

Titre: Affectation de cellules aux commutateurs dans les réseaux mobiles de troisième génération
Title: de troisième génération

Auteur: Mamadou Moudjitalba Diallo
Author:

Date: 2004

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Diallo, M. M. (2004). Affectation de cellules aux commutateurs dans les réseaux mobiles de troisième génération [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/7362/>
Citation:

Document en libre accès dans PolyPublie Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7362/>
PolyPublie URL:

Directeurs de recherche: Samuel Pierre, & Ronald Beaubrun
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

AFFECTATION DE CELLULES AUX COMMUTATEURS DANS
LES RÉSEAUX MOBILES DE TROISIÈME GÉNÉRATION

MAMADOU MOUDJITABA DIALLO
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)

AVRIL 2004



Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-01306-0

Our file Notre référence

ISBN: 0-494-01306-0

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

AFFECTATION DE CELLULES AUX COMMUTATEURS DANS
LES RÉSEAUX MOBILES DE TROISIÈME GÉNÉRATION

présenté par : DIALLO Mamadou Moudjitaba

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. QUINTERO Alejandro, Doct., président

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. BEAUBRUN Ronald, Ph.D., membre et codirecteur de recherche

M. GALINIER Philippe, Doct., membre

DÉDICACE

à mes parents...

REMERCIEMENTS

Tout d'abord, je tiens à remercier DIEU TOUT PUISSANT, miséricordieux, très miséricordieux, pour m'avoir permis de vivre sur cette terre et d'avoir veillé sur moi et sur mon entourage.

À mes parents bien aimés : Abdoulaye, mon père et Sadou, ma mère, pour m'avoir donné, grâce à DIEU TOUT PUISSANT, la vie et pour m'avoir guidé dans celle-ci. Spécialement à ma mère pour avoir sacrifié toute vie active pour rester à la maison et s'occuper de nous. « Bravo ! », toute la famille t'est très reconnaissante et nous t'avons tous promis qu'un jour tu verras les récompenses et ce jour est arrivé aujourd'hui. J'aimerais aussi exprimer ma profonde gratitude tout spécialement à mes frères, mes sœurs, mes belles sœurs et toutes les personnes que j'ai pu rencontrer jusque là dans ma vie pour leur amour, leur support et leur encouragement continual tout au long de ce travail et de mon séjour à Montréal.

Je voudrais remercier mon directeur de recherche, monsieur Samuel Pierre, professeur titulaire au département de génie informatique et directeur du Laboratoire de Réseautique et Informatique Mobile (LARIM) de l'École Polytechnique de Montréal, pour la qualité de son encadrement, sa compréhension, ses suggestions, sa disponibilité, sa patience, son encouragement et son support moral et financier tout au long de ce mémoire.

Je voudrais remercier aussi mon codirecteur de recherche, monsieur Ronald Beaubrun, professeur adjoint au département d'informatique et de génie logiciel de l'Université Laval, pour la qualité de son encadrement, sa disponibilité, sa patience, son encouragement et tout son soutien.

Je tiens aussi à exprimer ma sincère gratitude aux membres du LARIM pour leur soutien, leur humour. Bref, ils ont contribué à garder une bonne ambiance de travail, ce qui m'a été d'un grand support moral pour passer à travers les temps durs.

RÉSUMÉ

Depuis le début des années 80, on a assisté avec succès au déploiement des réseaux de communications personnelles dans les différentes régions du monde. La complexité des infrastructures ainsi que le souci de fournir une certaine qualité de service font que les opérateurs de ces réseaux investissent beaucoup de temps et d'argent dans la planification. Cette planification, jugée comme un problème complexe, est souvent divisée en plusieurs sous-problèmes dont la résolution peut être parfois très complexe. Un de ces sous-problèmes est l'affectation de cellules aux commutateurs. C'est un problème qui a été largement traité dans la première et deuxième génération de réseaux mobiles. Il consistait à trouver un patron d'affectation d'un ensemble n de cellules à un ensemble m de commutateurs tout en minimisant une fonction de coût intégrant les coûts de liaisons entre les équipements et les coûts de relève entre les cellules. Mais, l'introduction de la technologie de troisième génération et d'une nouvelle architecture de réseau avec des services de commutations de paquets (vidéophonie, Internet etc.), qui s'ajoutent aux services de commutations de circuits déjà offerts (voix), ont contribué à rendre le problème d'affectation des cellules encore plus difficile à résoudre.

Dans ce mémoire, nous proposons un modèle de formulation et une méthode de résolution du problème d'affectation des cellules aux commutateurs dans les réseaux mobiles de troisième génération. Pour y parvenir, nous avons tout d'abord apporté quelques précisions sur les concepts de *relève simple* et de *relève complexe* entre les cellules et sur le concept de *domiciliation* lors de l'affectation des cellules. Ensuite, nous avons formulé le problème d'affectation des cellules aux commutateurs sous forme de problème de minimisation d'une fonction de coût sujet à un ensemble de contraintes. Pour ce faire, nous avons considéré deux sous-affectations : la première qui consiste à affecter un ensemble n de cellules (Nœuds B) à un ensemble r de contrôleurs radio (RNC) et la deuxième qui consiste à affecter les r contrôleurs radio à un ensemble m de commutateurs MSC (prenant en charge le trafic de voix) et à un ensemble s de commutateurs SGSN (prenant en charge le trafic de données).

La fonction objectif à minimiser est constituée des coûts de liaison entre les équipements du réseau et des coûts de relève entre les cellules. Parmi les contraintes à satisfaire, il y a les contraintes de capacités sur les commutateurs (MSC et SGSN) et sur les RNC ainsi que les contraintes d'affectation unique des Nœuds B aux RNC et des RNC aux MSC et aux SGSN. Pour résoudre le problème, nous avons proposé une adaptation de la méta heuristique Recherche Taboue. Cette adaptation consiste essentiellement en la définition de mouvements permettant le déplacement lors de l'exploration de l'espace des solutions. Chaque mouvement est sanctionné par un gain qui nous permet d'évaluer l'amélioration apportée par ce mouvement sur le coût de la solution courante.

Après simulation de la méthode, les résultats obtenus montrent que l'adaptation faite des différents mécanismes de mémoire de la Recherche Taboue fonctionne parfaitement. En effet, le mécanisme de mémoire à long terme qui est le dernier à être appliqué arrive à améliorer dans un ordre de 20 à 40% le coût de la meilleure solution trouvée par le mécanisme de mémoire à court terme (premier mécanisme à être appliqué). Également, pour l'ensemble des séries de tests effectués, les résultats de comparaison des coûts trouvés par notre méthode avec une évaluation de la borne inférieure de chaque problème nous montrent en général, des écarts ne dépassant pas 30% et dans certains cas l'écart tombe en dessous de 1% pour des problèmes de petite taille.

ABSTRACT

The design of wireless networks is a difficult task that is generally decomposed into many sub problems, most of them still very hard to solve. Among those sub-problems, we have the assignment of cells to switches. In the first and second generation of wireless networks, the assignment of cells to switches consist of finding an optimal assignment of a set of n cells to a set of m switches, that minimizes the overall cost composed of handoffs cost between cells, and the cost of links between equipments. This assignment problem is known as being NP-hard; so forth heuristic methods are often used to solve it. But the introduction of third generation wireless networks with new equipment, new data services and new network architecture adds more challenges to the network planners some of them related to the cell assignment problem already known as NP-hard.

The main objective of this thesis is to study the whole problematic of cell assignment to switches in the 3G wireless networks. For that purpose, we developed a mathematic model for the assignment of cells to switches. To do so, we divided the main problem of assigning cells to switches in to two sub-assignment problems. The first one aim to assign a set of n cells to a set of r radio controllers (RNC) and the second one aim to assign those radio controllers at the same time to a set of k voice switches (MSC) and a set of s data switches (SGSN). The mathematic model proposed is formulated as an optimization problem subject to some constraints. The objective function is composed of two parts. The first one is the cost of links between cells to radio controllers and links between radio controllers to voice switches and to data switches. The second part is the cost of handoffs between cells. The constraints are mostly the capacity limitations on the switches and the radio controllers and the constraint of unique assignment of cells to radio controllers and radio controllers to the two types of switches.

To solve the problem, we proposed an adaptation of the meta-heuristic tabu search. The adaptation consists of defining moves we can use when exploring the set of

possible solutions. Each move has an associated gain that helps us assess on the degree of improvement of the current solution's cost.

Simulation results show that the long term memory mechanism of tabu search which is the last one applied, improve by 20 to 40% the cost of the best solution obtained by the short term memory mechanism (first mechanism to be applied). Also, we conducted a series of test on various 3G networks types and for each problem we evaluated a low boundary with which we compared the results obtained by our method. Those comparisons show that our results are in general within 30% of the low boundary sometimes that distance drops under 1% for smaller problems, which is very good.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS.....	v
RÉSUMÉ	vi
ABSTRACT.....	viii
TABLE DES MATIÈRES	x
LISTE DES FIGURES	xiii
LISTE DES TABLEAUX.....	xvi
LISTE DES SIGLES ET ABBRÉVIATIONS	xvii
LISTE DES ANNEXES	xix
CHAPITRE 1 : INTRODUCTION.....	1
1.1 Définitions et concepts de base.....	1
1.2 Éléments de la problématique.....	6
1.3 Objectifs de recherche.....	10
1.4 Principales contributions.....	11
1.5 Plan du mémoire	11
CHAPITRE 2 : AFFECTATION DE CELLULES AUX COMMUTATEURS DANS LES RÉSEAUX MOBILES DE DEUXIÈME GÉNÉRATION	13
2.1 Formulation du problème.....	13
2.1.1 Cas de la domiciliation simple.....	14
2.1.2 Cas de la domiciliation double.....	18
2.2 Méthodes de résolution	21
2.2.1 Méthodes classiques.....	21
2.2.2 Recueil simulé	22
2.2.3 Programmation par contraintes	23
2.2.4 Algorithmes génétiques	25
2.2.5 Recherche taboue	28

2.2.6	Algorithmes memétiques	31
2.3	Problématique des réseaux mobiles 3G	34
2.3.1	Acteurs principaux dans la normalisation des réseaux mobiles 3G.....	34
2.3.2	Architecture des réseaux mobiles 3G	35
2.3.3	Planification des réseaux 3G.....	39
CHAPITRE 3 : AFFECTATION DES CELLULES DANS LES RÉSEAUX MOBILES DE TROISIÈME GÉNÉRATION		45
3.1	Concepts de base.....	45
3.1.1	Choix de l'architecture du réseau	45
3.1.2	Concepts de relève simple et de relève complexe	47
3.1.3	Concepts de domiciliation simple et de domiciliation double.....	48
3.2	Formulation du problème.....	49
3.2.1	Hypothèses	50
3.2.2	Formulation mathématique proposé pour la domiciliation simple	50
3.2.3	Analyse de complexité du modèle mathématique	58
3.3	Adaptation de la Recherche Taboue	60
3.3.1	Principe de fonctionnement	61
3.3.2	Génération de la solution initiale	62
3.3.3	Mémoire à court terme.....	66
3.3.4	Mémoire à moyen terme	75
3.3.5	Mémoire à long terme.....	78
CHAPITRE 4 : IMPLÉMENTATION ET ANALYSE DES RÉSULTATS		79
4.1	Détails d'implémentation.....	79
4.1.1	Modélisation du trafic	79
4.1.2	Format des fichiers d'entrée et de sortie	82
4.1.3	Implémentation des principaux algorithmes	83
4.1.4	Structures de données utilisées	87
4.1.5	Environnement matériel et logiciel	92
4.2	Plan d'expériences	92

4.2.1	Comportement de la composante de mémoire à court terme.....	93
4.2.2	Comportement de la composante de mémoire à moyen terme	94
4.2.3	Comportement de la composante de mémoire à long terme.....	95
4.2.4	Comportement général de la méthode	95
4.2.5	Comparaison avec une borne inférieure	95
4.3	Simulations et analyse des résultats.....	96
4.3.1	Comportement de la composante de mémoire à court terme.....	97
4.3.2	Comportement de la composante de mémoire à moyen terme	104
4.3.3	Comportement de la composante de mémoire à long terme.....	107
4.3.4	Comportement général de la méthode	108
4.3.5	Comparaison des résultats avec une borne inférieure.....	118
	CHAPITRE 5 : CONCLUSION	124
5.1	Synthèse des travaux.....	124
5.2	Limitations des travaux.....	126
5.3	Indications de recherche future	127
	BIBLIOGRAPHIE.....	129
	ANNEXES	133

LISTE DES FIGURES

Figure 1.1 Architecture de base des réseaux mobiles 2G	3
Figure 1.2 Phénomène de relève dans les réseaux 2G	4
Figure 1.3 Architecture typique envisagée pour les réseaux mobiles 3G (UMTS)	5
Figure 1.4 Architecture typique du réseau d'accès (UTRAN)	6
Figure 1.5 Schéma d'affectation dans le 2G	8
Figure 1.6 Schéma désiré d'affectation dans le 3G	9
Figure 2.1 Représentation non binaire d'un chromosome (domiciliation simple)	27
Figure 2.2 Représentation non binaire d'un chromosome (domiciliation double)	27
Figure 2.3 Algorithme général de la recherche taboue	30
Figure 2.4 Fonctionnement général des AM.....	33
Figure 2.5 Architecture typique d'un réseau CDMA2000.....	38
Figure 3.1 Architecture typique d'un réseau UMTS	46
Figure 3.2 Interconnexion du réseau UMTS en relève simple	47
Figure 3.3 Interconnexion du réseau UMTS en relève complexe	48
Figure 3.4 Pseudo code modifié de génération de la solution initiale	64
Figure 3.5 Exemple de calcul du gain $\mathbf{G}_s^1(a, b)$	69
Figure 3.6 Exemple de calcul du gain $\mathbf{G}_s^2(b, c, d)$	72
Figure 4.1 Diagramme de fonctionnement du mécanisme de diversification	84
Figure 4.2 Diagramme de fonctionnement du mécanisme de mémoire à court terme	85
Figure 4.3 Diagramme de fonctionnement du mécanisme de mémoire à moyen terme... <td>86</td>	86
Figure 4.4 Diagramme des classes principales	88
Figure 4.5 Effet de la taille de la LT sur les solutions obtenues.....	98
Figure 4.6 Rapport de taille entre les deux sous-listes taboues	100
Figure 4.7 Effet du délai de déclenchement du mécanisme de « rappel ».....	101
Figure 4.8 Effet du poids de la sanction de la violation du trafic de voix par rapport au trafic de données	102

Figure 4.9 Effet de la variation de l'intensité maximale du mécanisme de « rappel »	103
Figure 4.10 Effet du délai de déclenchement des mouvements de réaffectation.....	105
Figure 4.11 Effet de la taille de la région d'intensification	106
Figure 4.12 Effet du nombre de redémarrage sur les solutions obtenues	108
Figure 4.13 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 1)	109
Figure 4.14 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 2)	110
Figure 4.15 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 3)	110
Figure 4.16 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 4)	111
Figure 4.17 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 5)	111
Figure 4.18 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 6)	112
Figure 4.19 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 7)	112
Figure 4.20 Pourcentage de solutions faisables pour chaque mécanisme (série n° 1)....	114
Figure 4.21 Pourcentage de solutions faisables pour chaque mécanisme (série n° 2)....	114
Figure 4.22 Pourcentage de solutions faisables pour chaque mécanisme (série n° 3)....	115
Figure 4.23 Pourcentage de solutions faisables pour chaque mécanisme (série n° 4)....	115
Figure 4.24 Pourcentage de solutions faisables pour chaque mécanisme (série n° 5)....	116
Figure 4.25 Pourcentage de solutions faisables pour chaque mécanisme (série n° 6)....	116
Figure 4.26 Pourcentage de solutions faisables pour chaque mécanisme (série n° 7)....	117
Figure B.1 Temps moyen d'exécution (série 1).....	136
Figure B.2 Temps moyen d'exécution (série 2).....	136
Figure B.3 Temps moyen d'exécution (série 3).....	137
Figure B.4 Temps moyen d'exécution (série 4).....	137

Figure B.5 Temps moyen d'exécution (série 5).....	138
Figure B.6 Temps moyen d'exécution (série 6).....	138
Figure B.7 Temps moyen d'exécution (série 7).....	139
Figure C.1 Exemple de fichier « .DON »	140
Figure C.2 Exemple de fichier « .CAP »	141
Figure C.3 Exemple de fichier «Result.ras »	141
Figure C.4 Exemple de fichier «TracePrg.tra »	142

LISTE DES TABLEAUX

Tableau 3.1 Résumé de la formulation du problème	59
Tableau 3.2 Coûts de liaison entre les Nœuds B et les RNC	62
Tableau 3.3 Coûts de liaison entre les RNC et les MSC	62
Tableau 3.4 Coûts de liaison entre les RNC et les SGSN.....	63
Tableau 3.5 Résultats de l'affectation des 10 Nœuds B aux 6 RNC	65
Tableau 3.6 Résultats de l'affectation des 6 RNC aux 3 MSC.....	66
Tableau 3.7 Résultats de l'affectation des 6 RNC aux 3 SGSN	66
Tableau 4.1 Cas tests utilisés pour le comportement des composantes de mémoire	96
Tableau 4.2 Séries de tests utilisées pour le comportement général de la méthode	109
Tableau 4.3 Écart par rapport à la borne inférieure pour la série n° 1	122
Tableau 4.4 Écart par rapport à la borne inférieure pour la série n° 2	122
Tableau 4.5 Écart par rapport à la borne inférieure pour la série n° 3	122
Tableau 4.6 Écart par rapport à la borne inférieure pour la série n° 4	122
Tableau 4.7 Écart par rapport à la borne inférieure pour la série n° 5	122
Tableau 4.8 Écart par rapport à la borne inférieure pour la série n° 6	122
Tableau 4.9 Écart par rapport à la borne inférieure pour la série n° 7	123
Tableau A.1 Série de tests n° 1	133
Tableau A.2 Série de tests n° 2	133
Tableau A.3 Série de tests n° 3	134
Tableau A.4 Série de tests n° 4	134
Tableau A.5 Série de tests n° 5	135
Tableau A.6 Série de tests n° 6	135
Tableau A.7 Série de tests n° 7	135

LISTE DES SIGLES ET ABBRÉVIATIONS

2G	: Réseaux mobiles de deuxième génération
3G	: Réseaux mobiles de troisième génération
AG	: Algorithmes Génétiques
AM	: Algorithmes Memétiques
BHCA	: Busy Hour Call Attempts
BSC	: Contrôleur de station de base (Base Station Controller)
BTS	: Station de Base (Base Transceiver Station)
GGSN	: Gateway GPRS Support Node
GMSC	: Gateway MSC
GPRS	: GSM Packet Radio Service
LT1	: Sous-liste Taboue d'affectation des Nœuds B aux RNC
LT2	: Sous-liste Taboue d'affectation des RNC aux MSC et aux SGSN
MSC	: Commutateur (Mobile Switching Center)
NB	: Nœud B
RCP	: Réseau de Communications Personnelles
RNC	: Radio Network Controller
RT	: Recherche Taboue
SGSN	: Commutateur (Serving GPRS Support Node)
UML	: Unified Modeling Language
UMTS	: Universal Mobile Telecommunications System

UTRAN : UMTS Terrestrial Radio Access Network

VLR : Visitor Location Register

VLSI : Very Large Scale Integrated

LISTE DES ANNEXES

ANNEXE A : COMPOSITION DES SÉRIES DE TESTS.....	133
ANNEXE B : GRAPHES DU TEMPS MOYEN D'EXÉCUTION.....	136
ANNEXE C : FICHIERS TYPIQUES D'ENTRÉE.....	140

CHAPITRE 1

INTRODUCTION

Depuis l'apparition des réseaux de télécommunications personnelles, c'est durant la dernière décennie qu'on a recensé la plus grande évolution en terme d'intégration de nouvelles technologies d'accès et de nouveaux services personnalisés. En effet, si dans les tout premiers réseaux mobiles, les usagers se contentaient de se servir de leur téléphone mobile juste pour parler, aujourd'hui les besoins en communications sont plus variés et intègrent de plus en plus l'échange de données et d'images à un débit assez élevé à même le téléphone mobile. On parle de plus en plus d'Internet et de multimédia n'importe où et n'importe quand sur le téléphone cellulaire. Mais l'intégration d'applications de type multimédia sur le téléphone cellulaire, en plus du trafic de voix déjà bien établi, ne va pas sans poser des défis supplémentaires aux planificateurs de réseaux de communications personnelles. Dans ce mémoire, nous traitons un des aspects de cette planification, soit la problématique de l'affectation des cellules aux différents commutateurs dans les réseaux mobiles de troisième génération. Dans ce chapitre d'introduction, nous commencerons tout d'abord par présenter quelques définitions et concepts de base sur les réseaux de communications personnelles et les éléments de la problématique qui motivent notre recherche. Ensuite, nous résumerons nos objectifs de recherche, les résultats attendus et notre contribution. Nous terminerons en présentant les grandes lignes du mémoire.

1.1 Définitions et concepts de base

Un *réseau cellulaire* est un réseau de communication dont l'étendue géographique de sa couverture est divisée en de petites surfaces communément appelées *cellules*. Celles-ci sont souvent représentées par des hexagones dont le rayon varie de quelques centaines de mètres à quelques kilomètres. C'est un réseau qui offre des services personnalisés, comme la transmission de voix, de messages numériques, de textes, de

courrier vocal, bientôt de services multimédias et une connexion à Internet. C'est pourquoi on l'appelle aussi *réseau de communications personnelles* (RCP) ou encore *réseau mobile* par abus de langage, puisque les usagers ont accès à leurs services partout et n'importe quand sur une unité mobile. Selon la technologie supportée et les services offerts, les réseaux cellulaires sont classés en *génération*. On a les tout premiers réseaux qui correspondent au 1G (*première génération*), ensuite ont suivi les réseaux 2G (*deuxième génération*) et aujourd'hui on est en train de concevoir les réseaux 3G (*troisième génération*).

Dans les réseaux cellulaires de *première* et de *deuxième génération*, chaque cellule est desservie par une *station de base* ou *BTS* (*Base Transceiver Station*) qui intègre une antenne assurant la transmission radio et la signalisation à l'intérieur de la cellule. Les stations de base sont à leur tour reliées à des *contrôleurs de station de base* ou *BSC* (*Base Station Controller*) qui gèrent les ressources radio ou bandes passantes des stations de base associées. Le regroupement formé par les stations de base (BTS) et les contrôleurs de stations de base (BSC) constitue le *sous-système radio* qui sert donc *d'interface radio* entre chaque unité mobile et le reste du réseau. Le *sous-système réseau*, quant à lui, est constitué de commutateurs appelés *MSC* (*Mobile service Switching Center*) qui sont installés à l'intérieur de quelques-unes des cellules choisies de manière stratégique. Ces commutateurs sont chargés entre autres de la gestion du trafic provenant des stations de base et des mécanismes de relève entre les cellules; reliés entre eux, ils constituent la partie fixe de l'infrastructure réseau souvent appelée *réseau central* ou *réseau cœur* (*Core Network*). La Figure 1.1 présente l'architecture typique d'un réseau 2G.

De par leur fonction, les nœuds de commutation que sont les MSC représentent des points centraux. Lorsqu'un utilisateur est à l'intérieur du réseau tel qu'ilustré sur la Figure 1.2 [31], son terminal est raccordé à une des stations de base positionnée dans une cellule en fonction de la puissance du signal qu'il reçoit. Lors du déplacement de l'utilisateur, si cette puissance tombe au-dessous d'un certain seuil, certaines opérations sont mises en œuvre pour relayer sa communication par une nouvelle cellule.

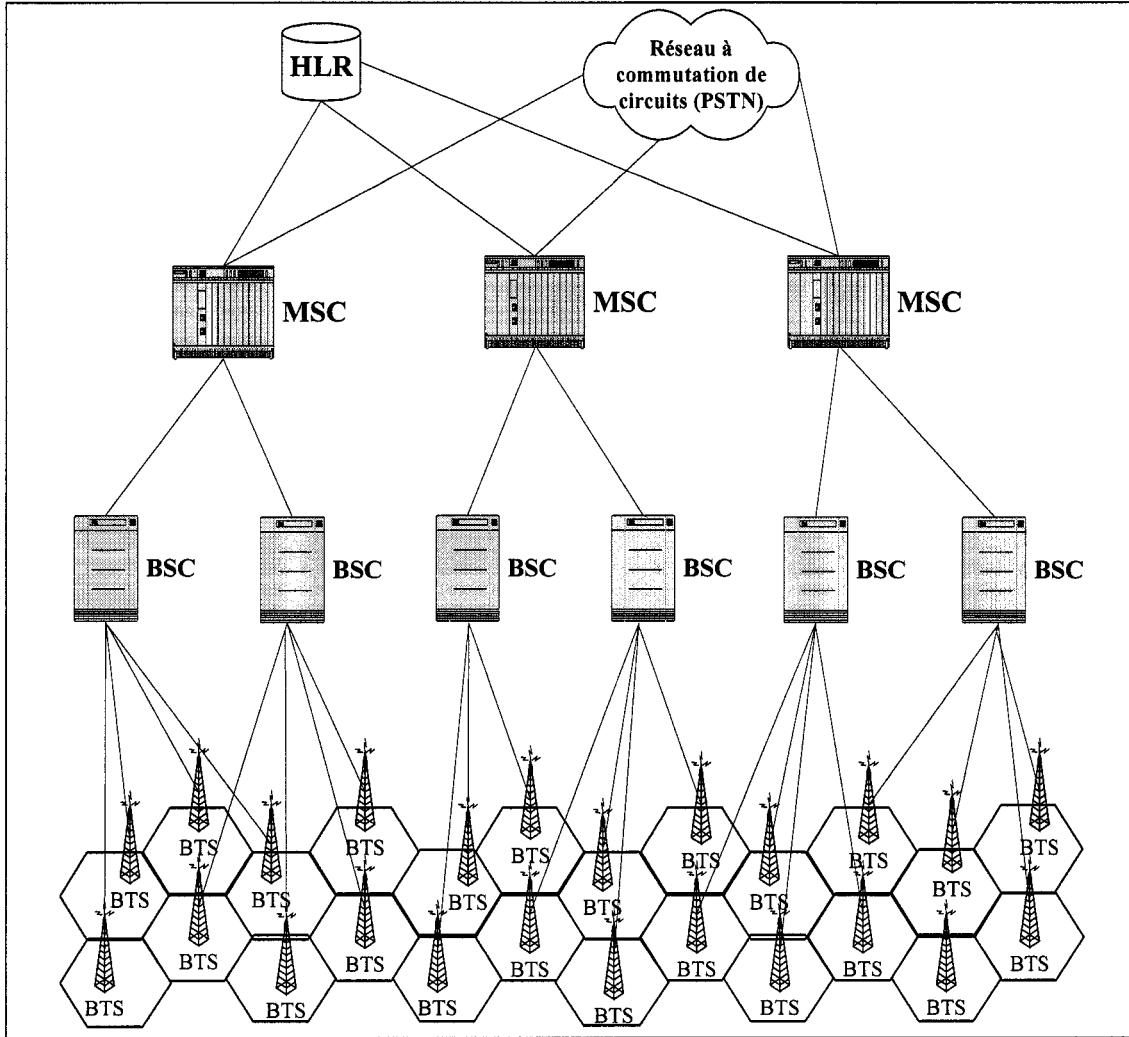


Figure 1.1 Architecture de base des réseaux mobiles 2G

Ce mécanisme est connu sous le nom de *relève* et doit s'effectuer de manière transparente à l'usager mobile. Selon la Figure 1.2 [31], lorsque l'usager passe de la cellule A à la cellule B, la *relève* est dite *simple* (*simple handoff*) car elle n'implique qu'un seul et même commutateur. Ainsi, aucune mise à jour n'est nécessaire. Par contre, lors du passage de l'usager de la cellule A à la cellule C, la *relève* est dite *complexe*

(*complex handoff*) car elle s'accompagne d'un changement de *MSC* : des mises à jour sont alors nécessaires et elles entraînent une forte consommation des ressources du réseau.

