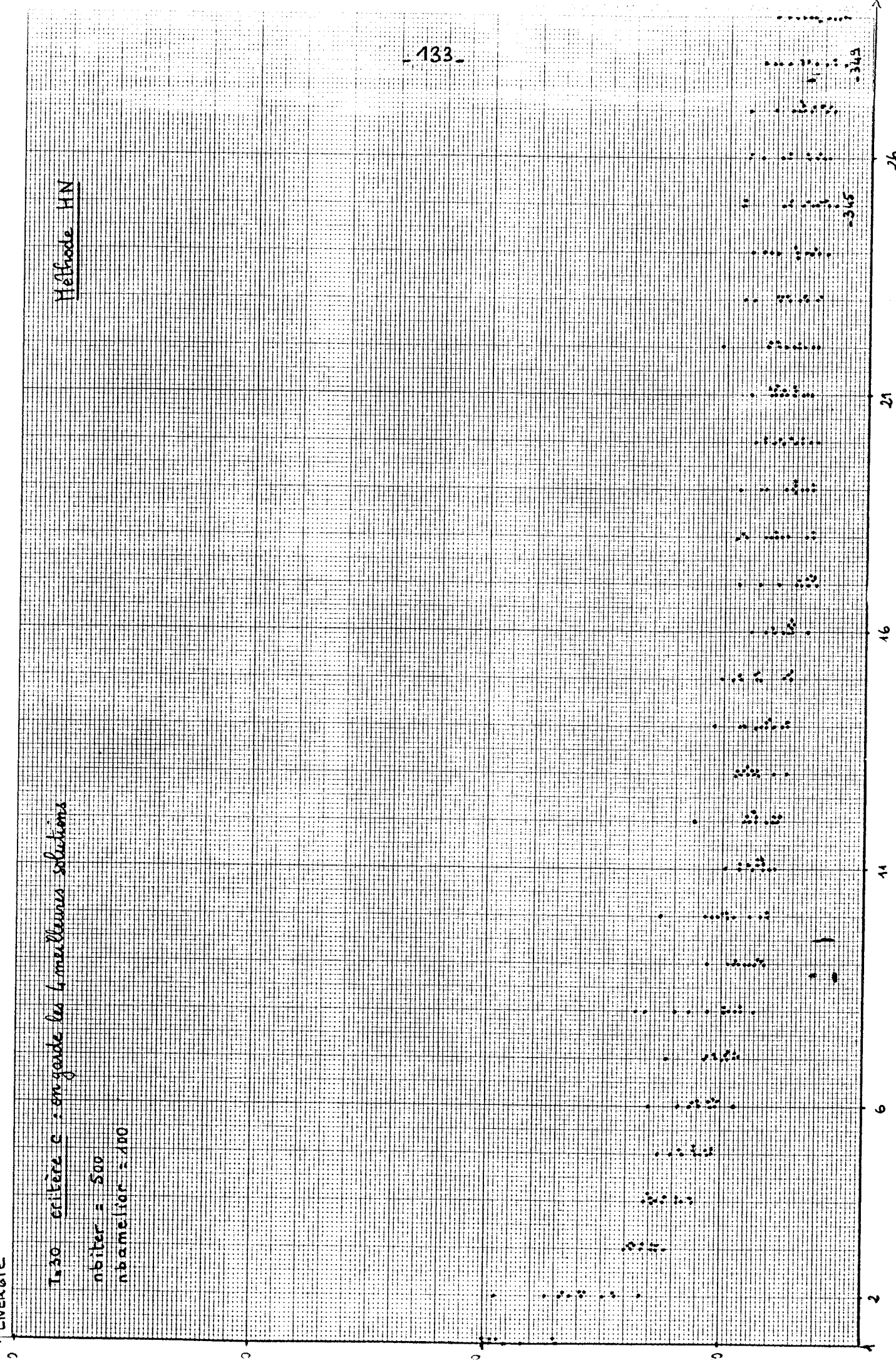


T=30 critère c : on garde les 4 meilleures solutions

nbiter = 500

nbamelior = 100

Méthode HN



-349

-345

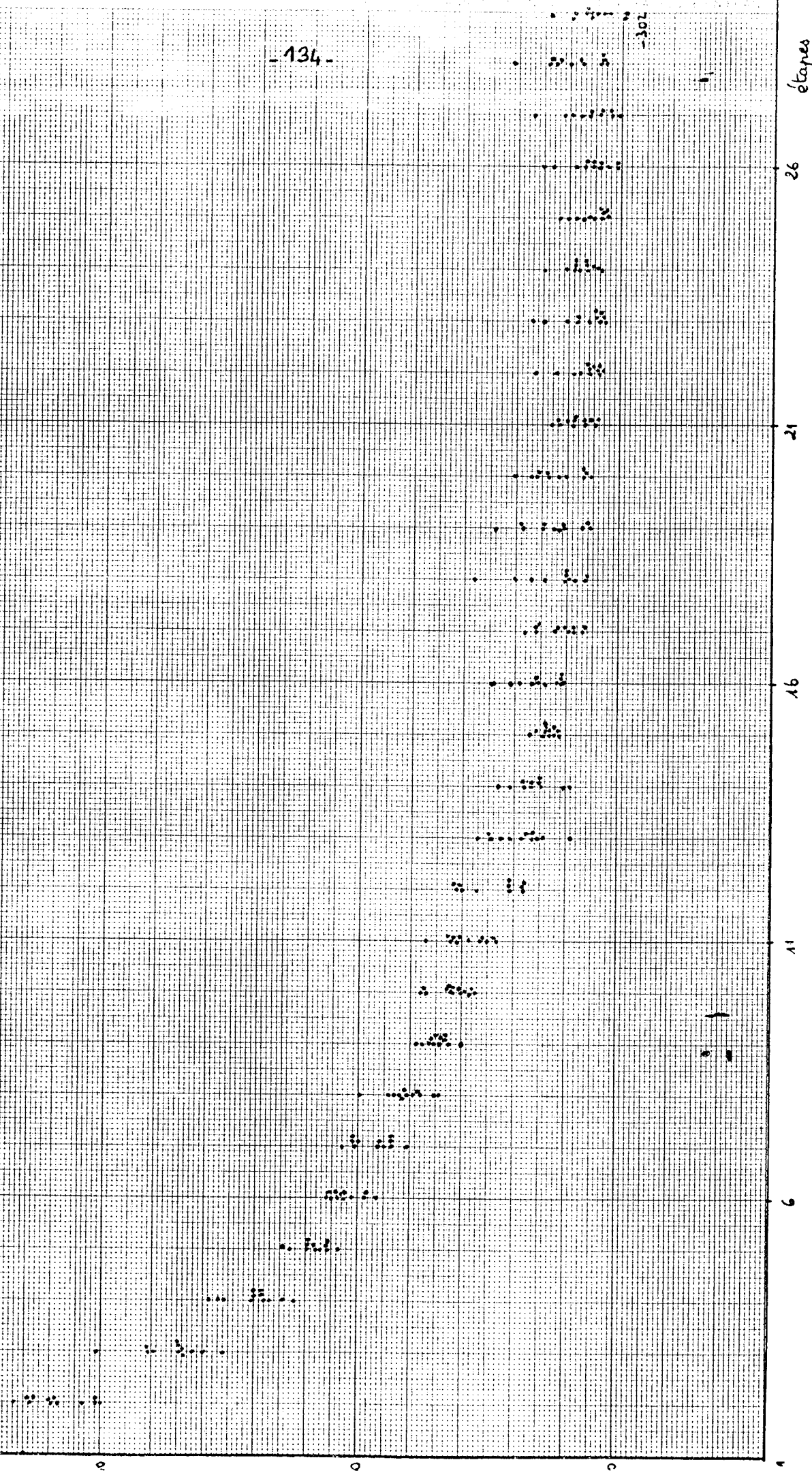
Méthode HN

T=20 critère a : on garde la meilleure solution

