

MEMOIRE

présenté en vue d'obtenir

**LE DIPLOME D'INGENIEUR I.I.E.**

Rapport Final

**Cédric BENTZ**

CONCEPTION DES ROULEMENTS POUR LE  
PERSONNEL DE L'ESCALE

Directeur de stage : M. Daniel GAUYACQ, Responsable de l'unité de  
recherche Génie Décisionnel Appliqué à la Direction de la Recherche  
et de la Technologie de la SNCF

# Table des matières

<b>1 CADRE DE LA RECHERCHE.....</b>	<b>4</b>
1.1 ACCUEIL ET ENCADREMENT.....	4
1.2 REMERCIEMENTS.....	5
1.3 PRÉSENTATION DU SUJET.....	6
1.3.1 <i>Notions élémentaires de planification de roulement</i> .....	6
1.3.2 <i>Description détaillée de la conception d'un roulement 3*8</i> .....	8
1.3.2.1 Cadre de travail – Hypothèses.....	8
1.3.2.2 Contraintes imposées par la législation ou l'expérience.....	8
1.3.2.3 Objectifs visés lors de la conception des roulements.....	9
1.4 INTÉRÊT DU TRAVAIL.....	10
1.4.1 <i>Etat actuel des connaissances</i> .....	10
1.4.2 <i>Utilité du travail</i> .....	10
<b>2 TRAVAIL RÉALISÉ.....</b>	<b>11</b>
2.1 REPRÉSENTATION DU PROBLÈME : DONNÉES ET VARIABLES UTILISÉES.....	11
2.1.1 <i>Représentation du tableau de roulement</i> .....	11
2.1.2 <i>Données du problème</i> .....	12
2.1.3 <i>Représentation des variables du problème</i> .....	13
2.1.3.1 Variables entières.....	13
2.1.3.2 Variables continues.....	13
2.1.3.3 Variables bivalentes.....	13
2.1.3.3.1 Périodes de travail.....	13
2.1.3.3.2 Périodes de repos.....	14
2.2 EXPRESSION FORMELLE DES CONTRAINTES ET OBJECTIFS DU PROBLÈME.....	16
2.2.1 <i>Expression des contraintes</i> .....	16
2.2.1.1 Contrainte d'unicité.....	16
2.2.1.2 Contraintes intra-GPT.....	17
2.2.1.3 Contraintes inter-GPT.....	17
2.2.1.4 Contraintes sur les repos.....	20
2.2.1.5 Contrainte de charge.....	23
2.2.1.6 Contraintes liées aux variables additionnelles.....	24
2.2.2 <i>Expression des préférences et des objectifs du problème</i> .....	26
2.2.2.1 Préférences liées à l'utilisation de la réserve.....	26
2.2.2.2 Préférences liées à l'attribution des repos.....	26
2.2.3 <i>Remarques</i> .....	27
2.2.3.1 Réduction du nombre de critères.....	27
2.2.3.2 Résultat important.....	27
2.3 VOLUMÉTRIE.....	28
2.3.1 <i>Nombre de variables (avec exemple pour un 3*8 monté avec 4 agents sur 4 semaines)</i> .....	28
2.3.2 <i>Nombre de contraintes (avec exemple pour un 3*8, avec 4 agents sur 4 semaines)</i> .....	29
2.4 IMPLANTATION DU MODÈLE ET EXPÉRIMENTATIONS.....	30
2.4.1 <i>Présentation du logiciel OPL Studio de la société ILOG</i> .....	30
2.4.1.1 Le fichier « .mod ».....	30
2.4.1.2 Le fichier « .dat ».....	30
2.4.2 <i>Validation préliminaire et enrichissement du modèle</i> .....	31
2.4.2.1 Hiérarchisation des critères : utilisation de coefficients non-compensatoires.....	31
2.4.2.1.1 Présentation de la méthode.....	31
2.4.2.1.2 Application au problème de la conception de roulement.....	32
2.4.2.2 Ajout de contraintes supplémentaires.....	33
2.4.2.2.1 Contrainte de type 1, sur l'utilisation quotidienne de réserve.....	33
2.4.2.2.2 Contrainte de type 2, sur la réserve globale.....	33
2.4.2.2.3 Contrainte de type 3, sur le nombre de jours de travail.....	33
2.4.2.2.4 Contrainte de type 4, sur la différence entre soirées et matinées de réserve.....	34
2.4.2.2.5 Bilan de l'apport des contraintes supplémentaires.....	35
2.4.2.3 Définition de priorités dans l'arbre de recherche.....	36
2.4.3 <i>Etude de cas concrets</i> .....	37

2.4.3.1 Roulement de type « 3*8 » sur 2 semaines.....	37
2.4.3.2 Roulement de type « 3*8 » sur 3 semaines.....	39
2.4.3.3 Roulement de type « 3*8 » sur 4 semaines.....	42
2.4.3.4 Roulement à besoins variables sur 2 semaines.....	45
2.4.3.5 Bilan.....	46
2.5 EVOLUTION DU MODÈLE .....	47
2.5.1 Conception de roulements « quelconques ».....	47
2.5.2 Groupes de roulements.....	47
2.5.3 Prise en compte des remarques des experts.....	48
<b>3 CONCLUSION.....</b>	<b>49</b>
<b>4 BIBLIOGRAPHIE.....</b>	<b>50</b>
<b>5 ANNEXES.....</b>	<b>52</b>
5.1 CALCUL DU NOMBRE D'AGENTS NÉCESSAIRES POUR TENIR UN 3*8.....	52
5.2 TABLEAU RÉCAPITULATIF DES CONTRAINTES ET CRITÈRES D'UN ROULEMENT DE TYPE « 3*8 ».....	53
5.3 TABLEAU ILLUSTRATIF DES DONNÉES.....	54
5.4 TABLEAU RÉCAPITULATIF DES PLAGES HORAIRES COUVERTES PAR UNE VARIABLE QK.....	54
5.5 TABLEAU RÉCAPITULATIF DES PLAGES HORAIRES COUVERTES PAR UNE VARIABLE CK.....	55
5.6 TABLEAU ILLUSTRANT LE PHÉNOMÈNE DE CYCLICITÉ DES ROULEMENTS.....	55
5.7 ILLUSTRATION D'UN CAS DE NON RESPECT DE LA PROPRIÉTÉ PROP1.....	56
5.8 TABLEAU ILLUSTRANT LES GPT COUVRANT UNE PLAGE HORAIRE DONNÉE.....	57
5.9 CALCUL DU RÉSULTAT IMPORTANT DU 2.2.3.2.....	57
5.10 LISTING DU FICHIER « ROULEMENTS.MOD ».....	58
5.11 LISTING DU FICHIER « ROULEMENTS.DAT ».....	62

# 1 Cadre de la recherche

## 1.1 Accueil et encadrement

**Directeur de stage :** M. Daniel Gauyacq, responsable de l'unité de recherche « Génie Décisionnel Appliqué ». Bureau 166 (téléphone : 01 53 42 92 74).

**Lieu de déroulement du stage :** Bureau 278 (téléphone : 01 53 42 92 46), dans les locaux de la Direction de la Recherche et de la Technologie, à Paris S<sup>t</sup>-Lazare.

**Etablissement et service d'accueil :** Le stage s'est effectué à la Société Nationale des Chemins de Fer Français (SNCF), et plus précisément au sein de l'unité de recherche « Génie Décisionnel Appliqué » de la Direction de la Recherche et de la Technologie (DRT).

La SNCF est un Etablissement Public Industriel et Commercial au service des clients, employant 175 000 personnes et réalisant un chiffre d'affaires annuel de 77 milliards de francs. Elle est chargée du transport ferroviaire de voyageurs et de fret sur le réseau national ainsi que de l'exploitation et de la gestion de l'infrastructure (voies, signaux ...). Elle répond aux besoins des clients en proposant des produits et des services de qualité à des prix compétitifs. Elle assure sa mission dans les meilleures conditions de sécurité, de confort et de ponctualité et réalise des performances techniques remarquables (TGV, record du monde de vitesse ...).

Afin d'assurer son développement et la reconquête de ses clients, la SNCF s'est engagée dans un **projet industriel**, qui repose sur trois thèmes : le client, l'entreprise et les hommes. Il prévoit un certain nombre d'actions à mener et propose à chaque cheminot de participer à leur mise en œuvre. Ces dernières doivent garantir le développement de l'activité ferroviaire et un redressement financier durable de la SNCF « **pour être, en 2002, l'entreprise de service public de référence en France et en Europe** ». La recherche est l'un des grands axes de ce projet industriel. La SNCF compte 240 chercheurs et ingénieurs de recherche, dont 70 à la DRT. Définie par le président Louis Gallois comme essentielle au renouveau de la SNCF, la DRT se voit confortée dans son rôle transversal au sein de l'entreprise. La recherche fonctionne désormais en réseau. Toute l'entreprise est donc concernée par la recherche. En particulier dans le cadre du projet industriel, un projet élaboré par l'entreprise à partir de 1996, dont l'idée maîtresse est de tourner la SNCF vers le client.

La mission de la Direction de la Recherche et de la Technologie consiste à :

- Gérer la politique de recherche dans l'entreprise et animer le réseau ;
- Elaborer avec l'ensemble des domaines de l'entreprise, la politique, le programme, le budget de recherche et développement de la SNCF et les proposer au Comité Exécutif ;
- Coordonner pour la SNCF les coopérations et partenariats aux niveaux national et international ;
- Assurer la diffusion de l'information issue de la veille technologique ;
- Contribuer à la mise en œuvre de la politique d'innovation de l'entreprise.

La DRT est organisée en cinq Unités de Recherche : l'unité « Génie Décisionnel Appliqué » a pour mission de développer des techniques de simulation, de génie décisionnel, d'aide à la décision et de traitement des données pour améliorer l'ensemble du système ferroviaire et gérer le trafic en temps réel en tenant compte des situations perturbées.

### **Personnes présentes lors des réunions d'avancement hebdomadaires :**

- Mme Véronique Labouisse : Techniques mathématiques de la prise de décision - Débit et Capacité du réseau ferroviaire – Systèmes d'aide à la décision pour l'opérationnel ;
- M. David De Almeida : Techniques mathématiques de la prise de décision - Systèmes d'aide à la décision pour l'opérationnel.

**Environnement technique :** Windows 9x/NT, Office 97, CPLEX. En outre, des serveurs tournant sous Unix sont disponibles au Centre De Calculs.

## 1.2 Remerciements

Je tiens tout d'abord à remercier mon directeur de stage, Daniel Gauyacq, pour m'avoir permis d'effectuer ce stage au sein de la DRT de la SNCF à la gare Paris Saint-Lazare, sur un sujet qui s'est révélé très enrichissant et grâce auquel j'ai pu approfondir mes connaissances en recherche opérationnelle.

Je tiens également à adresser un grand merci à Véronique Labouisse qui m'a conseillé et guidé tout au long de mon stage, et grâce à qui le travail avec les experts de la Direction des Ressources Humaines a été facilité.

Je tiens évidemment à remercier David De Almeida qui m'a été d'un grand secours pour toutes les questions d'ordre technique.

J'adresse aussi un grand merci à tous les experts de la Direction des Ressources Humaines qui ont bien voulu nous éclairer sur les notions pas toujours évidentes et parfois un peu obscures utilisées dans la conception des roulements de personnel, et dont l'aide, que ce soit en amont ou en aval de l'étude, a été déterminante.

Dans cette optique, je tiens donc à remercier Cécile Gourdon, de la DRH de Paris Saint-Lazare, et son adjoint, Loïc Vallée.

En ce qui concerne la DRH de la gare de Paris Montparnasse, je remercie vivement Alain Cormier et Virginie Molière.

Je tiens aussi à remercier ceux qui ont également participé à la bonne humeur ambiante, et notamment les autres stagiaires : Samuel, Lionel et Michel, avec qui je partageais mon bureau, César, Delphine et Laurent.

De la même façon, je remercie les prestataires qui m'ont fait, ainsi qu'aux autres stagiaires, un excellent accueil : Nicolas, Moktar, Serge, Vincent, Laurent, Denis, et tous ceux dont j'ai oublié les noms mais pas la sympathie.

Je voudrais également remercier André Cassan qui m'a accueilli à bras ouverts dans son bureau, et je lui souhaite bonne continuation.

Enfin, je tiens à remercier tous ceux de la DRT que je n'ai pas encore cité, Gisèle, Jean-Charles, Pascal Tariel et les autres.

## 1.3 Présentation du sujet

### 1.3.1 Notions élémentaires de planification de roulement

Les services offerts par la SNCF le sont quasiment sans interruption au cours d'une journée (24h) et au cours de l'année. Cela implique de mobiliser du personnel durant des plages horaires très variées pour permettre la réalisation des tâches de production et la prise des repos et congés. Afin d'assurer une gestion efficace des moyens, il faut planifier l'utilisation du personnel de telle sorte qu'à tout moment du personnel suffisant en nombre et en compétence soit disponible pour effectuer les tâches. Par ailleurs, il faut intégrer dans la planification le fait que le personnel tourne périodiquement sur les diverses tranches horaires (matinée, soirée et nuit, couvrant typiquement les intervalles respectifs 6h-14h, 14h-22h et 22h-6h) et prend des repos. La charge de travail d'un agent est limitée sur une année et aussi quotidiennement (8 heures). La succession des périodes de travail et de repos est aussi soumise à des règles strictes notamment pour prendre en compte les week-ends. Enfin, le roulement est cyclique, c'est-à-dire qu'au terme d'un intervalle donné (1 mois ou plus), la succession tâche/repos se répète, formellement, à l'identique.

Le cadre de l'étude est le suivant : il existe un certain nombre de postes prédéfinis, et chacun d'eux nécessite des compétences particulières. Avant d'élaborer les grilles de roulement du personnel, on détermine donc les grilles de besoin : pour chaque poste, on détermine le nombre d'agents dont on aura besoin pour chaque plage horaire de la journée (c'est-à-dire Matinée, Soirée ou Nuit). Par exemple, pour chaque guichet d'une gare et pour chaque jour, on détermine s'il a besoin d'être ouvert (et donc d'être tenu par un employé) le Matin (M) et/ou le Soir (S) et/ou la Nuit (N). On sait aussi quels sont les agents susceptibles d'être associés à chaque poste (ainsi, on ne pourra pas affecter n'importe quel employé à n'importe quel poste). Par exemple, une grille de besoins d'un roulement de type « 3\*8 » se présente comme suit (tableau T1) :

Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
M	M	M	M	M	M	M
S	S	S	S	S	S	S
N	N	N	N	N	N	N

Cette grille signifie que, pour chacun des 7 jours de la semaine, ce poste nécessite un agent le matin (M), un agent le soir (S) et un la nuit (N). En fait, la grille des besoins ne donne donc les besoins que sur une semaine : ils sont ensuite identiquement répétés d'une semaine à l'autre.

Chaque jour, un agent de roulement ne peut être affecté qu'à un poste et à une plage horaire. En outre, chaque agent a droit à des jours de repos réguliers (appelés repos périodiques), mais il doit travailler un minimum entre ces repos. L'intervalle entre deux repos périodiques successifs (par la suite, on notera **RP** un repos périodique) est appelé « Grande Période de Travail », ou **GPT**. La législation exige que la longueur d'une GPT se situe entre 3 et 6 jours : autrement dit, un agent doit travailler au moins 3 jours entre 2 RP successifs, mais il ne peut pas travailler plus de 6 jours d'affilée. D'autre part, en plus des jours de repos périodique, un agent a droit à un certain nombre de repos supplémentaires (notés **RU**), dont les conditions d'attribution diffèrent quelque peu des RP.

L'élaboration des roulements consiste donc en la planification de la suite des tâches et des repos que doit assurer un agent affecté à un poste de travail à horaires décalés, dans le respect de la réglementation du travail, mais également de façon à ce que la grille des besoins soit couverte au mieux. Le nombre d'agents (ou « rouleurs ») nécessaires pour tenir un roulement est obtenu par un calcul simple (voir annexe 5.1, p. 52), et l'objectif, une fois ce calcul effectué, est alors de répartir au mieux le travail entre les agents, suivant différents critères. Il faut noter qu'il arrive que, certains jours, les « rouleurs » soient en nombre insuffisant pour couvrir la totalité de la charge de travail. Ces situations peuvent se produire à cause du mode de calcul du nombre d'agents de roulement nécessaires : si le calcul donne 4.69 agents et qu'on n'en affecte que 4 au roulement, alors il y aura forcément, à certains moments, des pénuries de ressources. Dans ces cas-là, on fait appel aux agents de

**réserve** : ce sont des agents possédant plusieurs compétences, et qui sont donc susceptibles d'être affectés sur différents types de roulements. L'objectif est bien entendu de minimiser l'appel à la réserve, qui doit rester exceptionnel.

Ainsi, la contrainte de charge s'exprime, pour chaque plage horaire de chaque journée, de la façon suivante : **nombre d'agents présents pendant cette plage horaire + nombre d'agents de réserve affectés au roulement pendant cette plage horaire = besoin en agents pendant cette plage horaire.**

Enfin, il convient de remarquer que les roulements de personnel sont organisés en cycles, chaque cycle étant composé d'un certain nombre de semaines. Il y a en fait autant de cycles qu'il y a de « rouleurs » : ainsi, un roulement auquel 4 agents sont affectés sera composé de 4 cycles. Chaque cycle possède le même nombre de semaines, qui varie souvent entre 1 et 6.

Les deux paramètres importants du tableau de roulement sont donc : le nombre de semaines d'un cycle, que l'on notera **s**, et le nombre de cycles du roulement, que l'on notera **n**. Le tableau de roulement est parcouru de la façon suivante : on parcourt, pour chaque cycle, l'ensemble de ses semaines, puis on passe au cycle suivant, etc. Lorsque tous les cycles ont été parcourus, on revient au 1<sup>er</sup> qu'on a parcouru, et on recommence.

Le nombre de jours d'un cycle, qui est le nombre de jours par semaine multiplié par le nombre de semaines par cycle, soit  $7*s$ , correspond en fait au nombre de colonnes du tableau, alors que le nombre de cycles correspond au nombre de lignes. Le nombre total de jours est donc le nombre de jours par cycle multiplié par le nombre de cycles du roulement, soit  $7*s*n$ . Le tableau de roulement est donc un tableau de  $7*s*n$  éléments, ou cases : dans chaque case, on note le service que doit tenir l'agent pour cette journée. Un agent travaillant en matinée sera noté « M », en soirée « S », et la nuit « N ». Enfin, si ce jour est un jour de repos périodique pour l'agent, on le notera « RP », et si c'est un jour de repos supplémentaire, on le notera « RU ».

Voici un exemple de tableau de roulement (tableau **T2**) :

	Semaine 1							Semaine 2							...	Semaine s						
	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di		Lu	Ma	Me	Je	Ve	Sa	Di
Cycle 1	S	S	S	RP	RP	M	M	M	M	RP	RP	RP	N	N		M	M	M	M	M	RP	RP
Cycle 2	N	N	N	N	RP	RP	S	S	S	S	RP	M	M	M		RP	S	S	S	RP	RP	RU
...																						
Cycle n	RP	RP	M	M	M	M	RP	RP	RP	N	N	N	RP	RP		M	RP	RU	N	N	N	RP

Intéressons-nous à la façon dont ce tableau est parcouru : en fait, l'agent 1 effectue le cycle 1, puis le cycle 2, et continue jusqu'au cycle n. Lorsqu'il a fini de parcourir l'ensemble des n cycles, il recommence. L'agent 2 effectue le cycle 2, puis le cycle 3, continue jusqu'au cycle n, puis enchaîne

sur le cycle 1. De même, l'agent  $n$  effectue le cycle  $n$ , puis le cycle 1, et continue jusqu'au cycle  $n-1$ . Ainsi, pendant que l'agent 1 parcourt le cycle 1, l'agent 2 parcourt le cycle 2. De façon plus générale, étant donné un cycle  $c$ ,  $1 \leq c \leq n-1$ , pendant que l'agent 1 effectue le cycle  $c$ , l'agent 2 effectue le cycle  $c+1$ . Ainsi, avec 4 agents, les enchaînements effectués sont donnés par (tableau **T3**) :

Agent 1	Cycle 1	Cycle 2	Cycle 3	Cycle 4
Agent 2	Cycle 2	Cycle 3	Cycle 4	Cycle 1
Agent 3	Cycle 3	Cycle 4	Cycle 1	Cycle 2
Agent 4	Cycle 4	Cycle 1	Cycle 2	Cycle 3

Les agents effectuent donc tous les mêmes services, mais avec un décalage par rapport aux autres agents. On peut donc exprimer la contrainte de charge en colonne (c'est-à-dire en faisant intervenir les cycles). On peut l'écrire, pour chaque plage horaire : **nombre de fois où l'agent 1 est présent pendant cette plage horaire sur l'ensemble des cycles + nombre d'agents de réserve affectés au roulement pendant cette plage horaire = besoin en agents pendant cette plage horaire**. On ne va pour l'instant s'intéresser qu'aux roulements de type « 3\*8 », c'est-à-dire pour lesquels la grille des besoins stipule que, tous les jours, un agent doit être présent le Matin, le Soir et la Nuit.

### 1.3.2 Description détaillée de la conception d'un roulement 3\*8

Les contraintes et critères listés ci-dessous sont récapitulés dans le tableau de l'annexe 5.2, p. 53.