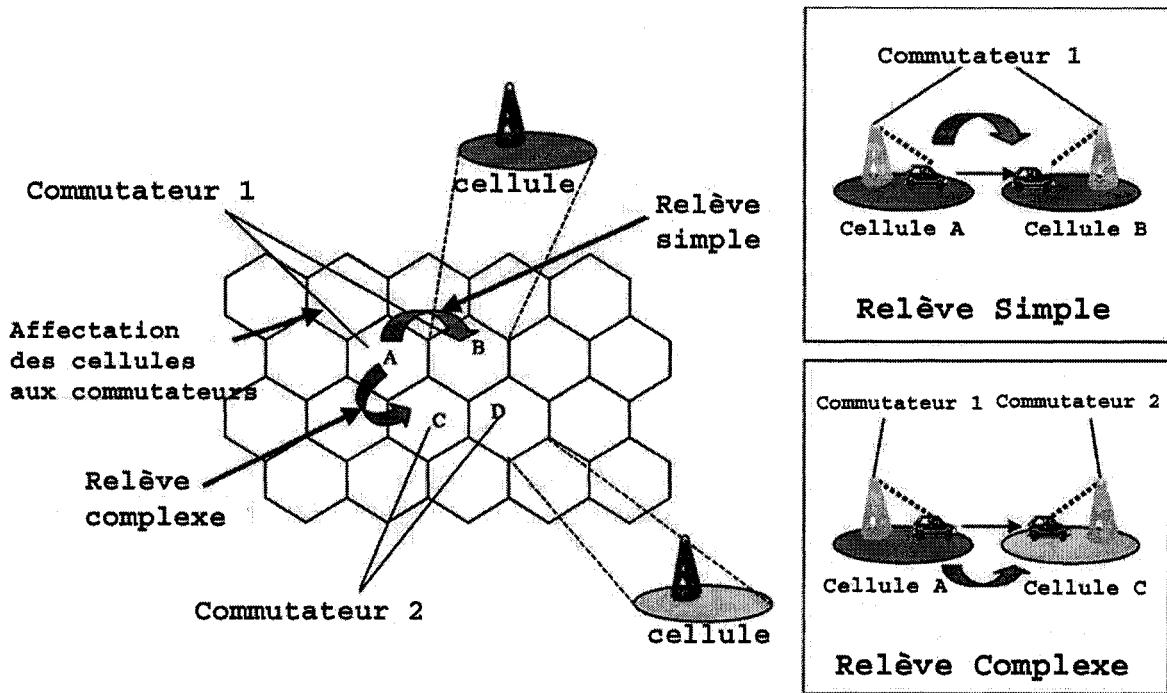


Figure 1.2 Phénomène de relève dans les réseaux 2G

Les réseaux de *troisième génération* quant à eux, se proposent de regrouper les divers environnements mobiles et incompatibles en une infrastructure capable d'offrir, avec une bonne qualité de service, toute une gamme de services de télécommunications à grande échelle. Tel qu'illustré à la Figure 1.3, il existe, pour ce qui est de l'architecture, une certaine similarité avec les réseaux mobiles 2G. La grande différence est l'apparition de nouveaux équipements qui prennent en charge le trafic de données. Il s'agit des *SGSN* (*Serving GPRS Support Node*) qui jouent le rôle de commutateurs, tout comme les *MSC* dans les réseaux 2G. Mais contrairement aux *MSC* qui sont dédiés à la voix (en commutation de circuits), les *SGSN* s'occupent des données (la commutation de paquets)

et sont reliés à des *GGSN* (*Gateway GPRS Support Node*) qui servent de pont (Gateway) avec les réseaux externes de données (ex. Internet).

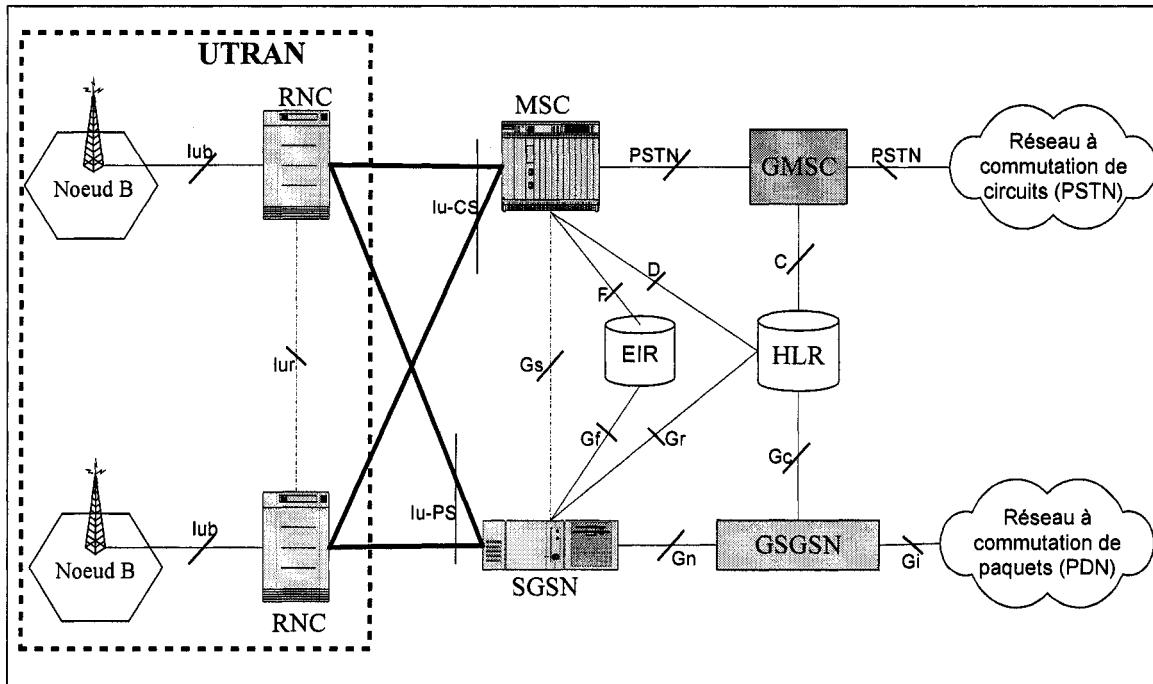


Figure 1.3 Architecture typique envisagée pour les réseaux mobiles 3G (UMTS)

Il faut remarquer aussi que les MSC sont toujours présents et cohabitent avec les SGSN. Les BTS des réseaux 2G sont remplacés par des « nœuds B » (de l'anglais *Node B*), alors que les BSC, quant à eux, sont remplacés par des *RNC* (*Radio Network Controller*). Les RNC constituent des équipements intermédiaires entre les commutateurs et les nœuds B. Ils contrôlent le trafic radio d'un certain nombre de nœuds B à l'intérieur d'une région bien définie. L'ensemble constitué par les Nœuds B et les RNC représente le sous-système radio (ou encore réseau d'accès) appelé *UTRAN* (*UMTS Terrestrial Radio Access Network*) tel que représenté à la Figure 1.4.

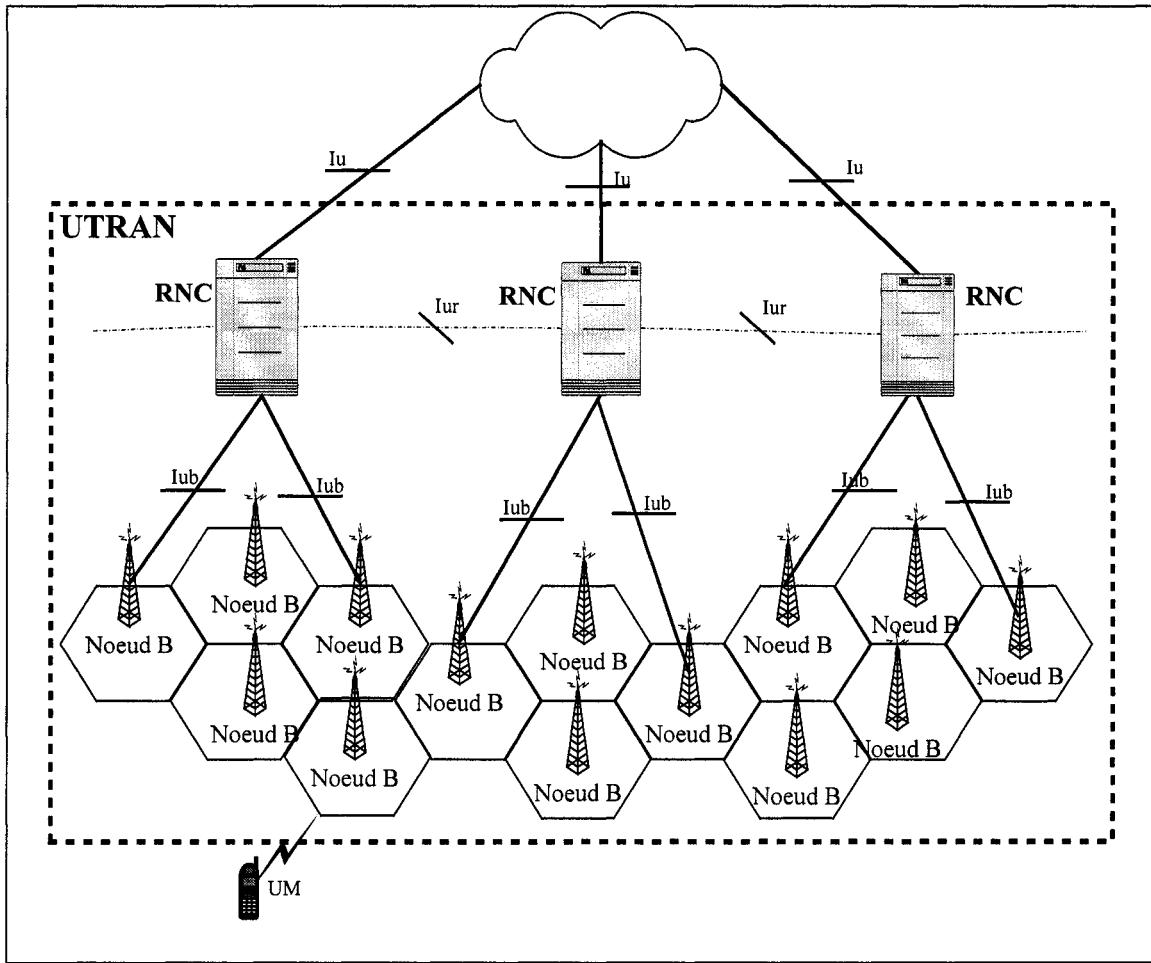


Figure 1.4 Architecture typique du réseau d'accès (UTRAN)

1.2 Éléments de la problématique

Chaque relève complexe met à contribution des ressources du réseau : mise à jour des données de localisation dans les bases de données, utilisation de protocoles de communications entre *MSC*, etc. Ainsi, des mécanismes ont été mis en place pour réduire au mieux les coûts qu'une telle opération engendre. Un des moyens consiste à optimiser l'affectation de cellules aux commutateurs. Dans ce cas, il importe de tenir compte des fréquences des relèves entre les différentes cellules du réseau. Ainsi, deux cellules

échangeant des quantités considérables d'informations seront affectées à un même commutateur pour minimiser le coût de relève.

Les travaux antérieurs ont démontré que le problème d'affectation des cellules aux commutateurs n'est pas un problème facile à résoudre. En effet, pour un réseau constitué de n cellules et de m commutateurs, l'utilisation d'un algorithme à énumération exhaustive nécessiterait l'examen de m^n combinaisons. Cela prendrait un temps fou pour trouver une bonne solution et peut facilement mener à une explosion combinatoire surtout lorsque le nombre de cellules et de commutateurs est grand. C'est un problème qui s'apparente beaucoup à certains types de problèmes bien connus en recherche opérationnelle, tels le partitionnement de graphes et la localisation d'entrepôts ou de concentrateurs. Le problème de localisation de p concentrateurs fixes (p -fixed hubs location) a été introduit dans [26] et [28] et consiste à affecter un ensemble N de nœuds à un ensemble H de concentrateurs dont les positions sont fixes et connues. Par analogie aux réseaux mobiles, les nœuds correspondent aux cellules du réseau mobile et les concentrateurs aux commutateurs.

Le problème de partitionnement de graphes, quant à lui, survient souvent en VLSI dans la disposition des composants et en calcul parallèle où on doit répartir des calculs entre un nombre donné de processeurs [17], [6], [25]. Il consiste, étant donné un graphe constitué de nœuds reliés par des arcs avec un poids sur chaque arc, à diviser l'ensemble N des nœuds du graphe en des sous-ensembles de cardinalité inférieure à un nombre maximal donné, de manière à minimiser le coût total des coupes (qui est la somme des coûts des arcs ayant leurs extrémités dans des sous-ensembles différents). Par analogie avec les réseaux mobiles, les nœuds du graphe à partitionner représentent les cellules du réseau et les arcs du graphe reliant les nœuds sont assimilés à des liens fictifs entre les cellules afin de représenter les coûts de relève entre les cellules (poids de chaque arc). Et un commutateur est associé à un nœud stratégique dans chaque partition. Les problèmes de partitionnement de graphes et de localisation d'entrepôts sont connus comme étant des problèmes NP-difficiles [7], [25], [26]. Le problème d'affectation de cellules pouvant être ramené à un de ces problèmes est aussi un problème NP-difficile.

Jusque là, le problème d'affectation a été traité seulement pour les réseaux mobiles de *première* et *deuxième génération*. Et dans ces réseaux, les équipements qui interviennent ont été divisés en deux niveaux : un premier niveau qui est constitué des stations de bases situées à l'intérieur des cellules (une station de base par cellule) et un deuxième niveau constitué des commutateurs MSC tel qu'illustré à la Figure 1.5. Dans ce cas, le problème d'affectation se résume à trouver un patron d'affectation des cellules aux commutateurs MSC. Ainsi, tous les modèles mathématiques développés dans la littérature se basent sur les hypothèses comme quoi, dans un patron d'affectation, les cellules sont affectées directement aux commutateurs et qu'il y a qu'un seul type de commutateur qui est le MSC.

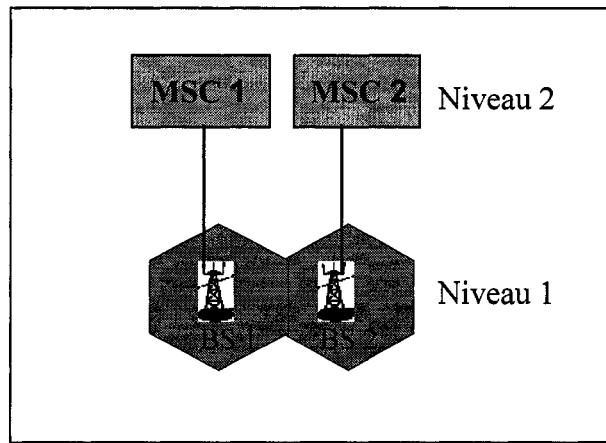


Figure 1.5 Schéma d'affectation dans le 2G

Maintenant, si l'on aborde le problème d'affectation dans le cas des réseaux mobiles de *troisième génération*, le problème ne se présentera pas de la même manière. En effet, vu l'architecture du réseau (Figure 1.3) et les fonctions des différents équipements, nous pouvons dégager trois niveaux d'emplacement des équipements : un premier niveau qui sera constitué des Nœuds B situés à l'intérieur des cellules (un Nœud B par cellule), un deuxième niveau constitué des contrôleurs RNC et un troisième niveau

constitué des deux types de commutateurs (MSC et SGSN). La Figure 1.6 présente une illustration des trois niveaux d'équipements pour un exemple d'affectation de cellules aux commutateurs dans un réseau mobile 3G. Ainsi, nous devons trouver des formules qui nous permettront d'affecter de manière optimale les Nœuds B aux contrôleurs RNC, puis d'affecter ces RNC aux commutateurs MSC et SGSN afin de pouvoir offrir simultanément aux unités mobiles les services de voix et de données : caractéristiques principales des réseaux de troisième génération.

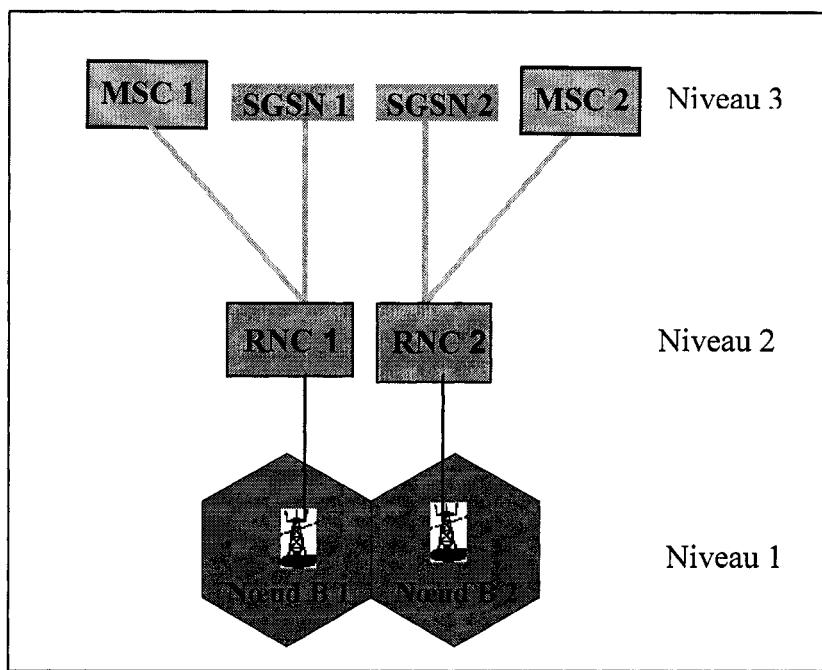


Figure 1.6 Schéma désiré d'affectation dans le 3G

En observant les schémas d'affectation pour le 2G (Figure 1.5) et pour le 3G (Figure 1.6), nous pouvons remarquer qu'il y a plusieurs différences. La première est qu'on a plus de niveaux d'équipement dans l'approche 3G par rapport au 2G. Par soucis de simplicité, les travaux effectués pour la résolution du problème d'affectation dans le 2G ont fait fi de la présence des contrôleurs de station de base (BSC) et les cellules ont

été affectées directement aux commutateurs MSC [2], [3], [13], [16], [20], [23], [31]. Dans le contexte de notre recherche, nous inclurons les RNC, non seulement parce que ce sont des équipements qui jouent un rôle important dans l'acheminement du trafic provenant des cellules, mais aussi parce que leur contrainte de capacité aura une incidence sur la recherche de la topologie optimale du réseau. La deuxième différence entre les deux schémas d'affectation est située au niveau des commutateurs. Dans les réseaux 3G, nous avons deux types de commutateurs : le MSC et le SGSN, alors que dans les réseaux 2G, nous avons simplement le MSC comme commutateur. Ces différences fondamentales font en sorte que les modèles développés jusqu'ici pour résoudre le problème d'affectation de cellules dans les réseaux 2G ne sont pas applicables pour la résolution du même problème dans les réseaux mobiles 3G.

Par ailleurs, certains concepts, comme le phénomène de relève décrits précédemment pour les réseaux 2G, ne s'interprètent pas de la même manière dans les réseaux 3G. Ainsi, pour le phénomène de relève, même si le principe reste le même, l'interconnexion du réseau et les équipements impliqués ne seront pas les mêmes. Il en résulte que les coûts de relève ne s'exprimeront pas de la même manière.

1.3 Objectifs de recherche

Dans ce mémoire, nous visons comme objectif principal l'étude de toute la problématique de l'affectation des cellules aux commutateurs dans les réseaux mobiles de troisième génération. Pour y parvenir, nous commencerons par choisir une architecture typique de réseaux cellulaires 3G parmi les architectures proposées dans la littérature sur laquelle nous nous baserons pour poser toutes nos hypothèses et contraintes. Ensuite, nous redéfinirons certains concepts de base des réseaux mobiles. Pour rechercher un patron d'affectation optimal des Nœuds B aux RNC et des RNC aux MSC et aux SGSN, nous élaborerons un modèle mathématique sous forme de problème d'optimisation qui explicitera la fonction objectif à minimiser en tenant compte de la diversité du trafic supporté par le réseau (voix, données etc.) et des contraintes associées liées entre autres

aux capacités des équipements RNC, MSC et SGSN. Après avoir étudié la complexité du modèle proposé, nous adapterons l'heuristique *recherche taboue* (RT) pour résoudre le problème. Les caractéristiques de robustesse et d'efficacité de cette heuristique à résoudre ce genre de problème ont motivé notre choix [16]. Nous définirons donc les paramètres d'adaptation des différents modules de la *recherche taboue*. Enfin, nous ferons une série de simulations de notre modèle mathématique sur plusieurs réseaux de test de taille et de topologies différentes afin de juger de la performance de l'adaptation faite de la RT et du bien fondé des hypothèses posées lors de la formulation du problème.

1.4 Principales contributions

Les principales contributions de ce mémoire sont au nombre de trois et consistent en :

- La définition des concepts et paramètres de base du problème d'affectation des cellules (Nœuds B) aux commutateurs dans les réseaux mobiles de troisième génération ;
- L'élaboration d'un modèle mathématique qui formule le problème d'affectation des cellules dans les réseaux mobiles 3G en intègre tous les nouveaux équipements, leurs contraintes, leurs capacités ainsi que leur interaction avec les équipements de la deuxième génération (qui continuent d'exister) ;
- L'adaptation de la *recherche taboue* pour la résolution du problème d'affectation de cellules aux commutateurs dans les réseaux mobiles de troisième génération.

1.5 Plan du mémoire

Ce mémoire est constitué de cinq chapitres. Le chapitre 2 présente une revue de littérature sur la problématique de l'affectation des cellules aux commutateurs dans le contexte des réseaux mobiles de deuxième génération, fait une revue sélective des méthodes utilisées pour résoudre ce problème et présente les problèmes de planification posés par les réseaux mobiles de troisième génération. Le chapitre 3 présente

l'architecture du réseau 3G sur laquelle se basera notre recherche, la formulation mathématique du problème d'affectation des nœuds B aux différents commutateurs et l'analyse de complexité du modèle mathématique résultant de cette formulation. Il présente aussi les paramètres de base de l'adaptation de la recherche taboue et les détails de fonctionnement de la méthode. Le chapitre 4 présente les détails d'implémentation du modèle ainsi qu'une analyse détaillée des résultats de simulation assortie de comparaisons avec une borne inférieure. Pour terminer, le chapitre 5 présente une conclusion qui résume les principaux aspects du mémoire, les résultats obtenus, les limitations de notre approche de résolution et les orientations de recherche.

CHAPITRE 2

AFFECTATION DE CELLULES AUX COMMUTATEURS

DANS LES RÉSEAUX MOBILES DE DEUXIÈME

GÉNÉRATION

Un des aspects fondamentaux de la conception et de la planification des réseaux cellulaires demeure l'affectation des cellules aux commutateurs. Le problème consiste à déterminer un patron d'affectation de cellules à des commutateurs dans le but de minimiser une certaine fonction de coût, tout en respectant un certain nombre de contraintes, dont une liée à la capacité des commutateurs. Dans ce chapitre, nous faisons une synthèse des travaux réalisés sur l'affectation des cellules aux commutateurs dans les réseaux cellulaires. Nous débuterons par la formulation actuelle du problème d'affectation pour les réseaux mobiles 2G. Par la suite, nous présenterons quelques méthodes connues de résolution de ce type de problème. Nous terminerons par une présentation des réseaux mobiles de troisième génération et des défis de planification qu'ils posent.

2.1 Formulation du problème

Dans les réseaux cellulaires de première et de deuxième génération, lors du processus d'affectation des cellules aux commutateurs (MSC), on a souvent le choix entre une domiciliation simple ou double pour les cellules. On parle de *domiciliation simple* lorsqu'une cellule est affectée à un seul commutateur. La domiciliation est *double* quand on affecte chaque cellule à deux commutateurs et ce, suivant les moments de la journée. Les deux commutateurs auxquels la cellule est reliée sont alors actifs de manière alternative, en fonction des périodes de la journée (un patron pour la matinée et un autre pour le soir). La capacité du MSC est généralement limitée par deux facteurs qui sont intimement liés : le nombre d'Erlangs et le nombre de BHCA (Busy Hour Call Attempts).

Alors que le Erlang reflète la capacité de commutation et la capacité des ports de communication du MSC, le BHCA quant à lui reflète la puissance de traitement du MSC. En général, le nombre d'Erlangs supportés peut être augmenté par l'ajout de cartes matérielles au MSC, alors que le nombre maximum de BHCA, pour un type donné de MSC, est pratiquement fixe. En conséquence, la véritable limite de capacité est déterminée par le BHCA [27].

Dans le processus d'affectation des cellules aux MSC, on parle souvent aussi de *patron*, de *patron d'appels* et de *patron d'affectation*. Le *patron* est un ensemble de données qui définit complètement l'état d'un attribut du réseau. Le *patron d'appels* définit le flot d'appels entre deux cellules quelconques du réseau. Le *patron d'affectation*, quant à lui, définit complètement le commutateur auquel est affectée chacune des cellules du réseau [16].