n biter = 20

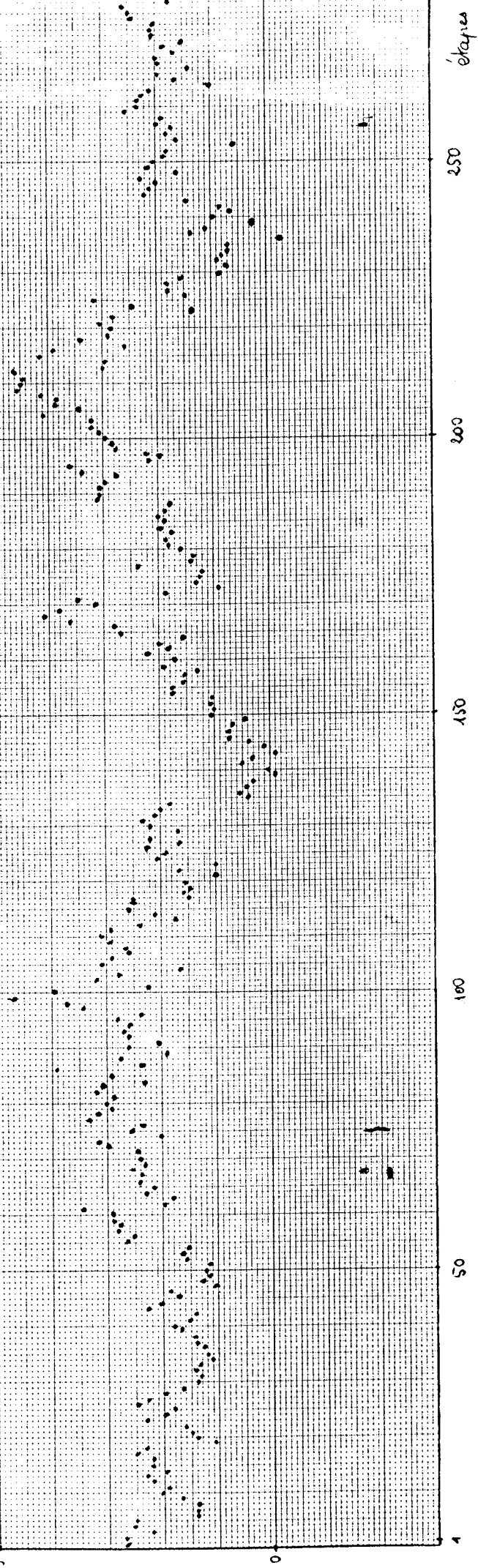
nb amplitud = 10

↑ Energie



ENERGIE

On laisse évoluer un seul processus
 On mémorise la moyenne solution de chq étape
 T=20 mbariter = 500
 nbarometre = 100



étapes
 250
 200
 150
 100
 50
 0