#### 1.3.2.1 Cadre de travail – Hypothèses.

Pour l'instant, l'étude va être restreinte à un cadre de travail précis. Voici la liste des hypothèses de travail qui seront utilisées dans toute la suite de l'étude :

- On ne considère que des roulements de type « 3\*8 », sauf précision contraire ;
- On suppose que le mode de calcul du nombre d'agents affectés est le mode par défaut (voir annexe 5.1, p. 52) : ainsi, il y a forcément un déficit de ressources, et l'appel à la réserve est inévitable ;
- On ne s'occupe pas des jours notés « d » (*disponibles*), qui imposent à un agent de venir travailler un jour donné sans poste prédéfini : il est alors affecté suivant les besoins constatés en temps réel ;
- Les « grandes soirées », qui couvrent typiquement l'intervalle 17h-2h, ne sont pas traitées ;
- On ne prend pas en compte les GPT de 6 jours : autrement dit, on ne considère que les GPT qui contiennent entre 3 et 5 jours.

#### 1.3.2.2 Contraintes imposées par la législation ou l'expérience

La conception des roulements doit se faire dans le respect de certaines contraintes, qui sont énumérées ci-dessous.

- Contraintes à respecter à l'intérieur d'une grande période de travail (ou GPT) :
  - (C1a) Une GPT contient entre 3 et 5 jours ;
  - (C1b) Une GPT n'est associée qu'à un seul type de plage horaire : Matinée, Soirée, ou Nuit. Ainsi, une GPT sera constituée uniquement de matins ou uniquement de soirées ou uniquement de nuits, selon le type de sa plage horaire de début.

- Contraintes liées à l'enchaînement des GPT :
  - (C2a) 2 GPT consécutives doivent être séparées par une période de repos d'au moins un jour, et même de 2 jours lorsque la première GPT dure 6 jours ;
  - (C2b) Les GPT s'enchaînent suivant le motif « Soirée – Matinée – Nuit » (contrainte spécifique aux roulements de type « 3\*8 »).
  
- Contraintes liées aux jours de repos des agents :
  - (C3a) Une période de repos ne peut pas être composée de plus de 3 jours : autrement dit, aucun agent ne peut avoir plus de 3 jours de repos consécutifs ;
  - (C3b) Un agent doit recevoir 132 jours de repos dans l'année ;
  - (C3c) Un agent doit recevoir 118 jours de repos périodique (RP) dans l'année ;
  - (C3d) Chaque agent a droit à au moins 52 repos doubles dans l'année (c'est-à-dire une période de repos d'au moins 2 jours comptés comme des RP) ;
  - (C3e) Chaque agent a droit à au moins 22 week-ends dans l'année, c'est-à-dire une période de repos d'au moins 2 jours contenant un dimanche, compté comme un RP, et un jour accolé (samedi ou lundi), compté comme un RP ou un RU ;
  - (C3f) Chaque agent a droit à au moins 12 samedi/dimanche dans l'année (les 2 jours étant comptés comme des RP et intégrés dans les 52 repos doubles obligatoires).
  
- Contrainte de charge :
  - (C4) A chaque plage horaire de chaque journée, le personnel doit être en nombre suffisant pour satisfaire la grille des besoins, qui s'exprime comme le nombre d'agents devant travailler pendant cette plage horaire. Par exemple, si la grille des besoins stipule que 3 agents devront être présents chaque jeudi soir, alors le roulement doit être fait de telle sorte que cette contrainte soit satisfaite. Chaque poste qui n'est pas assuré par un agent de roulement est systématiquement pris en charge par les agents dits « de réserve ».

### 1.3.2.3 Objectifs visés lors de la conception des roulements

La conception des roulements conduit aussi à prendre en compte, dans la mesure du possible, un certain nombre de critères. Ils ne constituent pas des contraintes absolues, mais plutôt des préférences.

- Préférences liées à l'utilisation de la réserve :
  - (P1a) Réduire le plus possible l'appel à la réserve ;
  - (P1b) Eviter de faire trop souvent appel à la réserve pour les postes de nuit ;
  - (P1c) Equilibrer le plus possible l'appel à la réserve entre les jours de la semaine ;
  - (P1d) Equilibrer le plus possible l'appel à la réserve entre les soirées et les matinées.
  
- Préférences liées à l'attribution des repos :
  - (P2a) Arriver à une répartition la plus équilibrée possible des Samedi/Dimanche sur l'année ;
  - (P2b) Arriver à une répartition la plus équilibrée possible des Week-ends sur l'année ;
  - (P2c) Réduire le plus possible le nombre de jours de repos isolés (c'est-à-dire non inclus dans une période de repos au moins double).

## 1.4 Intérêt du travail

### 1.4.1 Etat actuel des connaissances

Le problème de la planification des emplois du temps de salariés est un problème qui se pose dans de nombreux domaines, et qui a été étudié de façon spécifique : il n'existe pas de modèle générique pour un tel problème. Chaque domaine a ses contraintes, et la taille du problème elle-même est liée au nombre de contraintes.

Ainsi, l'affectation de personnel a fait l'objet d'études :

- dans le domaine hospitalier (des chercheurs danois se sont intéressés à la gestion des emplois du temps d'infirmières [Kragelund Mayoh 99]) ;
- dans le domaine du personnel volant (des chercheurs américains [Vance Barnhart Johnson Nemhauser 95] et autrichiens [Dawid König Strauss 00] ont étudié ce problème) ;
- au sein de la RATP (élaboration des tableaux de roulement des conducteurs de la RATP [Bertocchi Chouvaeff Lemoine 91]) ;
- pour le cas des conducteurs de bus [Fores 96] ;
- dans le domaine ferroviaire (des chercheurs ont étudié ce problème pour les cas du personnel roulant (c'est-à-dire non sédentaire) italien [Caprara Fischetti Toth Vigo 97] et hollandais [Kroon Fischetti 00]).

Les spécificités du personnel sédentaire, par contre (réserves, etc.), ne semblent pas avoir fait l'objet de recherches approfondies.

Les méthodes de résolution ont dû être adaptées aux modèles proposés : pour résoudre les programmes linéaires obtenus, des méthodes de **génération de colonnes** ont été utilisées (en Allemagne [Fahle Junker Karisch Kohl Sellmann Vaaben 99], et au Royaume-Uni [Fores 96]), ainsi que des **méthodes heuristiques basées sur une relaxation lagrangienne** du problème [Sanders Takkula Wedelin 99], pour des tailles importantes. On notera aussi l'utilisation de la **méthode de recherche tabou** [Chiarandini Schaerf Tiozzo 99].

Signalons aussi l'existence de certains logiciels, développés par des sociétés de service en informatique, qui se sont intéressées aux problèmes d'optimisation et de prise de décision inhérents à la conception de roulements de personnel. La société Eurodécision a notamment réalisé *LP-Roster*, un logiciel de planification de roulements, dont les méthodes de résolution sont basées sur la **programmation par contraintes**.

### 1.4.2 Utilité du travail

Le premier apport est évidemment de proposer un modèle réaliste du problème de la conception des roulements du personnel sédentaire de la SNCF, et d'acquérir une bonne maîtrise des notions nécessaires à la formalisation de ce problème. Le modèle implanté, une fois doté d'une interface graphique agréable, pourra amener à la création d'un véritable outil de planification automatique. L'objectif est, pour la Direction des Ressources Humaines, d'automatiser la conception des roulements de personnel.

Ainsi, le travail de l'expert en serait modifié : il ne s'agirait plus de manipuler à la main une combinatoire importante, mais d'utiliser un outil qui pourrait permettre de comparer rapidement plusieurs des solutions proposées. Son expertise serait de toute façon déterminante dans le choix de ces solutions et dans la bonne paramétrisation de l'outil.

## 2 Travail réalisé

### 2.1 Représentation du problème : données et variables utilisées

#### 2.1.1 Représentation du tableau de roulement

Dans le tableau, on décide de découper chaque jour en 3 **plages horaires** : la première correspond à la matinée, la seconde à la soirée, et la dernière à la nuit. Ainsi, chaque semaine comprend  $3 \times 7 = 21$  colonnes (au lieu de 7), un jour s'étalant sur 3 colonnes (une pour chaque plage horaire). Pour construire le tableau de roulement, on indiquera alors pour chaque plage horaire de chaque journée si l'agent travaille (cas que l'on notera 1) ou non (cas que l'on notera 0). Ainsi, si on reprend le tableau de roulement **T2** et qu'on examine la 2<sup>e</sup> semaine du 1<sup>er</sup> cycle (dont on rappelle que l'enchaînement des jours de travail et de repos est donné par M-M-RP-RP-RP-N-N), on pourrait la représenter sous la nouvelle forme suivante (tableau **T4**) :

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

Le tableau de roulement ainsi obtenu contient donc  $3 \times (7 \times s \times n) = 21 \times s \times n$  éléments, au lieu de  $7 \times s \times n$ . On décide de numéroter les plages horaires, dans l'ordre chronologique, de 1 à  $21 \times s \times n$ . Chaque élément du tableau de roulement est donc associé à une et une seule plage horaire, parmi ces  $21 \times s \times n$ .

Le tableau **T5** ci-dessous illustre cette numérotation : les cases de ce tableau sont remplies avec l'indice de la plage horaire correspondante. Ainsi, le lundi matin de la première semaine du premier cycle sera désigné par la plage horaire numéro 1 (en rouge dans le tableau), alors que le mercredi soir de la s<sup>e</sup> semaine du n<sup>e</sup> cycle sera désigné par la plage horaire numéro  $8 + 21 \times (n \times s - 1)$  (en bleu).

On remarque que l'indice k de la première plage horaire d'une journée (qui correspond à la matinée de cette journée) est toujours tel que  $k \equiv 1[3]$ . Par exemple, le premier lundi matin est indicé 1, le premier mardi matin est indicé 4, le premier mercredi matin 7, etc. De même, l'indice k de la deuxième plage horaire d'une journée (c'est-à-dire l'indice d'une soirée quelconque) est toujours tel que  $k \equiv 2[3]$ , et l'indice k de n'importe quelle nuit est tel que  $k \equiv 0[3]$ .

On peut donc parcourir le tableau avec une seule dimension, en ayant conscience qu'avancer dans le tableau de 21 plages horaires équivaut à avancer d'une semaine, alors qu'avancer dans le tableau de  $21 \times s$  plages horaires équivaut à considérer l'élément situé sur la même colonne, mais à la ligne suivante (et donc, dans le cycle suivant). Par exemple, le vendredi matin de la première semaine du premier cycle (en jaune dans le tableau ci-dessus) correspond à la plage horaire numéro 13, alors que le vendredi matin de la 1<sup>ère</sup> semaine du 2<sup>e</sup> cycle (en vert dans le tableau ci-dessus) correspond à la plage horaire numéro  $13 + 21 \times s$ .

D'autre part, on sera amené par la suite à numéroter les jours en fonction des plages horaires, et il faut noter, à cet effet, que le **p<sup>e</sup> jour du roulement est composé des plages horaires  $3 \times (p-1) + 1$ ,  $3 \times (p-1) + 2$  et  $3 \times p$ .**

		Semaine 1							Semaine s																							
		Lun		Mar		Mer		Jeu		Ven		Sam		Dim		Lun		Mar		Mer		Jeu		Ven		Sam		Dim				
		M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S	M	S			
Cycle 1																																
	Cycle 2																															
		Cycle n																														

En outre, indépendamment du découpage en temps, les semaines peuvent être numérotées en fonction des cycles. Par exemple, au lieu de parler de la  $j^e$  semaine du  $i^e$  cycle, on peut parler de la  $(j+(i-1)*s)^e$  semaine.

### 2.1.2 Données du problème

Le découpage temporel étant explicité, on peut introduire les notations qui seront utilisées pour désigner les données du problème :

- $\forall k \in \{1, \dots, 21*s\}$ ,  $w_k$  = besoin en agents pendant la plage horaire  $k$ . Cette donnée est établie à partir de la grille des besoins. En outre,  $\forall k \in \{1, \dots, 21*s\}$ ,  $w_k$  est un entier. Comme on l'a vu précédemment, chaque ligne du tableau de roulement correspond en fait à un agent. La contrainte de besoin va donc être exprimée par colonne, ce qui explique que l'on n'utilise que  $21*s$  (et non  $21*s*n$ ) quantités  $w_k$ , c'est-à-dire suffisamment pour couvrir un cycle (chaque cycle possédant  $21*s$  plages horaires).

A ce sujet, il faut se souvenir que la grille des besoins ne donne les besoins que sur une semaine : ils sont ensuite identiquement répétés d'une semaine à l'autre. On a donc :  $\forall k \in \{1, \dots, 21*s-1\}$ ,  $w_{(k+21)} = w_k$ . Ainsi, si les besoins en agents sont de 3 pour la nuit du jeudi de la première semaine, ils seront également de 3 pour la nuit du jeudi de la deuxième semaine et pour la nuit du jeudi de la  $s^e$  semaine. Le tableau en annexe 5.3, p. 54, illustre cette propriété.

- $s$  et  $n$  : le nombre de semaines  $s$  et le nombre de cycle  $n$  sont considérés comme des données dans cette étude. Le nombre de semaines fait en réalité partie des paramètres à déterminer, et sur lequel on peut jouer pour proposer différentes solutions à l'expert : on doit procéder par tâtonnement et résoudre le problème d'optimisation pour différentes valeurs de  $s$ . Pour déterminer le nombre de cycles  $c$  (et donc de rouleurs), on décide simplement si l'on veut travailler avec un excédent ou un déficit de ressources (rappelons que, pour l'instant, on se place systématiquement en déficit).

### 2.1.3 Représentation des variables du problème

#### 2.1.3.1 Variables entières

- $\forall k \in \{1, \dots, 21*s\}$ ,  $rs_k$  = le nombre d'agents de réserve affectés sur le roulement à la plage horaire  $k$ . Seules  $21*s$  variables sont nécessaires, car la contrainte de charge s'établit en colonne.
- $e$  = la valeur absolue de l'écart entre le nombre de matinées de réserve et le nombre de soirées de réserve. Cette variable sert lors de l'expression du critère ( $P1d$ ). La valeur de la variable  $e$  sera donnée par les contraintes ( $C5a$ ) et ( $C5b$ ).
- $\forall p \in \{1, \dots, n\}$ ,  $sa_p$  = le nombre de samedi/dimanche du cycle  $p$ .
- $\forall p \in \{1, \dots, n\}$ ,  $we_p$  = le nombre de week-ends du cycle  $p$ .

#### 2.1.3.2 Variables continues

- $\forall p \in \{1, \dots, 7*s\}$ ,  $ej_p$  = la valeur absolue de la différence entre l'appel à la réserve au jour  $p$  et l'appel à la réserve quotidien moyen sur un cycle. Ces variables servent lors de l'expression du critère ( $P1c$ ). La valeur de chaque  $ej_p$  sera donnée par les contraintes ( $C5c$ ) et ( $C5d$ ).
- $\forall p \in \{1, \dots, n\}$ ,  $ec1_p$  = la valeur absolue de la différence entre le nombre de samedi/dimanche du cycle  $p$  et le nombre moyen de samedi/dimanche par cycle sur l'ensemble du roulement. Ces variables servent lors de l'expression du critère ( $P2a$ ). La valeur de chaque  $ec1_p$  sera fixée par les contraintes ( $C5f$ ) et ( $C5g$ ).
- $\forall p \in \{1, \dots, n\}$ ,  $ec2_p$  = la valeur absolue de la différence entre le nombre de week-ends du cycle  $p$  et le nombre moyen de week-ends par cycle sur l'ensemble du roulement. Ces variables servent à exprimer le critère ( $P2b$ ). La valeur de chaque  $ec2_p$  sera fixée par les contraintes ( $C5i$ ) et ( $C5j$ ).

#### 2.1.3.3 Variables bivalentes

Elles vont servir à représenter les GPT et les périodes de repos. Néanmoins, au lieu de considérer chaque jour indépendamment de ses voisins, on va se servir de la contrainte forte qui impose que les GPT aient entre 3 et 5 jours ( $C1a$ ), et utiliser ainsi trois sortes de variables pour différencier les GPT de 3, 4 et 5 jours. En effet, dans une modélisation où l'on utilise une ou plusieurs variable(s) binaire(s) **par jour**, les contraintes portant sur les enchaînements de jours ne peuvent s'exprimer qu'à l'aide de multiplications de variables, ce qui rend ces contraintes non linéaires. Dans la mesure du possible, on essaye donc de s'orienter vers un programme linéaire, sans avoir à linéariser nos contraintes, ce qui serait un processus coûteux.

On fait de même pour différencier les périodes de repos de 1, 2 ou 3 jours (car la contrainte ( $C3a$ ) nous oblige à ne considérer que des périodes de repos d'au plus 3 jours). De cette façon, un certain nombre des contraintes citées dans la première partie seront implicitement exprimées par l'intermédiaire du choix des variables, comme c'est déjà le cas pour ( $C1a$ ) et ( $C3a$ ). Dans tous les cas, le choix effectué sera explicitement indiqué.

##### 2.1.3.3.1 Périodes de travail

- $\forall k \in \{1, \dots, 21*s*n\}$ ,  $t_k = 1$  si une période de 3 jours de travail commence à la plage horaire  $k$ , 0 sinon.

Ainsi, une GPT de 3 jours qui commence à la plage horaire  $k$  couvre les plages horaires  $k$ ,  $k+3$ , et  $k+6$ . En d'autres termes, une GPT de 3 jours commençant une matinée (ce qui correspond à  $k \equiv 1[3]$ ), couvrira les plages horaires  $k+3$  (qui est aussi une matinée, puisque  $k \equiv 1[3] \Rightarrow k+3 \equiv 1[3]$ ) et  $k+6$  (qui est également une matinée, car  $k \equiv 1[3] \Rightarrow k+6 \equiv 1[3]$ ). Il en est de même si la GPT commence un soir ou une nuit : elle ne se déroulera que sur un seul type de plage horaire. La contrainte (*Cib*) est donc implicitement exprimée pour les GPT de 3 jours par ce choix de variable. Le tableau **T6** ci-dessous résume ces propos. Les cases de même couleur correspondent aux plages horaires couvertes par une GPT de 3 jours, suivant le type de la plage horaire de départ. Une GPT de 3 jours couvre les plages horaires de couleur jaune si elle commence en  $k$ , celles de couleur rouge si elle commence en  $k+1$  et celles de couleur bleue si elle commence en  $k+2$ .

	Jour numéro $\frac{k-1}{3} + 1$			Jour numéro $\frac{k-1}{3} + 2$			Jour numéro $\frac{k-1}{3} + 3$		
Numéro de plage	$k$	$k+1$	$k+2$	$k+3$	$k+4$	$k+5$	$k+6$	$k+7$	$k+8$
Type de plage	M	S	N	M	S	N	M	S	N

- $\forall k \in \{1, \dots, 21*s*n\}$ ,  $q_k = 1$  si une période de 4 jours de travail commence à la plage horaire  $k$ , 0 sinon.

Ainsi, une GPT de 4 jours qui commence à la plage horaire  $k$  couvre les plages horaires  $k$ ,  $k+3$ ,  $k+6$  et  $k+9$ . En d'autres termes, une GPT de 4 jours commençant une matinée (ce qui correspond à  $k \equiv 1[3]$ ), couvrira les plages horaires de matinées d'indice  $k+3$ ,  $k+6$  et  $k+9$ . Il en est de même si la GPT commence un soir ou une nuit : elle ne se déroulera que sur un seul type de plage horaire. La contrainte (*Cib*) est donc implicitement exprimée pour les GPT de 4 jours par ce choix de variable. Le tableau en annexe 5.4, p. 54, résume ces propos.
- $\forall k \in \{1, \dots, 21*s*n\}$ ,  $c_k = 1$  si une période de 5 jours de travail commence à la plage horaire  $k$ , 0 sinon.

Ainsi, une GPT de 5 jours qui commence à une plage horaire  $k$  couvre les plages horaires  $k$ ,  $k+3$ ,  $k+6$ ,  $k+9$  et  $k+12$ . En d'autres termes, une GPT de 5 jours commençant une matinée (ce qui correspond à  $k \equiv 1[3]$ ), couvrira les plages horaires de matinée d'indice  $k+3$ ,  $k+6$ ,  $k+9$  et  $k+12$ . Il en est de même si la GPT commence un soir ou une nuit : elle ne se déroulera que sur un seul type de plage horaire. La contrainte (*Cib*) est donc implicitement exprimée pour les GPT de 5 jours par ce choix de variable. Le tableau en annexe 5.5, p. 55, résume ces propos.

### 2.1.3.3.2 Périodes de repos

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $r_k = 1$  si une période de 1 jour de repos de type RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si le  $k^{\text{ème}}$  jour du roulement est un RP), 0 sinon.

Seules  $7*s*n$  variables  $r_k$  sont nécessaires, et non  $21*s*n$ , car il n'y a qu'une variable par jour, et qu'il n'y a que  $7*s*n$  jours dans le roulement. Le tableau **T7** ci-dessous illustre les plages horaires couvertes par une variable  $r_k$  (en jaune).

	Jour numéro $k$		
Numéro de plage	$3*(k-1)+1$	$3*(k-1)+2$	$3*k$
Type de plage	M	S	N

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $rr_k = 1$  si une période de 2 jours de repos de type RP-RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k$  et  $k+1$  du roulement sont des RP), 0 sinon.