L'objectif principal du problème d'affectation est de trouver un patron d'affectation des cellules aux commutateurs qui minimise une certaine fonction de coût tout en respectant certaines contraintes. Nous formulerons le problème tel que présenté dans [20] dans le contexte des réseaux de communications personnelles de deuxième génération. Le problème sera formulé d'abord pour le cas de la domiciliation simple puis pour le cas de la domiciliation double. En même temps, nous montrerons qu'une adaptation directe du modèle de formulation pour chacune des domiciliations aux réseaux 3G n'est pas possible et qu'il sera nécessaire d'y apporter un certain nombre de modifications voire même une nouvelle formulation.

2.1.1 Cas de la domiciliation simple

Dans la domiciliation simple, chaque cellule du réseau est rattachée à un et un seul commutateur. Si on désigne par n le nombre de cellules à assigner et m le nombre de commutateurs du réseau. Supposons que l'emplacement des cellules et des commutateurs est fixe et connu. Soit x_{ik} une variable qui prend la valeur 1 si la cellule i ($i = 1, \dots, n$)

est reliée au commutateur k ($k=1, \dots, m$) et 0 autrement. Puisque chaque cellule doit être reliée à un et un seul commutateur, cela se traduit par la contrainte suivante :

$$\sum_{k=1}^m x_{ik} = 1 \quad \text{pour } i = 1, \dots, n \quad (2.1)$$

Soit c_{ik} le coût d'amortissement de la liaison entre la cellule i et le commutateur k , le coût total de câblage du réseau s'exprime de la manière suivante:

$$\sum_{i=1}^n \sum_{k=1}^m c_{ik} x_{ik}$$

Soit Cap_k la capacité d'un commutateur k mesurée en terme d'appels pris en charge par unité de temps et λ_i le nombre d'appels par unité de temps provenant de la cellule i . Ainsi, la capacité limitée des commutateurs impose la contrainte suivante :

$$\sum_{i=1}^n \lambda_i x_{ik} \leq Cap_k \quad \text{pour } k = 1, \dots, m \quad (2.2)$$

selon laquelle la charge totale de toutes les cellules assignées à un commutateur ne dépasse pas la capacité de ce dernier.

Pour formuler le coût total de relève, soit H_{ij} le coût par unité de temps d'une relève simple entre les cellules i et j impliquant un seul commutateur, et H'_{ij} le coût par unité de temps d'une relève complexe entre les cellules i et j ($i, j = 1, \dots, n$ avec $i \neq j$) impliquant deux commutateurs. H_{ij} et H'_{ij} sont proportionnels à la fréquence des relèves qui surviennent entre les cellules i et j .

Soient les variables z_{ijk} et y_{ij} suivantes:

$$z_{ijk} = x_{ik} x_{jk} \quad \text{pour } i, j = 1, \dots, n \text{ et } k = 1, \dots, m, \text{ avec } i \neq j. \quad (2.3)$$

$$y_{ij} = \sum_{k=1}^m z_{ijk} \quad \text{pour } i, j = 1, \dots, n, \text{ et } i \neq j \quad (2.4)$$

z_{ijk} et y_{ij} prennent la valeur 1 si les cellules i et j , avec $i \neq j$, sont reliées au même commutateur k et 0 autrement. Le coût total de relève par unité de temps s'exprime de la manière suivante :

$$\sum_{i=1}^n \sum_{j=1, j \neq i}^n H_{ij} (1 - y_{ij}) + \sum_{i=1}^n \sum_{j=1, j \neq i}^n H_{ij} y_{ij}$$

Ainsi, on obtient la fonction suivante du coût total par unité de temps à minimiser :

$$f = \sum_{i=1}^n \sum_{k=1}^m c_{ik} x_{ik} + \sum_{i=1}^n \sum_{j=1, j \neq i}^n H_{ij} (1 - y_{ij}) + \sum_{i=1}^n \sum_{j=1, j \neq i}^n H_{ij} y_{ij} \quad (2.5)$$

sujet aux contraintes (2.1) à (2.4).

Le premier terme de la fonction à minimiser est le coût de liaison, le deuxième terme prend en compte le coût des relèves complexes (impliquant deux commutateurs), et le troisième terme celui des relèves simples (impliquant un seul commutateur). Puisque la variable y_{ij} est une fonction quadratique des x_{ik} , on peut déduire que la fonction de coût total est aussi une fonction quadratique en x_{ik} . En rajoutant au besoin des cellules fictives avec des volumes d'appels fictifs non nuls, on peut ramener la contrainte (2.2) à une égalité. Ainsi formulé, le problème ne peut être résolu par les méthodes standard de programmation linéaire à cause de la non-linéarité de la contrainte (2.3). Cette dernière peut être remplacée par l'ensemble équivalent de contraintes suivant:

$$z_{ijk} \leq x_{ik} \quad (2.6)$$

$$z_{ijk} \leq x_{jk} \quad (2.7)$$

$$z_{ijk} \geq x_{ik} + x_{jk} - 1 \quad (2.8)$$

$$z_{ijk} \geq 0 \quad (2.9)$$

Il est également possible de simplifier encore plus la fonction de coût en reformulant la partie relative au coût de relève entre les cellules. En effet, on peut, au lieu de considérer séparément le coût dû à une relève simple et celui dû à une relève

complexe, définir une nouvelle variable h_{ij} qui désignera seulement le coût réduit par unité de temps d'une relève complexe entre les cellules i et j .

On a donc :

$$h_{ij} = H'_{ijk} - H_{ij}$$

Après quelques manipulations mathématiques, la nouvelle fonction de coût total à minimiser (équation 2.5) devient :

$$f = \sum_{i=1}^n \sum_{k=1}^m c_{ik} x_{ik} + \sum_{i=1}^n \sum_{\substack{j=1, j \neq i}}^n h_{ij} (1 - y_{ij}) + \sum_{\substack{i=1 \\ j=1 \\ i \neq j}}^n \sum_{\substack{j=1 \\ j \neq i}}^n H_{ij}$$

constante

En faisant abstraction de la partie constante, le problème d'affectation dans le cas de la domiciliation simple revient donc à résoudre le problème d'optimisation mathématique suivant soit :

Minimiser

$$f = \sum_{i=1}^n \sum_{k=1}^m c_{ik} x_{ik} + \sum_{i=1}^n \sum_{\substack{j=1, j \neq i}}^n h_{ij} (1 - y_{ij}) \quad (2.10)$$

sujet aux contraintes (2.1), (2.2), (2.4) et (2.6) à (2.9).

Ce problème peut être résolu à l'aide des méthodes usuelles de programmation en nombres entiers.

Maintenant, si l'on voulait adapter cette formulation à l'affectation des Nœuds B des réseaux 3G aux différents commutateurs, on serait confronté à un certain nombre de difficultés. D'abord, dans les réseaux 3G, on n'a plus seulement les MSC comme commutateurs car il y a aussi un autre type de commutateur qui est le SGSN. Et ces deux équipements n'ont pas les mêmes fonctions ni les mêmes capacités, ce qui invalide l'expression habituelle de la contrainte sur la capacité des commutateurs. De plus, tel qu'illustré dans le chapitre 1, dans l'architecture envisagée pour les réseaux 3G, chaque Nœud B du réseau est affecté à un RNC et chaque RNC est à son tour affecté en même

temps à un MSC et à un SGSN. Donc, il sera nécessaire de revoir la formulation des coûts de liaison et de relève, particulièrement dans le cas de la relève complexe, étant donnée qu'une relève complexe sera marquée par un changement de MSC et de SGSN et qu'une mise à jour sera en conséquence nécessaire pour associer le nouveau MSC et le nouveau SGSN. Enfin, il faudra aussi introduire de nouvelles variables définissant l'affectation d'un Nœud B à un RNC, puis d'un RNC à un MSC et à un SGSN.

2.1.2 Cas de la domiciliation double

La domiciliation double considère que chaque cellule du réseau est rattachée à deux commutateurs. Cela permet de définir deux patrons d'affectation pour chaque cellule, dépendamment de la période de la journée. Le problème d'affectation revient donc à optimiser simultanément les coûts par rapport aux deux patrons d'affectation. Les deux patrons étant liés, il est impossible de les minimiser séparément.

Certaines caractéristiques du problème de la domiciliation double comme le fait que dans chaque patron, chaque cellule soit reliée à un et un seul commutateur (comme dans le cas de la domiciliation simple), font qu'on peut l'assimiler à deux problèmes d'affectation simple. Ainsi, pour la formulation du problème dans ce cas ci, les mêmes variables que dans la domiciliation simple sont utilisées pour définir un des patrons d'affectation (par exemple, celui de la matinée). Pour le deuxième patron (celui de l'après-midi), des variables additionnelles sont introduites. Ainsi, λ' , désigne le nombre d'appels par unité de temps provenant de la cellule i et h'_{ij} le coût réduit par unité de temps d'une relève complexe entre les cellules i et j dans le deuxième patron d'affectation.

Pour ce qui est des autres variables, on a c_{ik} qui représente toujours le coût d'amortissement de la liaison entre la cellule i et le commutateur k et Cap_k la capacité maximale du commutateur k . Les valeurs de Cap_k et c_{ik} restent les mêmes pour les deux patrons d'affectation. Pour ce qui est du coût des liaisons, il a fallu éviter de compter en

double le coût de liaison c_{ik} dans le cas où une cellule reste connectée au même commutateur dans les deux patrons d'affectation.

Ainsi, les deux patrons d'affectation pour chacune des parties de la journée ont été formulés, en indiquant s'il est plus économique ou non d'avoir une cellule connectée à deux commutateurs avec un basculement d'un commutateur à un autre, ou alors d'avoir la cellule connectée à un seul commutateur. En se servant des variables et des relations exprimées dans le cas de la domiciliation simple, le premier patron d'affectation (matinée) devait satisfaire les contraintes (2.1), (2.2), (2.4) et (2.6) à (2.9). Quant au deuxième patron d'affectation (après-midi), de nouvelles variables $x'_{ik}, z'_{ijk}, y'_{ij}$ ayant les mêmes significations que les variables x_{ik}, z_{ijk}, y_{ij} , définies précédemment dans le cas de la domiciliation simple, ont été introduites. Ainsi, le deuxième patron doit aussi satisfaire les contraintes (2.1), (2.2), (2.4) et (2.6) à (2.9) où l'on remplace $\lambda_i, x_{ik}, z_{ijk}, y_{ij}$ par $\lambda'_i, x'_{ik}, z'_{ijk}, y'_{ij}$. Le coût de liaison entre la cellule i et le commutateur k est pris en compte si la cellule i est affectée au commutateur k dans au moins un des deux patrons d'affectation, c'est-à-dire si $x_{ik}=1$ ou $x'_{ik}=1$. La nouvelle variable w_{ik} qui est égale à 1 si $x_{ik}=1$ ou $x'_{ik}=1$ permet d'indiquer si l'on doit ou non compter le coût de liaison entre la cellule i et le commutateur k . Elle s'exprime de la manière suivante :

$$w_{ik} = x_{ik} \vee x'_{ik} \text{ pour } i = 1, \dots, n \text{ et } k = 1, \dots, m \quad (2.11)$$

où « \vee » désigne l'opérateur logique « ou ».

La fonction de coût à minimiser est formulée de la manière suivante:

Minimiser

$$f = \sum_{i=1}^n \sum_{k=1}^m c_{ik} w_{ik} + \sum_{i=1}^n \sum_{j=1, j \neq i}^n h_{ij} (1 - y_{ij}) + \sum_{i=1}^n \sum_{j=1, j \neq i}^n h'_{ij} (1 - y'_{ij}) \quad (2.12)$$

sous les contraintes (2.1), (2.2), (2.4) et (2.6) à (2.9), pour les variables $x_{ik}, z_{ijk}, y_{ij}, \lambda_i$ et h_{ij} d'une part, et $x'_{ik}, z'_{ijk}, y'_{ij}, \lambda'_i$ et h'_{ij} d'autre part. À tout cela, s'ajoute la contrainte supplémentaire (2.11).

De la même manière que dans la domiciliation simple, la non-linéarité de la contrainte (2.11) rend le problème difficile à résoudre. Ainsi, dans le but de linéariser le problème, la contrainte 2.11 est remplacée par les contraintes d'inégalité suivantes :

$$w_{ik} \geq x_{ik} \quad (2.13)$$

$$w_{ik} \geq x'_{ik} \quad (2.14)$$

$$w_{ijk} \leq x_{ik} + x'_{ik} \quad (2.15)$$

$$w_{ik} \leq 1 \quad (2.16)$$

Ainsi, le problème revient à :

Minimiser

$$f = \sum_{i=1}^n \sum_{k=1}^m c_{ik} w_{ik} + \sum_{i=1}^n \sum_{j=1, j \neq i}^n h_{ij} (1 - y_{ij}) + \sum_{i=1}^n \sum_{j=1, j \neq i}^n h'_{ij} (1 - y'_{ij}) \quad (2.12)$$

sujet aux contraintes (2.1), (2.2), (2.4), (2.6) à (2.9) et (2.13) à (2.16)

Pour les mêmes raisons qu'on a détaillées dans le cas de la domiciliation simple, il ne serait pas possible d'appliquer directement la formule (2.12) pour résoudre le problème d'affectation des Nœuds B aux commutateurs dans les réseaux de troisième génération. D'abord parce que dans les réseaux mobiles 3G, nous avons plus d'équipements qui entrent en jeu dans le processus d'affectation et également différents types d'interconnexion à considérer : l'interconnexion entre les Nœuds B et les RNC, l'interconnexion entre les RNC et les MSC et l'interconnexion entre les RNC et les SGSN. Ensuite, le type de trafic que l'on prend en charge dans les réseaux 3G est plus varié que celui qu'on a dans les réseaux 2G. En effet, en plus du trafic de voix par commutation de circuits, on a un nouveau trafic de type commutation de paquets, ce qui va certainement changer la manière de représenter les contraintes liées à la capacité des équipements à prendre en charge le trafic des usagers du réseau.

Et quand nous voudrions considérer une domiciliation double, contrairement à la domiciliation simple, il y a plusieurs possibilités qui pourraient s'offrir à nous. En effet,

nous pouvons considérer lors de notre affectation une domiciliation double pour l'affectation des Nœuds B aux RNC et une autre domiciliation double pour l'affectation des RNC aux MSC et aux SGSN, ou encore une domiciliation simple pour l'affectation des Nœuds B aux RNC et une domiciliation double pour l'affectation des MSC aux SGSN ou l'inverse. Cela complique davantage la formulation du problème et rend encore insuffisantes les contraintes et les variables dépendantes exprimées dans le cas de la domiciliation double pour les réseaux 2G. D'où la nécessité de trouver une nouvelle formulation des coûts de liaison et de relève en ajoutant éventuellement des variables et des contraintes supplémentaires.

2.2 Méthodes de résolution

Pour la résolution du problème d'affectation, beaucoup de méthodes ont été développées et sont classées en deux grands groupes : les *méthodes classiques* et les *méthodes heuristiques*. L'une des différences entre ces deux groupes est que les méthodes heuristiques se fondent souvent sur l'historique de l'exploration du domaine de recherche pour converger vers une «bonne» solution. Ce qui n'est pas le cas des méthodes classiques. Dans cette section, nous allons présenter d'abord les caractéristiques des méthodes classiques ensuite celles de quelques heuristiques soient : le *recuit simulé*, la *programmation par contraintes*, les *algorithmes génétiques*, la *recherche taboue* et les *algorithmes memétiques*.

2.2.1 Méthodes classiques

Parmi les méthodes classiques, on peut citer la méthode BB (Branch and Bound) et les algorithmes ADD et DROP. La méthode Branch and Bound est une technique de limitation d'exploration, elle permet de limiter le domaine de recherche grâce à une fonction de coût associée à chaque possibilité ou solution du domaine de recherche. Ainsi, on part d'une solution quelconque considérée comme point de départ, sur laquelle on effectue, parmi toutes les transformations possibles, la transformation qui minimise le

coût des efforts additionnels pour atteindre l'optimum [14]. Malheureusement, calculer le coût des efforts additionnels pour atteindre l'optimum revient presque à connaître cet optimum, ce qui est aussi difficile que le problème initial.

Les algorithmes ADD et DROP, quant à eux, prennent comme estimation du coût additionnel requis pour atteindre l'optimum, l'opposé du gain fait par rapport à la solution initiale. Ceci se justifie par le fait que plus on a déjà gagné par rapport à la solution de départ, moins il reste d'effort à faire pour parvenir à une solution [19]. Ces méthodes classiques, bien qu'assez efficaces et parvenant à une solution acceptable, comptent surtout sur une bonne solution initiale et sur une évaluation de l'optimum pour pouvoir se comparer tout au long du processus de recherche de la solution optimale. Certaines d'entre elles ne peuvent échapper au piège d'un minimum local lors de l'exploration de l'espace des solutions. C'est la raison pour laquelle, l'on s'est tourné vers les méthodes heuristiques qui permettent d'explorer efficacement l'espace des solutions.

2.2.2 Recuit simulé

Le recuit simulé, tel qu'introduit dans [18], est une technique d'optimisation qui permet d'échapper au piège des minima locaux en autorisant, avec une certaine probabilité, une augmentation de la fonction de coût d'une solution à une autre. C'est une technique inspirée de la thermodynamique. En effet, certains systèmes physiques utilisés en thermodynamique atteignent leur état d'équilibre (minimum d'énergie) malgré un grand nombre d'états métastables possibles. Au cours de cette opération, l'énergie peut localement augmenter avec une probabilité proportionnelle à la température du système suivant la loi de Boltzmann. Cette température diminue au cours du temps pour amener le système à l'état final. Ainsi, dans le recuit simulé, une diminution de la fonction sera toujours acceptée, alors qu'une augmentation de la fonction sera acceptée avec une certaine probabilité qui dépend d'un paramètre correspondant à la température. Un certain nombre de paramètres critiques doivent être déterminé tels que la solution initiale, la température initiale, le critère d'arrêt, etc.

Une adaptation du recuit simulé au problème d'affectation de cellules a été proposée et se résume de la manière suivante :

Étape 1 : Générer la topologie initiale au hasard sans tenir compte de la contrainte de capacité sur les commutateurs.

Étape 2 : Choisir une solution voisine à la solution courante.

Étape 3 : Si la solution est meilleure ou si la probabilité d'acceptation est satisfaite prendre cette solution comme nouvelle solution courante.

Étape 4 : Diminuer la température du système en la multipliant par le facteur de recuit.

Étape 5 : Si le critère d'arrêt n'est pas satisfait, aller à l'étape 2, sinon arrêter l'algorithme.

Les résultats de l'implémentation de cette méthode ont montré qu'on obtient souvent une solution qui est meilleure, comparativement à certains algorithmes comme celui développé dans [20].

2.2.3 Programmation par contraintes

La programmation par contraintes est une technique qui est basée sur les contraintes et les algorithmes de recherche de solutions satisfaisant ces contraintes. Pour résoudre un problème combinatoire, la méthode part d'une modélisation où les variables ainsi que leurs domaines sont définis de manière à limiter le nombre de combinaisons à considérer. On applique par la suite les contraintes aux variables de modélisation, puis on vérifie si elles sont satisfaites à l'aide de certains algorithmes propres à la programmation par contraintes. Enfin, on a recours à certaines heuristiques de contrôle de recherche pour trouver une affectation des valeurs à toutes les variables du problème avec le moins de conflits possibles. Ainsi, une solution s est déterminée parmi l'ensemble des solutions réalisables possibles T du problème, et la recherche se poursuit en ajoutant une contrainte supplémentaire $t < s$ ($t \in T$) sur toute nouvelle solution. Lorsque la recherche

se termine en échec, le dernier coût total calculé donne le coût optimal (global) du problème.

La programmation par contraintes a été adaptée à la résolution du problème d'affectation de cellules aux commutateurs dans le contexte des réseaux mobiles 2G, entre autres, dans [2] et [3]. Dans cette adaptation, chaque cellule i est représentée par une variable entière $Switch_i$ dont le domaine est déterminé par le nombre m de commutateurs disponibles dans le réseau :

$$Switch_i \in \{0, \dots, m-1\} \text{ pour } i = 0, 1, \dots, n - 1$$

À chacune de ces cellules sont associées deux variables de coût: le coût de câblage, $cCost[i, Switch_i]$, qui dépend de la cellule i et du commutateur auquel elle est affectée, et le coût de relève, $hCost[i]$, qui dépend des cellules n'appartenant pas au même commutateur. On veut alors minimiser la fonction objective suivante:

$$\sum(cCost[i, Switch_i] + hCost[i]), i = 0, 1, \dots, n - 1$$

sous la contrainte suivante:

$$\sum_{i \in Cells_j} \lambda_i \leq Cap_j$$

$Cells_j$ est une variable ensembliste symbolisant un commutateur auquel est affecté un ensemble de cellules. Une fois le problème modélisé, on applique et on propage les contraintes sur les variables choisies. Il y a deux contraintes fermes à respecter. Selon la première contrainte d'unicité de l'affectation, imposée par la domiciliation simple, chaque cellule est affectée à un et à un seul commutateur. Cette contrainte est respectée à coup sûr, car chaque variable entière représentant la cellule ne peut prendre qu'une seule valeur à la fois dans son domaine. La seconde contrainte concerne la capacité de chaque commutateur, qui est fixée et exprimée en volume d'appels par unité de temps. On exprime cette contrainte de manière simple et efficace à l'aide de la variable ensembliste:

$$Cells_j \subset \{0, 1, \dots, n - 1\}, j = 0, 1, \dots, m - 1$$

représentant chaque commutateur du réseau. L'objectif est alors de trouver, pour chaque commutateur, le sous-ensemble des éléments requis, c'est-à-dire des cellules qu'il peut gérer par unité de temps. Les résultats obtenus dans [2] et [3] ont montré que, malgré une certaine lenteur dans l'exécution, l'algorithme produit une solution qui est souvent excellente se situant à un faible pourcentage de l'estimation de l'optimum.

2.2.4 Algorithmes génétiques

Les algorithmes génétiques (AG) s'inspirent de la théorie de la survie des espèces de Charles Darwin, selon laquelle les spécimens jugés les plus forts peuvent se reproduire à un rythme plus rapide que les autres. On applique des opérateurs génétiques à des candidats choisis sur cette base, en espérant engendrer ainsi de nouveaux candidats plus performants.

En biologie, on manipule des gènes et des chromosomes; il en va de même dans le modèle des AG. Les problèmes et les solutions sont encodés et l'encodage prend souvent la forme d'une *chaîne de bits*. Ces chaînes de bits sont comparables aux *chromosomes* en biologie, et les bits ou caractères qui composent ces chaînes rappellent les *gènes*. L'ensemble de ces chaînes forme une *population*, alors qu'en biologie on parle de *génotype*. Dans la recherche de solutions à un problème, les AG utilisent une grande part de hasard. En effet, les candidats à la reproduction sont choisis de façon probabiliste; les chromosomes de la population sont croisés de façon aléatoire dans la progéniture; et les gènes d'un chromosome sont mutés selon une certaine probabilité. En appliquant ainsi, de génération en génération, les opérateurs génétiques sur des candidats jugés performants, on cherche à obtenir une progéniture plus performante que celle de la génération précédente, ce qui permet de s'approcher d'une solution optimale. L'algorithme génétique repose donc principalement sur des mécanismes tels la représentation des chromosomes, et les opérateurs génétiques qui sont au nombre de trois : *la sélection, le croisement et la mutation*.

La sélection

C'est le processus selon lequel des chaînes de la population sont choisies pour former une nouvelle génération sur la base de la valeur de la fonction objectif. Plus cette valeur est élevée, plus la chaîne en question a des chances d'être sélectionnée. Une partie des gènes des chromosomes choisis contribuera à une solution-chromosome dans les générations futures.

Le croisement

C'est le processus selon lequel les bits de deux chaînes sélectionnées sont intervertis: dans le langage génétique, on dira que ces chaînes sont croisées. Pour exécuter le croisement, on accouple des chaînes de la population au hasard. Chaque paire de longueur t subit le croisement comme suit: une position entière k est choisie uniformément entre 1 et $(t-1)$. Deux nouvelles chaînes sont créées en échangeant tous les gènes entre les positions $(k+1)$ et t de chaque paire considérée. Les nouvelles chaînes peuvent être totalement différentes de leurs parents. Toutefois, le croisement n'aura aucun effet sur un gène dont les parents ont la même valeur à la même position.

La mutation

C'est le processus selon lequel la valeur d'un gène choisi au hasard dans un chromosome est régénérée. En modifiant aléatoirement la valeur d'une position dans une chaîne, la mutation est utile pour ramener du matériel génétique qui aurait été oublié par les opérateurs de sélection et de croisement. Certaines implantations vont automatiquement changer la valeur du bit choisi, d'autres vont générer aléatoirement la nouvelle valeur du bit, ce qui n'entraîne à l'occasion aucune mutation.

Une adaptation de l'Algorithme Génétique à la résolution du problème d'affectation des cellules aux commutateurs dans les réseaux mobiles a été implantée dans [12] pour le cas de la domiciliation simple et dans [5] pour la domiciliation double. Dans cette adaptation, l'objectif est de trouver, à partir d'une population initiale de

chromosomes, la meilleure affectation, c'est-à-dire celle qui minimise le coût du réseau tout en respectant la contrainte de capacité des commutateurs et la contrainte d'affectation unique des cellules aux commutateurs (pour la domiciliation simple). Une représentation non binaire des chromosomes a été retenue. Chaque chromosome représente un schéma d'affectation spécifique. Les éléments de la chaîne représentant un chromosome sont des entiers. Ces entiers correspondent aux différents commutateurs, qui sont numérotés de 1 à m (cas d'un réseau à m commutateurs). La longueur des chaînes représentant les chromosomes est égale au nombre de cellules n en domiciliation simple et à $2n$ en domiciliation double. De même, la valeur maximale d'un gène de ces chromosomes est égale au nombre maximal de commutateurs. Certaines valeurs peuvent ne pas faire partie d'une chaîne, de sorte que le commutateur portant le numéro manquant ne figure pas dans le schéma d'affectation, donc n'est pas utilisé.

Exemple de codage des chromosomes

Soit un réseau composé de 8 cellules et 3 commutateurs potentiels. Les figures 2.1 et 2.2 illustrent respectivement la représentation du chromosome dans le cas d'une domiciliation simple et dans le cas d'une domiciliation double.

1	2	2	1	2	1	1	2
---	---	---	---	---	---	---	---

Figure 2.1 Représentation non binaire d'un chromosome (domiciliation simple)

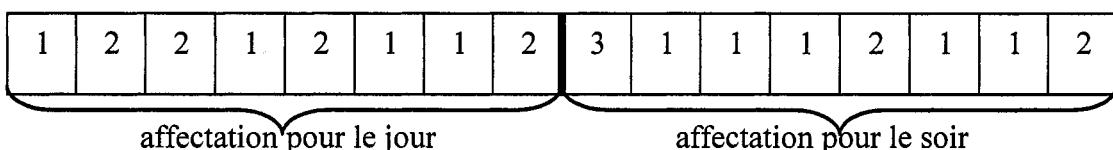


Figure 2.2 Représentation non binaire d'un chromosome (domiciliation double)

Le fonctionnement général de l'algorithme génétique est résumé dans les étapes suivantes :

Étape 1 : On génère une population initiale de taille n représentant le nombre de chromosomes, puis on choisit au hasard les gènes qui composent chaque chromosome: c'est la première génération de chromosomes.