Comme pour les variables  $r_k$ , seules  $7*s*n$  variables  $rr_k$  sont nécessaires. Le tableau **T8** ci-dessous illustre les plages horaires couvertes par une variable  $rr_k$  (en jaune), et par toute variable représentant une période de repos de 2 jours composée de RP et de RU.

	Jour numéro k			Jour numéro k+1		
Numéro de plage	$3*(k-1)+1$	$3*(k-1)+2$	$3*k$	$3*k+1$	$3*k+2$	$3*(k+1)$
Type de plage	M	S	N	M	S	N

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $ru_k = 1$  si une période de 2 jours de repos de type RP-RU commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si le  $k^{\text{ème}}$  jour du roulement est un RP et le  $k+1^{\text{ème}}$  un RU), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $ur_k = 1$  si une période de 2 jours de repos de type RU-RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si le  $k^{\text{ème}}$  jour du roulement est un RU et le  $k+1^{\text{ème}}$  un RP), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $rrr_k = 1$  si une période de 3 jours de repos de type RP-RP-RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k$ ,  $k+1$  et  $k+2$  du roulement sont des RP), 0 sinon.

Comme pour les variables  $r_k$  et  $rr_k$ , seules  $7*s*n$  variables  $rrr_k$  sont nécessaires. Le tableau **T9** ci-dessous illustre les plages horaires couvertes par une variable  $rrr_k$  (en jaune), et par toute variable représentant une période de repos de 3 jours composée de RP et de RU.

	Jour numéro k			Jour numéro k+1			Jour numéro k+2		
Numéro de plage	$3*(k-1)+1$	$3*(k-1)+2$	$3*k$	$3*k+1$	$3*k+2$	$3*(k+1)$	$3*k+4$	$3*k+5$	$3*(k+2)$
Type de plage	M	S	N	M	S	N	M	S	N

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $rru_k = 1$  si une période de 3 jours de repos de type RP-RP-RU commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k$  et  $k+1$  du roulement sont des RP et le jour  $k+2$  un RU), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $ruu_k = 1$  si une période de 3 jours de repos de type RP-RU-RU commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si le  $k^{\text{ème}}$  jour du roulement est un RP et les jours  $k+1$  et  $k+2$  des RU), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $zur_k = 1$  si une période de 3 jours de repos de type RP-RU-RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k$  et  $k+2$  du roulement sont des RP et le jour  $k+1$  un RU), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $uru_k = 1$  si une période de 3 jours de repos de type RU-RP-RU commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k$  et  $k+2$  du roulement sont des RU et le jour  $k+1$  un RP), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $uur_k = 1$  si une période de 3 jours de repos de type RU-RU-RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k$  et  $k+1$  du roulement sont des RU et le jour  $k+2$  un RP), 0 sinon.

- $\forall k \in \{1, \dots, 7*s*n\}$ ,  $urr_k = 1$  si une période de 3 jours de repos de type RU-RP-RP commence à la plage horaire  $3*(k-1)+1$  (c'est-à-dire si les jours  $k+1$  et  $k+2$  du roulement sont des RP et le jour  $k$  un RU), 0 sinon.

## 2.2 Expression formelle des contraintes et objectifs du problème

Par la suite, étant donné un réel  $a$  positif, on utilisera les notations  $\lfloor a \rfloor$  et  $\lceil a \rceil$  pour désigner respectivement la partie entière de  $a$  et la partie entière supérieure de  $a$  (qui vaut  $a$  si  $a$  est un entier,  $\lfloor a \rfloor + 1$  sinon).

### 2.2.1 Expression des contraintes

#### 2.2.1.1 Contrainte d'unicité

- Cette contrainte est une contrainte implicite mais fondamentale du problème. Elle peut s'exprimer ainsi : *un agent ne peut et ne doit faire qu'une seule chose à la fois*. En d'autres termes, elle signifie que, étant donné un jour quelconque du roulement, un agent doit être en train d'effectuer : **soit** une GPT de 3 jours, **soit** une GPT de 4 jours, **soit** une GPT de 5 jours, **soit** une période de repos de 1 jour, **soit** une période de repos de 2 jours, **soit** une période de repos de 3 jours.

Pour exprimer cette contrainte, on s'intéresse à un jour quelconque parmi les  $7*s*n$  du tableau de roulement, et on utilise le fait que ce jour doit être occupé par une et une seule « tâche » (notion regroupant les GPT et les périodes de repos). Pour cela, il convient d'abord de recenser l'ensemble de toutes les « tâches » auxquelles ce jour est susceptible d'appartenir.

Soit donc un jour d'indice  $p+1$  ( $p+1 \in \{1, \dots, 7*s*n\} \Rightarrow p \in \{0, \dots, 7*s*n-1\}$ ), composé des plages horaires  $3*p+1$ ,  $3*p+2$  et  $3*(p+1)$ . Il peut faire partie de toute GPT commençant pendant une de ces 3 plages horaires : en d'autres termes, il peut appartenir aux GPT associées aux variables  $t_{(3*p+1)}$  à  $t_{(3*(p+1))}$ ,  $q_{(3*p+1)}$  à  $q_{(3*(p+1))}$  et  $c_{(3*p+1)}$  à  $c_{(3*(p+1))}$ . Mais aussi à toute GPT de 3 jours ayant débuté il y a 2 jours ou moins ou à toute GPT de 4 jours ayant débuté il y a 3 jours ou moins, etc. On peut raisonner de la même façon pour les périodes de repos. Au final, l'ensemble des variables associées aux « tâches » dont le jour  $p+1$  peut faire partie est noté  $E$ . Le tableau **T10** ci-dessous récapitule l'ensemble de ces éléments. Pour chaque type d'éléments de  $E$  (GPT de 3, 4 ou 5 jours ; repos de 1, 2 ou 3 jours), on représente deux éléments : d'abord, l'élément dont la plage de départ (symbolisée par une étoile) est la plus éloignée du jour  $p+1$  (cet élément contenant le jour  $p+1$  puisque, par définition, tout élément de  $E$  le contient) ; puis l'élément dont la plage de départ est la plus proche de la dernière plage horaire du jour  $p+1$  (ie la plage  $3*(p+1)$ ). Ainsi, pour chaque type d'élément, toutes les variables associées aux éléments compris chronologiquement entre ces deux bornes appartiennent à  $E$ .

	Jour p-3			Jour p-2			Jour p-1			Jour p			Jour p+1		
	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N
	$3*p-11$	$3*p-10$	$3*p-9$	$3*p-8$	$3*p-7$	$3*p-6$	$3*p-5$	$3*p-4$	$3*p-3$	$3*p-2$	$3*p-1$	$3*p$	$3*p+1$	$3*p+2$	$3*p+3$
GPT 3 j.							*								
														*	
GPT 4 j.				*			*			*			*		
														*	
GPT 5 j.	*														*
RP 1 j.													*	*	*
										*	*	*	*	*	*
RP 2 j.										*	*	*	*	*	*
													*	*	*
RP 3 j.							*	*	*	*	*	*	*	*	*
													*	*	*

On note :  $r2_k = \{rr_k, ru_k, ur_k\}$  et  $r3_k = \{rrr_k, rru_k, rur_k, ruu_k, urr_k, uur_k, uru_k\}$ . L'ensemble E est donc défini par :  $E = \{t_k / k \in \{3*p-5, \dots, 3*(p+1)\}\} \cup \{q_k / k \in \{3*p-8, \dots, 3*(p+1)\}\} \cup \{c_k / k \in \{3*p-11, \dots, 3*(p+1)\}\} \cup \{r_{(p+1)}\} \cup r2_p \cup r2_{(p+1)} \cup r3_{(p-1)} \cup r3_p \cup r3_{(p+1)}$ .

Il reste à exprimer le fait qu'un et un seul élément de E peut et doit valoir 1 (c'est-à-dire que la « tâche » associée est active) : pour cela, on impose que la somme des variables associées aux éléments de E soit 1. Ainsi, on aura bien à la fois exclusion mutuelle entre les GPT d'une part, entre les périodes de repos d'autre part, et, pour finir, unicité de la « tâche » accomplie par un agent à un moment donné. Enfin, il ne faut pas oublier qu'en partant d'un jour p+1 quelconque, on peut, en considérant les jours p-1, p-2, etc., être en fait amené à considérer des jours qui précèdent p+1, à n cycles près (puisque le roulement est cyclique), c'est-à-dire à un roulement complet près. Par exemple, si le jour p+1 est le jour 1, alors le jour précédent sera en fait le jour d'indice  $7*s*n$ , à cause de la cyclicité du tableau (voir illustration à l'annexe 5.6, p. 55). Il faut donc prendre en compte ce phénomène dans la contrainte. On utilise donc les indices du tableau représentant les éléments de l'ensemble E, mais avec la précaution supplémentaire d'une expression par congruence  $21*s*n$  ou  $7*s*n$  (c'est-à-dire en les considérant modulo le nombre de plages horaires d'un roulement complet). Au final, cette contrainte, que l'on notera (C0), s'exprime donc :

$$(C0) : \forall p \in \{0, \dots, 7*s*n-1\}, \sum_{k=3*p-5}^{3*(p+1)} t(k[21*s*n]) + \sum_{k=3*p-8}^{3*(p+1)} q(k[21*s*n]) + \sum_{k=3*p-11}^{3*(p+1)} c(k[21*s*n]) + r(p+1) + \sum_{k=0}^1 rr((p+1-k)[7*s*n]) + \sum_{k=0}^1 ru((p+1-k)[7*s*n]) + \sum_{k=0}^1 ur((p+1-k)[7*s*n]) + \sum_{k=0}^2 rrr((p+1-k)[7*s*n]) + \sum_{k=0}^2 rru((p+1-k)[7*s*n]) + \sum_{k=0}^2 ruu((p+1-k)[7*s*n]) + \sum_{k=0}^2 rur((p+1-k)[7*s*n]) + \sum_{k=0}^2 uru((p+1-k)[7*s*n]) + \sum_{k=0}^2 uur((p+1-k)[7*s*n]) + \sum_{k=0}^2 urr((p+1-k)[7*s*n]) = 1$$

### 2.2.1.2 Contraintes intra-GPT

- (C1a) Une GPT contient entre 3 et 5 jours : comme nous l'avons vu dans la section précédente, cette contrainte est exprimée par le choix des variables  $t_k$ ,  $q_k$  et  $c_k$  qui représentent les périodes de travail, d'une part, et d'autre part par la contrainte (C0).
- (C1b) Une GPT n'est associée qu'à un seul type de plage horaire : comme nous l'avons vu dans la section précédente, cette contrainte est elle aussi exprimée par le choix des variables  $t_k$ ,  $q_k$  et  $c_k$  qui représentent les périodes de travail, d'une part, et d'autre part par la contrainte (C0).

### 2.2.1.3 Contraintes inter-GPT

- (C2a) 2 GPT doivent être séparées par une période de repos d'au moins un jour : en fait, cette contrainte peut se réécrire « 2 GPT sont séparées par une seule période de repos, qui contient soit 1, soit 2, soit 3 jours de repos ».

Pour exprimer cette contrainte, on va s'intéresser à un jour quelconque parmi les  $7*s*n$  du tableau de roulement, et on va simplement se servir du fait qu'il est un jour de début de période de repos si, et seulement si, il est précédé par le dernier jour d'une GPT de 3, 4 ou 5 jours. Or, si l'on note p+1 l'indice de ce jour quelconque ( $p+1 \in \{1, \dots, 7*s*n\} \Rightarrow p \in \{0, \dots, 7*s*n-1\}$ ), une GPT de 3 jours se terminant le jour p (c'est-à-dire la veille de p+1) commence forcément le jour p-2, c'est-à-dire soit pendant la plage horaire  $3*p-8$ , soit pendant la plage horaire  $3*p-7$ , soit pendant la plage horaire

3\*p-6. De même, une GPT de 4 jours se terminant le jour p commence forcément le jour p-3, et une GPT de 5 jours se terminant le jour p commence forcément le jour p-4. Le tableau **T11** ci-dessous résume l'ensemble des plages horaires susceptibles de convenir : les plages de la même couleur et placées sur la même ligne sont les plages appartenant à une même GPT et les plages marquées d'un « \* » sont les plages horaires de début de GPT.

	Jour p-4			Jour p-3			Jour p-2			Jour p-1			Jour p			Jour p+1		
	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N
	3*p-1 4	3*p-1 3	3*p-1 2	3*p-1 1	3*p-1 0	3*p-9	3*p-8	3*p-7	3*p-6	3*p-5	3*p-4	3*p-3	3*p-2	3*p-1	3*p	3*p+1	3*p+2	3*p+3
GPT 3 j.							*	*	*									
GPT 4 j.				*	*	*												
GPT 5 j.	*	*	*															

Si l'une des variables associées à ces GPT vaut 1, alors il y a forcément une période de repos qui commence au jour p+1. C'est soit une période de repos de 1 jour, soit une période de repos de 2 jours, soit une période de repos de 3 jours. En outre, il est important de se rappeler qu'il ne peut y avoir aucun autre cas dans lequel une période de repos commencerait le jour p+1. En effet, tels que les variables sont définies dans le modèle, on n'a pas le droit de considérer des enchaînements de la forme (RP-RU)-(RP), c'est-à-dire une période de repos de 2 jours de type RP-RU suivie d'un RP. Un tel enchaînement de jours ne peut être vu que sous la forme (RP-RU-RP), c'est-à-dire une période de repos de 3 jours de type RP-RU-RP. En d'autres termes, les variables représentant les périodes de repos de 3 (resp. 2) jours ont été définies de telle sorte qu'elles couvrent toutes les combinaisons possibles de périodes de repos de longueur inférieure, soit 1 jour ou 2 jours (resp. 1 jour), qui sont autorisées. C'est donc une condition nécessaire et suffisante : une GPT se termine le jour p  $\Leftrightarrow$  une période de repos commence le jour p+1. On va tout naturellement traduire cette équivalence par une égalité. Le membre de gauche est vrai si et seulement si l'une parmi les 6 GPT données dans le tableau **T11** est active, et le membre de gauche est vrai si et seulement si une période de repos d'un des 3 types existants est active à partir du jour p+1. Il y a donc égalité de ces 2 conditions. Remarquons que chacun des 2 membres vaut soit 0, soit 1, mais en aucun cas il ne peut valoir plus de 1 : la contrainte d'unicité (C0) impose qu'une et une seule « tâche » soit active à un moment donné. Enfin, il faut remarquer que, pour les mêmes raisons que celles évoquées dans l'expression de la contrainte (C0), les indices i tels que  $i < 3*p+1$  doivent être considérés modulo  $21*s*n$ , c'est-à-dire modulo un roulement complet. La contrainte (C2a) peut donc s'écrire :

$$(C2a) : \forall p \in \{0, \dots, 7*s*n-1\},$$

$$\sum_{k=1}^3 t_{(3*p-9+k)[21*s*n]} + q_{(3*p-12+k)[21*s*n]} + c_{(3*p-15+k)[21*s*n]} = (r_{(p+1)} + rr_{(p+1)} + ru_{(p+1)} + ur_{(p+1)} + rru_{(p+1)} + ruu_{(p+1)} + rur_{(p+1)} + rrr_{(p+1)} + uru_{(p+1)} + uur_{(p+1)} + urr_{(p+1)})$$

- (C2b) Les GPT s'enchaînent suivant le motif « Soirée – Matinée – Nuit ».

Pour exprimer cette contrainte, on la divise en trois sous-contraintes : (C2b.1), (C2b.2) et (C2b.3), qui signifient respectivement qu'une GPT de soirées est toujours suivie d'une GPT de matinées,

qu'une GPT de matinées est toujours suivie d'une GPT de nuits, et enfin qu'une GPT de nuits est toujours suivie d'une GPT de soirées. Cependant, compte tenu des ressemblances entre ces trois contraintes, le raisonnement n'est présenté qu'une seule fois.

L'équivalence que l'on a à traduire est : la  $m^{\text{e}}$  GPT est une GPT de soirées (resp. de matinées et de nuits)  $\Leftrightarrow$  la  $(m+1)^{\text{e}}$  GPT est une GPT de matinées (resp. de nuits et de soirées). Le problème, ici, est que les GPT ne sont pas numérotées. Pour résoudre ce problème, il suffit de s'inspirer à nouveau de la contrainte précédente. On va s'intéresser à l'ensemble des GPT de 3, 4 et 5 jours qui sont susceptibles de se terminer un jour donné. La GPT active parmi cet ensemble constitue celle que l'on a désignée ci-dessus comme étant la  $m^{\text{e}}$  GPT. La contrainte (C2b.1) traite le cas où cette GPT est une GPT de soirées, la contrainte (C2b.2) traite le cas où il s'agit d'une GPT de matinées, et enfin la contrainte (C2b.3) traite le cas d'une GPT de nuits. Ensuite, il convient de s'intéresser à la GPT placée juste après cette «  $m^{\text{e}}$  GPT », désignée précédemment par  $(m+1)^{\text{e}}$  GPT. Cependant, on ne connaît pas la plage horaire exacte à laquelle doit commencer cette  $(m+1)^{\text{e}}$  GPT. Par contre, **on peut déterminer un ensemble de jours qui contient à coup sûr la première plage horaire de la  $(m+1)^{\text{e}}$  GPT**. En effet, lorsque la  $m^{\text{e}}$  GPT se termine, elle est immédiatement suivie d'une période de repos de 1, 2 ou 3 jours, elle-même immédiatement suivie d'une nouvelle GPT : la  $(m+1)^{\text{e}}$  GPT. Donc, si la  $m^{\text{e}}$  GPT se termine le jour  $p+5$ , la  $(m+1)^{\text{e}}$  GPT commencera forcément entre le  $(p+7)^{\text{e}}$  jour et le  $(p+9)^{\text{e}}$  jour, comme le montre le tableau T12 ci-dessous (ce tableau montre seulement les jours et est donc valide pour les cas (C2b.1), (C2b.2) et (C2b.3)). La  $m^{\text{e}}$  GPT est représentée en jaune, rouge ou bleu selon qu'elle dure respectivement 3, 4 ou 5 jours. Le jour de début de la  $(m+1)^{\text{e}}$  GPT est indiquée dans tous les cas en vert, et les périodes de repos en gris.

	Jour p+1	Jour p+2	Jour p+3	Jour p+4	Jour p+5	Jour p+6	Jour p+7	Jour p+8	Jour p+9
GPT 3 j.						RP 1 j.			
						RP 2 j.			
						RP 3 j.			
GPT 4 j.						RP 1 j.			
						RP 2 j.			
						RP 3 j.			
GPT 5 j.						RP 1 j.			
						RP 2 j.			
						RP 3 j.			

En résumé, si l'on traduit ce tableau en plages horaires et non plus en jours, une GPT finissant le jour  $p+5$  doit :

- commencer le jour  $p+3$  soit à la plage horaire  $3*p+7$  (matinée), soit à la plage horaire  $3*p+8$  (soirée), soit à la plage horaire  $3*p+9$  (nuit) si c'est une GPT de 3 jours ;
- commencer le jour  $p+2$  soit à la plage horaire  $3*p+4$  (matinée), soit à la plage horaire  $3*p+5$  (soirée), soit à la plage horaire  $3*p+6$  (nuit) si c'est une GPT de 4 jours ;
- commencer le jour  $p+1$  soit à la plage horaire  $3*p+1$  (matinée), soit à la plage horaire  $3*p+2$  (soirée), soit à la plage horaire  $3*p+3$  (nuit) si c'est une GPT de 5 jours.

Encore une fois, c'est la contrainte (C0) qui nous assure qu'il ne peut y avoir qu'une seule GPT qui se termine le jour  $p+1$ . En outre, il est important de noter que, si l'on est sûr de ne trouver qu'une seule GPT débutant entre les jours  $p+7$  et  $p+9$ , c'est grâce, d'abord, à la contrainte (C0), puis à une propriété essentielle qui doit être vérifiée par les longueurs des GPT et des périodes de repos. Si l'on note  $\text{minGPT}$  la longueur minimale d'une GPT,  $\text{minRP}$  la longueur minimale d'une période de repos et  $\text{maxRP}$  la longueur maximale d'une période de repos, cette propriété **Prop1** s'écrit :  $2*\text{minRP} + \text{minGPT} > \text{maxRP}$ . Le non-respect de cette propriété entraînerait l'invalidité des expressions données

ci-dessous pour les contraintes (C2b.1), (C2b.2) et (C2b.3), car on ne pourrait alors plus garantir qu'une seule GPT est rencontrée lors de la recherche (voir annexe 5.7, p. 56).

A présent, on est en mesure d'énoncer les contraintes (C2b.1), (C2b.2) et (C2b.3) de la façon suivante : une GPT de soirées (resp. de matinées et de nuits) se termine au jour  $p+5 \Leftrightarrow$  une GPT de matinées (resp. de nuits et de soirées) commence entre le jour d'indice  $p+7$  et le jour d'indice  $p+9$ , c'est-à-dire entre la plage horaire d'indice  $3*p+19$  et la plage horaire d'indice  $3*p+27$ .

Enfin, il faut de nouveau remarquer que les indices  $i$  tels que  $i > 3*(p+1)$  doivent être considérés modulo  $21*s*n$ , c'est-à-dire modulo un roulement complet.