Étape 2 : On évalue chaque chromosome par la fonction objectif, ce qui permet de déduire sa valeur d'aptitude.

Étape 3 : Commence alors le cycle de génération des populations, chaque nouvelle génération remplace la précédente. Le nombre x de générations est déterminé au départ. Dans chaque génération, n chromosomes sont choisis pour se voir appliquer les différents opérateurs génétiques. Après chaque génération, les n nouveaux chromosomes ainsi créés remplacent la génération précédente. Après la création de la x^e génération, les chromosomes auront évolué de telle façon que cette dernière génération contiendra des chromosomes meilleurs à ceux des générations précédentes.

Les résultats de simulations obtenus dans [12] ont montré que le choix des paramètres des opérateurs génétiques influence beaucoup la solution obtenue. En effet, un choix judicieux des valeurs de ces paramètres offre plus de chances d'orienter les recherches dans une bonne direction. Également, on a constaté qu'une valeur élevée du nombre de générations ne garantit pas forcément l'obtention d'une meilleure solution. Il suffit en fait d'un croisement adéquat, qui peut d'ailleurs se produire à n'importe quelle génération, pour obtenir de bonnes solutions.

2.2.5 Recherche taboue

La Recherche Taboue (RT) est une technique adaptative introduite en optimisation combinatoire pour résoudre des problèmes NP-difficiles [8]. C'est une amélioration de l'algorithme général de descente. Pour éviter le piège des minima locaux, elle accepte de temps à autre des solutions qui n'améliorent pas la fonction objectif, en espérant ainsi parvenir plus tard à de meilleures solutions. Cependant, le fait de vouloir accepter des solutions pas forcément meilleures entraîne un risque de cycle, c'est-à-dire un retour vers des solutions déjà explorées. D'où l'idée de conserver une *liste taboue* T

des solutions déjà visitées. Ainsi, lors de la génération de l'ensemble V des solutions voisines candidates, on exclut toutes les solutions appartenant à la liste taboue.

Si la conservation d'une liste taboue permet d'éviter le risque de cycle, elle s'avère peu pratique par ailleurs. En effet, le stockage de toutes les solutions déjà visitées peut nécessiter beaucoup de mémoire. D'autre part, il peut s'avérer utile de revenir à une solution déjà visitée pour continuer la recherche dans une autre direction. Un compromis consiste à garder dans la liste taboue seulement les k dernières solutions. Ainsi, quand une nouvelle solution devient taboue, elle remplace la plus ancienne dans la liste, ce qui permet d'éviter des cycles de longueur inférieure ou égale à k . Toutefois, des cycles de longueur supérieure à k peuvent toujours survenir. L'algorithme RT s'arrête quand il n'y a eu aucune amélioration depuis un nombre k_{max} d'itérations, ou si toutes les solutions voisines candidates sont taboues, c'est-à-dire $V \cap T = \emptyset$, où V représente l'ensemble des solutions voisines candidates et T l'ensemble des solutions taboues.

La Figure 2.3 expose les grandes lignes de l'algorithme général de la RT. C'est une méthode itérative qui part d'une solution initiale et essaie d'atteindre un optimum global par l'application de mouvements permettant de passer d'une solution s_i à une solution s_{i+1} choisie dans le voisinage $N(s_i)$ de s_i . À chaque itération i , le voisinage où les solutions sont sélectionnées est redéfini comme le voisinage de la solution courante s_i . À cela s'ajoute le fait que les conditions taboues changent et que les solutions admissibles ne sont plus les mêmes.

On peut représenter l'exploration de l'espace de recherche dans la RT par un graphe $G=(X,A)$, où X désigne l'ensemble des solutions et A l'ensemble des arcs $(x,m(x))$, $m(x)$ étant la solution obtenue en appliquant le mouvement m à x . Un mouvement donné équivaudra donc à l'ensemble des arcs $\{(x,m(x)), x \in X\}$. Le graphe G est symétrique, car pour chaque arc $(x, m(x))$, il existe un arc $(m(x), x)$ obtenu en appliquant le mouvement inverse m^{-1} à $m(x)$. L'algorithme de la recherche taboue part d'une solution initiale x_0 , nœud du graphe G , et va chercher dans G un chemin x_0, x_1, \dots, x_k , où $x_i = m(x_{i-1})$ avec $i=1, \dots, k$. On trouve chaque solution intermédiaire du chemin en appliquant un

mouvement à la précédente. Les arcs (x_i, x_{i+1}) du chemin sont choisis en résolvant le problème d'optimisation :

$$f(x_{i+1}) = \min f(x_i) \text{ avec } x_i \in V - T.$$

```

Initialisation : Choisir une solution initiale  $s$  dans  $X$ .
 $s^* := s$  ( $s^*$  est la meilleure solution obtenue jusqu'ici)
 $nbiter := 0$  ( $nbiter$  est le compteur des itérations)
 $bestiter := 0$  ( $bestiter$  est le numéro de l'itération à laquelle on a obtenu la dernière amélioration, le dernier  $s^*$ )
 $T := \emptyset$  ( $T$  est la liste taboue)

Tant que ( $nbiter - bestiter > kmax$ ) ou ( $V - T = \emptyset$ ) alors
     $nbiter := nbiter + 1$ 
    Générer  $V \subseteq N(s)$ 
    Trouver la meilleure solution  $s'$  dans  $V$  ( $f(s') = \min f(s)$  avec  $s \in V - T$ )
     $s := s'$ 
    Mise à jour de  $T$ 
    Si  $f(s') < f(s^*)$  alors  $s^* := s'$ 
     $Bestiter := nbiter$ 
    Fin si
Fin tant que

```

Figure 2.3 Algorithme général de la recherche taboue

La RT se distingue donc de la méthode de descente générale par l'utilisation d'une structure de mémoire à court terme qui permet d'accepter des solutions moins bonnes pour sortir des optima locaux, tout en évitant les cycles. Elle implémente également un mécanisme de mémoire à moyen terme afin d'intensifier la recherche localement dans des régions prometteuses et un mécanisme de mémoire à long terme pour amener la recherche dans des régions jusqu'ici peu explorées en régénérant à chaque cycle de redémarrage une nouvelle solution initiale.

La recherche taboue a été adaptée au problème d'affectation de cellules aux commutateurs dans les réseaux mobiles 2G (en domiciliation simple) dans [15]. Dans cette adaptation, l'espace de recherche est libre des contraintes sur la capacité des commutateurs mais respecte la contrainte sur l'affectation unique des cellules aux commutateurs. Ainsi, la solution obtenue n'est pas toujours faisable. La solution initiale est obtenue en affectant chaque cellule au commutateur le plus proche. Ensuite, un tableau de gain est généré pour chaque solution courante en calculant pour chaque cellule le gain qu'on obtiendra en la réaffectant à chacun des commutateurs du réseau. Pour un réseau de n cellules et m commutateurs, le tableau de gain sera de taille $n \times m$. Pour chaque solution courante, une évaluation des solutions du voisinage est effectuée en utilisant le gain associé à chaque mouvement effectué pour obtenir la solution voisine en question. Le mouvement ayant le gain le plus faible est alors sélectionné lors de la recherche, puis une mise à jour très simple et rapide du tableau des gains est effectuée.

Un mécanisme de *rappel* est utilisé pour ramener l'exploration vers des meilleures solutions lorsque l'algorithme s'attarde sur des solutions qui ne respectent pas la contrainte de capacité. Pour y parvenir, après un nombre donné de solutions consécutives non faisables, on pénalise les gains des mouvements menant à des solutions non faisables. L'adaptation de la RT a donné des résultats assez satisfaisants par rapport aux autres heuristiques [15]. Son efficacité et sa robustesse ont été établies. Ce qui nous a motivé à la choisir pour traiter notre problème d'affectation des cellules dans les réseaux mobiles de troisième génération. Les détails de son adaptation à notre problème seront présentés dans le chapitre 3.

2.2.6 Algorithmes memétiques

Les algorithmes memétiques (AM) sont des heuristiques utilisées dans la résolution des problèmes d'optimisation combinatoire et sont basés sur la théorie de l'évolutivité des espèces. Ce sont des algorithmes qui s'inspirent de la notion de « *mème* » définie par Richard Dawkins en 1976 comme étant une unité d'information qui

se reproduit lors de l'évolution d'une espèce [21]. En effet, d'après Dawkins, tout comme les *gènes*, les *mèmes* évoluent dans le temps ; les meilleurs survivent alors que les moins bons risquent de disparaître. Ainsi, le « *mème* » est souvent modifié par le spécimen *parent* avant d'être transmis à la prochaine génération. Les AM sont une combinaison entre les méthodes de recherche globale basées sur la théorie de la survie des espèces (comme les algorithmes génétiques) et une recherche locale qui met l'emphase sur l'évolution isolée de chaque spécimen de l'espèce [21]. Le fonctionnement des AM consiste en la création, pour un problème d'optimisation donné, d'une certaine population initiale soit sur une base aléatoire ou alors suivant une certaine procédure d'initialisation définie au préalable. Une fois créée, chaque individu de la population initiale suit son propre évolution en faisant une recherche locale autour de lui pour s'améliorer. Lorsqu'il atteindra un certain stade de développement, il pourra à ce moment là interagir avec les autres membres de la population qui eux aussi auraient suivi le même processus. Pour effectuer la recherche locale, il est possible d'utiliser d'autres heuristiques comme la recherche taboue, le recuit simulé etc.

Une adaptation des AM pour la résolution du problème d'affectation des cellules aux commutateurs dans les réseaux mobiles 2G a été effectuée dans [23]. Dans cette adaptation, une notation similaire à celle utilisée dans [13], et présentée à la section 2.2.4, a été introduite pour représenter les cellules et les commutateurs et pour encoder les chromosomes et les gènes. Dans cette représentation, les gènes représentent les cellules et la longueur des chromosomes est égale au nombre de cellules du réseau et les valeurs qu'ils contiennent représentent le commutateur auquel est affectée chaque cellule (voir les figures 2.1 et 2.2). Les AM sont contrôlés par plusieurs paramètres dont : la taille de la population, le nombre de génération, le nombre de cycles, la probabilité de croisement et de mutation, qui ont un effet sur la précision et l'efficacité de l'algorithme. La Figure 2.4 présente le mode de fonctionnement des AM. À remarquer que pour la recherche locale, ils ont utilisé dans [23], la Recherche Taboue dont l'algorithme est présenté à la Figure 2.3 témoignant ainsi du caractère hybride des AM.

```

Initialisation: Choisir une population (gen).

Pour chaque individu i ∈ gen
    Faire i = Recherche_tabou (i)
Fin Pour

Tant que Continuer
    répéter
        sélectionner deux individus i, j ∈ gen
        appliquer Croisement (i, j) et générer les enfant (c)
        ajouter Recherche_tabou (enfant (c)) à gen
    jusqu'à croisement := faux
    Pour chaque individu i ∈ gen Faire
        Si probabilite-mutation := vrai alors faire i = Recherche_tabou (Mutation(i))
    Fin Pour
    gen = Selection (gen)
Fin tant que

```

Figure 2.4 Fonctionnement général des AM

Afin de déterminer l'efficacité des AM, Pierre et Quintero [23] ont effectué des tests de comparaison avec d'autres heuristiques telles l'heuristique H, implémentée dans [20], et l'adaptation de la Recherche Taboue faite dans [15]. Les résultats obtenus ont montré que, pour les différentes séries de tests effectués, les AM apportent une amélioration sur la fonction de coût obtenue par l'heuristique H et la RT. Ainsi, en terme de coûts, cette amélioration se situe aux alentours de 1.96% par rapport à la Recherche Taboue et de 2.21% par rapport à l'heuristique H. Et en terme de temps de calcul, pour un grand nombre de cellules, les résultats obtenus ont montré que la Recherche Taboue est un peu plus rapide que l'heuristique H et les AM. Et, pour des réseaux de taille moyenne, les AM se sont avérés plus lents que l'heuristique H et la Recherche Taboue. Le rapport du temps de calcul requis entre les AM et la Recherche Taboue ou les AM et l'heuristique H est approximativement de 10.

2.3 Problématique des réseaux mobiles 3G

Le désir d'introduire des nouveaux services particulièrement gourmands en bande passante et de nouvelles techniques de gestion de réseaux ont amené les chercheurs à se pencher sur la ré-ingénierie des réseaux mobiles existants. C'est ainsi que des architectures de réseau GPRS (*GSM Paquet Radio Service*) ont été développées pour les réseaux dits 2.5G, devant assurer la transition vers des réseaux de troisième génération qui sont en processus de normalisation. Cependant, le déploiement de ces types de réseaux nécessite une planification minutieuse. Ainsi, avant de présenter les éléments essentiels à considérer lors de la planification de ces réseaux, en particulier le réseau UMTS, nous allons nous intéresser aux acteurs principaux dans le processus de normalisation et aux détails d'architecture de ces réseaux mobiles 3G.

2.3.1 Acteurs principaux dans la normalisation des réseaux mobiles 3G

Le processus de normalisation des systèmes mobiles de troisième génération a été initié par l'UIT (*Union Internationale des Télécommunications*) avec pour but de définir une norme mondiale qui remplacerait les systèmes existants. Ce système, baptisé *International Mobile Telecommunication 2000* (IMT-2000) par l'UIT, a pour but de fournir des services mobiles multimédias à haut débit (ex : la messagerie, vidéophonie en temps réel, l'Internet, transmission d'images...) tout en permettant une mobilité globale de ses utilisateurs. L'UMTS (*Universal Mobile Telecommunications System*) est un des systèmes proposés par IMT-2000 pour succéder à la norme GSM (*Global System for Mobile communications*). Le Groupe de travail 3GPP (*Third Generation Partnership Project*) créé par l'institut européen de normalisation ETSI (*European Telecommunications Standards Institute*) est le principal architecte de la normalisation UMTS.

Depuis 1998, 3GPP développe les spécifications techniques de l'UMTS pour les proposer ensuite à l'UIT. 3GPP distingue cinq domaines principaux de normalisation

UMTS : le réseau d'accès radio, le réseau cœur, les terminaux mobiles, les services et les systèmes de gestion d'accès aux réseaux [24]. Parallèlement, l'institut américain de normalisation ANSI (*American National Standards Institute*) a créé un projet semblable au 3GPP, appelé le 3GPP2, pour mettre au point les spécifications techniques d'une architecture appelée CDMA2000 [27].

2.3.2 Architecture des réseaux mobiles 3G

Parmi les architectures de réseaux développées jusqu'ici pour les réseaux mobiles 3G, on a les réseaux UMTS et les réseaux CDMA2000. Ces deux architectures, même si elles présentent des similarités quant à la disposition de certains des équipements, ne mettent pas en jeu les mêmes équipements ni les mêmes technologies d'accès et de gestion de réseau.

Architecture des réseaux UMTS

Les réseaux UMTS constituent une évolution de la norme GSM et une continuation des réseaux GPRS communément appelés aussi réseaux 2.5G. Nous avons présenté dans le chapitre 1 (Figure 1.3) l'architecture du réseau UMTS pour illustrer les différences marquantes par rapport aux réseaux mobiles 2G et qui motivent le choix de notre sujet de recherche. Dans cette section, nous allons présenter les informations complémentaires relatives à l'architecture du réseau. Ainsi, dans un réseau typique UMTS, on peut distinguer essentiellement deux niveaux hiérarchiques : le niveau 1 qui constitue le réseau d'accès radio aussi appelé UTRAN (*UMTS Terrestrial Radio Access Network*), et le niveau 2 qui est le sous-système réseau ou encore réseau cœur. Au niveau de l'UTRAN, dans chacune des cellules du réseau est placée une station de base appelée *Nœud B* qui constitue le point initial d'accès des usagers au réseau. Les fonctions du *Nœud B* sont :

- la gestion de la couche physique de l'interface radio en assurant la transmission et la réception radio ;

- la Modulation / Démodulation ;
- le codage du canal physique ;
- le contrôle d'erreur ;
- le contrôle de puissance en boucle fermée.

Un ou plusieurs de ces *Nœuds B* sont connectés à un contrôleur de station de base appelé RNC (*Radio Network Controller*) qui assure les fonctions suivantes :

- la gestion du protocole *Radio Resource Control* (RRC) définissant les procédures et les messages entre le mobile et l'UTRAN ;
- le contrôle d'admission ;
- l'allocation des canaux ;
- l'interfaçage avec le sous-système réseau pour les transmissions en mode paquet (Iu PS) et en mode circuit (Iu CS) ;
- le contrôle de charge et de congestion des différents *Nœuds B* ;
- la signalisation.

L'interconnexion entre le sous-système radio (UTRAN) et le réseau cœur est assurée par les RNC. Le réseau cœur est constitué de l'ensemble des équipements qui assurent le contrôle des appels, le contrôle de la sécurité, la gestion des interfaces avec les réseaux externes, etc. Il est responsable de la commutation et du routage des communications (voix et données) vers les réseaux externes.

Dans une première phase de déploiement, le système UMTS s'appuiera sur le réseau cœur du système GPRS qui se décompose en deux domaines : le domaine paquet et le domaine circuit. Le domaine circuit permet de gérer les services temps réels correspondant aux conversations téléphoniques, à la vidéo-téléphonie, aux jeux vidéo, aux streaming et aux applications multimédia. Ces applications nécessitent un temps de transfert faible, et seul le domaine circuit sera capable de gérer ces flux lors de

l'introduction de l'UMTS [4]. L'infrastructure s'appuiera sur un MSC/VLR (*Mobile Switching Centre/Visitor Location Register*) correspondant au commutateur (MSC) et à la base de données (VLR), et sur un GMSC (Gateway MSC), commutateur connecté directement au réseau externe. Le serveur MSC assure toutes les fonctions liées à la mobilité et au contrôle logique en interagissant avec la base de données située au niveau du VLR.

Le domaine paquet permet de gérer les services non-temps réels correspondant à la navigation sur Internet, aux jeux en réseau et aux emails, entre autres. Ces applications sont moins sensibles au temps de transfert et ces données pourront transiter en mode paquet [4]. Le débit supporté pourra atteindre 2 Mb/s. Le domaine paquet s'appuiera sur un SGSN (*Serving GPRS Support Node*) correspondant au MSC/VLR en mode circuit et sur un GGSN (*Gateway GPRS Support Node*) correspondant au GMSC en mode circuit, et commutera vers le réseau Internet et vers d'autres réseaux publics ou privés de transmission de données. Le sous-système réseau intègre aussi une base de données nominale HLR (*Home Location Register*) qui gère les informations sur tous les abonnés du réseau.

Architecture des réseaux CDMA2000

L'architecture CDMA2000 comporte tous les éléments qui font partie des réseaux de voix actuels, auxquels s'ajoutent plusieurs nouveaux équipements qui assurent l'interconnexion du réseau d'accès aux réseaux de données. Cette architecture, illustrée à la Figure 2.5, utilise une composante appelée PDSN (*Packet Data Serving Node*), essentielle au support des services de données [27]. Une telle composante communique avec les nœuds du réseau de voix (HLR et VLR) en utilisant un serveur d'authentification appelé AAA (*Authentication, Authorization Accounting*). Ce dernier identifie les abonnés, leur octroie des autorisations et aide à gérer leurs comptes. Un agent local (*Home Agent*) garde la trace des usagers mobiles qui changent de zones de localisation de manière à s'assurer que les paquets sont bien acheminés à leurs destinataires. En outre, on utilise des routeurs pour faciliter l'échange de paquets entre les différentes entités du

système, ainsi que l'échange entre le système et les autres réseaux fixes à commutation de paquets. Notons également qu'un coupe-feu (*firewall*) est également inséré dans l'architecture CDMA2000 pour en assurer la sécurité.

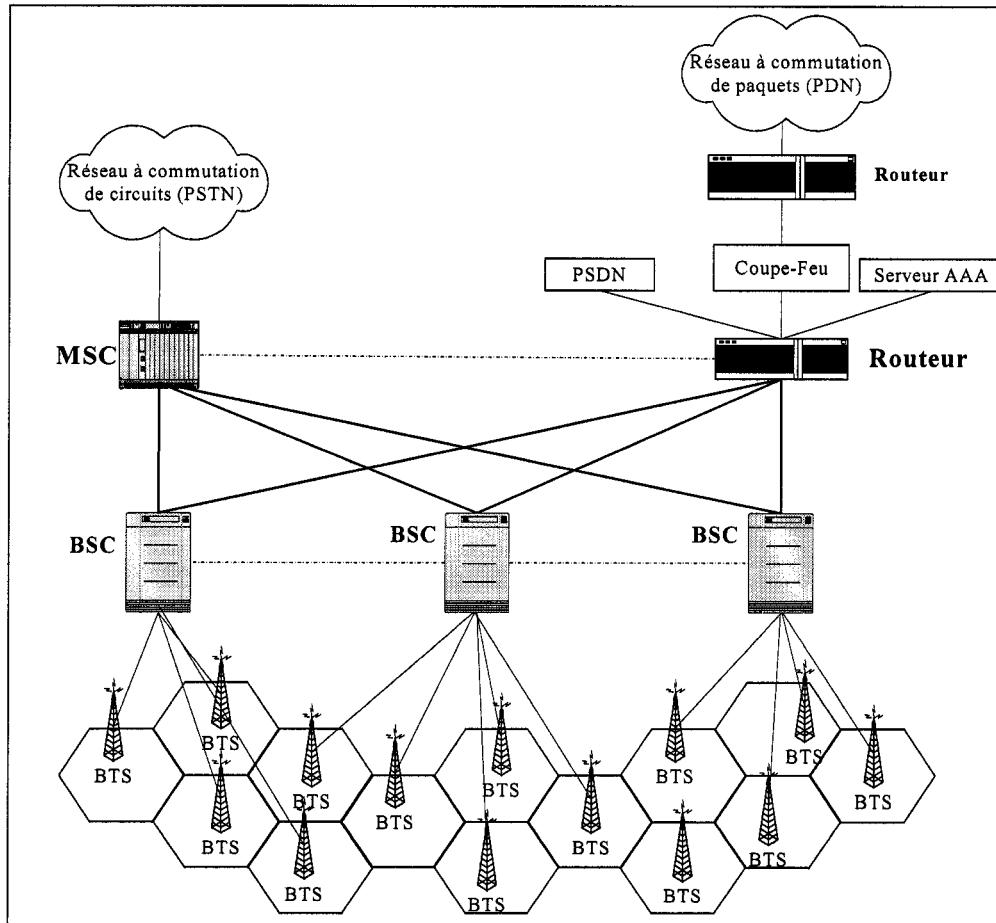


Figure 2.5 Architecture typique d'un réseau CDMA2000

Au niveau du réseau d'accès, nous retrouvons, sous d'autres vocables, les mêmes éléments que dans l'UTRAN de l'UMTS. Plus spécifiquement, les Nœuds B deviennent des BTS (*Base Transceiver Station*), alors que les RNC deviennent des BSC (*Base Station Controller*).

2.3.3 Planification des réseaux 3G

La planification d'un réseau cellulaire a pour but de minimiser les coûts de déploiement et d'exploitation du réseau tout en maintenant une communication de qualité et une capacité élevée. Le réseau doit garantir une bonne qualité de service par l'utilisation de liens radio relativement fiables et disponibles dans les zones de service des usagers. Il s'agit de trouver la configuration des sous-systèmes réseau et radio la plus économique, tout en maintenant une bonne performance et une utilisation efficace des ressources du réseau.

Éléments essentiels de la planification

La planification d'un réseau mobile 3G nécessite de bien connaître entre autres:

- les caractéristiques de l'environnement à couvrir : particularités géographiques, types de propagation radio ;
- le profil des usagers à desservir : densité, comportement statistique ;
- les bandes de fréquences allouées ;
- d'autres informations telles : données du marché, étude de marché (business plan), types de services à fournir, plan de financement, etc.

Comme nous l'avons mentionné précédemment, un réseau UMTS est une infrastructure constituée typiquement de deux parties : l'UTRAN (ou *sous-système radio*) et le Réseau Cœur (ou *sous-système réseau*). Ainsi, le processus de planification se déroule en deux phases:

1. **La planification du sous-système radio (UTRAN)** qui consiste à sélectionner les sites radio (Nœuds B) à l'aide d'outils de planification, à localiser et dimensionner les RNC et connecter les Nœuds B aux RNC et les RNC entre eux ;

2. **La planification du sous-système réseau** qui comprend la localisation et le dimensionnement des différents nœuds du réseau (RNC, MSC et SGSN).

La planification du *sous-système radio* inclut la détermination du modèle de mobilité et de trafic, la définition de la couverture radio, le dimensionnement des cellules, et la planification des fréquences. La planification du *sous-système réseau* inclut la définition du réseau de commutation, la conception des interfaces, la planification du réseau de signalisation et des bases de données (HLR et VLR).

La planification de ces deux sous-systèmes ne se fait pas de manière indépendante car les RNC qui contrôlent le trafic au niveau du sous-système radio sont à leurs tours connectés à deux types de commutateurs que sont les MSC et SGSN. Chacun de ces commutateurs a une capacité limitée en terme de prise en charge du trafic. Ce sont les liaisons entre ces équipements de niveaux hiérarchiques différents qui assurent l'interconnectivité entre le sous-système radio et le sous-système réseau. Il est important de planifier rigoureusement cette portion du réseau, sinon on risquerait de demander aux commutateurs de prendre en charge du trafic qui dépasserait leur capacité. Ce problème est encore plus susceptible de se produire si l'on prend en compte le fait que les usagers sont mobiles et changent constamment de cellule donc de Nœud B et de RNC d'attache. Il faudra donc optimiser les opérations de relève entre les différents commutateurs lorsque les usagers connectés se déplacent. C'est à cette étape particulière de la planification que nous nous attarderons dans ce mémoire pour trouver une nouvelle formulation de la fonction de coûts à minimiser tout en tenant compte des différentes contraintes imposées par le réseau ou par les équipements. Puisqu'on a montré dans les sections précédentes, notamment pour les réseaux mobiles 2G, que les modèles de formulation existants ne sont pas applicables directement aux réseaux mobiles 3G, de nouveaux modèles et méthodes de résolution s'avèrent donc nécessaires. Dans ce sens, des travaux ont été réalisés par Pierre et Wu [30] pour traiter une partie de la planification des réseaux mobiles 3G.