Les contraintes (C2b.1), (C2b.2) et (C2b.3) peuvent donc s'écrire :

- (C2b.1 : une GPT de soirées est toujours suivie d'une GPT de matinées) :

$$\begin{aligned} & \forall p \in \{0, \dots, 7*s*n-1\}, t((3*p+8) \lfloor 21*s*n \rfloor) + q((3*p+5) \lfloor 21*s*n \rfloor) + c(3*p+2) \\ & \leq \sum_{l=0}^2 \left( t((3*p+19+l*3) \lfloor 21*s*n \rfloor) + q((3*p+19+l*3) \lfloor 21*s*n \rfloor) + c((3*p+19+l*3) \lfloor 21*s*n \rfloor) \right) \end{aligned}$$

- (C2b.2 : une GPT de matinées est toujours suivie d'une GPT de nuits) :

$$\begin{aligned} & \forall p \in \{0, \dots, 7*s*n-1\}, t((3*p+7) \lfloor 21*s*n \rfloor) + q((3*p+4) \lfloor 21*s*n \rfloor) + c(3*p+1) \\ & \leq \sum_{l=0}^2 \left( t((3*p+21+l*3) \lfloor 21*s*n \rfloor) + q((3*p+21+l*3) \lfloor 21*s*n \rfloor) + c((3*p+21+l*3) \lfloor 21*s*n \rfloor) \right) \end{aligned}$$

- (C2b.3 : une GPT de nuits est toujours suivie d'une GPT de soirées) :

$$\begin{aligned} & \forall p \in \{0, \dots, 7*s*n-1\}, t((3*p+9) \lfloor 21*s*n \rfloor) + q((3*p+6) \lfloor 21*s*n \rfloor) + c(3*p+3) \\ & \leq \sum_{l=0}^2 \left( t((3*p+20+l*3) \lfloor 21*s*n \rfloor) + q((3*p+20+l*3) \lfloor 21*s*n \rfloor) + c((3*p+20+l*3) \lfloor 21*s*n \rfloor) \right) \end{aligned}$$

#### 2.2.1.4 Contraintes sur les repos

- (C3a) Une période de repos ne peut pas être composée de plus de 3 jours : autrement dit, aucun agent ne peut avoir plus de 3 jours de repos consécutifs : comme nous l'avons vu dans la section précédente, cette contrainte est exprimée par le choix des variables  $r_k, rr_k, ru_k, ur_k, rru_k, ruu_k, rur_k, rrr_k, uru_k, uur_k,$  et  $urr_k$  qui représentent les périodes de repos. Rappelons également que c'est la contrainte (C2a) qui nous assure qu'on ne rencontrera jamais, dans un roulement, deux périodes de repos consécutives.
- (C3b) Un agent doit recevoir 132 jours de repos dans l'année.

Comme l'on travaille sur  $s*n$  semaines et non sur 52, cela revient, par proportionnalité, à octroyer à chaque agent un nombre de jours de repos égal à  $\left\lfloor \frac{132*s*n}{52} \right\rfloor$  sur l'ensemble du roulement (les experts en roulement préférant attribuer moins de repos que trop).

Or, pour compter le nombre de repos que l'on a accordé, il suffit de compter pour deux toute variable représentant une période de repos de 2 jours et pour trois toute variable représentant une période de repos de 3 jours. Chaque période de repos de 1 jour ne sera évidemment comptée que pour un.

La contrainte (C3b) peut donc s'écrire :

$$(C3b) : 2 * \sum_{k=1}^{7*s*n} (rr_k + ru_k + ur_k) + 3 * \sum_{k=1}^{7*s*n} (rru_k + ruu_k + rur_k + rrr_k + uru_k + uur_k + urr_k) + \sum_{k=1}^{7*s*n} r_k = \left\lfloor \frac{132 * s * n}{52} \right\rfloor$$

- (C3c) Un agent doit recevoir 118 RP dans l'année.

Comme l'on travaille sur  $s*n$  semaines et non sur 52, cela revient à octroyer à chaque agent un nombre de RP égal à  $\left\lfloor 118 * \frac{s * n}{52} \right\rfloor$  (les experts en roulement préférant attribuer plus de RP que pas assez). Or, pour compter le nombre de RP que l'on a accordé, il suffit de compter pour deux toute variable représentant une période de repos contenant 2 RP et pour trois toute variable représentant une période de repos contenant 3 RP. La contrainte (C3c) peut donc s'écrire :

$$(C3c) : \sum_{k=1}^{7*s*n} (r_k + ru_k + ur_k + ruu_k + uru_k + uur_k) + 2 * \sum_{k=1}^{7*s*n} (rr_k + rur_k + urr_k + rru_k) + 3 * \sum_{k=1}^{7*s*n} (rrr_k) = \left\lfloor 118 * \frac{s * n}{52} \right\rfloor$$

- (C3d) Chaque agent a droit à au moins 52 repos doubles dans l'année (c'est-à-dire une période de repos d'au moins 2 jours comptés comme des RP).

Chaque période de repos de 3 jours contenant 2 RP successifs compte évidemment pour un repos double. Pour exprimer cette contrainte, il suffit donc de compter le nombre de  $rr_k$ , de  $rru_k$ , de  $rrr_k$  et de  $urr_k$  que l'on a dans les  $s*n$  semaines du tableau de roulement.

La contrainte (C3d) peut donc s'écrire :

$$(C3d) : \sum_{k=1}^{7*s*n} (rr_k + rru_k + rrr_k + urr_k) \geq s * n$$

- (C3e) Chaque agent a droit à au moins 22 week-ends dans l'année.

Pour exprimer cette contrainte, il suffit d'identifier les périodes de repos qui peuvent constituer des « week-ends ». Rappelons qu'un week-end est une période de repos d'au moins 2 jours et contenant un dimanche comptabilisé comme un RP.

Les périodes de repos qui satisfont cette définition sont donc :

- les périodes de repos de 3 jours qui commencent le vendredi, de type RP/RU-RP/RU-RP ;
- les périodes de repos de 2 ou 3 jours qui débutent le samedi, de type RP/RU-RP(-RP/RU) ;
- les périodes de repos de 2 ou 3 jours qui débutent le dimanche, de type RP-RP/RU(-RP/RU).

Le tableau **T13** ci-dessous résume l'ensemble de ces périodes et montre que chacune d'elles couvre au moins 2 des 3 jours samedi-dimanche-lundi, le dimanche étant 1 RP. Seuls les jours sont représentés ; les RP sont représentés en jaune, et les RU en rouge.

	Vendredi	Samedi	Dimanche	Lundi	Mardi
RP 2 jours		RP			
		RU	RP		
			RP	RU	
			RP		
RP 3 jours	RP				
	RP	RU	RP		
	RU	RP			
	RU		RP		
		RP			
		RP		RU	
		RU	RP		
		RU	RP	RU	
			RP		
			RP		RU
			RP	RU	RP
			RP	RU	

Il ne reste plus qu'à exhiber les indices des jours de départ de ces périodes de repos. Il suffit pour cela de rappeler que le vendredi, le samedi et le dimanche de la semaine  $se$ ,  $1 \leq se \leq s^*n$ , sont les jours d'indices respectifs  $5 + 7*(se-1)$ ,  $6 + 7*(se-1)$  et  $7*se$ . Enfin, il faut se rapporter à  $s^*n$  semaines, dans lesquelles on doit donc avoir  $\left\lceil 22 * \frac{s^*n}{52} \right\rceil$  week-ends.

La contrainte (C3e) peut donc s'écrire :

$$(C3e) : \sum_{k=0}^{s^*n-1} \left( \sum_{q=0}^1 (rur(5+q*2+k*7)) + \sum_{q=0}^1 rru(6+q+k*7) + \sum_{q=0}^1 (urr(5+q+k*7)) + uur(5+k*7) + uru(6+k*7) + ruu((k+1)*7) \right) + \sum_{k=0}^{s^*n-1} \left( \sum_{q=0}^2 (rrr(5+q+k*7)) + \sum_{q=0}^1 (rr(6+q+k*7)) + ur(6+k*7) + ru((k+1)*7) \right) \geq \left\lceil 22 * \frac{s^*n}{52} \right\rceil$$

- (C3f) Chaque agent a droit à au moins 12 samedi/dimanche dans l'année.

Pour exprimer cette contrainte, on procède de façon analogue. Les périodes de repos qui constituent des samedi/dimanche sont :

- les périodes de repos de 3 jours qui commencent le vendredi, de type RP/RU-RP-RP ;
- les périodes de repos de 2 ou 3 jours qui commencent le samedi, de type RP-RP(-RP/RU).

Le tableau **T14** correspondant est donc le même que pour la contrainte précédente, sans les périodes de repos commençant le dimanche. Les jours de repos périodiques (RP) sont représentés en jaune, et les jours de repos supplémentaire en rouge.

	Vendredi	Samedi	Dimanche	Lundi
RP 2 jours		RP		
RP 3 jours	RU	RP		
		RP		
		RP		RU
		RP		

Les indices des plages horaires qui nous intéressent sont les mêmes que ceux calculés dans la contrainte précédente. On doit également se rapporter à  $s*n$  semaines, dans lesquelles on doit ainsi avoir  $\left\lceil 12 * \frac{s * n}{52} \right\rceil$  samedi/dimanche. La contrainte (C3f) peut donc s'écrire :

$$(C3f) : \sum_{k=0}^{s*n-1} \left( rrr(6+k*7) + urrr(5+k*7) + rru(6+k*7) + \sum_{q=0}^1 (rrr(5+q+k*7)) \right) \geq \left\lceil 12 * \frac{s * n}{52} \right\rceil$$

### 2.2.1.5 Contrainte de charge

- (C4) A chaque plage horaire de chaque journée, le personnel doit être en nombre suffisant pour satisfaire la grille des besoins.

Pour exprimer cette contrainte, la première chose à noter est qu'on fait l'hypothèse que la charge de travail est couverte au plus juste, et qu'il n'y a pas d'excès de ressources. Ensuite, il convient de déterminer quelles sont les différentes combinaisons de GPT qui permettent de couvrir la charge. En effet, pour qu'une plage horaire  $k$  d'un type quelconque soit une plage horaire pendant laquelle l'agent travaille, il faut :

- qu'une GPT de 3 jours commence à la plage horaire  $k$ , à la plage horaire  $k-3$  (c'est-à-dire la veille, sur le même type de plage horaire) ou à la plage horaire  $k-6$  (c'est-à-dire l'avant veille, sur le même type de plage horaire).
- ou bien qu'une GPT de 4 jours commence à la plage horaire  $k$ , à la plage horaire  $k-3$ , à la plage horaire  $k-6$  ou à la plage horaire  $k-9$ .
- ou encore qu'une GPT de 5 jours commence à la plage horaire  $k$ , à la plage horaire  $k-3$ , à la plage horaire  $k-6$ , à la plage horaire  $k-9$  ou à la plage horaire  $k-12$ .

Le tableau en annexe 5.8, p. 57, résume l'ensemble des GPT couvrant une plage horaire  $k$ .

Une seule de ces 12 GPT peut être active, à cause de la contrainte (C0). Enfin, il faut se rappeler que la charge se calcule en colonne, c'est-à-dire qu'on considère la  $k^e$  plage horaire de **chaque** cycle (c'est-à-dire l'ensemble des plages horaires éloignées d'un nombre entier de cycles, ie d'un multiple de  $21*s$  plages horaires, de la  $k^e$  plage horaire du premier cycle). On considère donc l'ensemble des GPT du tableau de l'annexe 5.8 relativement à un cycle, puis on somme sur l'ensemble des cycles du roulement, et on complète la charge non couverte dans le roulement par des agents de réserve. Remarquons aussi que tous les indices  $i$  différents de  $k$  doivent être considérés modulo un roulement complet (ie modulo  $21*s*n$  plages horaires).

La contrainte (C4) peut donc s'écrire :

$$(C4) : \forall k \in \{1, \dots, 21*s\},$$

$$\sum_{q=0}^{n-1} \sum_{p=0}^2 t((k-3*p+21*q*s)[21*s*n]) + \sum_{q=0}^{n-1} \sum_{p=0}^3 q((k-3*p+21*q*s)[21*s*n]) + \sum_{q=0}^{n-1} \sum_{p=0}^4 c((k-3*p+21*q*s)[21*s*n]) + rs_k = w_k$$

### 2.2.1.6 Contraintes liées aux variables additionnelles

- la variable  $e$  représente la valeur absolue de  $\sum_{k=0}^{7*s-1} (rs(3*k+1) - rs(3*k+2))$ .

$\forall a$  réel,  $x = |a| \Leftrightarrow x = \max(a, -a) \Leftrightarrow x = \min\{y / y \geq a \text{ et } y \geq -a\}$ . La définition de  $e$  peut donc s'écrire à l'aide des contraintes (C5a) et (C5b) et du critère (P1d), détaillé dans la section suivante :

$$(C5a) : e \geq \sum_{k=0}^{7*s-1} (rs(3*k+1) - rs(3*k+2)) .$$

$$(C5b) : e \geq \sum_{k=0}^{7*s-1} (rs(3*k+2) - rs(3*k+1)) .$$

- chaque variable  $ej_p$  représente la valeur absolue de  $\sum_{k=1}^3 rs(3*(p-1)+k) - \frac{\sum_{q=1}^{21*s} rs_q}{7*s}$ .

Par analogie avec (C5a) et (C5b), on peut traduire cette propriété à l'aide de deux contraintes, (C5c) et (C5d), et du critère (P1c), détaillé dans la section suivante :

$$(C5c) : \forall p \in \{0, \dots, 7*s-1\}, ej_{(p+1)} \geq \sum_{k=1}^3 rs(3*p+k) - \frac{\sum_{q=1}^{21*s} rs_q}{7*s} .$$

$$(C5d) : \forall p \in \{0, \dots, 7*s-1\}, ej_{(p+1)} \geq \frac{\sum_{q=1}^{21*s} rs_q}{7*s} - \sum_{k=1}^3 rs(3*p+k) .$$

- (C5e) : chaque variable  $sa_p$  représente le nombre de samedi/dimanche du cycle  $p$ .

La contrainte (C5e) peut s'écrire :

$$(C5e) : \forall p \in \{0, \dots, n-1\},$$

$$sa_{(p+1)} = \sum_{k=0}^{s-1} \left( rrr(6+k*7+p*7*s) + urr(5+k*7+p*7*s) + rru(6+k*7+p*7*s) + \sum_{l=0}^1 (rrr(5+l+k*7+p*7*s)) \right) .$$

- chaque variable  $ec1_p$  représente la valeur absolue de  $sa_p - \frac{\sum_{p=1}^n sa_p}{n}$ .

Par analogie avec (C5a) et (C5b), on peut traduire cette propriété à l'aide de deux contraintes, (C5f) et (C5g), et du critère (P2a), détaillé dans la section suivante :

$$(C5f) : \forall p \in \{1, \dots, n\}, ec1_p \geq sa_p - \frac{\sum_{p=1}^n sa_p}{n} .$$

$$(C5g) : \forall p \in \{1, \dots, n\}, ec1_p \geq \frac{\sum_{p=1}^n sa_p}{n} - sa_p.$$

- **(C5h)** : chaque variable  $we_p$  représente le nombre de week-end du cycle  $p$ .

Par analogie avec (C5e) et d'après le résultat vu en (C3e), la contrainte (C5h) peut s'écrire :

$$(C5h) : \forall p \in \{0, \dots, n-1\},$$

$$we_{(p+1)} = \sum_{k=0}^{s-1} \left( \sum_{q=0}^2 (rrr(5+q+(k+p*s)*7)) + \sum_{q=0}^1 (rr(6+q+(k+p*s)*7)) + ur(6+(k+p*s)*7) + ru((k+1+p*s)*7) \right) +$$

$$\sum_{k=0}^{s-1} \left( \sum_{q=0}^1 (rur(5+q*2+(k+p*s)*7)) + uur(5+(k+p*s)*7) + uru(6+(k+p*s)*7) + ruu((k+1+p*s)*7) \right) +$$

$$\sum_{k=0}^{s-1} \left( \sum_{q=0}^1 rru(6+q+(k+p*s)*7) + \sum_{q=0}^1 urr(5+q+(k+p*s)*7) \right)$$

- chaque variable  $ec2_p$  représente la valeur absolue de  $we_p - \frac{\sum_{p=1}^n we_p}{n}$ .

Par analogie avec (C5a) et (C5b), on peut traduire cette propriété à l'aide de deux contraintes, (C5i) et (C5j), et du critère (P2b), détaillé dans la section suivante :

$$(C5i) : \forall p \in \{1, \dots, n\}, ec2_p \geq we_p - \frac{\sum_{p=1}^n we_p}{n}.$$

$$(C5j) : \forall p \in \{1, \dots, n\}, ec2_p \geq \frac{\sum_{p=1}^n we_p}{n} - we_p.$$

## 2.2.2 Expression des préférences et des objectifs du problème

Chaque critère doit être associé à un paramètre ( $\alpha, \beta, \gamma, \dots$ ), qui représente l'importance relative de ce critère par rapport aux autres, et qui doit être défini par l'expert en fonction de ses propres préférences (par exemple, on peut lui demander de classer les critères par ordre d'importance).

### 2.2.2.1 Préférences liées à l'utilisation de la réserve

- **(P1a)** Réduire le plus possible l'appel à la réserve.

$$(P1a) : \min \alpha^* \sum_{k=1}^{21*s} rS_k$$

- **(P1b)** Eviter de faire trop souvent appel à la réserve pour les postes de nuit.

Les postes de nuit correspondent aux plages horaires  $k$  telles que  $k \equiv 0[3]$ . Le critère (P1b) peut donc s'écrire :

$$(P1b) : \min \beta^* \sum_{k=1}^{7*s} rS(3*k)$$

- **(P1c)** Equilibrer le plus possible l'appel à la réserve entre les jours de la semaine.

D'après la définition de la variable  $e_{j_p}$ , le critère (P1c) s'écrit :

$$(P1c) : \min \gamma^* \sum_{p=1}^{7*s} e_{j_p}$$

- **(P1d)** Equilibrer le plus possible l'appel à la réserve entre les soirées et les matinées.

D'après la définition de la variable  $e$ , le critère (P1d) s'écrit :

$$(P1d) : \min \delta^* e$$

### 2.2.2.2 Préférences liées à l'attribution des repos

- **(P2a)** Arriver à une répartition la plus équilibrée possible des Samedi/Dimanche sur l'année (on équilibre par cycle).

D'après la définition de la variable  $ec1_p$ , le critère (P2a) s'écrit :

$$(P2a) : \min \varepsilon^* \sum_{p=1}^n ec1_p$$

- **(P2b)** Arriver à une répartition la plus équilibrée possible des Week-end sur l'année (on équilibre par cycle).

D'après la définition de la variable  $ec2_p$ , le critère (P2b) s'écrit :

$$(P2b) : \min \zeta^* \sum_{p=1}^n ec2_p$$

- **(P2c)** Réduire le plus possible le nombre de jours de repos isolés (c'est-à-dire non inclus dans une période de repos au moins double).

Les seules périodes de repos qui vérifient cette propriété sont celles qui sont liées à une variable de type  $r_k$ . Il suffit donc de minimiser la somme des variables de ce type. Le critère (P2c) peut donc s'écrire :

$$(P2c) : \min \eta * \sum_{k=1}^{7*s*n} r_k$$

### 2.2.3 Remarques

#### 2.2.3.1 Réduction du nombre de critères

On peut éventuellement se ramener à une expression par contraintes pour les critères (P1c) et (P1d) :

- **(P1c)** : l'utilisation de la réserve sur une journée entière ne peut pas dépasser la moyenne de plus de deux :  $\forall p \in \{0, \dots, 7*s-1\}$ ,  $\sum_{k=1}^3 rs(k+3*p) * 21*s \leq \sum_{k=1}^{21*s} rsk + 42*s-2$  et  $\sum_{k=1}^{21*s} rsk \leq \sum_{k=1}^3 rs(k+3*p) * 21*s + 42*s-2$ .
- **(P1d)** : l'utilisation de la réserve en matinée ne peut pas dépasser de plus d'un l'utilisation de la réserve en soirée :  $\sum_{p=0}^{7*s-1} rs(1+3*p) \leq \sum_{p=0}^{7*s-1} rs(2+3*p) + 1$  et  $\sum_{p=0}^{7*s-1} rs(2+3*p) \leq \sum_{p=0}^{7*s-1} rs(1+3*p) + 1$ .

On peut aussi avoir une approche en écart absolu, et non plus relatif. Par exemple, en posant que l'utilisation de la réserve ne peut pas dépasser  $R$  ( $R$  entier) par jour :  $\forall p \in \{0, \dots, 7*s-1\}$ ,  $\sum_{k=1}^3 rs(k+3*p) \leq R$ .

#### 2.2.3.2 Résultat important

De (C0) et (C4), on tire (le calcul est détaillé en annexe 5.9, p. 57) :  $\sum_{k=1}^{21*s} w_k - \sum_{k=1}^{21*s} rsk = 7*s*n - \sum_{k=1}^{7*s*n} (r_k + 2*(rr_k + ru_k + ur_k) + 3*(rru_k + ruu_k + rur_k + rrr_k + uru_k + uur_k + urr_k))$ .

Cette égalité signifie : la différence entre la charge totale de travail et l'appel global à la réserve est égal à la différence entre le nombre de jours du roulement et le nombre total de jours de repos.

Le nombre de repos étant fixé par la contrainte (C3b), le nombre total d'appels à la réserve l'est donc également, et la préférence (P1a) est finalement inutile.

## 2.3 Volumétrie

### 2.3.1 Nombre de variables (avec exemple pour un 3\*8 monté avec 4 agents sur 4 semaines)

	21*s variables $rs_k$	$21*4 = 84$ (entières)
+	21*s*n variables $t_k$	$21*4*4 = 336$ (bivalentes)
+	21*s*n variables $q_k$	$21*4*4 = 336$ (bivalentes)
+	21*s*n variables $c_k$	$21*4*4 = 336$ (bivalentes)
+	7*s*n variables $r_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $rr_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $ru_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $ur_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $rrr_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $rru_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $ruu_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $rur_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $uru_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $uur_k$	$7*4*4 = 112$ (bivalentes)
+	7*s*n variables $urr_k$	$7*4*4 = 112$ (bivalentes)
+	1 variable e	1 (entière)
+	n variables $sa_p$	4 (entières)
+	n variables $we_p$	4 (entières)
	7*s variables $ej_p$	$7*4 = 28$ (continues)
+	$n+n = 2*n$ variables $ec1_p + ec2_p$	8 (continues)
	<b><math>140*s*n + 4*n + 28*s + 1</math></b>	<b>2369</b>

Le problème comporte donc  $(140*n+28)*s + 4*n + 1$  variables, soit :

- **$140*s*n$  variables bivalentes (2240 dans notre exemple) ;**
- **$21*s + 2*n + 1$  variables entières (93 dans notre exemple) ;**
- **$2*n + 7*s$  variables continues (36 dans notre exemple).**

2.3.2 Nombre de contraintes (avec exemple pour un 3\*8, avec 4 agents sur 4 semaines)

	7*s*n contraintes d'unicité (C0)	7*4*4 = 112
+	7*s*n contraintes (C2a)	7*4*4 = 112
+	7*s*n contraintes (C2b.1)	7*4*4 = 112
+	7*s*n contraintes (C2b.2)	7*4*4 = 112
+	7*s*n contraintes (C2b.3)	7*4*4 = 112
+	1 contrainte (C3b)	1
+	1 contrainte (C3c)	1
+	1 contrainte (C3d)	1
+	1 contrainte (C3e)	1
+	1 contrainte (C3f)	1
+	21*s contraintes de charge (C4)	21*4 = 84
+	1 contrainte (C5a)	1
+	1 contrainte (C5b)	1
+	7*s contraintes (C5c)	7*4 = 28
+	7*s contraintes (C5d)	7*4 = 28
+	n contraintes (C5e)	4
+	n contraintes (C5f)	4
+	n contraintes (C5g)	4
+	n contraintes (C5h)	4
+	n contraintes (C5i)	4
+	n contraintes (C5j)	4
	<b>35*s*n + 35*s + 6*n + 7</b>	<b>731</b>

Le problème comporte donc **(35\*s+6)\*(n+1)+1 contraintes** : la taille du problème, tant au niveau des variables qu'au niveau des contraintes, nous permet donc d'envisager la possibilité d'implanter et de tester le modèle. C'est ce que nous allons étudier en détails dans la section suivante.

## 2.4 Implantation du modèle et expérimentations

Cette section présente l'implantation du modèle proposé précédemment. Elle a été réalisée à l'aide du logiciel OPL Studio.

### 2.4.1 Présentation du logiciel OPL Studio de la société ILOG

OPL est un « langage de spécification » permettant de décrire les modèles avec un formalisme de haut niveau (proche du formalisme mathématique). Les modèles ainsi spécifiés peuvent ensuite être étudiés dans le cadre de la programmation linéaire (CPLEX) ou de la programmation par contraintes (SOLVER). Dans cette étude, nous utiliserons la programmation linéaire, avec recherche de solutions entières optimales par un algorithme de Branch & Bound.

Les programmes OPL se présentent sous la forme de projets, composés de deux fichiers :

- un fichier avec une extension « .mod »
- un fichier avec une extension « .dat »

#### 2.4.1.1 Le fichier « .mod »

Ce fichier contient :

- toutes les déclarations de **variables** ;
- l'expression de la **fonction objectif** en fonction des variables définies ;
- l'expression de la totalité des **contraintes**, exprimables en exploitant des quantificateurs.

L'avantage de ce langage réside dans son haut niveau d'abstraction : le modèle peut être implanté quasiment tel quel, les contraintes sont exprimées sous forme mathématique et l'on n'a pas besoin de définir la matrice des coefficients des variables.

Par exemple, la contrainte

$$\forall k \in \{1..n\}, x_k + y_k \leq 1$$

s'écrira, avec OPL :

```
forall(k in 1..n)
  x[k]+y[k] <= 1 ;
```

Il est donc extrêmement facile et rapide de modifier le modèle pour effectuer des tests, par exemple. Le listing du fichier « roulements.mod » est fourni en annexe 5.10, p.58.

#### 2.4.1.2 Le fichier « .dat »

Ce fichier contient une partie ou la totalité des **données** du problème. Un fichier « .mod » peut ainsi être associé à plusieurs fichiers « .dat », organisés selon la source des données.

Le listing du fichier « roulements.dat » est fourni en annexe 5.11, p. 62.

## 2.4.2 Validation préliminaire et enrichissement du modèle

Les premiers tests effectués sur le modèle implanté tel quel ont montré, assez rapidement :

- l'adéquation du modèle théorique : obtention de solutions « correctes » du point de vue de la législation et des contraintes informelles fixées (voir le tableau de l'annexe 5.2, p. 53) ;
- la difficulté du processus de recherche de solutions, en raison, notamment, du caractère multi-critères du problème étudié.

En effet, il se passe parfois beaucoup de temps entre le moment où une solution a été trouvée, et le moment où la preuve que cette solution est optimale est faite.

Ces premiers tests ont donc permis de valider l'expression formelle du modèle, ainsi que son implantation. La seconde étape a consisté à rechercher des « techniques » efficaces pour guider la recherche et améliorer le temps de résolution. Dans la suite, nous présenterons les trois axes de recherche qui ont été développés.

### 2.4.2.1 Hiérarchisation des critères : utilisation de coefficients non-compensatoires

#### 2.4.2.1.1 Présentation de la méthode

Comme son nom l'indique, cette méthode n'est utilisable que si une hiérarchie nette et précise peut être établie entre les différents critères (au nombre de six dans notre cas).

Supposons donc que l'on ait  $n$  ( $n \geq 2$ ) critères  $C_i(x)$ , fonctions du vecteur des variables  $x$  et numérotés de 1 à  $n$ , dans l'ordre de hiérarchisation (c'est-à-dire que le critère  $i$  est plus important que le critère  $j$ , pour tout  $i > j$ ). Cette méthode consiste à assigner à chaque critère un poids  $p_i$  tel que :

$$\forall x, \forall i \in \{1, \dots, n-1\}, p_i * C_i(x) > M * p_{i+1} * C_{i+1}(x),$$

où  $M$  est une constante à définir.

En pratique, on définit un ordre de grandeur  $g_i$  pour chaque critère, et on a donc  $p_1, \dots, p_n$  tels que :

$$\forall i \in \{1, \dots, n-1\}, p_i * g_i > M * p_{i+1} * g_{i+1}.$$

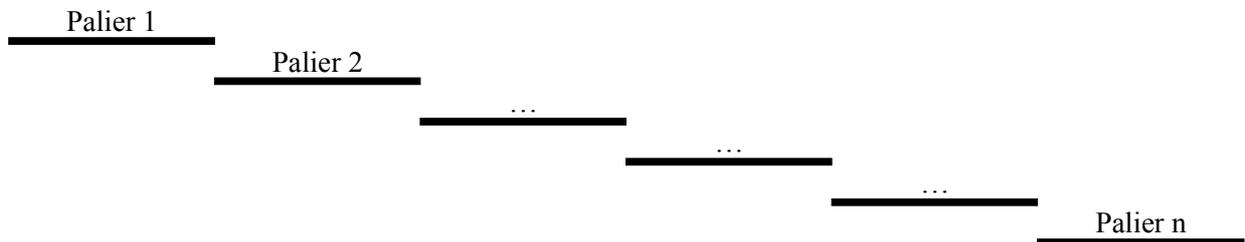
Si les intervalles de variation des critères sont trop importants, on définit, pour chaque critère, deux bornes,  $C_{\min_i}$  et  $C_{\max_i}$ , et  $p_1, \dots, p_n$  vérifient alors :

$$\forall i \in \{1, \dots, n-1\}, p_i * C_{\min_i} > p_{i+1} * C_{\max_{i+1}}.$$

Ainsi, on limite les « interactions » entre les critères : si on cherche à atteindre un minimum, une solution où le critère  $i$  prend une valeur  $v_i$  sera forcément meilleure qu'une solution où le critère  $i$  vaut  $v_{i+1}$ , et ce, **quelles que soient les valeurs des critères  $C_j$ ,  $i+1 \leq j \leq n$ .**

Prenons un exemple : supposons que l'on ait trois critères,  $C_1$ ,  $C_2$  et  $C_3$  qui varient entre 1 et 5. On peut alors (par exemple) adopter la hiérarchisation suivante :  $p_1 = 10^2$ ,  $p_2 = 10$ ,  $p_3 = 1$ . On a bien :  $1 * 10^2 > 5 * 10$  et  $1 * 10 > 5 * 1$ .

Pour la fonction objectif, qui s'écrit  $\sum_{i=1}^n p_i * C_i(x)$ , cela revient à avoir une évolution de la valeur de la fonction en marches d'escalier, ou paliers.



Chaque palier correspond à une valeur du critère de plus fort poids ( $C_1$ ) ; sur chaque palier, on a ensuite, au même titre, des petites variations dues aux critères de plus faibles poids. Par exemple, le palier 1 n'est pas une droite, mais est également composé d'une succession de paliers qui correspondent aux différentes valeurs prises par le critère  $C_2$ , etc. On voit bien que la valeur d'une solution est donc très influencée par le palier sur lequel elle se trouve (une solution sur le palier 2 sera forcément moins bonne qu'une solution sur le palier 1).

En pratique, on fait en sorte que les paliers soient suffisamment éloignés les uns des autres, pour qu'il soit nécessaire de faire « un grand saut » pour passer de l'un à l'autre. En d'autres termes, en supposant que le critère  $C_1$  ne prenne que des valeurs entières et puisque l'on perd 1 sur le critère  $C_1$  (et donc  $p_1$  sur la fonction objectif) en passant du palier 1 au palier 2, alors pour qu'un point du palier 2 soit presque au niveau de points du palier 1, il faut que le gain global au niveau des critères  $C_2$  à  $C_n$  soit de l'ordre de  $p_1$ .

La hiérarchisation permet donc :

- de guider le processus de recherche ;
- de définir une topologie sur l'espace {solutions, coûts} qui favorise la vitesse de la recherche.

#### 2.4.2.1.2 Application au problème de la conception de roulement

Pour appliquer cette méthode à notre problème, il a suffi de définir un ordre de priorité entre les critères. En effet, les ordres de grandeurs ne posent pas vraiment de problème, chacun étant entre 1 et 10, et même entre 1 et 5 pour la plupart.

L'ordre qui a été retenu est :

1. Le nombre de réserve de nuits
2. Le nombre de repos isolés
3. L'écart quotidien de réserve par rapport à la moyenne
4. La différence entre le nombre de matinées de réserve et le nombre de soirées de réserve
5. L'écart par cycle du nombre de week-end par rapport à la moyenne
6. L'écart par cycle du nombre de samedi/dimanche par rapport à la moyenne

### 2.4.2.2 Ajout de contraintes supplémentaires

La deuxième piste explorée concerne l'ajout de contraintes. Cependant, l'objectif n'est pas de restreindre l'ensemble des solutions admissibles : les contraintes rajoutées sont des contraintes **implicites** du problème. Leur explicitation doit permettre au logiciel de résolution d'atteindre **plus vite** une solution admissible. Les contraintes qui ont été ajoutées sont de quatre types.

#### 2.4.2.2.1 Contrainte de type 1, sur l'utilisation quotidienne de réserve

Le premier type de contraintes ajoutées concerne l'utilisation quotidienne de la réserve. En effet, on sait qu'elle ne peut jamais dépasser le nombre de lignes du roulement (c'est même une majoration assez large), soit  $n$ .

Elle s'écrit donc :

$$\forall p \in \{0, \dots, 7*s-1\}, \sum_{k=1}^3 rs(3*p+k) \leq n.$$

On ajoute ainsi  $7*s$  contraintes, soit 28 dans le cas d'un roulement de type « 3\*8 » construit sur 4 semaines.

#### 2.4.2.2.2 Contrainte de type 2, sur la réserve globale

La deuxième contrainte concerne l'utilisation globale de la réserve sur l'ensemble du roulement. En effet, on sait (voir le résultat important du 2.2.3.2, p. 27) que le nombre total d'appels à la réserve est fixe.

On a donc :

$$\sum_{k=1}^{21*s} rs(k) = \sum_{k=1}^{21*s} w_k - 7*s*n + \left\lfloor \frac{132 * s * n}{52} \right\rfloor.$$

On ajoute ainsi une contrainte.

#### 2.4.2.2.3 Contrainte de type 3, sur le nombre de jours de travail

La troisième contrainte concerne le nombre total de jours de travail sur l'ensemble du roulement. En effet, on connaît le nombre de jours de repos à octroyer sur un roulement, ainsi bien sûr que le nombre total de jours d'un roulement. On peut donc en déduire que le nombre de jours de travail est la différence entre les deux.

On a donc :

$$\sum_{k=1}^{21*s*n} (3*t(k) + 4*q(k) + 5*c(k)) = 7*s*n - \left\lfloor \frac{132 * s * n}{52} \right\rfloor.$$

On ajoute ainsi une contrainte.

#### 2.4.2.2.4 Contrainte de type 4, sur la différence entre soirées et matinées de réserve

La quatrième contrainte ajoutée n'a pas pour but de permettre de trouver plus vite une solution, mais de s'arrêter plus vite lorsqu'une solution a été trouvée. Pour cela, on va guider la recherche en proposant un ou plusieurs « critères d'arrêt » : c'est une caractéristique a priori de toute solution, qui dépend directement et uniquement des données. Par exemple, on est en mesure d'énoncer une propriété intéressante concernant la différence entre le nombre de soirées et de matinées de réserve, moyennant un certain nombre d'hypothèses. D'abord, on a vu que le nombre total d'appels à la réserve est donné par :

$$\sum_{k=1}^{21*s} rS(k) = \sum_{k=1}^{21*s} w_k - 7*s*n + \left\lfloor \frac{132*s*n}{52} \right\rfloor.$$

Or, si l'on considère un roulement de type « 3\*8 » à 4 agents, la quantité  $7*s*n$  est paire, puisque  $n = 4$ .

D'autre part,  $\sum_{k=1}^{21*s} w_k$  peut se réécrire  $s*\sum_{k=1}^{21} w_k$ , soit  $21*s$ , pour un roulement de type « 3\*8 ».

Donc, la parité de la quantité  $\sum_{k=1}^{21*s} w_k$  est la même que celle de  $s$ .

Enfin, considérons la quantité  $\left\lfloor \frac{132*s*n}{52} \right\rfloor$ . Pour un roulement de type « 3\*8 » à 4 agents, elle vaut  $\left\lfloor \frac{132*s*4}{52} \right\rfloor$ , soit  $\lfloor 10.15*s \rfloor$ . Pour  $s$  compris entre 1 et 6, cette quantité est donc paire.

Au final, on s'aperçoit donc que la parité du nombre total d'appels à la réserve est la même que celle du nombre de semaines  $s$ , dans le cas d'un roulement de type « 3\*8 », monté avec 4 agents sur un nombre de semaines compris entre 1 et 6. On peut considérer que tous les roulements de type « 3\*8 » étudiés vérifient ces hypothèses, et donc également cette propriété. Supposons à présent que l'on n'ait aucune réserve de nuit (cas idéal). Si on note  $nbRS_{matins}$  le nombre de matins en réserve et  $nbRS_{soirs}$  le nombre de soirs en réserve, alors  $nbRS_{matins} + nbRS_{soirs} = \sum_{k=1}^{21*s} rS(k)$ . Comme on veut équilibrer la

différence entre  $nbRS_{matins}$  et  $nbRS_{soirs}$ , le cas idéal serait  $nbRS_{matins} = nbRS_{soirs} = \frac{\sum_{k=1}^{21*s} rS(k)}{2}$ .

Dans le cas où  $\sum_{k=1}^{21*s} rS(k)$  est pair, c'est théoriquement possible, et donc :  $\min_{théorique} |nbRS_{matins} - nbRS_{soirs}| = 0$ .

Dans le cas où  $\sum_{k=1}^{21*s} rs(k)$  est impair, c'est impossible, et donc, on a, au mieux :

$$nbRS_{matins} = \frac{\sum_{k=1}^{21*s} rs(k) - 1}{2} \quad \text{ou bien} \quad nbRS_{matins} = \frac{\sum_{k=1}^{21*s} rs(k) + 1}{2}, \quad \text{c'est-à-dire, dans tous les cas :}$$

$$\min_{\text{théorique}} |nbRS_{matins} - nbRS_{soirs}| = 1.$$

Dans le cas d'un roulement de type « 3\*8 » monté sur 3 semaines, si une solution est trouvée avec  $|nbRS_{matins} - nbRS_{soirs}| = 1$ , on aura ainsi un moyen de signaler que c'est le mieux que l'on puisse faire sur ce critère (toujours en gardant l'hypothèse qu'il n'y a aucune réserve de nuit), et qu'il est impossible d'atteindre  $|nbRS_{matins} - nbRS_{soirs}| = 0$ .

Il reste à exprimer cette propriété sous forme de contrainte. En fait, on note nbRS le nombre total d'appels à la réserve (c'est une quantité calculable à partir des données), et on a donc :

$$\text{diffMatSoir} \geq \left| \frac{nbRS}{2} - \left\lfloor \frac{nbRS}{2} \right\rfloor \right| - \sum_{p=1}^{7*s} rs(3*p).$$

Lorsque nbRS est pair, cette contrainte n'est pas active. De même, lorsque le nombre de nuits de réserve vaut au moins 1, elle n'est pas active. Mais si le nombre de nuits de réserve est nul et que la quantité  $s$  est impaire (soit nbRS impair), alors cette contrainte impose :  $\text{diffMatSoir} \geq 1$ .

On ajoute donc encore une contrainte.

#### 2.4.2.2.5 Bilan de l'apport des contraintes supplémentaires

Au final, on a donc ajouté  $7*s + 3$  contraintes. Si on se ramène à l'exemple du roulement de type « 3\*8 » monté avec 4 agents sur 4 semaines, on ajoute donc 31 contraintes.

Au cours des tests, on s'est aperçu que deux contraintes étaient particulièrement efficaces : les contraintes de type 2 et 3. Pour un roulement de type « 3\*8 » monté sur 2 semaines, leur introduction permet d'obtenir une solution admissible en moins d'une minute, alors qu'auparavant aucune solution n'était trouvée en moins de 5 minutes.

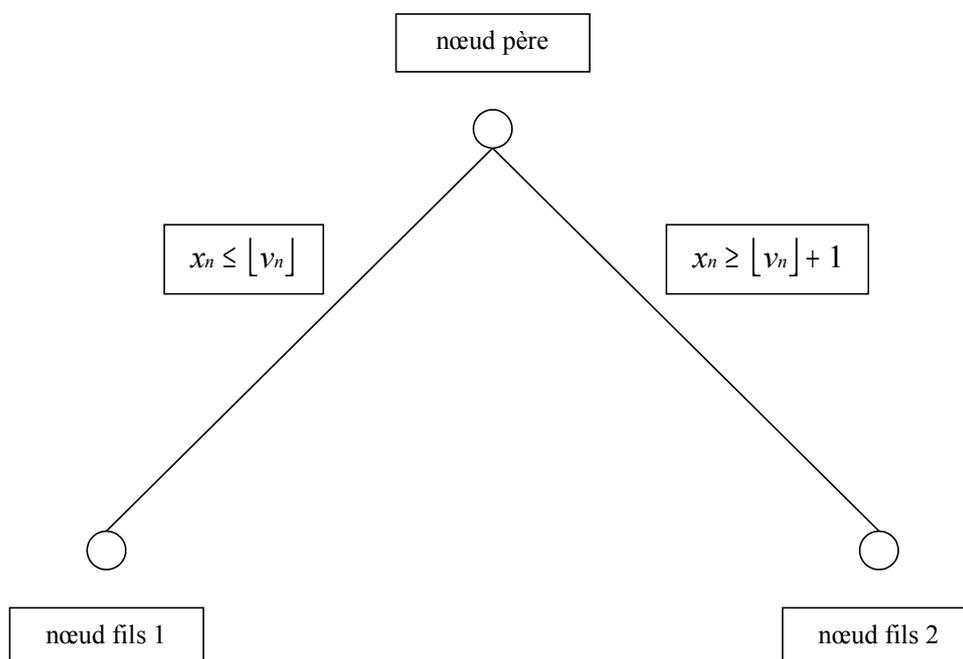
D'autre part, la contrainte de type 1 semble moins intéressante, mais ne doit pas être négligée, car elle permet tout de même de gagner plusieurs dizaines de secondes lors de la recherche d'une solution admissible.

### 2.4.2.3 Définition de priorités dans l'arbre de recherche

La troisième et dernière voie qui a été explorée concerne la mise en place de priorités entre les variables dans le parcours de l'arbre de recherche.