Travaux réalisés sur la planification de l'UTRAN

Un des aspects de la planification du sous-système radio comprend le dimensionnement des cellules et leur affectation aux contrôleurs radio que sont les RNC. Il consiste plus en détails à déterminer le type de RNC requis, leur emplacement possible et de trouver un patron d'affectation qui va spécifier pour chaque Nœud B du réseau son RNC d'attache. Pierre et Wu [30] ont présenté un modèle basé sur la programmation par contraintes pour traiter un des problèmes de planification de l'UTRAN, soit l'affectation des Nœuds B vers les RNC. Leur objectif était de déterminer un patron d'affectation des Nœuds B vers les RNC dont les emplacements sont non fixés d'avance (donc à déterminer) et dont les sites potentiels d'emplacement ont un coût mensuel de location. Ils ont formulé leur modèle de la manière suivante : un tableau *affect* [] dont la taille est égale au nombre de Nœuds B du réseau renferme pour chaque Nœud B, le numéro du RNC auquel il est affecté. Un tableau de valeur binaire (0 ou 1) dénoté *disponible* [] de taille égale au nombre de sites potentiels d'emplacement des RNC du réseau est utilisé pour indiquer si un site potentiel de RNC est disponible ou occupé. Un tableau appelé *coûtsliens* [][] est utilisé pour contenir les coûts mensuels de location permettant l'établissement de liens physiques entre un Nœud B donné et son RNC d'attache. Un tableau *coûtsrelèves* [][] contient les coûts de relève entre les différentes cellules (Nœuds B). Et enfin, un tableau *coûtsites* [] contient les coûts mensuels d'amortissement pour chaque site d'emplacement de RNC. Ainsi, pour un réseau constitué de n Nœuds B et de m sites potentiels de localisation de RNC, on obtient la relation suivante :

$$t_{ij} = (\text{affect}[i] == \text{affect}[j]) \quad \text{pour } i, j = 1, \dots, n$$

où t_{ij} prend la valeur 1 si les Nœuds B i et j sont tous les deux connectés au même RNC et 0 sinon. L'opérateur logique « == » est utilisé pour comparer les valeurs dans *affect*[i] et *affect*[j] et le résultat de la comparaison est bien sûr un booléen (0 ou 1). La contrainte pour s'assurer qu'un RNC ne soit plus disponible une fois qu'un Nœud B lui est affecté s'exprime par :

$$\text{disponible}[affect[i]] = 1 \quad \text{pour } i = 1, \dots, n$$

Si λ_i désigne le volume du trafic par unité de temps destiné à la cellule i , les contraintes au niveau du RNC en terme de capacité de prise en charge du trafic qui lui est destiné et d'occupation du nombre total de ses ports s'expriment par :

$$\sum_{i=1}^n \lambda_i (affect[i] == k) \leq M_k \text{disponible}[k] \quad \text{pour } k = 1, \dots, m$$

$$\sum_{i=1}^n (affect[i] == k) \leq N_k \text{disponible}[k] \quad \text{pour } k = 1, \dots, m$$

où M_k désigne la capacité limitée du RNC k en terme de prise en charge de trafic et N_k désigne le nombre total de ports du RNC. La première contrainte stipule que le trafic total provenant des cellules affectées à un RNC k ne doit pas dépasser la capacité M_k de ce RNC. Et la deuxième contrainte stipule que le nombre de connexions actives dirigées vers un RNC k ne doit pas dépasser le nombre total de ports N_k de ce RNC.

Ainsi, la fonction objectif à minimiser s'exprime par :

$$f = \sum_{i=1}^n \text{coûtsliens}[i, affect[i]] + \sum_{k=1}^m \text{disponible}[k] \cdot \text{coûtsites}[k] + \\ \sum_{i=1}^n \sum_{j=1}^n (1 - t_{ij}) \cdot \text{coûtsrelève}[i, j]$$

Cette fonction intègre les différentes contraintes du problème de telle sorte que toute modification à la fonction objectif, se propagera sur les différentes variables définissant les contraintes et de la même manière, toute modification aux contraintes va se refléter sur la fonction objectif. Cela permet d'éliminer, lors de la recherche de solution, certaines zones de recherche qui ne produiront pas de meilleures solutions [29].

Pour simuler leur modèle, Pierre et Wu [30] ont pris une approche hybride en utilisant dans un premier temps la technique de satisfaction de contraintes [29] combinée à un guide de recherche pour problème spécifique. Ensuite, ils ont appliqué la technique

de recherche locale [1] pour améliorer la solution. Et enfin, ils se sont servi de la solution trouvée avec la technique de recherche locale comme borne supérieure et ont appliqué la programmation par contraintes [22] pour chercher une meilleure solution. Le guide de recherche pour problème spécifique consiste à choisir un Nœud B et le plus petit indice dans le tableau des RNC disponibles (C'est à dire le RNC ayant la capacité suffisante pour prendre en charge le trafic total du Nœud B choisi). Une fois le Nœud B sélectionné, utiliser le plus faible coût de liaison pour affecter ce Nœud B. Par contre, si le choix du RNC échoue, ce dernier est retiré du domaine de recherche et le prochain RNC à coût minimum est choisi et le processus reprend. Pour démontrer l'efficacité de leur méthode hybride de résolution, Pierre et Wu [30] ont effectué des tests de comparaison avec l'heuristique H dans [20] et l'adaptation de la Recherche Taboue faite dans [15] dans le contexte des réseaux 2G où ils ont remplacé les MSC du modèle 2G par les RNC de leur modèle 3G et les cellules sont restées les même en tant qu'entité mais bien sûr dans le 3G ils ont un Nœud B dans chaque cellule au lieu des stations de base (BTS) du 2G. Leurs résultats ont montré que chaque solution obtenue est en moyenne entre 5.77% et 7.48 % de sa borne inférieure et que pour des réseaux de grande taille (exemple 300 cellules et 5 RNC) leur méthode hybride améliore la solution de 2.15% par rapport à la Recherche Taboue et de 3.96% par rapport à l'heuristique H.

Maintenant, si l'on veut situer le travail effectué par Pierre et Wu [30] pour la planification du réseau d'accès dans les réseaux mobiles 3G par rapport à nos objectifs pour ce mémoire, nous pouvons dire qu'ils ne traitent qu'une partie du travail qu'on veut effectuer. En effet, ils traitent un des deux sous problèmes d'affectation mentionnés dans le chapitre 1 en particulier l'affectation des Nœuds B vers les RNC. Et notre objectif principal est de traiter le problème d'affectation des cellules dans son entier en considérant non seulement l'affectation des Nœuds B vers les RNC mais aussi l'affectation des RNC vers les commutateurs (MSC et SGSN). Dans leur modèle, ils ne font aucune différence entre le trafic de données et le trafic de voix qui est une caractéristique importante des réseaux 3G. Nous avons l'intention pour la formulation de notre modèle d'en tenir compte et de voir pour un trafic (voix et/ou données) provenant

d'un Nœud B, comment ce dernier est pris en charge par le réseau du Nœud B d'origine jusqu'au dernier équipement du réseau qui le prend en charge avant de l'acheminer vers le destinataire. D'après l'architecture des réseaux 3G, ce dernier équipement est souvent un commutateur (MSC ou SGSN). Également, le modèle proposé par Pierre et Wu [30] tient compte, dans la fonction objectif, des sites potentiels d'emplacement des RNC avec des coûts mensuels d'amortissement, ce que nous n'avons pas l'intention de faire dans notre modèle. En effet, nous allons considérer plutôt des Nœuds B, RNC, MSC et SGSN dont les emplacements sont fixes et connus.

CHAPITRE 3

AFFECTATION DES CELLULES DANS LES RÉSEAUX MOBILES DE TROISIÈME GÉNÉRATION

Une affectation optimale des cellules aux commutateurs dans un réseau mobile permet une utilisation plus efficace des ressources et contribue à améliorer le service offert aux utilisateurs. Il y a eu beaucoup de travaux de recherche pour résoudre le problème d'affectation dans un contexte de réseau 2G et ce chapitre aborde ce problème dans un contexte de réseaux mobiles de troisième génération (3G). Nous allons donc, commencer par présenter certains concepts de base qui nous seront utiles dans la formulation du problème, puis nous proposerons une formulation mathématique pour l'affectation des cellules d'un réseau 3G dans un cas de domiciliation simple. Nous terminerons en proposant une adaptation de l'heuristique Recherche Tabou (RT) pour résoudre le problème.

3.1 Concepts de base

Le problème d'affectation des cellules dans un contexte de réseaux mobiles 3G fait intervenir plusieurs facteurs qu'il convient de préciser dans cette section, entre autres le type de réseau mobile, les équipements qui composent ce réseau, la définition de certains concepts comme le phénomène de relève et le type de domiciliation à considérer. Ainsi, puisqu'on a plus d'une architecture pour les réseaux mobiles de troisième génération, il importe d'effectuer un choix d'architecture de réseau.

3.1.1 Choix de l'architecture du réseau

Présentement, il existe deux propositions d'architecture de réseau 3G qui font l'objet d'étude. Ce sont : l'architecture UMTS et l'architecture CDMA2000 [24]. En analysant ces deux types d'architectures (Figure 2.5 et 3.1), on peut remarquer qu'ils

n'ont pas beaucoup de différences pour ce qui est de leur structuration. En effet, les équipements situés dans le sous-système radio et qui assurent l'inter-connectivité entre les deux sous-systèmes (radio et réseau) sont reliés en même temps à deux équipements de fonctionnalités différentes dans le sous-système réseau. En effet, dans le réseau CDMA2000, on a le BSC qui est relié à un MSC et à un routeur, alors que dans le réseau UMTS, on a le RNC qui est relié à un MSC et à un SGSN. Ainsi, le traitement du problème d'affectation sera semblable dans un cas comme dans l'autre. Sans perte de généralité, nous nous référerons à l'architecture de la norme UMTS pour traiter notre problème d'affectation des cellules aux commutateurs.

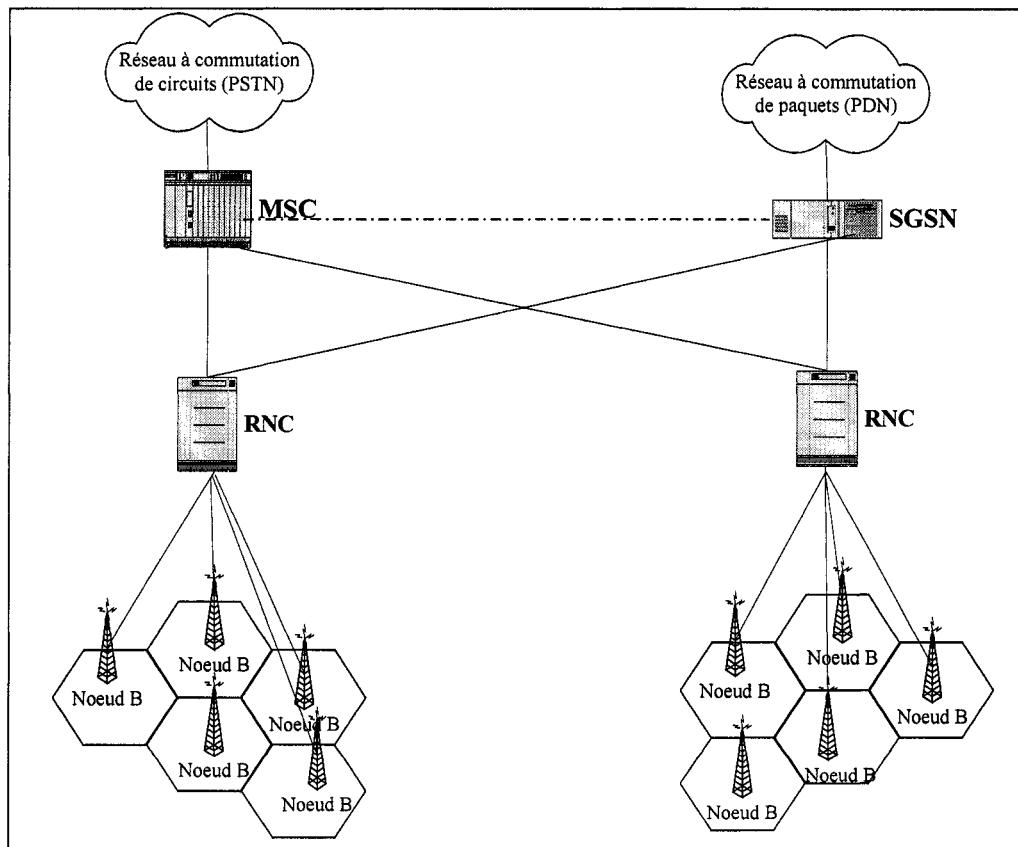


Figure 3.1 Architecture typique d'un réseau UMTS

3.1.2 Concepts de relève simple et de relève complexe

Dans le contexte des réseaux mobiles 2G, les phénomènes de *relève simple* et de *relève complexe* ont été définis en tenant compte seulement de ce qui se passe au niveau des cellules et des commutateurs. Dans le cadre des réseaux mobiles 3G, puisque nous désirons tenir compte des contrôleurs radio (RNC) dans la formulation du problème d'affectation, il nous faut donc préciser ce qui se passe à leur niveau lorsqu'il y a relève.

Ainsi, pour qu'il y ait *relève simple* dans un réseau mobile 3G, il faut qu'il y ait changement de Nœuds B, que ces deux Nœuds B soient reliés à deux RNC différents et que ces deux RNC partagent le même commutateur MSC et le même commutateur SGSN. En effet, ce sont les commutateurs qui déclenchent le mécanisme de relève et pour cela, il faut qu'il y ait changement de RNC. C'est pour cette raison que lors de la relève simple, il faut que le changement de Nœud B soit suivi d'un changement de RNC. Ainsi, les cas d'échanges entre deux Nœuds B i et i' , contrôlés par un même RNC j ne seront pas considérés comme une relève simple car le MSC et le SGSN qui gèrent le RNC j ne voient aucun changement et continuent de recevoir indifféremment le trafic provenant du RNC. La Figure 3.2 illustre le cas d'une relève simple.

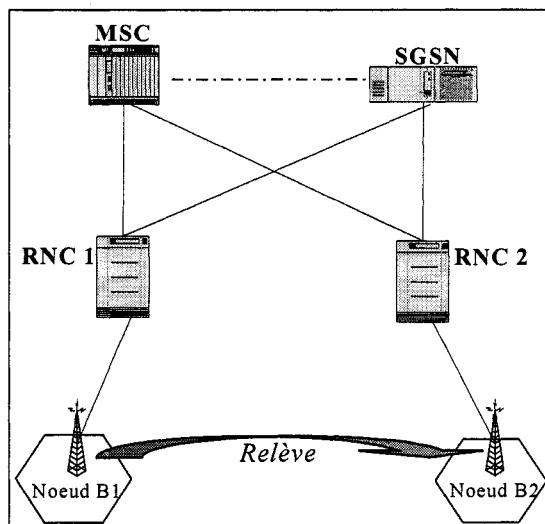


Figure 3.2 Interconnexion du réseau UMTS en relève simple

Pour la *relève complexe*, elle résulte d'échanges entre deux Nœuds B reliés à deux RNC différents et que ces deux RNC ne partagent pas les mêmes commutateurs MSC et SGSN. La Figure 3.3 illustre le cas d'une relève complexe.

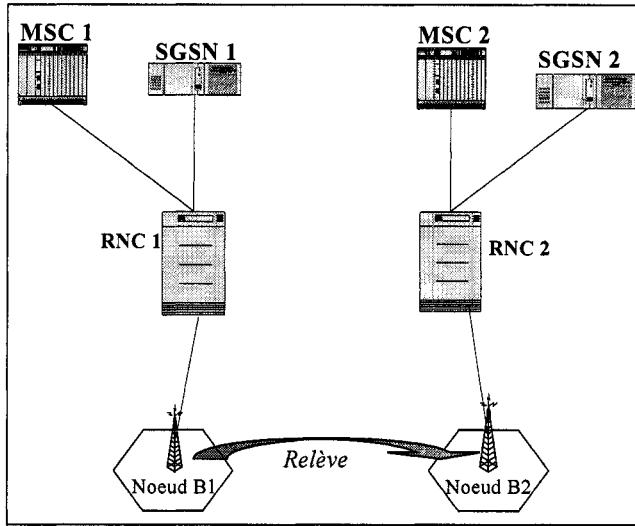


Figure 3.3 Interconnexion du réseau UMTS en relève complexe

3.1.3 Concepts de domiciliation simple et de domiciliation double

Comme dans les réseaux de deuxième génération, l'affectation des cellules aux commutateurs dans les réseaux 3G peut se faire de deux manières. En effet, on peut considérer dans la formulation du problème une domiciliation simple ou une domiciliation double pour chacune de nos cellules. En général, on parle de domiciliation simple lorsque, pour deux groupes d'équipements de niveaux d'emplacement différents qu'on veut interconnecter, chaque équipement du niveau inférieur est affecté à un et un seul équipement du niveau supérieur. En domiciliation double, chaque équipement du niveau inférieur est affecté à deux équipements du niveau supérieur mais avec un seul équipement actif à la fois.

L'architecture du réseau UMTS comporte trois niveaux d'équipements : le premier niveau est constitué des Nœuds B, le deuxième niveau des contrôleurs radio

(RNC) et le troisième niveau des commutateurs (MSC et SGSN). Ainsi, puisque nous voulons interconnecter ces trois niveaux d'équipements, cela nous donne quatre cas possibles de domiciliation en considérant ce qui se passe dans le réseau tout entier. En effet, on peut avoir le premier cas où chaque Nœud B est affecté à un seul RNC et que chaque RNC est affecté à un seul MSC et à un seul SGSN. On peut aussi avoir le deuxième cas où chaque Nœud B est affecté à deux RNC et que chaque RNC est à son tour affecté à deux MSC et à deux SGSN. Le troisième cas est celui où chaque Nœud B est affecté à un seul RNC et que chaque RNC est affecté à deux MSC et à deux SGSN. Enfin, le dernier cas est celui où chaque Nœud B est affecté à deux RNC et chaque RNC est affecté à un MSC et à un SGSN. En considérant les cas de figure mentionnés ci-haut, on peut dégager trois types de domiciliation : la *domiciliation simple*, la *domiciliation double* et la *domiciliation mixte*. Ainsi, on parle de *domiciliation simple* lorsque chacun des Nœuds B est affecté à un seul RNC et que chaque RNC est affecté à son tour à un MSC et à un SGSN. En *domiciliation double*, chacun des Nœuds B est affecté à deux RNC et chacun des RNC est affecté à deux MSC et à deux SGSN (un seul RNC, un seul MSC et un seul SGSN sont actifs à la fois). Enfin en *domiciliation mixte*, comme son nom l'indique c'est une combinaison des deux premiers cas de domiciliation. En effet, on a soit le cas où chaque Nœud B est affecté à un seul RNC et chaque RNC est affecté à deux MSC et à deux SGSN, ou alors le cas où chaque Nœud B est affecté à deux RNC et chaque RNC est affecté à un MSC et à un SGSN. Dans ce mémoire, nous nous concentrerons sur la formulation du problème pour le cas de la domiciliation simple.

3.2 Formulation du problème

Le problème d'affectation de cellules dans les réseaux de troisième génération consiste à déterminer un patron d'affectation des Nœuds B, placés dans les cellules, aux contrôleurs radio (RNC) et de ces RNC aux commutateurs (MSC et SGSN) dans le but de minimiser une certaine fonction de coût, tout en respectant un certain nombre de contraintes. Mais avant de présenter notre formulation du problème, nous poserons quelques hypothèses qui seront utiles à l'expression de la fonction de coût à minimiser,

ensuite nous présenterons les détails de la formulation pour le cas de la domiciliation simple.

3.2.1 Hypothèses

Nous poserons les hypothèses suivantes concernant l'architecture du réseau UMTS :

- Le réseau est constitué de n Nœuds B installés chacun dans une des cellules du réseau, r contrôleurs RNC, m commutateurs MSC et s commutateurs SGSN;
- Chaque Nœud B est connecté à un et un seul RNC;
- Chaque RNC est à la fois connecté à un MSC et à un SGSN;
- On fait abstraction des autres équipements du réseau car certains d'entre eux, en particulier les GGSN, sont situés à un niveau supérieur dans le réseau. Le problème d'interconnexion de ces équipements avec ceux pris en compte dans ce mémoire pourrait faire l'objet d'une autre étude.

Également, il est important de remarquer que la décomposition de l'architecture du réseau en niveaux d'équipements influencera la formulation du problème. On aura donc deux sous-problèmes d'affectation : l'affectation des Nœuds B aux RNC, puis l'affectation des RNC aux MSC et aux SGSN.

3.2.2 Formulation mathématique proposée pour la domiciliation simple

Soit un réseau de n Nœuds B à affecter à r contrôleurs RNC et ces r RNC à affecter à m commutateurs MSC et à s commutateurs SGSN. La localisation des Nœuds B, des RNC, des MSC et des SGSN est fixe et connue. Soient $I = \{1, 2, 3, 4, \dots, n\}$ l'ensemble des Nœuds B de notre réseau, $J = \{1, 2, 3, 4, \dots, r\}$ l'ensemble des RNC, $K = \{1, 2, \dots, m\}$ l'ensemble des MSC/VLR et $L = \{1, 2, 3, 4, \dots, s\}$ l'ensemble des SGSN. Soient c_{ij} le coût d'amortissement de la liaison entre le Nœud B i ($i \in I$) et le RNC j ($j \in J$), c'_{jk} le coût d'amortissement de la liaison entre le RNC j et le MSC k

$(k \in K)$ et c_{jl}'' le coût d'amortissement de la liaison entre le RNC j et le SGSN l ($l \in L$).

Soient Cap_j la capacité en commutation de circuit du RNC j , Cap_j' la capacité en commutation de paquets (bps) du RNC j , Cap_k'' la capacité du MSC k en nombre d'appels par unité de temps et Cap_l''' la capacité du SGSN l en bits par seconde. Soient f_{jk} l'intensité du trafic de voix provenant du RNC j et destiné au MSC k , f_{jl}' l'intensité du trafic de données provenant du RNC j et destiné au SGSN l , f_{ij}^{voix} l'intensité du trafic de voix provenant du Nœud B i et destiné au RNC j et $f_{ij}^{données}$ l'intensité du trafic de données provenant du Nœud B i et destiné au RNC j . Soit $H_{ii'}^{msc}$ le coût par unité de temps d'une relève simple entre les Nœuds B i et i' impliquant un seul MSC, $H_{ii'}^{msc'}$ le coût par unité de temps d'une relève complexe entre les Nœuds B i et i' impliquant deux MSC, $H_{ii'}^{sgsn}$ le coût par unité de temps d'une relève simple entre les Nœuds B i et i' impliquant un seul SGSN, et $H_{ii'}^{sgsn'}$ le coût par unité de temps d'une relève complexe entre les Nœuds B i et i' impliquant deux SGSN.

Soient :

$$x_{ij} = \begin{cases} 1 & \text{si le Nœud B } i \text{ est relié au RNC } j \text{ (} i \in I \text{ et } j \in J), \\ 0 & \text{sinon} \end{cases}$$

$$x_{jk}' = \begin{cases} 1 & \text{si le RNC } j \text{ est relié au MSC } k \text{ (} j \in J \text{ et } k \in K) \\ 0 & \text{sinon} \end{cases}$$

$$x_{jl}'' = \begin{cases} 1 & \text{si le RNC } j \text{ est relié au SGSN } l \text{ (} j \in J \text{ et } l \in L), \\ 0 & \text{sinon} \end{cases}$$

Soient $Z_{ii'j}$ et $y_{ii'}$ définis par :

$$z_{ii'j} = x_{ij}x_{i'j} \quad \text{avec } i, i' \in I \text{ et } j \in J, \text{ et } i \neq i'$$

$$y_{ii'} = \sum_{j \in J} z_{ii'j} \quad \text{avec } i, i' \in I, \text{ et } i \neq i'$$

$z_{ii'j}$ est égale à 1 si les Nœuds B i et i' , avec $i \neq i'$, sont tous deux connectés au même RNC j et 0 autrement. $y_{ii'}$ prend la valeur 1 si les Nœuds B i et i' sont tous deux connectés seulement à un seul et même RNC parmi l'ensemble des RNC, et la valeur 0 s'ils sont connectés à des RNC différents. Soient $Z_{jj'k}$ et $Y_{jj'}^{msc}$ définis par :

$$Z_{jj'k} = x_{jk}^{\prime} x_{j'k}^{\prime} \quad \text{avec } j, j' \in J \text{ et } k \in K, \text{ et } j \neq j'$$

$$Y_{jj'}^{msc} = \sum_{k \in K} Z_{jj'k} \quad \text{avec } j, j' \in J, \text{ et } j \neq j'$$

$Z_{jj'k}$ est égale à 1 si les RNC j et j' , avec $j \neq j'$, sont tous deux connectés au même MSC k et est égale à 0 s'ils sont connectés à des MSC différents. $Y_{jj'}^{msc}$ prend la valeur 1 si les RNC j et j' sont tous deux connectés au même MSC, et 0 sinon.

Soient $Z_{jj'l}$ et $Y_{jj'}^{sgsn}$ définis par :

$$Z_{jj'l} = x_{jl}^{\prime\prime} x_{j'l}^{\prime\prime} \quad \text{avec } j, j' \in J \text{ et } l \in L, \text{ et } j \neq j'$$

$$Y_{jj'}^{sgsn} = \sum_{l \in L} Z_{jj'l} \quad \text{avec } j, j' \in J, \text{ et } j \neq j'$$

$Z'_{jj'}$ est égale à 1 si les RNC j et j' , avec $j \neq j'$, sont tous deux connectés au même SGSN l et est égale à 0 s'ils sont connectés à des SGSN différents. $Y_{jj'}^{sgsn}$ prend la valeur 1 si les RNC j et j' sont tous deux connectés au même SGSN, et 0 sinon.

La présence de deux niveaux d'interconnexion dans l'architecture du réseau nous amène à considérer une double affectation : d'abord l'affectation des Nœuds B aux RNC puis l'affectation des RNC aux MSC et aux SGSN. Ainsi, la fonction de coût total sera divisée en deux parties. La première portion du coût constituera le coût pour affecter les Nœuds B aux RNC. Dans ce coût, on comptabilisera le coût de câblage pour relier chaque Nœud B à un RNC. Normalement, on devrait aussi comptabiliser les coûts engendrés par les changements de RNC résultant des relèves fréquentes entre les Nœuds B. Mais comme au niveau des RNC, on n'a pas de mécanisme de gestion des relèves, il est donc impossible de faire une évaluation exacte du coût de relève. Nous allons donc considérer que ce sont des relèves à coûts minimes par rapport aux coûts des relèves qui se font au niveau des commutateurs (MSC et SGSN). De plus, les RNC ne sont que des contrôleurs radio, donc il n'y a pas de base de données à manipuler, contrairement aux MSC et aux SGSN. Ainsi, on aura dans l'expression de la première portion de la fonction de coût, seulement le coût de câblage pour relier les Nœuds B aux RNC.