Cette mise en place de priorités est très facile à définir via OPL : on utilise l'instruction **setPriority**. Plus une variable a un taux de priorité élevé, plus on essaie de la fixer tôt dans l'arbre de recherche.

Par exemple, supposons qu'un problème comporte  $n$  variables entières  $x_1, \dots, x_n$ , et que chaque variable a un taux de priorité égal à 1, sauf  $x_n$  dont le taux de priorité vaut 5. Dans la solution optimale du problème relaxé qui est résolu au premier nœud de l'arborescence, si la variable  $x_n$  n'est pas entière et vaut  $v_n$ , c'est elle que l'on essaiera de fixer lors du premier branchement ; soit :



Sans précision de priorité, l'algorithme aurait pu choisir n'importe quelle variable pour ce nœud.

Dans notre problème, on choisit de fixer en premier les réserves (priorité la plus importante), puis les repos isolés. Les variables correspondant aux GPT et aux autres types de repos viennent ensuite, toutes à la même priorité. Enfin, toutes les variables liées aux contraintes additionnelles dépendent des variables de GPT ou de repos et sont donc fixées en dernier.

### 2.4.3 Etude de cas concrets

Cette section présente les résultats obtenus à l'aide d'OPL. La plupart des résultats porte sur des roulements de type « 3\*8 », que ce soit sur 2, 3 ou 4 semaines. Un roulement à besoins variables a aussi été testé.

De façon générale, la première chose que l'on peut tirer est qu'un roulement « 3\*8 » est infaisable sur une semaine (tel que défini dans notre étude). La recherche de solutions avec notre modèle a mis en évidence cette propriété.

Par ailleurs, les tests n'ont pas été poussés au-delà de 4 semaines pour des raisons de difficultés combinatoires, et parce que les résultats sur 2, 3 et 4 semaines donnaient déjà une bonne vision des possibilités du modèle.

Enfin, les tests ont également permis de constater que les contraintes *C2b.1*, *C2b.2*, *C2b.3* qui imposent l'enchaînement « Soirées – Matinées – Nuits » sont des contraintes très dures, puisqu'elles augmentent considérablement le temps de résolution.

Nous allons étudier, en détails et tour à tour, les résultats sur 2, 3 et 4 semaines, puis les résultats sur un roulement à besoins variables.

#### 2.4.3.1 Roulement de type « 3\*8 » sur 2 semaines

Ce roulement a été obtenu après 25 secondes de calcul.

##### Grille des besoins

Lu	Ma	Me	Je	Ve	Sa	Di
M	M	M	M	M	M	M
S	S	S	S	S	S	S
N	N	N	N	N	N	N

##### Tableau de roulement

Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di
S	S	S	S	RP	RP	RP	M	M	M	RP	RP	N	N
N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	M
M	RP	RP	N	N	N	N	RP	RP	S	S	S	S	RP
RP	RP	M	M	M	RU	RP	N	N	N	N	N	RP	RP
M				S	M	M	S						S

Regardons d'abord les indicateurs numériques :

	Valeurs du roulement étudié	Minimum légal
Nombre de repos doubles sur le roulement (sur l'année)	8 (52)	8 (52)
Nombre de samedi/dimanche sur le roulement (sur l'année)	2 (13)	2 (13)
Nombre de week-ends sur le roulement (sur l'année)	4 (26)	4 (26)
Nombre de repos (RP/RU) sur le roulement (sur l'année)	20 (130)	20 (130)

Les indicateurs sur les repos doubles, les samedi/dimanche et les week-end sont donc respectés au plus juste.

Examinons à présent les différents critères :

	Attributs de la solution étudiée	Optimum théorique
Nombre de réserves de nuit	0	0
Nombre de repos isolés	0	0
Somme des écarts quotidiens de réserve par rapport à la moyenne	6.857	6.857
Différence entre le nombre de matinées de réserve et le nombre de soirées de réserve	0	0
Somme des écarts par cycle du nombre de week-ends, par rapport à la moyenne (répartition entre les cycles)	2 (1-0-1-2)	0 (1-1-1-1)
Somme des écarts par cycle du nombre de samedi/dimanche, par rapport à la moyenne (répartition entre les cycles)	2 (1-0-0-1)	2 (1-0-0-1)

Cette solution semble donc excellente, puisque seul le critère 5 n'atteint pas son minimum, et d'ailleurs en est très proche. Après 400 secondes de calcul, on arrive d'ailleurs à montrer que cette solution est la solution optimale du problème.

### 2.4.3.2 Roulement de type « 3\*8 » sur 3 semaines

Deux solutions ont été retenues. La première est obtenue en arrêtant la recherche au bout de 20 minutes de calcul (sans preuve d'optimalité, donc), la seconde après plusieurs heures. Nous allons les examiner en parallèle et montrer leurs intérêts respectifs.

#### Grille des besoins

Lu	Ma	Me	Je	Ve	Sa	Di
M	M	M	M	M	M	M
S	S	S	S	S	S	S
N	N	N	N	N	N	N

#### Tableau de roulement : solution 1

Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di
S	RP	RP	M	M	M	RP	RU	N	N	N	N	RP	RP	S	S	S	S	S	RU	RP
M	M	M	RP	N	N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	RU	RP	RP
N	N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	M	RP	RP	N	N	N	N
RP	RP	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	N	N	RP	RP	S	S
S					S	M			S	M			S					M	M	M

#### Tableau de roulement : solution 2

Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di
N	N	N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	N
RP	RP	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	N	N	RP	RU	S	S
S	RP	RP	M	M	M	RP	RP	N	N	N	N	RP	RP	S	S	S	S	S	RP	RU
M	M	M	RP	RU	N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	M	RP	RP
S					S	MS			M	M		S						M	M	

Regardons tout d'abord les indicateurs numériques :

	Valeurs du roulement 1	Valeurs du roulement 2	Minimum légal
Nombre de repos doubles sur le roulement (sur l'année)	12 (52)	12 (52)	12 (52)
Nombre de samedi/dimanche sur le roulement (sur l'année)	3 (13)	4 (17.33)	3 (13)
Nombre de week-ends sur le roulement (sur l'année)	6 (26)	6 (26)	6 (26)
Nombre de repos (RP/RU) sur le roulement (sur l'année)	30 (130)	30 (130)	30 (130)

Les indicateurs sur les repos doubles, les samedi/dimanche et les week-ends sont donc respectés au plus juste au plus juste pour le roulement 1.

Pour le roulement 2, il y a plus de samedi/dimanche que nécessaire (un de plus sur le roulement).

Examinons à présent les différents critères :

	Attributs de la solution 1	Attributs de la solution 2	Optimum théorique
Nombre de réserves de nuit	0	0	0
Nombre de repos isolés	1	0	0
Somme des écarts quotidiens de réserve par rapport à la moyenne	10.286	11.143	10.286
Différence entre le nombre de matinées de réserve et le nombre de soirées de réserve	1	1	1 (d'après le 2.4.2.2.4, p. 34)
Somme des écarts par cycle du nombre de week-ends, par rapport à la moyenne (répartition entre les cycles)	4 (3-2-1-0)	2 (1-1-2-2)	2 (2-2-1-1 ou bien 1-1-2-2)
Somme des écarts par cycle du nombre de samedi/dimanche, par rapport à la moyenne (répartition entre les cycles)	1.5 (1-1-0-1)	0 (1-1-1-1)	1.5 (1-1-0-1) avec 3 samedi/dimanche et 0 (1-1-1-1) avec 4.

La première solution, mis à part le repos isolé, semble donc aussi bonne que celle de la section la précédente (sur 2 semaines). La deuxième solution est optimale sur tous les points, sauf en ce qui concerne la répartition de la réserve : on a une matinée et une soirée en réserve le premier dimanche, soit deux appels à la réserve le même jour.

On peut faire une remarque intéressante en comparant les deux solutions. En effet, dans la première solution, on avait trois samedi/dimanche répartis comme suit : 1-1-0-1. Donc, en réalité, la somme des écarts à la moyenne (qui était de  $\frac{3}{4} = 0.75$ ) était égale à  $(1-\frac{3}{4})+(1-\frac{3}{4})+\frac{3}{4}+(1-\frac{3}{4}) = 1.5$ . Avec trois samedi/dimanche (c'est-à-dire le nombre minimum nécessaire à placer sur un roulement), cette somme ne peut pas être inférieure à 1.5. Par contre, dans la deuxième solution, quatre samedi/dimanche ont été octroyé au lieu de trois. Cela a eu pour effet de rendre la somme des écarts à la moyenne nulle : la résolution a donc fait apparaître le fait qu'une bonne répartition n'était pas suffisante pour atteindre le minimum dans ce critère. Le nombre de samedi/dimanche doit aussi être adapté, si possible, au nombre de semaines, pour obtenir une moyenne **entière**.

La deuxième solution est donc excellente (mais longue à obtenir), mises à part deux utilisations de la réserve un même jour : elle est meilleure que la première sur les critères 2, 5 et 6, et moins bonne sur le critère 3.

### 2.4.3.3 Roulement de type « 3\*8 » sur 4 semaines

Tout d'abord, précisons que cette solution a été obtenue en neutralisant la totalité des critères, car le temps de calcul en tenant compte des critères était prohibitif. Elle a donc pour seule ambition de permettre d'examiner une solution admissible donnée par le programme sur quatre semaines. Précisons aussi que cette solution admissible a été obtenue au bout 340 secondes environ (presque 6 minutes).

#### Grille des besoins

Lu Ma Me Je Ve Sa Di

M	M	M	M	M	M	M
S	S	S	S	S	S	S
N	N	N	N	N	N	N

#### Tableau de roulement

Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di								
RP	RP	M	M	M	RP	RP	N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	RU	RP	RP	
S	S	S	S	RP	M	M	M	RP	RP	N	N	N	RP	RP	S	S	S	S	S	S	RP	M	M	M	RP	N	N	N
N	RU	RP	RP	S	S	S	RP	RP	M	M	M	M	RP	RP	RU	N	N	N	RP	S	S	S	S	RP	M	M	M	
RP	RP	N	N	N	RP	RP	S	S	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	RP	RP	S	S	S	S	
M	M				N	N	M					M	N	N			M				M		M					
	N											N																

A première vue, cette solution est donc assez mauvaise, et les experts en roulements de la SNCF nous ont montré des roulements de type « 3\*8 » sur 4 semaines beaucoup plus intéressants. Cependant, il faut remarquer que, moyennant un certain nombre d'opérations manuelles, ce roulement peut être sensiblement amélioré, et donc servir de base pour en construire un bien meilleur.

Nous allons voir quelles sont les substitutions simples que l'on peut effectuer et montrer ainsi à quel point il est difficile d'évaluer une solution : un roulement qui semble mauvais en terme de critères peut servir de base pour un meilleur roulement obtenu après quelques modifications rapides effectuées manuellement par un expert.

Les modifications à effectuer sont indiquées en couleur dans le tableau précédent. Les appels à la réserve indiqués en rouge sont ceux qui doivent « remonter », c'est-à-dire être assurés non plus par les agents de réserve, mais par les « rouleurs ». De même, les plages horaires remplies en rouge sont celles qui doivent « tomber » à la réserve, et ne plus être assurées par les « rouleurs ». Les repos isolés indiqués en vert sont ceux qui vont disparaître (et donc vont devenir des repos doubles), ceux indiqués en jaune sont ceux qui vont rester isolés.

Pour bien comprendre la façon dont les choses fonctionnent, on va détailler chaque étape.

- on fait remonter la nuit de réserve du premier mardi à la troisième ligne du tableau : cela est possible car, d'une part, la GPT de nuits dans laquelle va s'intégrer cette nuit contient quatre jours (donc il y a encore une place au maximum, puisque les GPT ne doivent pas faire plus de cinq jours), et d'autre part elle remplacera un RU intégré dans un repos triple ;
- on fait descendre la soirée du premier jeudi du deuxième cycle : cela est possible car cette soirée appartient à une GPT de quatre jours (une GPT ne peut pas faire moins de trois jours) et qu'elle est à la fin de la GPT. Ainsi, à la place de cette soirée, on place le RU récupéré ci-dessus, et on élimine un repos isolé ;

- on fait remonter la nuit de réserve du troisième mardi à la troisième ligne du tableau : cela est possible car, d'une part, la GPT de nuits dans laquelle va s'intégrer cette nuit contient quatre jours (donc il y a encore une place au maximum, puisque les GPT ne doivent pas faire plus de cinq jours), et d'autre part elle remplacera un RU intégré dans un repos triple ;
- on fait descendre la soirée du troisième samedi du deuxième cycle : cela est possible car cette soirée appartient à une GPT de cinq jours (une GPT ne peut pas faire moins de trois jours) et qu'elle est à la fin de la GPT. Ainsi, à la place de cette soirée, on place le RU récupéré ci-dessus, et on élimine un autre repos isolé ;
- enfin, on inverse le RU du quatrième vendredi du premier cycle avec la nuit du quatrième vendredi du deuxième cycle, et on élimine encore un repos isolé.

Le roulement finalement obtenu (roulement 2) est :

Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di	
RP	RP	M	M	M	RP	RP	N	N	N	RP	RP	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	RP	RP		
S	S	S	RU	RP	M	M	M	RP	RP	N	N	N	RP	RP	S	S	S	S	RU	RP	M	M	M	RP	RU	N	N	
N	N	RP	RP	S	S	S	RP	RP	M	M	M	M	RP	RP	N	N	N	N	RP	S	S	S	S	RP	M	M	M	
RP	RP	N	N	N	RP	RP	S	S	S	S	S	RP	RP	M	M	M	RP	RP	N	N	N	RP	RP	S	S	S	S	
M	M	S	N	N	M							M	N						S	M			M					

Les repos doubles obtenus lors des modifications sont indiqués en rouge et les repos qui sont restés isolés sont indiqués en jaune.

Comparons d'abord les indicateurs numériques :

	Valeurs du roulement 1	Valeurs du roulement 2	Minimum légal
Nombre de repos doubles sur le roulement (sur l'année)	16 (52)	16 (52)	16 (52)
Nombre de samedi/dimanche sur le roulement (sur l'année)	4 (13)	4 (13)	4 (13)
Nombre de week-ends sur le roulement (sur l'année)	7 (22.75)	8 (26)	7 (22.75)
Nombre de repos (RP/RU) sur le roulement (sur l'année)	40 (130)	40 (130)	40 (130)

Les indicateurs sur les repos doubles, les samedi/dimanche et les week-ends sont donc respectés au plus juste au plus juste pour le roulement 1.

Pour le roulement 2, il y a plus de week-ends que nécessaire (un de plus sur le roulement).

Comparons à présent les différents critères :

	Attributs de la solution 1	Attributs de la solution 2	Optimum théorique
Nombre de réserves de nuit	6	4	0
Nombre de repos isolés	5	2	0
Somme des écarts quotidiens de réserve par rapport à la moyenne	15.43	14.57	13.71
Différence entre le nombre de matinées de réserve et le nombre de soirées de réserve	6	4	0
Somme des écarts par cycle du nombre de week-ends, par rapport à la moyenne (répartition entre les cycles)	3.5 (3-2-0-2)	2 (3-2-1-2)	1.5 (2-2-1-2) avec 7 week-ends et 0 (2-2-2-2) avec 8.
Somme des écarts par cycle du nombre de samedi/dimanche, par rapport à la moyenne (répartition entre les cycles)	4 (2-0-0-2)	4 (2-0-0-2)	0 (1-1-1-1)

La deuxième solution obtenue est donc bien meilleure que la première, même si elle n'est probablement pas optimale. Les quelques changements effectués ont tout de même réussi à :

- améliorer sensiblement les critères 1, 2, 3, 4 et 5 ;
- conserver la même performance pour le critère 6.

### 2.4.3.4 Roulement à besoins variables sur 2 semaines

Le roulement considéré est le suivant :

#### Grille des besoins

Lu	Ma	Me	Je	Ve	Sa	Di
M	M	M	M	M	M	M
S	S	S	S	S	S	S
S	S	S	S	S	S	S
N	N	N	N	N	N	N

Les besoins diffèrent donc suivant les plages horaires : tous les jours, on a besoin d'un agent le matin et la nuit, mais de deux agents le soir. Remarquons que la contrainte d'enchaînement « Soirées – Matinées – Nuits » a été désactivée.

Le roulement suivant a été construit en 35 secondes environ.

#### Tableau de roulement

Lu	Ma	Me	Je	Ve	Sa	Di	Lu	Ma	Me	Je	Ve	Sa	Di
S	RP	RP	RP	S	S	S	S	RP	RP	M	M	M	RP
RP	S	S	S	S	S	RP	RP	S	S	S	S	RP	RP
M	M	M	M	M	RP	RP	N	N	N	N	N	RU	RP
RP	S	S	S	RP	RP	RP	S	S	S	RP	RP	N	N
N	N	N	RP	RP	M	M	M	M	M	RP	RU	S	S
S	RP	RP	N	N	N	N	RP	RP	RP	S	S	S	S
						S							M

Regardons d'abord les indicateurs numériques :

	Valeurs du roulement étudié	Minimum légal
Nombre de repos doubles sur le roulement (sur l'année)	12 (52)	12 (52)
Nombre de samedi/dimanche sur le roulement (sur l'année)	3 (13)	3 (13)
Nombre de week-ends sur le roulement (sur l'année)	6 (26)	6 (26)
Nombre de repos (RP/RU) sur le roulement (sur l'année)	30 (130)	30 (130)

Les indicateurs sur les repos doubles, les samedi/dimanche et les week-end sont donc respectés au plus juste.

Examinons à présent les différents critères :

	Attributs de la solution étudiée	Optimum théorique
Nombre de réserves de nuit	0	0
Nombre de repos isolés	0	0
Somme des écarts quotidiens de réserve par rapport à la moyenne	3.43	3.43
Différence entre le nombre de matinées de réserve et le nombre de soirées de réserve	0	0
Somme des écarts par cycle du nombre de week-ends, par rapport à la moyenne (répartition entre les cycles)	4 (1-2-2-1-0-0)	0 (1-1-1-1-1-1)
Somme des écarts par cycle du nombre de samedi/dimanche, par rapport à la moyenne (répartition entre les cycles)	3 (0-1-1-1-0-0)	3 (0-1-1-1-0-0)

La solution obtenue semble donc excellente : seul le critère 5 n'atteint pas son optimum théorique.

#### 2.4.3.5 Bilan

Dans la plupart des cas, il faut remarquer que, même si une « bonne » solution est trouvée en un temps raisonnable, la preuve de son optimalité n'est apportée que beaucoup plus tard. Par exemple, pour le roulement de type « 3\*8 » construit sur 2 semaines, la solution optimale est trouvée au bout de 25 secondes, mais la preuve de son optimalité n'est apportée qu'au bout de 400 secondes environ. Il vaut donc mieux faire part à l'expert des solutions trouvées au fur et à mesure, pour qu'il les évalue (notamment à l'aide des indicateurs numériques que l'on a déjà utilisé précédemment et que l'on peut aisément lui fournir), plutôt que de laisser l'algorithme essayer de prouver l'optimalité des solutions obtenues.

Concernant les résultats, on peut également faire remarquer que les experts de la Direction des Ressources Humaines des gares de Paris Montparnasse et Paris Saint-Lazare les ont trouvés plutôt satisfaisants. Les remarques portaient principalement sur les repos isolés et l'utilisation des RU à Paris Montparnasse, et les particularités locales pourraient nécessiter l'ajout de certaines contraintes et/ou critères, comme nous le verrons dans la section suivante.

## 2.5 Evolution du modèle

Cette section concerne des modifications qu'il serait intéressant d'effectuer sur le modèle, mais qui n'ont pas encore été étudiées dans le détail. Il convient également de noter à ce niveau qu'il pourrait éventuellement être intéressant de tester les contraintes proposées au 2.2.3.1, p. 27, pour réduire le nombre de critères, même si le modèle se comporte assez bien tel qu'il est.

### 2.5.1 Conception de roulements « quelconques »

La première modification à effectuer sur le modèle consiste à pouvoir considérer des roulements quelconques, c'est-à-dire autres que ceux de type « 3\*8 ». Pour cela, il faut, dans un premier temps, s'intéresser à la succession des GPT. En effet, dans un roulement de type « 3\*8 », l'ordre d'enchaînement des GPT est imposé : le motif « Soirées – Matinées – Nuits » se répète sur tout le roulement. Dans les autres types de roulement, aucune succession n'est imposée. Il faut donc vérifier, d'une façon ou d'une autre, que les agents n'effectuent pas leurs services toujours sur les mêmes plages horaires (soirées, matinées ou nuits) sur plusieurs semaines d'affilée.

A cet égard, des tests préliminaires ont été menés sur des roulements à besoins variables (notamment celui donné au 2.4.3.4, p. 45), en désactivant simplement les contraintes d'enchaînement de GPT (C2b.1), (C2b.2) et (C2b.3). Or, on observe que les GPT alternent entre elles (on n'observe pas 3 GPT de soirées à la suite, par exemple). Il est donc probable que cette alternance, voulue par les experts et les agents, ne soit pas un réel problème. Cependant, il faudrait effectuer des tests plus poussés pour valider cette conjecture et, éventuellement, ajouter quelques contraintes et/ou critères le cas échéant.