La deuxième portion du coût sera le coût d'affectation des RNC aux MSC et aux SGSN. Dans ce coût, on tiendra compte des coûts de câblage des RNC vers les MSC et des coûts de câblage des RNC vers les SGSN. Mais contrairement au premier sous-problème d'affectation, ici, les MSC et les SGSN déclenchent les mécanismes de relève pour assurer les communications et les services des usagers lorsque ces derniers se déplacent à l'intérieur du réseau. Ainsi, en plus du coût des liaisons, on tiendra aussi compte des coûts de relève engendrés par le changement de MSC et/ou de SGSN. On notera f_1 , f_2 respectivement la première et la deuxième portion de la fonction totale de coût à minimiser. f_1 s'exprime de la manière suivante:

$$f_1 = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

et représente le coût total d'interconnexion des Nœuds B aux RNC. La fonction de coût f_2 représentant le coût d'affectation des RNC aux MSC et aux SGSN s'exprime de la manière suivante :

$$\begin{aligned} f_2 = & \sum_{j \in J} \sum_{k \in K} c'_{jk} x'_{jk} + \sum_{j \in J} \sum_{l \in L} c''_{jl} x''_{jl} + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} H_{ii'}^{msc} (1 - y_{ii'}) Y_{jj'}^{msc} + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} H_{ii'}^{msc} (1 - y_{ii'}) (1 - Y_{jj'}^{msc}) \\ & + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} H_{ii'}^{sgsn} (1 - y_{ii'}) Y_{jj'}^{sgsn} + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} H_{ii'}^{sgsn} (1 - y_{ii'}) (1 - Y_{jj'}^{sgsn}) \end{aligned}$$

Les deux premiers termes de f_2 sont respectivement les coûts totaux des liaisons entre les RNC et les MSC et des liaisons entre les mêmes RNC et les SGSN. Les troisième et quatrième termes représentent respectivement le coût des relèves simples et complexes entre les Nœuds B impliquant des MSC; les cinquième et sixième termes représentent respectivement le coût des relèves simples et complexes entre les Nœuds B impliquant des SGSN. On note que pour comptabiliser un coût de relève, il faut que les deux Nœuds B soient reliés à des RNC différents ($y_{ii'} = 0$) pour avoir $(1 - y_{ii'}) = 1$ et les RNC à leur tour peuvent être, soit reliés au même commutateur (MSC ou SGSN) pour la relève simple (dans ce cas $Y_{jj'}^{msc} = 1$ ou $Y_{jj'}^{sgsn} = 1$) ou à des commutateurs différents pour la relève complexe (dans ce cas, $Y_{jj'}^{msc} = 0$ ou $Y_{jj'}^{sgsn} = 0$; d'où $(1 - Y_{jj'}^{msc}) = 1$ ou $(1 - Y_{jj'}^{sgsn}) = 1$). Ce qui est consistant avec la définition donnée plutôt de la relève simple et de la relève complexe. Également, on remarque que la fonction f_2 est quadratique en x_{ij} , x'_{jk} et x''_{jl} car $y_{ii'}$ est une fonction quadratique en x_{ij} , $Y_{jj'}^{msc}$ est quadratique en x'_{jk} et $Y_{jj'}^{sgsn}$ est quadratique en x''_{jl} . Le problème général revient à minimiser la fonction de coût total :

$$f = f_1 + f_2$$

L'affectation des cellules aux commutateurs est sujette à un certain nombre de contraintes. En effet, chaque Nœud B doit être affecté à un et un seul RNC. Cela se traduit par la relation suivante :

$$\sum_{j \in J} x_{ij} = 1 \quad \text{avec } i \in I \quad (3.1)$$

Également, chaque RNC doit être affecté à un et un seul MSC et à un et un seul SGSN. Cela se traduit respectivement par les deux relations suivantes :

$$\sum_{k \in K} x'_{jk} = 1 \quad \text{avec } j \in J \quad (3.2)$$

$$\sum_{l \in L} x''_{jl} = 1 \quad \text{avec } j \in J \quad (3.3)$$

Puisque dans les réseaux de troisième génération, les usagers produisent deux types de trafic, soient le trafic conversationnel et le trafic de données, alors l'ensemble des deux trafics provenant des Nœuds B reliés au RNC j ne doit pas dépasser la capacité de ce RNC en terme de prise en charge du trafic de voix et du trafic de données. Cela se traduit par les deux contraintes suivantes :

$$\sum_{i \in I} f_{ij}^{voix} x_{ij} \leq Cap_j \quad \text{avec } j \in J \quad (3.4)$$

$$\sum_{i \in I} f_{ij}^{données} x_{ij} \leq Cap'_j \quad \text{avec } j \in J \quad (3.5)$$

De la même manière, l'ensemble du trafic de voix provenant des RNC reliés au MSC k ne doit pas dépasser la capacité de ce MSC. Cela se traduit par la contrainte suivante:

$$\sum_{j \in J} f_{jk} x'_{jk} \leq Cap''_k \quad \text{avec } k \in K \quad (3.6)$$

Également, l'ensemble du trafic de données par unité de temps provenant des RNC reliés au SGSN l ne doit pas dépasser la capacité de ce SGSN. Cela se traduit par la contrainte suivante:

$$\sum_{j \in J} f'_{jl} x''_{jl} \leq Cap'''_l \quad \text{avec } l \in L \quad (3.7)$$

Enfin, pour compléter les contraintes du problème, on a :

$$z_{ii'j} = x_{ij}x_{i'j} \quad \text{avec } i, i' \in I \text{ et } j \in J, \text{ et } i \neq i' \quad (3.8)$$

$$y_{ii'} = \sum_{j \in J} z_{ii'j} \quad \text{avec } i, i' \in I, \text{ et } i \neq i' \quad (3.9)$$

$$Z_{jj'k} = x_{jk}^{\top} x_{j'k}^{\top} \quad \text{avec } j, j' \in J \text{ et } k \in K, \text{ et } j \neq j' \quad (3.10)$$

$$Y_{jj'}^{msc} = \sum_{k \in K} Z_{jj'k} \quad \text{avec } j, j' \in J, \text{ et } j \neq j' \quad (3.11)$$

$$Z_{jj'l}^{\top} = x_{jl}^{\top} x_{j'l}^{\top} \quad \text{avec } j, j' \in J \text{ et } l \in L, \text{ et } j \neq j' \quad (3.12)$$

$$Y_{jj'}^{sgsn} = \sum_{l \in L} Z_{jj'l}^{\top} \quad \text{avec } j, j' \in J, \text{ et } j \neq j' \quad (3.13)$$

En résumé, le problème revient à minimiser la fonction de coût total $f = f_1 + f_2$ sous les contraintes (3.1) à (3.13). Ainsi formulé, le problème ne peut être résolu avec les méthodes traditionnelles de programmation linéaire à cause de la non linéarité des contraintes (3.8), (3.10) et (3.12). On va donc remplacer chacune de ces contraintes par un ensemble de contraintes afin de linéariser le problème. Ainsi, (3.8) sera remplacée par l'ensemble des contraintes suivant :

$$z_{ii'j} \leq x_{ij} \quad (3.14)$$

$$z_{ii'j} \leq x_{i'j} \quad (3.15)$$

$$z_{ii'j} \geq x_{ij} + x_{i'j} - 1 \quad (3.16)$$

$$z_{ii'j} \geq 0 \quad (3.17)$$

La contrainte (3.10) sera remplacée par :

$$Z_{jj'k} \leq x'_{jk} \quad (3.18)$$

$$Z_{jj'k} \leq x'_{j'k} \quad (3.19)$$

$$Z_{jj'k} \geq x'_{jk} + x'_{j'k} - 1 \quad (3.20)$$

$$Z_{jj'k} \geq 0 \quad (3.21)$$

Et la contrainte (3.12) sera remplacée par :

$$Z'_{jj'l} \leq x''_{jl} \quad (3.22)$$

$$Z'_{jj'l} \leq x''_{j'l} \quad (3.23)$$

$$Z'_{jj'l} \geq x''_{jl} + x''_{j'l} - 1 \quad (3.24)$$

$$Z'_{jj'l} \geq 0 \quad (3.25)$$

Il est également possible de simplifier davantage la fonction de coût f_2 , en reformulant la partie relative au coût de relève entre les Nœuds B. En effet, on peut, au lieu de considérer séparément le coût dû à une relève simple et celui dû à une relève complexe, définir de nouvelles variables qui constitueront le coût réduit par unité de temps d'une relève complexe entre les Nœuds B i et i' . Puisqu'on a des relèves qui impliquent des MSC et/ou des SGSN, alors on notera $h_{ii'}^{msc}$ et $h_{ii'}^{sgsn}$ respectivement le coût réduit par unité de temps d'une relève complexe entre les Nœuds B i et i' impliquant deux MSC et le coût réduit par unité de temps d'une relève complexe entre les Nœuds B i et i' impliquant deux SGSN. On a les expressions suivantes de $h_{ii'}^{msc}$ et $h_{ii'}^{sgsn}$ par rapport aux coûts de relève simple $H_{ii'}^{msc}$ et $H_{ii'}^{sgsn}$ et de relève complexe $H'_{ii'}^{msc}$ et $H'_{ii'}^{sgsn}$:

$$h_{ii'}^{msc} = H'_{ii'}^{msc} - H_{ii'}^{msc}$$

$$h_{ii'}^{sgsn} = H'_{ii'}^{sgsn} - H_{ii'}^{sgsn}$$

La fonction de coût f_2 peut donc être réécrite de la manière suivante :

$$\begin{aligned} f_2 = & \sum_{j \in J} \sum_{k \in K} c_{jk}^1 x_{jk}^1 + \sum_{j \in J} \sum_{l \in L} c_{jl}^2 x_{jl}^2 + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{msc} (1 - y_{ii'}) (1 - Y_{jj'}^{msc}) \\ & + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{sgsn} (1 - y_{ii'}) (1 - Y_{jj'}^{sgsn}) \end{aligned}$$

Et la fonction de coût total f à minimiser devient :

$$\begin{aligned} f = & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} c_{jk}^1 x_{jk}^1 + \sum_{j \in J} \sum_{l \in L} c_{jl}^2 x_{jl}^2 + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{msc} (1 - y_{ii'}) (1 - Y_{jj'}^{msc}) \\ & + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{sgsn} (1 - y_{ii'}) (1 - Y_{jj'}^{sgsn}) \end{aligned}$$

Le Tableau 3.1 présente le résumé de la formulation mathématique du problème d'affectation. En dépit des transformations, le problème d'affectation demeure assez complexe à résoudre. Dans la prochaine section, nous allons justement étudier sa complexité et voir quelle méthode est plus appropriée pour le résoudre.

3.2.3 Analyse de complexité du modèle mathématique

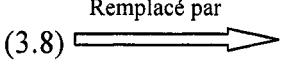
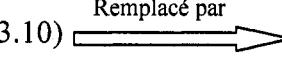
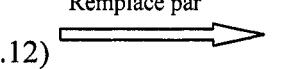
Afin d'être en mesure de choisir une méthode de résolution pour notre problème d'affectation, nous devons étudier sa complexité. Le problème général d'affectation des Nœuds B aux RNC et des RNC aux MSC et SGSN, tel que nous l'avons formulé à la section 3.2.2, est difficile à résoudre. En effet, il s'agit de faire une double affectation d'abord des Nœuds B aux RNC et des RNC aux MSC et aux SGSN du réseau. Et en analysant la première sous-affectation, soit des Nœuds B aux RNC, on constate qu'elle ressemble beaucoup au problème d'affectation des cellules aux commutateurs dans les réseaux mobiles de deuxième génération présenté dans le chapitre 2. En effet, on avait n cellules qu'il fallait affecter à m commutateurs et il a été démontré qu'un algorithme à énumération exhaustive examinait m^n combinaisons, ce qui nécessitait un temps considérable pour trouver une solution. C'est un problème qui a été prouvé NP-difficile.

Tableau 3.1 Résumé de la formulation du problème

Minimiser :

$$f = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} c'_{jk} x'_{jk} + \sum_{j \in J} \sum_{l \in L} c''_{jl} x''_{jl} + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{msc} (1 - y_{ii'}) (1 - Y_{jj'}^{msc}) \\ + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{sgsn} (1 - y_{ii'}) (1 - Y_{jj'}^{sgsn})$$

Sujet à :

$\square \quad \sum_{j \in J} x_{ij} = 1 \quad \text{avec } i \in I$	(3.1)	$\sum_{i \in I} f_{ij}^{\text{dommées}} x_{ij} \leq Cap_j' \quad \text{avec } j \in J$	(3.5)
$\square \quad \sum_{k \in K} x'_{jk} = 1 \quad \text{avec } j \in J$	(3.2)	$\sum_{j \in J} f_{jk} x'_{jk} \leq Cap_k'' \quad \text{avec } k \in K$	(3.6)
$\square \quad \sum_{l \in L} x''_{jl} = 1 \quad \text{avec } j \in J$	(3.3)	$\sum_{j \in J} f_{jl} x''_{jl} \leq Cap_l''' \quad \text{avec } l \in L$	(3.7)
$\square \quad \sum_{i \in I} f_{ij}^{\text{voix}} x_{ij} \leq Cap_j \quad \text{avec } j \in J$	(3.4)		
$\square \quad z_{ii'j} = x_{ij} x_{i'j}$	(3.8)  Remplacé par	$z_{ii'j} \leq x_{ij}$	(3.14)
		$z_{ii'j} \leq x_{i'j}$	(3.15)
		$z_{ii'j} \geq x_{ij} + x_{i'j} - 1$	(3.16)
		$z_{ii'j} \geq 0$	(3.17)
$\square \quad y_{ii'} = \sum_{j \in J} z_{ii'j} \quad \text{avec } i, i' \in I, \text{ et } i \neq i'$	(3.9)		
$\square \quad Z_{jj'k} = x'_{jk} x'_{j'k}$	(3.10)  Remplacé par	$Z_{jj'k} \leq x'_{jk}$	(3.18)
		$Z_{jj'k} \leq x'_{j'k}$	(3.19)
		$Z_{jj'k} \geq x'_{jk} + x'_{j'k} - 1$	(3.20)
		$Z_{jj'k} \geq 0$	(3.21)
$\square \quad Y_{jj'}^{msc} = \sum_{k \in K} Z_{jj'k} \quad \text{avec } j, j' \in J, \text{ et } j \neq j'$	(3.11)		
$\square \quad Z_{jj'l} = x''_{jl} x''_{j'l}$	(3.12)  Remplacé par	$Z_{jj'l} \leq x''_{jl}$	(3.22)
		$Z_{jj'l} \leq x''_{j'l}$	(3.23)
		$Z_{jj'l} \geq x''_{jl} + x''_{j'l} - 1$	(3.24)
		$Z_{jj'l} \geq 0$	(3.25)
$\square \quad Y_{jj'}^{sgsn} = \sum Z_{jj'l} \quad \text{avec } j, j' \in J, \text{ et } j \neq j'$	(3.13)		

Dans notre cas, nous avons dans la première sous-affectation n Nœuds B que nous voulons affecter à r RNC, ce qui nécessitera avec un algorithme à énumération exhaustive l'examen de r^n combinaisons pour résoudre seulement une partie de notre problème. Ensuite, il faudra aussi affecter ces r RNC à m commutateurs MSC et à s commutateurs SGSN, ce qui nécessitera d'examiner un autre $(m+s)^r$ combinaisons. Avec ce temps de calcul qui croît exponentiellement, on ne serait pas à mesure de trouver une solution en un temps raisonnable. Ce qui nous amène à dire que, dans l'ensemble, le problème est NP-difficile et requiert l'utilisation d'une approche heuristique pour sa résolution pratique. Parmi les heuristiques connues et présentées dans le chapitre 2, nous avons choisi d'utiliser la Recherche Taboue dont nous présentons dans la section suivante les adaptations à apporter pour la résolution de notre problème.

3.3 Adaptation de la Recherche Taboue

Tel que présentée dans [15], l'adaptation de la recherche taboue à la résolution du problème d'affectation fonctionne en appliquant séquentiellement trois mécanismes de mémoire : *la mémoire à court terme*, *la mémoire à moyen terme* et *la mémoire à long terme*. Dans les deux premiers mécanismes, la méthode effectue des opérations appelées *mouvements* pour passer d'une solution à une autre dans sa recherche de la solution optimale. Enfin, pour éviter de refaire les mêmes mouvements, la méthode garde une liste de mouvements dits *tabous*. Dans cette section, notre objectif est de préciser spécifiquement pour chaque mécanisme de mémoire, comment l'adapter à notre modèle de résolution du problème d'affectation en définissant entre autres les types de mouvements possibles et les gains qui leurs sont associés. Mais, tout d'abord, en fonction de notre réseau 3G, nous allons définir le principe général de fonctionnement de l'algorithme RT et présenter l'algorithme de génération de la solution initiale.

3.3.1 Principe de fonctionnement

Notre adaptation de la recherche taboue construit au départ une solution initiale à partir des données du problème (à noter qu'il sera aussi possible de fournir une solution initiale déjà construite, cela n'aura aucun effet sur le fonctionnement de l'algorithme RT ni sur la solution finale qu'on obtiendra). Pour effectuer les affectations, nous procéderons du niveau d'équipement le plus bas, c'est à dire les Nœuds B, vers le niveau le plus haut constitué des commutateurs MSC et SGSN. Pour la génération de nouvelles solutions, la RT effectuera des mouvements à l'intérieur d'un espace de recherche défini (on verra plus tard en quoi consistent ces mouvements). Afin de pouvoir examiner le plus de solutions possibles et augmenter les chances d'aboutir à de bonnes solutions, nous libérerons l'espace de recherche des contraintes de capacité sur les contrôleurs (RNC) et sur les commutateurs (MSC et SGSN). Cela affectera peut être la faisabilité des solutions obtenues au cours de la recherche, surtout lors de l'application du mécanisme de mémoire à court terme, mais c'est un compromis à faire. Cependant, nous verrons que l'application du mécanisme de mémoire à moyen terme permettra d'effectuer des mouvements pour rétablir ces contraintes de capacité. Et pour diminuer le risque d'avoir trop des solutions non-faisables, nous introduirons, comme dans [15], une sanction en terme de coût fixe additionnel pour le non respect des contraintes de capacité. À noter que les contraintes d'affectation unique, quant à elles, seront toujours respectées dans chacun des patrons d'affectations.

Ainsi, à chaque solution, la méthode va associer deux valeurs numériques dont la première est le coût intrinsèque de la solution, calculé à partir de la fonction objectif, et la deuxième comprend le coût de la fonction objectif plus la sanction pour le non respect des contraintes de capacité. Après chaque étape, la méthode choisira la solution ayant la meilleure évaluation. Lorsqu'elle tombe sur un optimum local, l'algorithme va choisir la solution voisine qui dégradera le moins la fonction objectif. Pour éviter les cycles autour de cet optimum, une liste taboue gardera les k dernières solutions et interdira momentanément un retour vers ces solutions. Les solutions taboues seront libérées après

I_{max} itérations ou lorsqu'elles satisferont un certain critère d'aspiration. La composante de mémoire à moyen terme gardera les m dernières meilleures solutions pour, plus tard, intensifier la recherche dans leurs voisinages. En se basant sur les statistiques de recherche, la composante de mémoire à long terme déterminera un nouveau point de départ de la recherche contrastant le plus possible avec les solutions déjà visitées.

3.3.2 Génération de la solution initiale

À défaut d'une solution initiale déjà construite, il faut en construire une en fonction des paramètres d'entrée du réseau comme les coûts des liaisons, les capacités des équipements etc. Lors de la génération de cette solution initiale, l'algorithme procédera d'abord à l'affectation des Nœuds B aux contrôleurs RNC puis à l'affectation des RNC aux MSC et aux SGSN. Le domaine d'exploitation sera libre des contraintes de capacité sur les contrôleurs radio (RNC) et sur les commutateurs (MSC et SGSN). Ainsi, la solution initiale doit seulement respecter les contraintes d'affectation unique des Nœuds B aux RNC et d'affectation unique des RNC aux MSC et aux SGSN. Afin d'illustrer comment se présentera notre solution initiale, nous allons utiliser comme exemple un réseau de 10 Nœuds B, 6 RNC, 3 MSC et 3 SGSN dont les coûts des liaisons entre les différents équipements sont présentés aux tableaux 3.2, 3.3 et 3.4.

Tableau 3.2 Coûts de liaison entre les Nœuds B et les RNC

Nœud B	RNC					
	0	1	2	3	4	5
0	1	2	2	3	3	2
1	0	1	1	0	0	0
2	2	3	3	1	1	1
3	1	0	1	1	1	1
4	1	1	0	1	1	1
5	1	1.3	1	1	1	0
6	0	1	2	2	1	0
7	0	0	0	0	1	0
8	1	1	1	0	0	1
9	1.1	0	0.2	0.3	0.9	0.2

Tableau 3.3 Coûts de liaison entre les RNC et les MSC

RNC	MSC		
	0	1	2
0	2	2	0
1	0	2	0
2	3	0	2
3	2	0	1
4	2	1	1
5	2	0	1

Tableau 3.4 Coûts de liaison entre les RNC et les SGSN

RNC	SGSN		
	0	1	2
0	0	2	0
1	1	1	0
2	2	1	3
3	0	0	2
4	2	1	0
5	1	3	1

Nous allons légèrement modifier l'algorithme de génération d'une solution initiale pour la recherche taboue présenté dans [16], dans le contexte des réseaux 2G, afin de l'adapter à notre problème. Le nouvel algorithme est présenté à la Figure 3.4. Il commence d'abord par affecter chaque Nœud B au RNC le plus proche en terme de distance les séparant; ensuite, de la même manière, il affecte les RNC aux commutateurs MSC et SGSN les plus proches. Entre les MSC et les SGSN, il n'y a aucun ordre de priorité d'affectation car cela n'aura aucune incidence sur les résultats. Si un Nœud B se trouve à égale distance de deux RNC, il est affecté au premier RNC dans l'ordre spécifié par le problème. Le même principe s'applique lors de l'affectation des RNC aux MSC et aux SGSN. Pour les MSC et SGSN, nous utiliserons une seule variable pour représenter la capacité résiduelle de l'équipement. Quant aux RNC, nous utiliserons deux variables : une pour mesurer la capacité résiduelle en trafic de voix et une autre pour le trafic de données. Au départ de la simulation, chaque variable résiduelle sera initialisée à la pleine

capacité de l'équipement en question. Ainsi, lorsqu'on affecte un Nœud B à un RNC, les variables de capacité résiduelle en trafic de voix et de données du RNC seront décrémentées respectivement des valeurs du volume du trafic de voix et de données produit par le Nœud B. Pour les MSC et SGSN, lorsqu'on leur affecte un RNC, leurs capacités résiduelles seront décrémentées respectivement de la valeur du volume du trafic de voix et de celle du volume du trafic de données produit par le RNC.

```

Initialisation : Obtenir les matrices des coûts de liaison
Pour chaque Nœud B i faire :
    Meilleur := +infini;
    Meilleur_Rnc := 0;
    Pour chaque RNC j faire :
        Si  $c_{ij} < \text{Meilleur}$ 
            alors Meilleur :=  $c_{ij}$  ;
            Meilleur_Rnc :=  $j$ ;
        Fin si
    Fin pour
    Affecter le Nœud B  $i$  au RNC Meilleur_Rnc;
Fin pour
Pour chaque RNC j faire :
    Meilleur := +infini;
    Meilleur_Msc := 0;
    Pour chaque MSC k faire :
        Si  $c'_{jk} < \text{Meilleur}$ 
            alors Meilleur :=  $c'_{jk}$  ;
            Meilleur_Msc :=  $k$ ;
        Fin si
    Fin pour
    Affecter le RNC  $j$  au MSC Meilleur_Msc;
    Meilleur := +infini;
    Meilleur_Sgsn := 0;
    Pour chaque SGSN l faire :
        Si  $c''_{jl} < \text{Meilleur}$ 
            alors Meilleur :=  $c''_{jl}$  ;
            Meilleur_Sgsn :=  $l$ ;
        Fin si
    Fin pour
    Affecter le RNC  $j$  au SGSN Meilleur_Sgsn;
Fin pour
```

Figure 3.4 Pseudo code modifié de génération de la solution initiale

En appliquant l'algorithme de la Figure 3.4 avec les données des tableaux 3.2, 3.3 et 3.4, on obtient la solution initiale illustrée aux tableaux 3.5, 3.6 et 3.7. La solution fournit l'information sur l'interconnexion du réseau et peut être interprétée de la manière suivante :

- pour l'affectation des Nœuds B aux RNC : si un Nœud B à la ligne i du Tableau 3.5 est affecté à un RNC de la colonne j , la valeur 1 est inscrite dans la case (i, j) , autrement la valeur 0 y est inscrite (dans notre formule présentée au Tableau 3.1, cela correspond à la variable x_{ij}) ;
- pour l'affectation des RNC aux commutateurs MSC : si un RNC à la ligne j du Tableau 3.6 est affecté au MSC de la colonne k , la valeur 1 est inscrite dans la case (j, k) , autrement c'est la valeur 0 qui y est inscrite (dans notre formule présentée au Tableau 3.1, cela correspond à la variable x_{jk}) ;
- pour l'affectation des RNC aux commutateurs SGSN : si un RNC à la ligne j du Tableau 3.7 est affecté au SGSN de la colonne l , la valeur 1 est inscrite dans la case (j, l) , autrement c'est la valeur 0 qui y est inscrite (dans notre formule présentée au Tableau 3.1, cela correspond à la variable x_{jl}) .

Tableau 3.5 Résultats de l'affectation des 10 Nœuds B aux 6 RNC

Nœud B	RNC					
	0	1	2	3	4	5
0	1	0	0	0	0	0
1	1	0	0	0	0	0
2	0	0	0	1	0	0
3	0	1	0	0	0	0
4	0	0	1	0	0	0
5	0	0	0	0	0	1
6	1	0	0	0	0	0
7	1	0	0	0	0	0
8	0	0	0	1	0	0
9	0	1	0	0	0	0

Tableau 3.6 Résultats de l'affectation des 6 RNC aux 3 MSC

RNC	MSC		
	0	1	2
0	0	0	1
1	1	0	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	1	0

Tableau 3.7 Résultats de l'affectation des 6 RNC aux 3 SGSN

RNC	SGSN		
	0	1	2
0	1	0	0
1	0	0	1
2	0	1	0
3	1	0	0
4	0	0	1
5	1	0	0

3.3.3 Mémoire à court terme

La mémoire à court terme est le mécanisme de base de la RT dans le sens que, dans la séquence d'application des trois mécanismes de mémoire, c'est le premier à être appliqué. Il permet de passer itérativement d'une solution à une autre par l'application de mouvements, tout en interdisant un retour vers les k dernières solutions déjà visitées. Tout au long de son application, il garde un certain nombre de bonnes solutions qu'il transmettra au mécanisme de mémoire à moyen terme pour l'intensification. Il garde également des statistiques sur les différentes solutions explorées qu'il transmettra au mécanisme de mémoire à long terme. Dans cette sous-section, nous allons définir les caractéristiques de la mémoire à court terme soient : les types de mouvements possibles,

leurs gains ainsi que la liste taboue à utiliser et le critère d'aspiration permettant d'annuler le caractère tabou d'un mouvement.