L'autre problème à considérer pour un roulement quelconque concerne la définition des plages horaires. Par exemple, on peut considérer des roulements en « 2\*8 », chaque jour ne comptant plus que deux plages horaires ; faisons l'hypothèse qu'il s'agit de « matinée » et « soirée ». La première solution consiste à garder le modèle tel quel, en ajoutant les contraintes suivantes :

- $\forall p \in \{1, \dots, 7*s*n\}, t_{(3*k)} = 0$  ;
- $\forall p \in \{1, \dots, 7*s*n\}, q_{(3*k)} = 0$  ;
- $\forall p \in \{1, \dots, 7*s*n\}, c_{(3*k)} = 0$  ;
- $\forall p \in \{1, \dots, 7*s\}, r_{S(3*k)} = 0$ .

Ces contraintes expriment le fait que, quel que soit le jour considéré, on ne peut pas avoir de GPT de nuits, ni de réserve de nuit.

Néanmoins, il s'agit d'une approche maladroite, puisqu'on ajoute des contraintes et on conserve des variables inutiles. L'autre solution, bien plus intéressante, est de ne définir initialement que deux plages horaires par jour au lieu de trois. Il faut alors reprendre l'expression du modèle en utilisant un paramètre  $ph$  qui désigne le nombre de plages horaires par jour : ainsi, au lieu d'avoir 21 plages horaires par semaines, on en aurait  $7*ph$ . Il faudrait aussi généraliser l'expression du modèle en utilisant ce paramètre  $ph$ .

### 2.5.2 Groupes de roulements

La seconde modification concerne l'interaction entre les roulements. En effet, comme on l'a déjà vu, les agents de réserve tournent sur plusieurs roulements. Pour améliorer la répartition globale de la réserve sur l'ensemble des roulements (et pas seulement individuellement sur chaque roulement), il

serait donc intéressant de considérer des « groupes de roulements », constitués des roulements partageant la même réserve. Cependant, le traitement d'un seul roulement posant déjà des problèmes hautement combinatoires, il est peut-être utopique de considérer plusieurs roulements **en même temps**. Face à ce constat, une solution serait peut-être de considérer les roulements successivement, tout en faisant en sorte que les résultats d'un roulement influent sur le suivant, en tant que données du problème.

Ainsi, pour réduire l'écart quotidien d'utilisation de la réserve par rapport à la moyenne, on ne considère plus l'ensemble des réserves du roulement courant, que l'on notera le roulement  $j$ , mais l'ensemble des réserves de tous les roulements  $i$  tels que  $i \leq j$ . Il faut donc, pour toutes les variables, ajouter un deuxième indice qui désigne le numéro de roulement.

Par exemple, considérons un groupe de  $r$  roulements. L'expression des contraintes (C5a), (C5b), (C5c) et (C5d) devient, pour le roulement  $j$ ,  $2 \leq j \leq r$  :

$$(C5c) : \forall p \in \{0, \dots, 7*s-1\}, e_{j(p+1)} \geq \sum_{k=1}^3 \left( \sum_{i=1}^{j-1} (rs(3*p+k)_i) + rs(3*p+k)_j \right) - \frac{\sum_{q=1}^{21*s} \left( \sum_{i=1}^{j-1} (rs_{qi}) + rs_{qj} \right)}{7*s} ;$$

$$(C5d) : \forall p \in \{0, \dots, 7*s-1\}, e_{j(p+1)} \geq \frac{\sum_{q=1}^{21*s} \left( \sum_{i=1}^{j-1} (rs_{qi}) + rs_{qj} \right)}{7*s} - \sum_{k=1}^3 \left( \sum_{i=1}^{j-1} (rs(3*p+k)_i) + rs(3*p+k)_j \right) ;$$

$rs_{qi}$  étant considérée comme une donnée,  $1 \leq q \leq 21*s$ ,  $1 \leq i \leq j-1$ .

Cette option ne donnerait probablement pas la solution optimale globale sur l'ensemble des roulements. Par contre, elle fournirait peut-être une bonne solution à moindres frais. Il faudrait effectuer des tests et évaluer les résultats.

### 2.5.3 Prise en compte des remarques des experts.

Les remarques des experts portent sur deux points. D'abord, il leur arrive peu fréquemment d'accoler des RU à un seul RP. Donc, les meilleures configurations sont RP-RP-RU ou RU-RP-RP. Cela pourrait justifier le fait d'ajouter un critère sur les variables  $ru$ ,  $ur$ ,  $rur$ ,  $ruu$ ,  $uru$  et  $uur$  :

$$\min \theta^* \sum_{k=1}^{7*s*n} (ru_k + ur_k + rur_k + ruu_k + uru_k + uur_k)$$

D'autre part, les repos isolés sont quasiment interdits. Cela justifierait donc, en plus du critère portant sur les repos isolés, l'ajout d'une contrainte sur le nombre total de repos isolés. En effet, un expert sait, par exemple, que compte-tenu des accords passés avec les partenaires sociaux, il refusera catégoriquement tout roulement en excédent sur le quota de repos isolés négocié (par exemple, pas plus d'un repos isolé sur un roulement de type « 3\*8 » monté sur 2 semaines).

Il suffit donc d'ajouter une contrainte qui interdit plus de  $ri$  repos isolés,  $ri$  étant un paramètre défini par l'expert :

$$\sum_{k=1}^{7*s*n} r_k \leq ri .$$

### 3 Conclusion

La conception des roulements de personnel est, en définitive, un problème relativement complexe à modéliser. Le modèle finalement implanté se présente sous la forme d'un programme linéaire en variables entières (bivalentes, même, pour la plupart), avec un nombre de contraintes et un nombre de variables relativement raisonnables. Pourtant, les temps de calcul ne sont pas toujours négligeables. Par exemple, pour un roulement de type « 3\*8 », passer de 2 semaines à 3 semaines augmente considérablement le temps de résolution. D'un autre côté, la solution sur 2 semaines est excellente, et peut-être n'est-il pas nécessaire de s'intéresser aux solutions sur 3 semaines. D'autant que c'est la démarche suivie actuellement par les experts : ils commencent par essayer de monter leurs roulements sur 2 ou 3 semaines et augmentent progressivement le nombre de semaines, s'ils ne trouvent pas de solutions satisfaisantes. Si l'on pouvait toujours se ramener à 2 semaines, le temps de calcul ne poserait plus de problème. Cependant, on l'a vu pour le cas des roulements de type « 3\*8 » : ils sont infaisables sur 1 semaine. On peut donc imaginer qu'il existe, de la même façon, un type de roulements infaisables sur 1 ou 2 semaine(s). Pour l'instant en tout cas, rien ne prouve le contraire.

Tout ceci nous pousse à nous interroger sur les suites prévisibles de cette étude. En effet, l'un des points les plus importants à ce niveau est : comment faire pour diminuer encore un peu plus les temps de calcul ? Pour l'instant, une solution sur 2 semaines s'obtient en moins d'une minute, alors qu'il faut attendre 20 minutes pour une solution sur 3 semaines, bonne au demeurant. L'un des objectifs de l'outil d'aide automatisée à la conception de roulements dans le cadre duquel cette étude s'insère, c'est de fournir rapidement différentes solutions (optimales ou non) à l'expert : c'est pourquoi il est aussi important de chercher à améliorer les performances. Pour cela, on pourra peut-être s'inspirer des quelques pistes encore inexplorées qui ont été évoquées dans cette étude (notamment la réduction du nombre de critères par ajout de contraintes), ou bien on essaiera de chercher à perfectionner les méthodes déjà développées. Il serait également très intéressant de chercher à en développer de nouvelles.

Du point de vue de la modélisation, la généralisation du modèle à des roulements quelconques est forcément la première évolution à envisager, en suivant les consignes exposées dans la section précédente. De la même façon, il faudrait étudier en détails la prise en compte des groupes de roulement, en suivant les pistes lancées dans cette étude ou non.

Par ailleurs, il y a un point qui n'a pas encore été abordé. En effet, nous ne nous sommes intéressés qu'à la planification des roulements de personnel, donc à une gestion à long terme. Tout reste donc à faire dans le domaine de la gestion opérationnelle des agents. Mais pour étudier une gestion à plus court terme, le plus important est peut-être, dans un premier temps, de réduire les temps de résolution : c'est même probablement une condition nécessaire.

Au final, on peut remarquer que cette première étude a fourni des résultats plutôt encourageants du point de vue de la SNCF : c'est pour cette raison qu'une suite lui sera probablement donnée. Il est intéressant de préciser que, si un tel outil d'aide à la décision venait à être finalisé, il s'insérerait probablement dans une application intranet, ou en tout cas facilement accessible par tous et aisément utilisable. La qualité première d'un tel outil doit être son interactivité, et sa faculté à se laisser guider par l'utilisateur final, c'est-à-dire l'expert en roulements : c'est probablement lui qui devrait être le plus à même d'influer sur la qualité des solutions obtenues.

## 4 Bibliographie

- [Bertocchi Chouvaeff Lemoine 91] BERTOCCHI R., CHOUVAEFF J.-M. et LEMOINE J.-F. — Elaboration des tableaux de roulement des conducteurs de la RATP : le logiciel CATAR. Revue Générale des Chemins de Fer, Décembre 1991.
- [Caprara Fischetti Toth Vigo 95] CAPRARA A., FISCHETTI M., TOTH P., VIGO D. — Modeling and solving the crew rostering problem. Rapport technique, Université de Bologne (Italie), Département d'électronique - informatique, 1995.
- [Caprara Fischetti Toth Vigo 97] CAPRARA A., FISCHETTI M., TOTH P., VIGO D. — Solution of Large-Scale Railway Crew Planning Problems : The Italian Experience. Rapport technique, Université de Bologne (Italie), Département d'électronique - informatique, Mai 1997.
- [Kroon Fischetti 00] KROON L., FISCHETTI M. — Crew Scheduling for Netherlands Railways. Rapport de Recherche, Erasmus Research Institute of Management (Pays-Bas), Décembre 2000.
- [Fores 96] FORES S. — Column Generation Approaches to Bus Driver Scheduling. Université de Leeds (Royaume-Uni), Mars 1996.
- [Sanders Takkula Wedelin 99] SANDERS P., TAKKULA T., WEDELIN D. — High Performance Integer Optimization for Crew Scheduling. HPCN Europe, 1999.
- [Vance Barnhart Johnson Nemhauser 95] VANCE P., BARNHART C., JOHNSON E., NEMHAUSER G. — Airline Crew Scheduling : A New Formulation and Decomposition Algorithm. Institut de Technologie de Georgie (Etats-Unis), Ecole de Génie Industriel et Systémique, Mai 1995.
- [Dawid König Strauss 00] DAWID H., KONIG J., STRAUSS C. — An Enhanced Rostering Model for Airline Crews. Université de Vienne (Autriche), Département des Sciences du Management, 2000.
- [Kragelund Mayoh 99] KRAGELUND L., MAYOH B — Nurse Scheduling Generalised. Université d'Aarhus (Danemark), Département des Sciences Informatiques, Juin 1999.
- [Chiarandini Schaerf Tiozzo 99] CHIARANDINI M., SCHAERF A., TIOZZO F. — Solving Employee Timetabling Problems with Flexible Workload using Tabu Search. Université d'Udine (Italie), Département d'électronique, d'informatique de gestion et de génie mécanique, 1999.
- [Gunes 99] GUNES D. — Workforce Scheduling. Université de Bilkent (Turquie), Département des Sciences Industrielles, Avril 1999.
- [Schaerf Meisels 99] SCHAERF A., MEISELS A. — Solving Employee Timetabling Problems by Generalized Local Search. Université de Néguev (Israël), Département de Mathématiques et de Sciences Informatiques, 1999.
- [König Strauss 00] KÖNIG J., STRAUSS C. — Rostering-integrated Services and Crew Efficiency. Université de Vienne (Autriche), Département des Sciences du Management, 2000.
- [Lau Lua 97] LAU H. C., LUA S. C. — Efficient Multi-Skill Crew Rostering via Constrained Sets. Institut des Technologies de l'Information (Singapour), 1997.
- [Fahle Junker Karisch Kohl Sellmann Vaaben 99] FAHLE T., JUNKER U., KARISCH S. E., KOHL N., SELLMANN M., VAABEN B. — Constraint Programming Based Column Generation for Crew Assignment. Université de Paderborn (Allemagne), Département de Mathématiques et de Sciences Informatiques, Novembre 1999.

[Hertz Robert Berthod 97] HERTZ A., ROBERT V., BERTHOD V. — Planification des emplois du temps et de la formation au sein d'une grande entreprise. Ecole Polytechnique Fédérale de Lausanne (Suisse), Département de Mathématiques, Juillet 1997.

[Boizumault Delon Périddy 95] BOIZUMAULT P., DELON Y., PERIDY L. — Constraint Logic Programming for Examination Timetabling. *J. Logic Programming*, p.1-17, 1995.

[Gabriel Roy 90] GABRIEL T., ROY J. — Une approche de recouvrement d'ensembles pour l'établissement d'horaires de chauffeurs dans le transport routier de charges partielles. *Operations Research*, volume 24 n°2, p. 123-158, 1990.

[Steiger 61] STEIGER F. — Optimization of Swiss Air's Crew Scheduling by an Integer Linear Programming Model. Département de recherche opérationnelle, Société Swissair 1961.

[DRH SNCF 00] Direction des Ressources Humaines de la SNCF — Réglementation du travail, document RH-0077, Février 2000.

[DRH SNCF 99] Direction des Ressources Humaines de la SNCF — Réglementation du travail, document RH-0677, Décembre 1999.

[Minoux 90] MINOUX M. — Programmation mathématique : théorie et algorithme. Tomes 1 et 2, Collection Technique et Scientifique des Télécommunications – Dunod 1990.

[Partouche 97] PARTOUCHE A. — A hybrid method for the general rostering problem with a mixed homogeneous workforce. LAMSADE, Université Paris IX Dauphine (France), Cahier n°151, Décembre 1997.

[Partouche 98] PARTOUCHE A. — Planification d'horaires de travail : méthodologie, modélisation et résolution à l'aide de la programmation linéaire en nombres entiers et de la programmation par contraintes. Thèse de doctorat, UFR Sciences des Organisations, Université Paris IX Dauphine (France), Janvier 1998.

[Jacquet-Lagrèze Montaut Partouche 97] JACQUET-LAGREZE E., MONTAUT D., PARTOUCHE A. — The shift scheduling problem : different formulations and solutions methods. Cahier du Lamsade, Université Paris IX Dauphine (France), Juillet 1997.

[Gabteni 99] GABTENI S. — Heuristiques et méthodes de décomposition appliquées à des problèmes d'optimisation technique et commerciale à la SNCF. Thèse de doctorat, UFR Sciences des Organisations, Université Paris IX Dauphine (France), Janvier 1999.

[Wren 96] WREN A. — Scheduling, Timetabling and Rostering : a special relationship. Ecole des Sciences Informatiques, Université de Leeds (Royaume-Uni), 1996.

[Smith 86] SMITH B. M. — Bus crew scheduling using linear programming. Thèse de l'Université de Leeds, 1986.

[Barnhart Shenoï 96] BARNHART C., SHENOI R.G.— An alternate model and solution approach for the long haul crew pairing problem. MIT, Centre des Etudes sur le Transport (Etats-Unis), Juillet 1996.

## 5 Annexes

### 5.1 Calcul du nombre d'agents nécessaires pour tenir un 3\*8

- 132 jours de repos.
- Jours de travail requis sur une année :  $52$  (nombre de semaines / an) \*  $7$  (nombre de jours / semaine) \*  $3$  (nombre de plages / jour) =  $1092$ .
- Jours travaillés par un agent sur une année :  $365 - 132$  (jours de repos) =  $233$ .

$$\Rightarrow \text{nombre d'agents nécessaires pour assurer le } 3*8 = \frac{1092}{233} = 4.686.$$

En fait, si on a  $n$  lignes dans la grille des besoins, on a en tout (au maximum)  $7*n$  besoins dans la semaine, soit  $364*n$  dans l'année. Par exemple, pour un roulement 3\*8, on a 3 lignes dans la grille des besoins, et on a donc  $7*3 = 21$  besoins dans la semaine.

Ainsi, pour que les ressources couvrent à peu près les besoins, on doit avoir :  $\frac{233*(n + p)}{364*n} \approx 1$ ,

soit  $0.64 + \frac{233*p}{364*n} \approx 1$ ,  $p$  étant le nombre d'agents supplémentaires nécessaires (c'est-à-dire en plus

des  $n$  premiers). D'où :  $\frac{p}{n} \approx 0.36 * \frac{364}{233} \approx 0.56$ . On a donc :  $p \approx 0.56*n$  (au maximum). Ensuite, se

pose la question de savoir si l'on doit utiliser  $\lfloor p \rfloor$  (mode de calcul par défaut) ou  $\lceil p \rceil$  (mode de calcul par excès) agents supplémentaires. Par exemple, pour un roulement de type 3\*8,  $p = 0.56*3 = 1.68$ . On a alors le choix d'utiliser 4 ( $n + \lfloor p \rfloor$ ) ou 5 ( $n + \lceil p \rceil$ ) agents, sachant qu'avec 5, aucun appel à la réserve ne devrait être fait.

## 5.2 Tableau récapitulatif des contraintes et critères d'un roulement de type « 3\*8 »

Ce tableau a été établi en collaboration avec les experts en planification de roulements de personnel de la Direction des Ressources Humaines de la SNCF.

	<b>Contraintes</b>	<b>Critères (préférences)</b>
<b>Intra – GPT</b>	<ul style="list-style-type: none"> <li>- Une GPT contient 3, 4 ou 5 jours.</li> <li>- Une GPT n'est associée qu'à un seul type de plage horaire : matinée, soirée ou nuit.</li> </ul>	
<b>Inter – GPT</b>	<ul style="list-style-type: none"> <li>- Il doit y avoir au moins un jour de repos entre 2 GPT.</li> <li>- Les GPT s'enchaînent suivant le motif Soirée – Matinée – Nuit.</li> </ul>	
<b>Réserve</b>		<ul style="list-style-type: none"> <li>- Réduire le plus possible l'appel à la réserve.</li> <li>- Eviter de faire trop souvent appel à la réserve pour les postes de nuit.</li> <li>- Equilibrer le plus possible l'appel à la réserve entre les jours de la semaine.</li> <li>- Equilibrer le plus possible l'appel à la réserve entre les soirées et les matinées.</li> </ul>
<b>Repos</b>	<ul style="list-style-type: none"> <li>- Il faut au moins 12 Samedi-Dimanche de type RP-RP dans l'année.</li> <li>- Il faut au moins 22 week-ends de type RP/RU-RP-RP/RU dans l'année.</li> <li>- Il faut au moins 52 repos doubles (RP-RP) dans l'année.</li> <li>- Il faut 132 jours de repos (RP+RU) dans l'année.</li> <li>- Il faut 118 RP dans l'année.</li> <li>- Pas plus de 3 jours de repos consécutifs.</li> </ul>	<ul style="list-style-type: none"> <li>- Arriver à la répartition la plus équilibrée possible des Samedi-Dimanche sur l'année.</li> <li>- Arriver à la répartition la plus équilibrée possible des Week-end sur l'année.</li> <li>- Réduire le plus possible le nombre de jours de repos isolés (c'est-à-dire non inclus dans une période de repos au moins double).</li> </ul>
<b>Charge</b>	<ul style="list-style-type: none"> <li>- La charge de travail doit être couverte, éventuellement à l'aide de la réserve.</li> </ul>	

### 5.3 Tableau illustratif des données

Chaque case de ce tableau contient le numéro de la plage horaire qui lui correspond, et les plages horaires nécessitant le même nombre d'agents sont représentées avec la même couleur (seules les quatre premières colonnes ont été considérées). En outre, la notion de cycle étant absente de l'expression des besoins, les semaines sont représentées les unes en-dessous des autres.

	Lundi			Mardi			Mercredi			Jeudi			Vendredi			Samedi			Dimanche		
	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N
Semaine 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Semaine 2	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Semaine 3	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

### 5.4 Tableau récapitulatif des plages horaires couvertes par une variable $q_k$

Les cases de même couleur correspondent aux plages horaires couvertes par une GPT de 4 jours, suivant le type de la plage horaire de départ. Une GPT de 4 jours couvre les plages horaires de couleur jaune si elle commence en  $k$ , celles de couleur rouge si elle commence en  $k+1$  et celles de couleur bleue si elle commence en  $k+2$ .

	Jour numéro $\frac{k-1}{3} + 1$			Jour numéro $\frac{k-1}{3} + 2$			Jour numéro $\frac{k-1}{3} + 3$			Jour numéro $\frac{k-1}{3} + 4$		
Numéro de plage	k	k+1	k+2	k+3	k+4	k+5	k+6	k+7	k+8	k+9	k+10	k+11
Type de plage	M	S	N	M	S	N	M	S	N	M	S	N

### 5.5 Tableau récapitulatif des plages horaires couvertes par une variable $c_k$

Les cases de même couleur correspondent aux plages horaires couvertes par une GPT de 5 jours, suivant le type de la plage horaire de départ. Une GPT de 5 jours couvre les plages horaires de couleur jaune si elle commence en  $k$ , celles de couleur rouge si elle commence en  $k+1$  et celles de couleur bleue si elle commence en  $k+2$ .