Mouvements et gains

Le but de la composante de mémoire à court terme est d'améliorer la solution courante, soit en diminuant son coût ou alors en diminuant les pénalités encourues. Pour y parvenir, elle aura le choix entre différents mouvements qui consistent à faire des réaffectations des Nœuds B vers les RNC et des RNC vers les MSC et les SGSN. Ainsi, pour passer de la solution courante à une nouvelle solution, nous avons le choix d'effectuer trois types de mouvements que nous pourrons par la suite réduire à deux grands types. Nous dénotons les trois types de mouvements par $M_1(a, b)$, $M_2(b, c)$ et $M_3(b, d)$, avec :

- $M_1(a, b)$: *réaffectation du Nœud B **a** au contrôleur RNC **b*** ;
- $M_2(b, c)$: *réaffectation du contrôleur RNC **b** au commutateur MSC **c*** ;
- $M_3(b, d)$: *réaffectation du contrôleur RNC **b** au commutateur SGSN **d***.

Afin de limiter l'ensemble des solutions possibles et surtout de respecter le schéma de relève complexe (Figure 3.3), nous regroupons les deux derniers mouvements en un seul appelé $M_2(b, c, d)$. Ce mouvement oblige, à chaque fois que l'on veut réaffecter un RNC, à le faire en changeant ses deux commutateurs d'attache (MSC et SGSN). Nous aurons donc finalement, les deux types de mouvements suivants pour générer une nouvelle solution :

- $M_1(a, b)$: *réaffectation du Nœud B **a** au contrôleur RNC **b*** ;
- $M_2(b, c, d)$: *réaffectation du contrôleur RNC **b** aux commutateurs MSC **c** et SGSN **d***.

Chacun de ces mouvements va être accompagné d'un certain gain par rapport à la solution courante. Lors de la simulation, nous appliquerons successivement les deux types de mouvements pour générer l'ensemble des solutions voisines à la solution

courante. Ainsi, le voisinage $N(S)$ d'une solution S est défini par toute solution accessible à partir de S par l'application de $M_1(a, b)$ et $M_2(b, c, d)$ à S . Pour évaluer rapidement les solutions du voisinage $N(S)$, nous définissons les deux gains $G_S^1(a, b)$ et $G_S^2(b, c, d)$ associés respectivement aux mouvements $M_1(a, b)$ et $M_2(b, c, d)$ et à la solution S .

$G_S^1(a, b)$ s'exprime par :

$$G_S^1(a, b) = \begin{cases} \sum_{i=1, i \neq a}^n \sum_{k=1, k \neq k_0}^m \sum_{l=1, l \neq l_0}^s (h_{ia} + h_{ai}) Y_{b_0 k_0} Y_{b_0 l_0} Y_{bk} Y_{bl} (x_{ib_0} - x_{ib}) + c_{ab} - c_{ab_0} & \text{si } b \neq b_0 \\ 0 & \text{sinon} \end{cases}$$

où :

- c_{ab} désigne le coût de liaison entre le Nœud B a et le RNC b ;
- b_0 représente le RNC d'attache du Nœud B a dans la solution S avant l'application du mouvement $G_S^1(a, b)$;
- h_{ia} désigne le coût total de relève entre les Nœuds B i et a ;
- x_{ib} prend la valeur 1 si les Nœuds B i est relié au contrôleur RNC b ;
- Y_{bk} prend la valeur 1 si le RNC b est relié au MSC k ;
- Y_{bl} prend la valeur 1 si le RNC b est relié au SGSN l ;
- k_0 et l_0 représentent respectivement le MSC et le SGSN d'attache du RNC b_0 ;
- k et l représentent respectivement le MSC et le SGSN d'attache du RNC b .

Puisque dans la formulation du problème nous avions décidé de ne considérer le phénomène de relève complexe entre deux Nœuds B que seulement lorsqu'il est suivi non seulement d'un changement de RNC mais aussi d'un changement de commutateurs (MSC et SGSN), nous avons ajouté des conditions dans la formule du gain $G_S^1(a, b)$ pour

tenir compte de cela. Ainsi, le gain obtenu en réaffectant le Nœud B a vers le RNC b correspond à la différence entre la sommation des coûts de relève complexe entre le Nœud B a et tous les autres Nœuds B reliés à son ancien RNC (qui est b_0 dans ce cas-ci) et la sommation des coûts de relève complexe entre le même Nœud B a et tous les Nœuds B reliés à son nouveau RNC (qui est b). Cette différence est calculée seulement si les RNC b_0 et b sont reliés à des MSC et des SGSN différents. À cette différence, il faut ajouter la différence entre le coût de la liaison entre le Nœud B a et le RNC b et celui de la liaison entre le Nœud B a et le RNC b_0 . La Figure 3.5 illustre un exemple de calcul du gain $G_S^1(a, b)$. Mentionnons que pour $b = b_0$ (c'est à dire que le nouveau RNC auquel on veut réaffecter le Nœud B a est égale à son ancien RNC), le mouvement $M_1(a, b)$ ne modifie pas la solution S et le gain est donc nul.

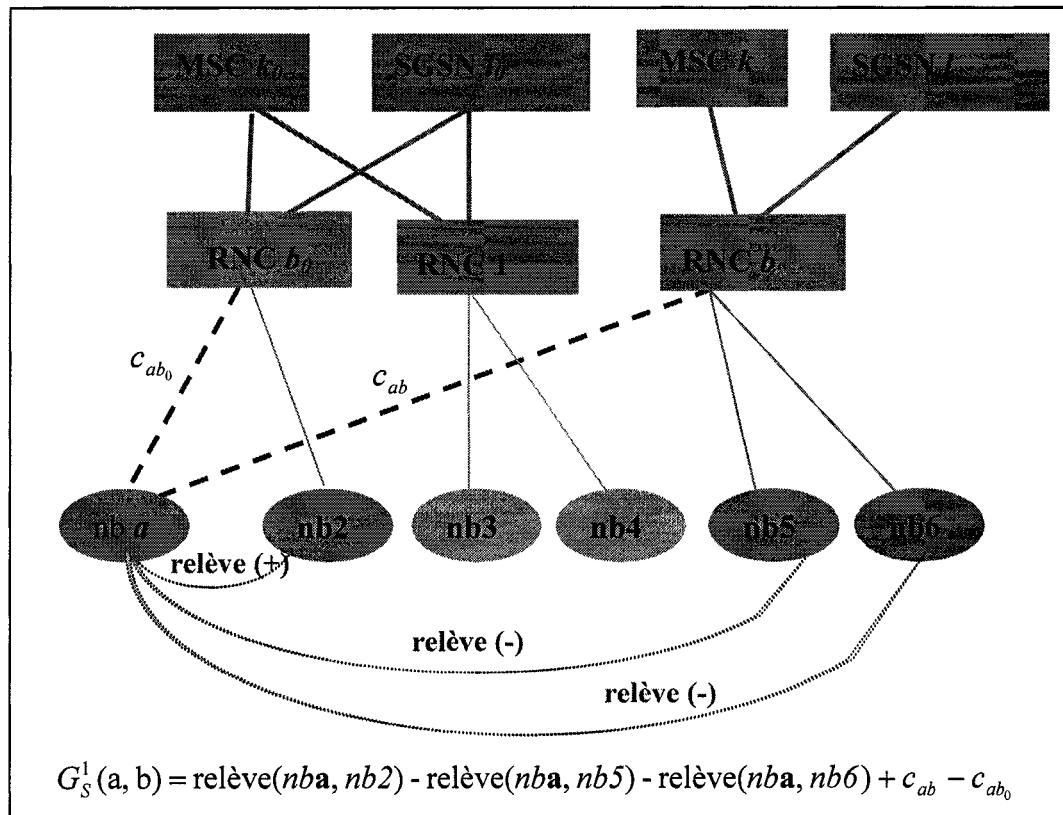


Figure 3.5 Exemple de calcul du gain $G_S^1(a, b)$

Quant au gain $G_S^2(b, c, d)$, il s'exprime par:

$$G_S^2(b, c, d) = \begin{cases} \sum_{j=1, j \neq b}^r \sum_{i=1}^n \sum_{i'=1}^n (h_{ii'} + h_{i'i})(1 - y_{ii'})(Y_{jc_0} Y'_{jd_0} - Y_{jc} Y'_{jd}) + c_{bc} + c_{bd} - c_{bc_0} - c_{bd_0} & \text{si } c \neq c_0 \text{ et } d \neq d_0 \\ 0 & \text{sinon} \end{cases}$$

où :

- $h_{ii'}$ désigne le coût total de relève entre les Nœuds B i et i' ;
- $y_{ii'}$ prend la valeur 1 si les Nœuds B i et i' sont reliés au même contrôleur RNC ;
- Y_{jc} prend la valeur 1 si le RNC j est relié au MSC c ;
- Y'_{jd} prend la valeur 1 si le RNC j est relié au SGSN d ;
- c_0 et d_0 représentent respectivement le MSC et le SGSN d'attache du RNC b dans la solution S , c'est-à-dire avant l'application du mouvement $G_S^2(b, c, d)$;
- c_{bc} et c_{bd} représentent respectivement le coût de liaison entre le RNC b et le MSC c et entre le même RNC et le SGSN d .

Comme on peut le constater sur la formule de $G_S^2(b, c, d)$, le calcul du gain pour le mouvement $M_2(b, c, d)$ est un peu plus complexe car, même si $M_2(b, c, d)$ consiste à changer les commutateurs d'attache d'un RNC, les variables de coût de relève font intervenir les Nœuds B qui sont à un niveau inférieur par rapport aux RNC. Ainsi, lorsqu'on a un RNC b qui est relié à un MSC c_0 et à un SGSN d_0 et qu'on veut le réaffecter à un nouveau MSC c et à un nouveau SGSN d , le gain $G_S^2(b, c, d)$ est obtenu en faisant la différence entre la sommation des coûts de relève entre le RNC b et tous les autres RNC reliés au MSC c_0 et au SGSN d_0 et la sommation des coûts de relève entre

le RNC b et tous les autres RNC reliés aux nouveaux commutateurs MSC c et SGSN d . Mais comme les relèves se font réellement au niveau des Nœuds B, pour avoir par exemple le coût de relève complexe entre le RNC b et un RNC j , il faudra sommer les coûts de relève entre chaque Nœud B du RNC b et tous les Nœuds B du RNC j .

Le calcul du gain $G_S^2(b, c, d)$ sera complété en faisant la différence entre la somme des coûts de liaison entre le RNC b et ses anciens commutateurs (MSC et SGSN) et la somme des coûts de liaison entre le même RNC b et ses nouveaux commutateurs d'attache. On obtient donc la valeur du gain $G_S^2(b, c, d)$ sur le coût $f(S)$ pour $c \neq c_0$ et $d \neq d_0$. De la même manière que dans le cas du mouvement $M_1(a, b)$, pour $c = c_0$ et $d = d_0$ le gain vaut 0. La Figure 3.6 montre un exemple de calcul du gain $G_S^2(b, c, d)$.

Ainsi, après l'application des deux mouvements ($M_1(a, b)$ et $M_2(b, c, d)$), le coût de la nouvelle solution S' est obtenu à partir de la formule suivante :

$$f(S') = f(S) + G_S^1(a, b) + G_S^2(b, c, d), \text{ où } f(S) \text{ est le coût de la solution courante } S.$$

Lors de la simulation, une fois le type de mouvement choisi, un ou deux tableaux contenant tous les gains seront générés. Le mouvement $M_1(a, b)$ nécessitera un tableau de gain de taille $n \times r$, où n désigne le nombre de Nœuds B et r le nombre de RNC. Chaque ligne du tableau représentera un Nœud B et sur cette ligne seront inscrits les gains obtenus en réaffectant ce Nœud B à chaque RNC du réseau. Le mouvement $M_2(b, c, d)$ nécessitera, quant à lui, deux tableaux dont l'un sera de taille $r \times k$, où r désigne le nombre de RNC et k le nombre de MSC, et l'autre de taille $r \times s$, où r désigne le nombre de RNC et s le nombre de SGSN. Après chaque mouvement, le ou les tableaux de gains sont mis à jour. Et à chaque itération, l'algorithme sélectionnera, parmi les mouvements possibles, celui ayant le gain minimum. Mais, lorsque le seul mouvement possible de type $M_1(a, b)$ est celui pour $b = b_0$ donc de gain nul et que le seul mouvement possible de type $M_2(b, c, d)$ est celui pour $c = c_0$ et $d = d_0$ (gain nul), alors pour éviter de boucler sur

cette solution, nous attribuerons à chacun des gains de ces deux mouvements une valeur M arbitrairement grande. De cette manière, lors de la prochaine étape, l'algorithme choisira d'autres mouvements de réaffectation qui auront un gain minimum et qui ne seront pas des mouvements avec $b = b_0$, $c = c_0$ et $d = d_0$.

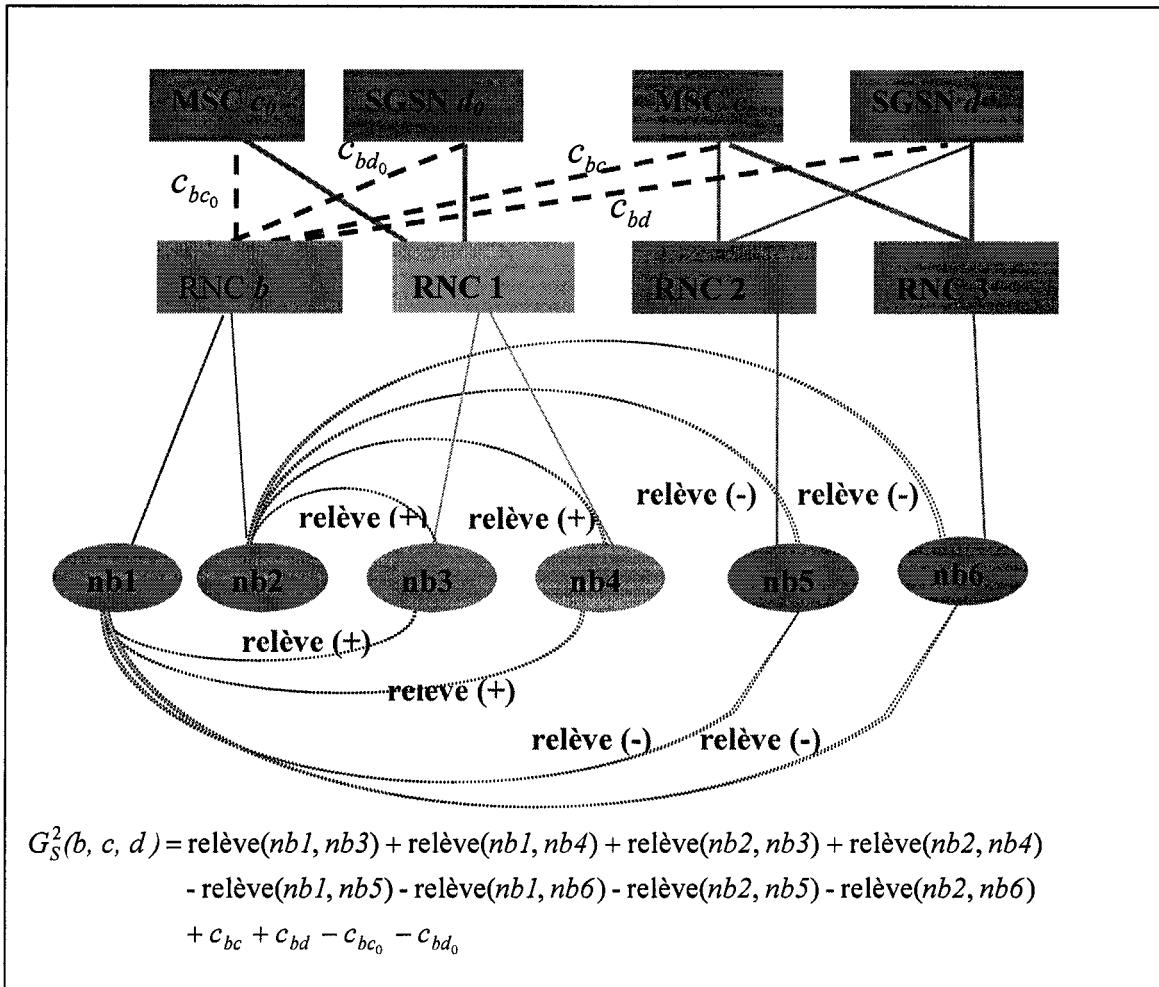


Figure 3.6 Exemple de calcul du gain $G_s^2(b, c, d)$

Liste taboue et critère d'aspiration

Pour chaque itération, une fois une solution S déterminée, la RT cherchera à améliorer celle-ci en étudiant les solutions candidates contenues dans le sous-ensemble de son voisinage. Mais l'une des difficultés dans la RT est de déterminer la liste des solutions candidates. Car en utilisant les différentes combinaisons de mouvements, nous risquons d'avoir un sous-ensemble très grand de solutions voisines (et donc nécessitant beaucoup de temps de calcul) et aussi, nous risquons de faire des boucles en retournant sur des solutions déjà explorées. Pour éviter cela, nous choisirons de garder certaines solutions déjà visitées dans une liste taboue de taille finie et nous limiterons le nombre d'itérations possibles pour rechercher une meilleure solution dans le voisinage d'une solution courante.

Pour ce qui est de la liste taboue, nous devons définir son contenu, sa taille et sa durée de vie. Telle que l'adaptation de la RT est présentée dans [15], on a le choix de garder dans la liste taboue soit un numéro associé à chaque mouvement pour désigner son statut tabou ou encore de garder dans la liste les k dernières solutions générées pour les exclure de la liste des futures solutions candidates. Enfin, le dernier choix est de garder dans la liste des mouvements dits tabous. Pour ce faire, à chaque fois qu'un mouvement est effectué, son inverse est gardé dans la liste pour éviter plus tard de revenir sur des solutions obtenues en faisant le même mouvement. Ce dernier choix constitue un bon compromis entre les deux premiers, car au lieu de garder un numéro du mouvement (ce qui ne donne pas beaucoup d'information sur le mouvement lui-même) ou encore de garder toute la solution (ce qui peut s'avérer lourd à un moment donné et l'information sur le mouvement qui a mené à cette solution sera toujours manquante), il est plus judicieux de garder l'information sur le mouvement lui-même, d'autant plus que les mouvements constituent l'élément central du fonctionnement de l'algorithme RT.

Ainsi, pour chacun des deux types de mouvement $M_1(a, b)$ et $M_2(b, c, d)$, les mouvements inverses $M_1(a, b_0)$ et $M_2(b, c_0, d_0)$ seront gardés dans la liste taboue où b_0 désigne le RNC du Nœud B a avant l'application du mouvement $M_1(a,b)$; c_0 et d_0

désignent respectivement le MSC et le SGSN du RNC b avant l'application du mouvement $M_2(b, c, d)$. Afin de différencier les mouvements de réaffectation des Nœuds B (mouvements de type $M_1(a, b)$) des mouvements de réaffectation des RNC (mouvements de type $M_2(b, c, d)$), nous créerons deux sous-listes taboues, dont une appelée LT_mvt1 pour contenir les mouvements de type $M_1(a, b)$ et l'autre appelée LT_mvt2 pour contenir les mouvements de type $M_2(b, c, d)$. Les deux sous-listes seront de taille $TAILLELT$.

Également, pour éviter de trop s'éloigner des solutions courantes avec peu de chance d'y revenir, nous arrêterons l'exploration du sous-ensemble des solutions voisines de la solution courante après K_{max} itérations. Lorsque l'exploration s'éloignera des solutions réalisables, nous utiliserons un mécanisme de rappel pour ramener l'exploration vers les solutions réalisables. Ce mécanisme de rappel consistera à infliger une pénalité aux gains des mouvements menant à des solutions non-réalisables après un certain nombre (*DRESPECT*) de solutions consécutives non-réalisables. La pénalité sera composée d'une partie fixe appelée *PRIMECAPAFIXE* et d'une partie variable appelée *PRIMECAPA* et sera appliquée au gain des mouvements qui tendront à affecter un Nœud B à un RNC ayant déjà une capacité résiduelle négative ou alors à affecter un RNC à un MSC ou à un SGSN ayant déjà une capacité résiduelle négative. Mais, à la première solution faisable, le dispositif de rappel sera désactivé.

Lorsque la RT tombe sur un optimum local, il peut arriver lors de l'exploration de l'espace des solutions aux alentours de cet optimum, le cas où, après un certain nombre d'itérations qui n'améliorent pas beaucoup la solution courante, tout à coup l'application d'un nouveau mouvement améliore la solution mais que ce dernier figure dans la liste taboue. Dans ce cas, afin d'annuler temporairement le critère tabou de ce mouvement et le rendre admissible, nous intégrerons un critère d'aspiration dans l'application de la composante de mémoire à court terme. Ainsi, lors de l'exploration de l'espace de recherche, la méthode vérifiera toujours si le mouvement qu'elle vient d'effectuer est tabou ou non et, dans le cas où il l'est et qu'il conduit à une solution dont l'évaluation du

coût est inférieure à celle de la meilleure solution actuellement connue, son critère tabou sera annulé.

3.3.4 Mémoire à moyen terme

La composante de mémoire à moyen terme de la RT a pour but d'intensifier la recherche localement autour de régions prometteuses. Dans notre adaptation, le mécanisme de mémoire à court terme garde une liste *FIFO* de taille *ITAILLEBEST* des dernières meilleures solutions qu'il transmet au mécanisme de mémoire à moyen terme. C'est autour de ces solutions que le mécanisme de mémoire à moyen terme va intensifier la recherche. Les solutions sont gardées avec tous les paramètres de l'état de l'exploration pour permettre de restaurer le contexte de la recherche au moment opportun. Parmi les informations sauvegardées pour chaque bonne solution, il y a la topologie de la solution, son coût, la meilleure évaluation de la fonction de coût trouvée à cette étape et les tableaux des gains. Pour effectuer l'intensification, il peut y avoir divers mouvements ou combinaisons de mouvements possibles.

Mouvements, liste taboue et critère d'aspiration

Afin d'explorer d'autres régions de recherche, il existe une multitude de mouvements possibles que nous pouvons effectuer vu que nous avons deux niveaux d'affectation dans notre problème. À noter que ces mouvements seront différents de ceux qu'on avait définis pour la mémoire à court terme. Pour ne pas trop surcharger notre adaptation de la RT, nous allons nous en tenir à des mouvements basés sur deux opérations de base, soient la *permutation* et le *déplacement*. Ainsi, nous définissons les mouvements suivants :

- $m_1(a_1, a_2)$: permutation des Nœuds B a_1 et a_2 selon les plus faibles gains ;
- $m_2(b_1, b_2)$: permutation des RNC b_1 et b_2 selon les plus faibles gains ;
- $m_3(a, b)$: déplacement du Nœud B a vers le RNC b en vue de rétablir les contraintes de capacité ;

- $m_4(b, c, d)$: déplacement du RNC b vers le MSC c et le SGSN d en vue de rétablir les contraintes de capacité.

Permutation : Mouvements de permutation $m_1(a_1, a_2)$ et $m_2(b_1, b_2)$

Le but de ce type de mouvement est d'améliorer l'évaluation des solutions en diminuant le coût associé. Ces deux mouvements seront appliqués généralement sur des solutions déjà faisables. Chaque permutation peut être divisée en deux sous-mouvements de réaffectation consécutifs. Mais, contrairement aux mouvements de réaffectation définis dans le cas de la mémoire à court terme, ici, lors du processus de sélection des deux Nœuds B (ou RNC) à permuter, entre la sélection du premier, sa réaffectation et la sélection du deuxième, le tableau des gains ne sera pas mis à jour. Cela permet de stabiliser la topologie du réseau de la solution courante. Le fait que nous avons deux sous-mouvements consécutifs de réaffectation nous oblige, pour le seul mouvement de permutation $m_1(a_1, a_2)$ (ou $m_2(b_1, b_2)$), à inscrire dans la liste taboue deux mouvements inverses aux sous-mouvements de réaffectation.

Ainsi, pour l'application de la mémoire à moyen terme, nous aurons besoin de deux sous-listes taboues de taille double ($2 * TAILLELT$) par rapport à celles définies dans le cas de la mémoire à court terme. Nous appellerons la première sous-liste $LT11$ pour les mouvements de type $m_1(a_1, a_2)$ et la deuxième $LT1$ pour ceux de type $m_2(b_1, b_2)$. Un mouvement de permutation sera tabou si l'un au moins des sous-mouvements de réaffectation l'est. L'application des mouvements de permutation sera libre des contraintes de capacité sur les RNC, MSC et SGSN et s'appuiera exclusivement sur une estimation des gains pour le choix de la permutation. Le critère d'aspiration est défini de la même manière que dans le cas de la composante de mémoire à court terme; il consiste à accepter une permutation taboue si elle mène à une solution meilleure en terme de coût que celles déjà obtenues. Dès qu'on aura un certain nombre de solutions non faisables consécutives, nous appliquerons les mouvements de déplacement $m_3(a, b)$ et $m_4(b, c, d)$.

Déplacement : Mouvements de déplacement $m_3(a, b)$ et $m_4(b, c, d)$

Puisque jusque là, les opérations de recherche de solution sont libres des contraintes de capacité sur les équipements du réseau, nous utiliserons ces deux mouvements de déplacement dans le but de restaurer les contraintes de capacité et du même coup diminuer les pénalités applicables sur l'évaluation de la solution. Le mouvement $m_3(a, b)$ consiste à :

- déterminer le RNC que l'on appellera b' et qui a la capacité résiduelle totale (en appels de voix et données) minimale ;
- trouver le Nœud B a , affecté à b' , qui génère le volume d'appel total (voix + données) minimal ;
- affecter ce Nœud B a au RNC b qui a une capacité totale résiduelle (en appels de voix et données) suffisante et qui permet d'obtenir le gain minimal.