	Jour numéro $\frac{k-1}{3} + 1$			Jour numéro $\frac{k-1}{3} + 2$			Jour numéro $\frac{k-1}{3} + 3$			Jour numéro $\frac{k-1}{3} + 4$			Jour numéro $\frac{k-1}{3} + 5$		
Numéro de plage	k	k+1	k+2	k+3	k+4	k+5	k+6	k+7	k+8	k+9	k+10	k+11	k+12	k+13	k+14
Type de plage	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N

### 5.6 Tableau illustrant le phénomène de cyclicité des roulements

		Semaine 1							Semaine s							
1 roulement complet	Cycle 1	Jour	1	2	3	4	5	6	7	$7*s-6$	$7*s-5$	$7*s-4$	$7*s$	$7*s-2$	$7*s-1$	$7*s$
	Cycle n	Jour	$1+7*s*(n-1)$	$2+7*s*(n-1)$	$3+7*s*(n-1)$	$4+7*s*(n-1)$	$5+7*s*(n-1)$	$6+7*s*(n-1)$	$7+7*s*(n-1)$	$7*s*n-6$	$7*s*n-5$	$7*s*n-4$	$7*s*n-3$	$7*s*n-2$	$7*s*n-1$	$p=7*s*n$
Nouveau roulement	Cycle 1	Jour	$p+1=1$	2	3	4	5	6	7	$7*s-6$	$7*s-5$	$7*s-4$	$7*s-3$	$7*s-2$	$7*s-1$	$7*s$

## 5.7 Illustration d'un cas de non respect de la propriété *prop1*

Rappelons que si l'on note  $\text{minGPT}$  la longueur minimale d'une GPT,  $\text{minRP}$  la longueur minimale d'une période de repos et  $\text{maxRP}$  la longueur maximale d'une période de repos, la propriété **prop1** qui doit être vérifiée par ces grandeurs s'écrit :  $2 \cdot \text{minRP} + \text{minGPT} > \text{maxRP}$ . Examinons un cas ne respectant pas cette propriété, en posant par exemple que les périodes de repos de 5 jours sont autorisées. On a :  $2 \cdot \text{minRP} + \text{minGPT} = 2 \cdot 1 + 3 = 5$  et  $\text{maxRP} = 5$ .

Il faut donc **explorer jusqu'au jour  $p+11$  pour être sûr de trouver la  $(m+1)^{\text{e}}$  GPT**. Mais alors, supposons qu'un seul jour de repos suive la  $m^{\text{e}}$  GPT, qu'une GPT de 3 jours (la  $(m+1)^{\text{e}}$  GPT) commence juste après ce jour de repos, et que cette GPT soit elle-même suivie d'un seul jour de repos. Une GPT (la  $(m+2)^{\text{e}}$  GPT) pourrait ensuite commencer le jour  $p+11$ , et en explorant les jours  $p+7$  à  $p+11$ , on trouverait non pas une, mais 2 GPT, ce qui ne doit jamais être le cas pour que l'expression de la contrainte soit valide.

Les deux tableaux ci-dessous résument ce problème. Le premier tableau représente le cas extrême (5 jours de repos) qui nous oblige à explorer jusqu'au jour  $p+11$  pour trouver une GPT :

Jour $p+3$	Jour $p+4$	Jour $p+5$	Jour $p+6$	Jour $p+7$	Jour $p+8$	Jour $p+9$	Jour $p+10$	Jour $p+11$
m <sup>e</sup> GPT			RP 5 j.					(m+1) <sup>e</sup> GPT

Le second tableau représente le cas où 2 GPT sont rencontrées entre les jours  $p+7$  et  $p+11$ :

Jour $p+3$	Jour $p+4$	Jour $p+5$	Jour $p+6$	Jour $p+7$	Jour $p+8$	Jour $p+9$	Jour $p+10$	Jour $p+11$
m <sup>e</sup> GPT			RP 1 j.	(m+1) <sup>e</sup> GPT		RP 1 j.	(m+2) <sup>e</sup> GPT	
AMPLITUDE DE LA RECHERCHE								

### 5.8 Tableau illustrant les GPT couvrant une plage horaire donnée

Ce tableau résume l'ensemble des GPT couvrant une plage horaire  $k$ . La plage horaire  $k$  considérée est ici telle que  $k \equiv 2[3]$ , c'est-à-dire que cette plage horaire  $k$  correspond à une soirée. Les GPT de 3 jours sont représentées en jaune, celles de 4 jours en rouge, et celles de 5 jours en bleu. Un « \* » indique la plage de début d'une GPT. Remarquons aussi que tous les indices  $i$  différents de  $k$  doivent être considérés modulo un roulement complet (ie modulo  $21*s*n$  plages horaires).

	Jour numéro $\frac{k+1}{3} - 4$			Jour numéro $\frac{k+1}{3} - 3$			Jour numéro $\frac{k+1}{3} - 2$			Jour numéro $\frac{k+1}{3} - 1$			Jour numéro $\frac{k+1}{3}$		
Numéro de plage	k-13	k-12	k-11	k-10	k-9	k-8	k-7	k-6	k-5	k-4	k-3	k-2	k-1	k	k+1
Type de plage	M	S	N	M	S	N	M	S	N	M	S	N	M	S	N
GPT 3 jours														*	
											*				
								*							
GPT 4 jours														*	
								*			*				
					*				*			*			
GPT 5 jours														*	
								*			*				
		*			*				*			*			

### 5.9 Calcul du résultat important du 2.2.3.2

- Pour  $(C0)$ , on somme sur  $p$  allant de 0 à  $7*s*n-1$ .
- Pour  $(C4)$ , on fait la somme pour  $k$  allant de 1 à  $21*s$ .

- A partir de là, la quantité  $( \sum_{p=0}^{7*s*n-1} ( \sum_{k=3*p-5}^{3*(p+1)} t(k[21*s*n]) + \sum_{k=3*p-8}^{3*(p+1)} q(k[21*s*n]) + \sum_{k=3*p-11}^{3*(p+1)} c(k[21*s*n]) ) )$  est commune aux deux égalités, et on peut donc alors identifier les deux expressions.

## 5.10 Listing du fichier « roulements.mod »

```
/* debut parametres */

float nbSemainesFloat = ...;
float nbAgentsFloat = ...;
int nbSemaines = ftoi(nbSemainesFloat);
int nbAgents = ftoi(nbAgentsFloat);
float nbReposParAn = ...;
float nbRPParAn = ...;
int B = ...// Reserves de nuits
int C = ...// Ecart quotidien de reserve
int D = ...// Ecart Soirees-Matinee de reserve
int E = ...// Ecart Samedi/Dimanche par rapport a la moyenne
int F = ...// Ecart Week-Ends par rapport a la moyenne
int G = ...// Nombre repos isoles

/* fin parametres */

/* debut ensembles de variation */

range plaHorParCycle 0..21*nbSemaines-1; /* nombre de plages horaires par cycle */
range plaHorParRoul 0..21*nbSemaines*nbAgents-1; /* nombre de plages horaires par roulement */
range jourParCycle 0..7*nbSemaines-1; /* nombre de jours par cycle */
range jourParRoul 0..7*nbSemaines*nbAgents-1; /* nombre de jours par roulement */
range semParRoul 0..nbSemaines*nbAgents-1; /* nombre de semaines par roulement */
range cycleParRoul 0..nbAgents-1; /* nombre de cycles par roulement */
range semParCycle 0..nbSemaines-1; /* nombre de semaines par cycle */

/* fin ensembles de variation */

/* debut donnees */

int work[i in plaHorParCycle] = 1;
float workFloat[i in plaHorParCycle] = 1.0;

float reposRoul = nbReposParAn*nbSemainesFloat*nbAgentsFloat/52.0; /* nombre de repos a placer dans le roulement */
float RPRoul = nbRPParAn*nbSemainesFloat*nbAgentsFloat/52.0; /* nombre de RP (repos periodiques) a placer dans le roulement */
float weRoul = 22.0*nbSemainesFloat*nbAgentsFloat/52.0; /* nombre de week-ends a placer dans le roulement */
float sadiRoul = 12.0*nbSemainesFloat*nbAgentsFloat/52.0; /* nombre de samedi/dimanche a placer dans le roulement */
int roul = 7*nbSemaines*nbAgents; /* nombre de jours d'un roulement */
float nbRSFloat = (sum(k in plaHorParCycle) workFloat[k]) - 7.0*nbSemainesFloat*nbAgentsFloat + floor(reposRoul); /* nombre de reserves a placer dans le roulement */
int nbRS = ftoi(nbRSFloat);

/* fin donnees */

/* debut variables principales */

var int rs[plaHorParCycle] in 0..nbAgents; /* agents de reserve : au grand maximum, nb de lignes du tableau */
var int t[plaHorParRoul] in 0..1; /* GPT de 3 jours */
var int q[plaHorParRoul] in 0..1; /* GPT de 4 jours */
var int c[plaHorParRoul] in 0..1; /* GPT de 5 jours */
var int r[jourParRoul] in 0..1; /* RP */
var int rr[jourParRoul] in 0..1; /* RP-RP*/
```

```

var int ru[jourParRoul] in 0..1; /* RP-RU*/
var int ur[jourParRoul] in 0..1; /* RU-RP*/
var int rrr[jourParRoul] in 0..1; /* RP-RP-RP*/
var int rru[jourParRoul] in 0..1; /* RP-RP-RU*/
var int rur[jourParRoul] in 0..1; /* RP-RU-RP*/
var int ruu[jourParRoul] in 0..1; /* RP-RU-RU*/
var int uru[jourParRoul] in 0..1; /* RU-RP-RU*/
var int urr[jourParRoul] in 0..1; /* RU-RP-RP*/
var int uur[jourParRoul] in 0..1; /* RU-RU-RP*/

```

*/\* fin variables principales \*/*

*/\* debut variables additionnelles \*/*

**var** int diffMatSoir **in** 0..nbRS; /\* valeur absolue de la difference entre le nombre de matins de reserve et le nombre de soirs de reserve \*/

**var** int sadi[cycleParRoul] **in** 0..**ftoi**(**ceil**(sadiRoul)); /\* nombre de samedi/dimanche de chaque cycle \*/

**var** int we[cycleParRoul] **in** 0..**ftoi**(**ceil**(weRoul)); /\* nombre de week-ends de chaque cycle \*/

**var** int absVarRs[jourParCycle] **in** 0..7\*nbSemaines\*nbAgents; /\* valeurs absolues de la difference entre l'appel quotidien a la reserve et l'appel moyen sur le roulement, multipliee par le nombre de jours d'un roulement\*/

**var** int absVarSadi[cycleParRoul] **in** 0..**ftoi**(**ceil**(sadiRoul)); /\* valeurs absolues de la difference entre le nombre de samedi/dimanche du cycle et le nombre moyen de samedi/dimanche sur le roulement, multipliee par le nombre de cycles d'un roulement\*/

**var** int absVarWe[cycleParRoul] **in** 0..**ftoi**(**ceil**(weRoul)); /\* valeurs absolues de la difference entre le nombre de week-ends du cycle et le nombre moyen de we sur le roulement, multipliee par le nombre de cycles d'un roulement\*/

*/\* fin variables additionnelles \*/*

*/\* debut definition des priorites sur les variables \*/*

```

setting mipsearch {
  forall(k in plaHorParCycle) {
    setPriority(rs[k],10);
  };
  forall(k in plaHorParRoul) {
    setPriority(t[k], 5);
    setPriority(q[k], 5);
    setPriority(c[k], 5);
  };
  forall(k in jourParRoul) {
    setPriority(r[k], 9);
    setPriority(rr[k], 5);
    setPriority(ru[k], 5);
    setPriority(ur[k], 5);
    setPriority(rrr[k], 5);
    setPriority(rur[k], 5);
    setPriority(rru[k], 5);
    setPriority(urr[k], 5);
    setPriority(uur[k], 5);
    setPriority(uru[k], 5);
    setPriority(ruu[k], 5);
  };
  setPriority(diffMatSoir, 8);
  forall(k in cycleParRoul) {
    setPriority(sadi[k], 4);
    setPriority(we[k], 4);
    setPriority(absVarSadi[k], 2);
    setPriority(absVarWe[k], 2);
  };

```

```

};
forall(k in jourParCycle) {
  setPriority(absVarRs[k], 8);
};
};

/* fin definition des priorites sur les variables */

/* debut fonction objectif */

minimize B*(sum(k in jourParCycle) rs[2+3*k]) + C*(sum(k in jourParCycle) absVarRs[k]) + D*diffMatSoir
+ E*(sum(k in cycleParRoul) absVarSadi[k]) + F*(sum(k in cycleParRoul) absVarWe[k]) + G*(sum(k in
jourParRoul) r[k])

/* fin fonction objectif */

subject to {

/* debut contraintes */

/* debut contrainte d'unicite */

forall(p in jourParRoul) sum(k in 0..8) t[(3*p-6+k+3*roul) mod (3*roul)] + sum(k in 0..11)
q[(3*p-9+k+3*roul) mod (3*roul)] + sum(k in 0..14) c[(3*p-12+k+3*roul) mod (3*roul)] + r[p] + sum(k in
0..1) (rr[(p+roul-k) mod roul]+ru[(p+roul-k) mod roul]+ur[(p+roul-k) mod roul]) + sum(k in 0..2) (rrr[(p+roul-
k) mod roul]+rru[(p+roul-k) mod roul]+rur[(p+roul-k) mod roul]+ruu[(p+roul-k) mod roul]+urr[(p+roul-k)
mod roul]+uru[(p+roul-k) mod roul]+uur[(p+roul-k) mod roul]) = 1; /* C0 */

/* fin contrainte d'unicite */

/* debut contraintes inter-GPT */

forall(p in jourParRoul) sum(k in 1..3) (t[((3*p)-10+k+(3*roul)) mod (3*roul)]+q[((3*p)-13+k+(3*roul)) mod
(3*roul)]+c[((3*p)-16+k+(3*roul)) mod (3*roul)]) = r[p]+rr[p]+ru[p]+ur[p]+rrr[p]+rru[p]+rur[p]+ruu[p]+urr[p]
+uru[p]+uur[p]; /* C2a */

forall(p in jourParRoul) t[((3*p)+7) mod (3*roul)]+q[((3*p)+4) mod (3*roul)]+c[(3*p)+1] <= sum(l in 0..2)
(t[((3*p)+18+(1*3)) mod (3*roul)]+q[((3*p)+18+(1*3)) mod (3*roul)]+c[((3*p)+18+(1*3)) mod (3*roul)]); /*
C2b.1 */

forall(p in jourParRoul) t[((3*p)+6) mod (3*roul)]+q[((3*p)+3) mod (3*roul)]+c[(3*p)] <= sum(l in 0..2)
(t[((3*p)+20+(1*3)) mod (3*roul)]+q[((3*p)+20+(1*3)) mod (3*roul)]+c[((3*p)+20+(1*3)) mod (3*roul)]); /*
C2b.2 */

forall(p in jourParRoul) t[((3*p)+8) mod (3*roul)]+q[((3*p)+5) mod (3*roul)]+c[(3*p)+2] <= sum(l in 0..2)
(t[((3*p)+19+(1*3)) mod (3*roul)]+q[((3*p)+19+(1*3)) mod (3*roul)]+c[((3*p)+19+(1*3)) mod (3*roul)]); /*
C2b.3 */

/* fin contraintes inter-GPT */

/* debut contraintes sur les repos */

sum(k in jourParRoul) (2*rr[k]+2*ru[k]+2*ur[k]+3*rrr[k]+3*rru[k]+3*rur[k]+3*ruu[k]+3*urr[k]+ 3*uur[k]
+3*uru[k]+r[k]) = ftoi(floor(reposRoul)); /* C3b */

sum(k in jourParRoul) (2*rr[k]+ru[k]+ur[k]+3*rrr[k]+2*rru[k]+2*rur[k]+ruu[k]+2*urr[k]+uur[k]+uru[k]+r[k])
<= ftoi(ceil(RPRoul)); /* C3c.1 */

sum(k in jourParRoul) (2*rr[k]+ru[k]+ur[k]+3*rrr[k]+2*rru[k]+2*rur[k]+ruu[k]+2*urr[k]+uur[k]+uru[k]+r[k])
>= ftoi(floor(RPRoul)); /* C3c.2 */

sum(k in jourParRoul) (rr[k]+rrr[k]+rru[k]+urr[k]) >= nbSemaines*nbAgents; /* C3d */

sum(k in semParRoul) (sum(q in 0..1) rur[4+q*2+k*7]+sum(q in 0..1) rru[5+q+k*7]+sum(q in 0..1)
urr[4+q+k*7] + uur[4+k*7]+uru[5+k*7]+ruu[6+k*7] + sum(q in 0..2) rrr[4+q+k*7] + sum(q in 0..1)
rr[5+q+k*7] + ur[5+k*7]+ru[6+k*7]) >= ftoi(ceil(weRoul)); /* C3e */

```

```

sum(k in semParRoul) (rru[5+k*7]+urr[4+k*7] + sum(q in 0..1) rrr[4+q+k*7] + rr[5+k*7]) >=
ftoi(ceil(sadiRoul)); /* C3f */

/* fin contraintes sur les repos */

/* debut contrainte de charge */

forall(k in plaHorParCycle)
  sum(l in cycleParRoul) (sum(p in 0..2) t[(k-(3*p)+(21*1*nbSemaines)+(3*roul)) mod (3*roul)] + sum(p in
  0..3) q[(k-(3*p)+(21*1*nbSemaines)+(3*roul)) mod (3*roul)] + sum(p in 0..4) c[(k-(3*p)+(21*1*nbSemaines)
  +(3*roul)) mod (3*roul)]) + rs[k] = work[k]; /* C4 */

/* fin contrainte de charge */

/* debut contraintes pour reduire le temps de recherche */

diffMatSoir >= ftoi(ceil((nbRSFloat/2.0)-floor((nbRSFloat/2.0)))) - (sum(k in jourParCycle) rs[2+(3*k)]);

forall(p in jourParCycle) sum(k in 0..2) rs[(p*3)+k] <= nbAgents;

sum (k in plaHorParRoul) ((3*t[k])+(4*q[k])+(5*c[k])) = 7*nbSemaines*nbAgents - ftoi(floor(reposRoul));

sum(k in plaHorParCycle) rs[k] = nbRS;

/* fin contraintes pour reduire le temps de recherche */

/* debut contraintes sur variables additionnelles */

diffMatSoir >= sum(k in jourParCycle) (rs[(3*k)]-rs[(3*k)+1]); /* C5a */
diffMatSoir >= sum(k in jourParCycle) (rs[(3*k)+1]-rs[(3*k)]); /* C5b */

forall(p in jourParCycle) absVarRs[p] >= 7*nbSemaines*(sum(k in 0..2) rs[(3*p)+k]) - nbRS; /* C5c */
forall(p in jourParCycle) absVarRs[p] >= nbRS - 7*nbSemaines*(sum(k in 0..2) rs[(3*p)+k]); /* C5d */

forall(p in cycleParRoul)
  sadi[p] = sum(k in semParCycle) (rru[5+(k*7)+(p*nbSemaines*7)]+urr[4+(k*7)+(p*nbSemaines*7)]+sum(l
  in 0..1) rrr[4+l+(k*7)+(p*nbSemaines*7)]+rr[5+(k*7)+(p*nbSemaines*7)]); /* C5e */

forall(p in cycleParRoul) absVarSadi[p] >= nbAgents*sadi[p] - (sum(i in cycleParRoul) sadi[i]); /* C5f */
forall(p in cycleParRoul) absVarSadi[p] >= (sum(i in cycleParRoul) sadi[i]) - nbAgents*sadi[p]; /* C5g */

forall(p in cycleParRoul)
  we[p] = sum(k in semParCycle) (sum(l in 0..1) rur[4+(1*2)+(k*7)+(p*nbSemaines*7)]+sum(l in 0..1)
  rru[5+l+(k*7)+(p*nbSemaines*7)]+sum(l in 0..1) urr[4+l+(k*7)+(p*nbSemaines*7)]+
  uur[4+(k*7)+(p*nbSemaines*7)]+uru[5+(k*7)+(p*nbSemaines*7)]+ruu[6+(k*7)+(p*nbSemaines*7)]+sum(l in 0..2) rrr[4+l+
  (k*7)+(p*nbSemaines*7)] +sum(l in 0..1) rr[5+l+(k*7)+(p*nbSemaines*7)]+ur[5+(k*7)+(p*nbSemaines*7)]
  +ru[6+(k*7)+(p*nbSemaines*7)]); /* C5h */

forall(p in cycleParRoul) absVarWe[p] >= nbAgents*we[p] - (sum(i in cycleParRoul) we[i]); /* C5i */
forall(p in cycleParRoul) absVarWe[p] >= (sum(i in cycleParRoul) we[i]) - nbAgents*we[p]; /* C5j */

/* fin contraintes sur variables additionnelles */

/* fin contraintes */

```

```
};
```

## 5.11 Listing du fichier « roulements.dat »

```
nbSemainesFloat = 2.0;
nbAgentsFloat = 4.0;
nbReposParAn = 132.0;
nbRPParAn = 118.0;
B=10000001;//Reserves de nuits 1000000
C=100;//Ecart quotidien de reserve 10000
D=1000;//Ecart Soirees-Matinees de reserve 1000
E=12;//Ecart Samedi/Dimanche par rapport a la moyenne 100
F=13;//Ecart Week-Ends par rapport a la moyenne 100
G=1000000;//Nombre repos isoles 100000
```