Le mouvement $m_4(b, c, d)$, quant à lui, consiste à :

- déterminer les MSC et SGSN que l'on appellera respectivement c' et d' qui ont la capacité résiduelle minimale ;
- trouver le RNC b affecté à c' et d' et par qui passe le volume d'appel total (voix + données) minimal ;
- l'affecter au MSC c et au SGSN d qui ont une capacité résiduelle suffisante et qui permet d'obtenir le gain minimal.

Ces deux mouvements sont donc fondés à la fois sur les contraintes de capacité et sur les gains. Dans le processus d'implémentation de la composante de mémoire à moyen terme, ce sont ces mouvements de déplacement qui seront activés une fois que nous aurons un certain nombre (dénommé par la variable *IDRESPECT*) de solutions consécutives non faisables. Et leur application demeurera active tant que les solutions trouvées ne seront pas faisables. On peut remarquer que les mouvements de déplacement $m_3(a, b)$ et $m_4(b, c, d)$ sont des mouvements de réaffectation comme les mouvements

$M_1(a, b)$ et $M_2(b, c, d)$ de la composante de mémoire à court terme sauf que dans ce cas-ci, ils ne sont pas basés uniquement sur le gain mais plutôt sur le gain et les contraintes de capacité. Une fois les deux mouvements $m_3(a, b)$ et $m_4(b, c, d)$ appliqués, nous garderons dans la liste taboue respective les mouvements inverses $m_3(a, b')$ et $m_4(b, c', d')$ où b' désigne l'ancien RNC du Nœud B a avant l'application du mouvement $m_3(a, b)$ et c' et d' désignent respectivement l'ancien MSC et l'ancien SGSN du RNC b avant l'application du mouvement $m_4(b, c, d)$. Nous utiliserons ici aussi deux sous-listes taboues de taille $TAILLET$, une appelée $LT22$ pour les mouvements de type $m_3(a, b)$ et l'autre appelée $LT2$ pour les mouvements de type $m_4(b, c, d)$. Pour l'application des mouvements de déplacement, aucun critère d'aspiration ne sera mis en œuvre.

La composante de mémoire à moyen terme, tout comme celle de la mémoire à court terme, s'arrête si la meilleure solution n'a pas été améliorée pendant les K_{max} dernières itérations ou alors s'il n'y a plus de mouvements possibles. Afin de continuer l'exploration, il faut faire appel à d'autres mécanismes tels la diversification qui fera l'objet de la section suivante.

3.3.5 Mémoire à long terme

Pour diversifier la recherche, la RT utilise une structure de mémoire à long terme afin d'amener la recherche dans des régions jusqu'ici peu explorées. Ceci nécessite souvent la génération de nouvelles solutions de départ. Dans notre adaptation, pour générer cette solution de départ, nous allons nous servir des statistiques sur les solutions déjà explorées afin de voir, par exemple pour l'affectation des Nœuds B aux RNC, laquelle des combinaisons d'affectation (*Nœud B, RNC*) apparaît le moins de fois ou pas du tout dans la topologie des solutions déjà visitées et ainsi générer une nouvelle solution de départ contenant cette combinaison. Le même processus sera appliqué pour l'affectation des RNC aux MSC et aux SGSN. Et une fois une nouvelle solution initiale générée, une nouvelle recherche redémarre en appliquant de nouveau les mécanismes de mémoire à court et moyen termes. Le nombre de cycles de redémarrage est fixé d'avance à travers une variable appelée *NBSTART*.

CHAPITRE 4

IMPLÉMENTATION ET ANALYSE DES RÉSULTATS

La fonctionnalité et la performance de l'adaptation de la RT reposent sur une implémentation optimale sous forme de code exécutable et sur un bon choix de certains paramètres clés de la méthode. Dans ce chapitre, nous présenterons d'abord les détails de l'implémentation de la RT pour résoudre le problème d'affectation des cellules aux commutateurs dans les réseaux mobiles 3G. Ensuite, nous analyserons la performance de la méthode en la soumettant à une série de tests pour calibrer les paramètres clés de chacune des composantes de mémoire et évaluer l'efficacité de la méthode à résoudre le problème.

4.1 Détails d'implémentation

Pour son fonctionnement, le programme requiert des données d'entrée sur le type de réseau 3G que nous voulons simuler. Nous présenterons donc le modèle de génération des fichiers d'entrée ainsi que leurs formats et décrirons les algorithmes des principales parties de l'adaptation proposée ainsi que les classes utilisées.

4.1.1 Modélisation du trafic

Pour la génération des fichiers de données d'entrée servant à notre simulation, nous allons réutiliser le programme *Matlab* utilisé dans [15]. Mais, puisque le programme générait des fichiers tests pour des réseaux 2G, nous allons modifier certaines fonctions pour introduire le trafic de données et aussi les capacités des RNC et leurs coûts de liaison aux MSC et aux SGSN. Nous allons faire l'analogie avec le 2G en considérant que les Nœuds B correspondent aux cellules dans le 2G, si bien qu'on supposera que notre réseau est constitué de cellules disposées sur une grille hexagonale de longueur et de largeur à peu près égales et que, dans chaque cellule, il y a un Nœud B. Les antennes

sont situées au centre des cellules et distribuées uniformément sur la grille. Toutefois après une distribution des antennes sur la grille, lorsque deux ou plusieurs antennes sont trop proches, la disposition des antennes est rejetée et on refait de nouveau une autre distribution. Le coût de liaison d'une cellule à un contrôleur RNC est proportionnel à la distance les séparant. Nous avons pris un coefficient de proportionnalité égal à l'unité. Pour le calcul des coûts de liaison entre les RNC et les MSC et SGSN, nous avons considéré un réaménagement de la grille de notre réseau en formant des clusters de cellules. Chaque cluster renferme un RNC qui gère les cellules qu'il contient. Les MSC et SGSN sont placés de manière stratégique dans certaines de ces clusters, au niveau de l'emplacement de certaines antennes (pour des raisons d'économie de coût de location des lieux d'entreposage), afin d'assurer le contrôle du trafic transitant par les RNC. Ainsi, le coût de liaison d'un contrôleur RNC à un MSC ou à un SGSN est proportionnel à la distance les séparant. Nous avons pris dans ce cas aussi, pour chaque liaison (RNC vers MSC ou RNC vers SGSN), un coefficient de proportionnalité égal à l'unité.

De la même manière que dans [16], nous allons considérer que le taux d'appels en trafic de voix f_i^{voix} provenant d'un Nœud B i suit une loi gamma de moyenne et de variance égales à l'unité. Pour ce qui est du taux d'appels en trafic de données $f_i^{données}$ provenant du Nœud B i , nous posons l'hypothèse qu'il suit également une loi gamma de moyenne et de variance égales à l'unité. Car jusqu'ici, il n'y a pas de loi mathématique qui permet de modéliser de manière exacte le comportement du trafic en commutations de paquets dans un réseau mobile 3G (d'ailleurs c'est un sujet qui fait l'objet de beaucoup de discussions dans le milieu de la recherche).

Les temps de service (ou temps de séjour) des appels (voix et données) à l'intérieur des cellules sont distribués selon une loi exponentielle de paramètre 1 [16]. Si un Nœud B situé dans la cellule j possède k voisins, l'intervalle $[0,1]$ est divisé en $k+1$ sous-intervalles en choisissant k nombres aléatoires distribués uniformément entre 0 et 1. À la fin de la période de service dans la cellule j , l'appel peut être, soit transféré au $i^{ème}$ voisin ($i=1, \dots, k$) avec une probabilité de relève r_{ij} égale à la longueur du $i^{ème}$ intervalle,

soit terminé avec une probabilité égale à la longueur du $k+1^{\text{ème}}$ intervalle. Pour trouver des volumes d'appel et des taux de relève cohérents, les cellules sont considérées comme des files d'attente $M/M/1$ formant un réseau de Jackson. Les taux d'arrivées α_i^{voix} et $\alpha_i^{\text{données}}$ dans les cellules à l'équilibre sont obtenus en résolvant les systèmes suivants :

$$\alpha_i^{\text{voix}} - \sum_{j=1}^n \alpha_j^{\text{voix}} r_{ji} = f_i^{\text{voix}} \quad \text{avec } i = 1, 2, 3, \dots, n$$

$$\alpha_i^{\text{données}} - \sum_{j=1}^n \alpha_j^{\text{données}} r_{ji} = f_i^{\text{données}} \quad \text{avec } i = 1, 2, 3, \dots, n$$

où n est le nombre total de cellules du réseau, r_{ij} la probabilité de relève entre les cellules i et j et f_i^{voix} , $f_i^{\text{données}}$ respectivement les taux d'appels en trafic de voix et de données provenant de la cellule i . On choisit comme volume d'appel d'un Nœud B j , que ce soit la voix ou les données, la longueur moyenne de sa file d'attente correspondant au type de trafic. Les taux de relève h_{ij}^{voix} et $h_{ij}^{\text{données}}$ sont définis par :

$$h_{ij}^{\text{voix}} = f_i^{\text{voix}} \cdot r_{ij}$$

$$h_{ij}^{\text{données}} = f_i^{\text{données}} \cdot r_{ij}$$

Tous les r contrôleurs RNC ont la même capacité Cap en appel circuit et la même capacité Cap' en commutation de paquets calculées comme suit pour les besoins de la simulation :

$$Cap = \frac{1}{r} \left(1 + \frac{k}{100}\right) \sum_{i=1}^n f_i^{\text{voix}}$$

$$Cap' = \frac{1}{r} \left(1 + \frac{k}{100}\right) \sum_{i=1}^n f_i^{\text{données}}$$

où k est uniformément réparti entre 10 et 50, ce qui assure un surplus global de 10 à 50% de la capacité des contrôleurs par rapport au volume d'appel des Nœuds B.

Tous les m commutateurs MSC ont la même capacité Cap'' calculée comme suit :

$$Cap'' = \frac{1}{m} \left(1 + \frac{k}{100}\right) \sum_{j=1}^r Cap'_j$$

Et tous les s commutateurs SGSN ont la même capacité Cap''' calculée comme suit :

$$Cap''' = \frac{1}{s} \left(1 + \frac{k}{100}\right) \sum_{j=1}^r Cap'_j$$

où k est encore une fois uniformément réparti entre 10 et 50, ce qui assure un surplus global de 10 à 50% de la capacité des commutateurs par rapport au volume d'appel provenant des contrôleurs RNC.

4.1.2 Format des fichiers d'entrée et de sortie

Pour résoudre le problème d'affectation, notre programme doit d'abord acquérir les spécifications du problème. Celles-ci sont fournies dans deux types de fichiers :

- Le premier, appelé « *fichier de données* » avec l'extension « *.DON* », fournit sur sa première ligne, le nombre n de Nœuds B, le nombre r de RNC, le nombre m de MSC et le nombre s de SGSN du réseau à simuler. Cette première ligne est suivie de cinq matrices de données (Figure C.1 de l'annexe C). La première de taille $n \times r$ contient les coûts de liaison entre les Nœud B et les RNC, où chaque ligne i donne les coûts de liaison du Nœud B i aux r RNC. La deuxième matrice de taille $r \times m$ contient les coûts de liaison entre les RNC et les MSC, où chaque ligne j donne les coûts de liaison du RNC j aux m commutateurs MSC. Et la troisième matrice de taille $r \times s$ contient, quant à elle, les coûts de liaison entre les RNC et les SGSN. La quatrième et la cinquième matrice contiennent les coûts de relève entre les Nœuds B. Elles sont toutes les deux de taille $n \times n$ et chaque ligne i présente les coûts de relève du Nœud B i par rapport aux $n-1$ autres Nœuds B.

Ainsi, la quatrième contient les coûts de relève entre Nœuds B impliquant des commutateurs de type MSC et la cinquième les coûts de relève entre Nœuds B impliquant des commutateurs de type SGSN.

- Le deuxième fichier appelé « *fichier de capacité* » avec l’extension « *.CAP* », donne sur sa première ligne le volume de trafic en commutation de circuit (trafic de voix) produit par chaque Nœud B et sur la deuxième le trafic en commutation de paquets produit par chaque Nœud B. La troisième et la quatrième ligne contiennent respectivement les capacités en commutation de circuit et en commutation de paquets des r RNC. La cinquième et la sixième ligne contiennent, quant à elles, respectivement les capacités des m commutateurs MSC et des s commutateurs SGSN.

À l’exécution, le programme produit des résultats, notamment l’évaluation de la solution, le temps de calcul ainsi que le patron d’affectation sous forme de trois matrices contenant des valeurs binaires (voir la Figure C.3 et les tableaux 3.5, 3.6 et 3.7) : la première de taille $n \times r$ pour l’affectation des Nœuds B aux RNC, la deuxième de taille $r \times m$ pour l’affectation des RNC aux MSC et la troisième de taille $r \times s$ pour l’affectation des RNC aux SGSN. Ces résultats sont inscrits dans le fichier « *Result.ras* ». Afin de suivre la trace du programme, nous avons ajouté un fichier de sortie appelé « *TracePrg.tra* ». Ainsi, pendant son exécution, le programme inscrit dans ce fichier le mécanisme de mémoire en cours d’exécution, les mouvements effectués, l’évaluation de la fonction objectif de la solution courante, le meilleur coût trouvé jusque là et la faisabilité de la solution résultant des mouvements effectués. Des exemples de fichiers de types « *.DON* », « *.CAP* », « *Result.ras* » et « *TracePrg.tra* » sont présentés à l’annexe C.

4.1.3 Implémentation des principaux algorithmes

Dans son fonctionnement, la RT applique en premier la mémoire à court terme, suivie de la mémoire à moyen terme (intensification) et enfin de la mémoire à long terme (diversification). En réalité, la diversification ne fait que redémarrer le processus en

appliquant à nouveau les mémoires à court et moyen termes. La Figure 4.1 illustre le diagramme de fonctionnement du mécanisme de diversification. Sur cette figure, le bloc de génération de la solution de départ est rappelé à chaque redémarrage. Cela permet à la méthode de partir à chaque fois vers des régions non encore visitées car la solution de redémarrage est construite en consultant les statistiques des solutions déjà visitées pour éviter de refaire la recherche autour de ces dernières. Les figures 4.2 et 4.3 présentent respectivement les diagrammes de fonctionnement des mécanismes de mémoire à court et moyen termes.

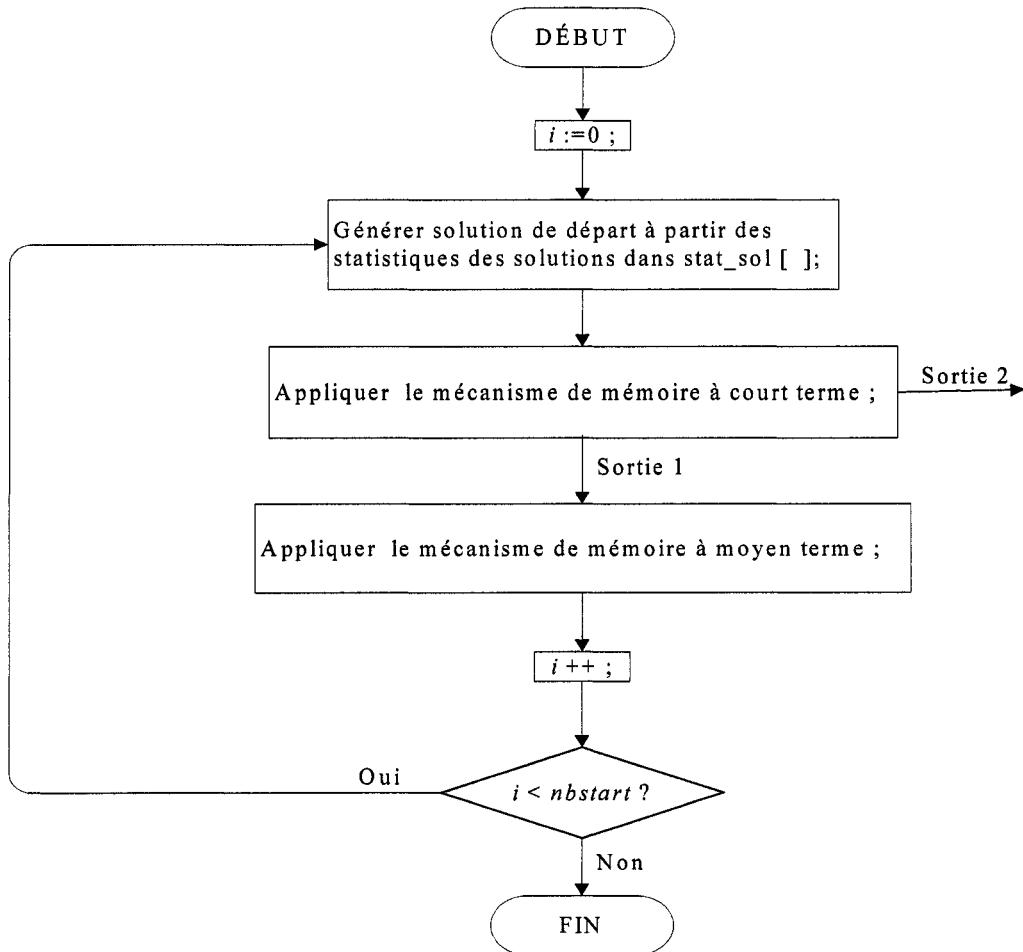


Figure 4.1 Diagramme de fonctionnement du mécanisme de diversification

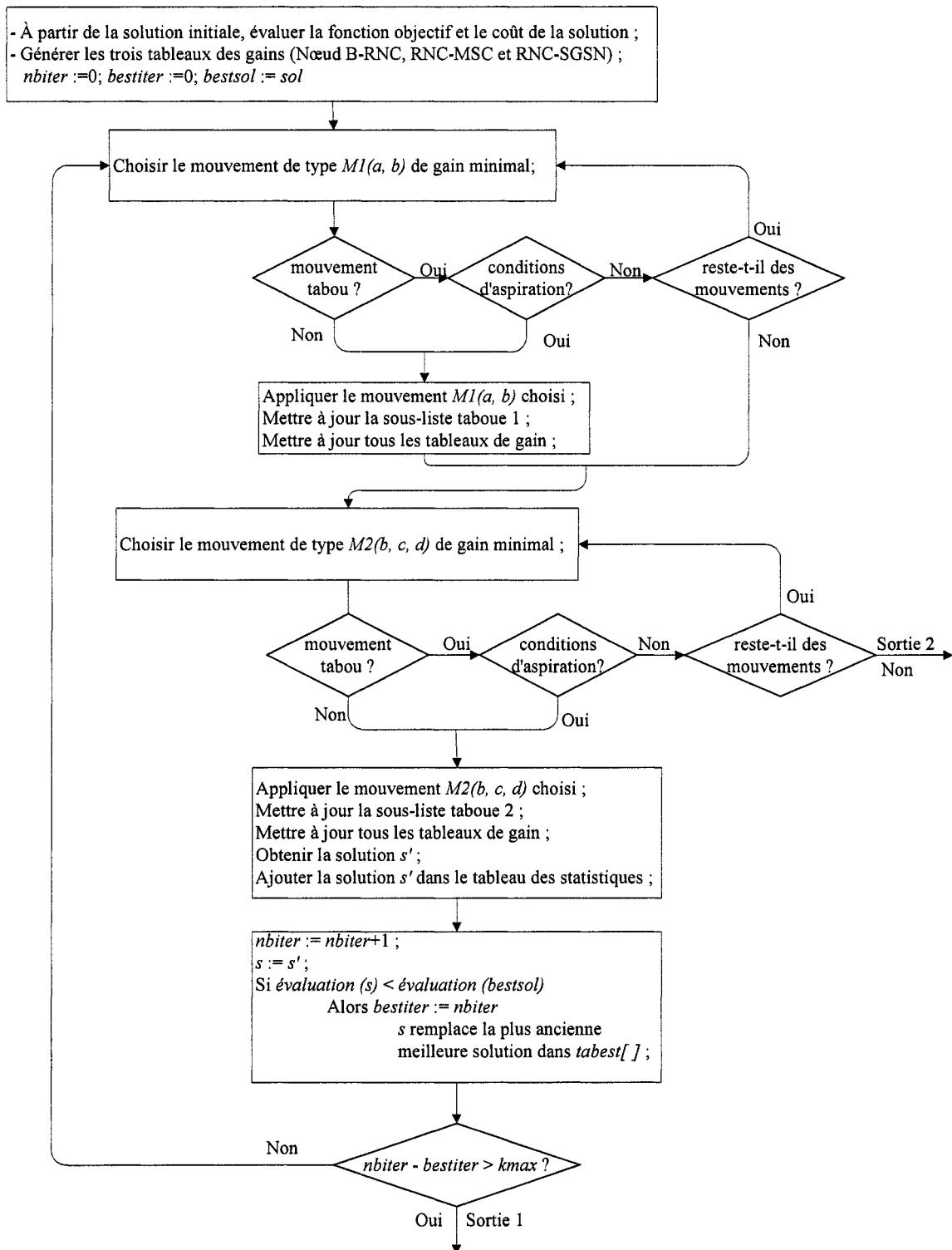


Figure 4.2 Diagramme de fonctionnement du mécanisme de mémoire à court terme

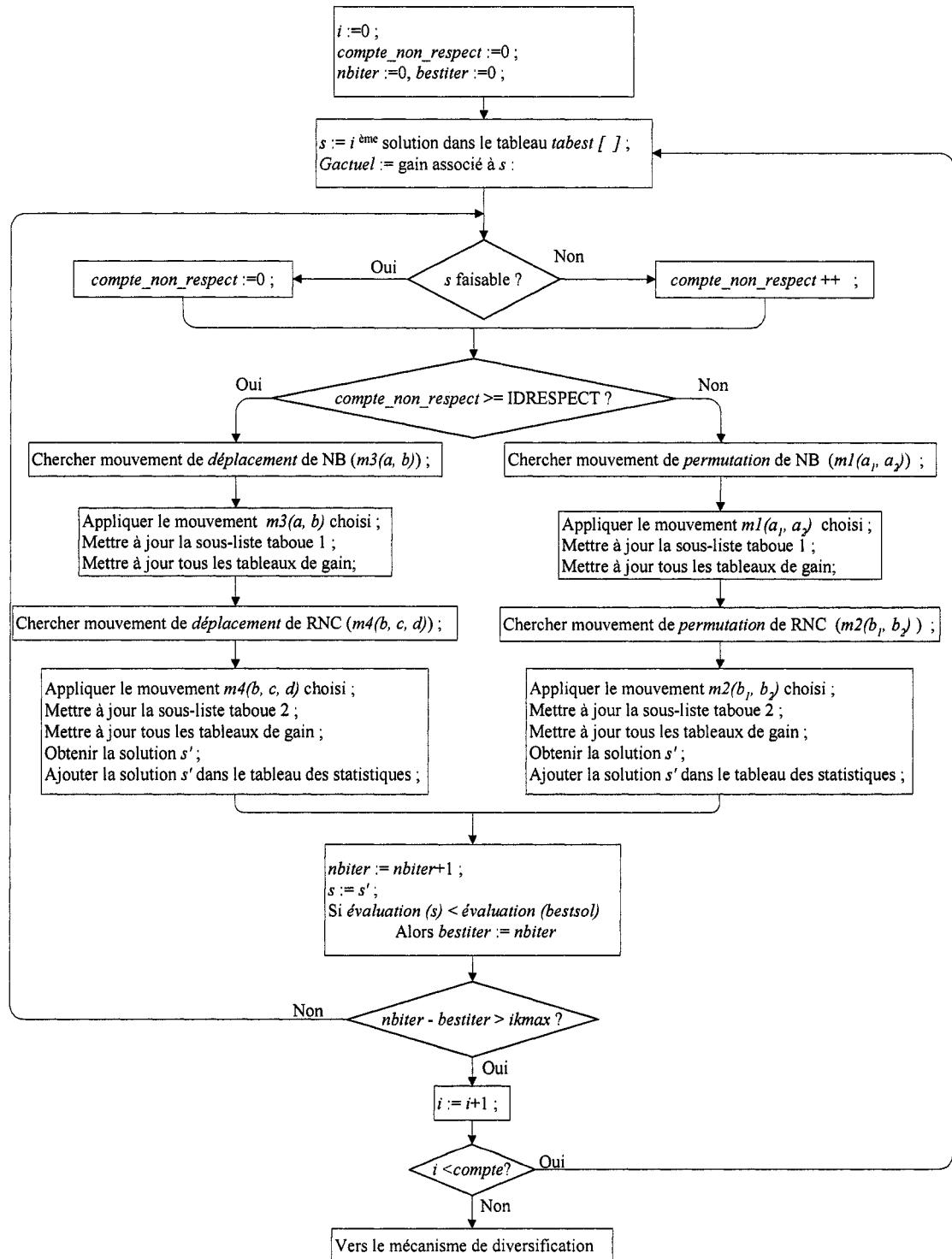


Figure 4.3 Diagramme de fonctionnement du mécanisme de mémoire à moyen terme

4.1.4 Structures de données utilisées

Nous avons implémenté la RT sous forme de programme Orienté Objet. La Figure 4.4 présente le diagramme UML (*Unified Modeling Language*) des principales classes utilisées dans notre adaptation. Chaque classe effectue une gestion efficace de la mémoire à l'aide de méthodes de création et de destruction d'objets. Pour chaque classe, nous présenterons les principales tâches qu'elle accomplit ainsi que ses méthodes et paramètres principaux.

La classe « *Problem* »

C'est la classe qui contient le code principal de la Recherche Taboue. Elle applique le mécanisme de mémoire à court terme, notamment le choix des mouvements à effectuer, génère les solutions et les gains après chaque mouvement et appelle les mécanismes de mémoire à moyen et long terme en leur transmettant les résultats du mécanisme de mémoire à court terme. C'est donc la classe la plus globale car regroupant toutes les autres classes. Elle contient les méthodes suivantes :

- *choix_mvt1 ()* qui choisit pour la mémoire à court terme, le prochain mouvement possible de type $M_1(a, b)$ en tenant compte des contraintes de la liste taboue et du gain minimum ;
- *choix_mvt2 ()* qui choisit toujours pour la mémoire à court terme, le prochain mouvement possible de type $M_2(b, c, d)$ en tenant compte des contraintes de la liste taboue et du gain minimum ;
- *appliquer_mvt1 ()* et *appliquer_mvt2 ()* qui permettent d'appliquer les deux types de mouvements choisis par les deux méthodes précédentes et d'évaluer la solution obtenue ;
- *Cmtaboue ()* applique la composante de mémoire à court terme et appelle la composante de mémoire à moyen terme en lui passant les résultats obtenus ;

- *Rtaboue ()* appelle d'abord la méthode *Cmtaboue ()* pour appliquer les deux composantes de mémoire (court et moyen termes), puis applique la diversification (mémoire à long terme) pour raffiner la solution.

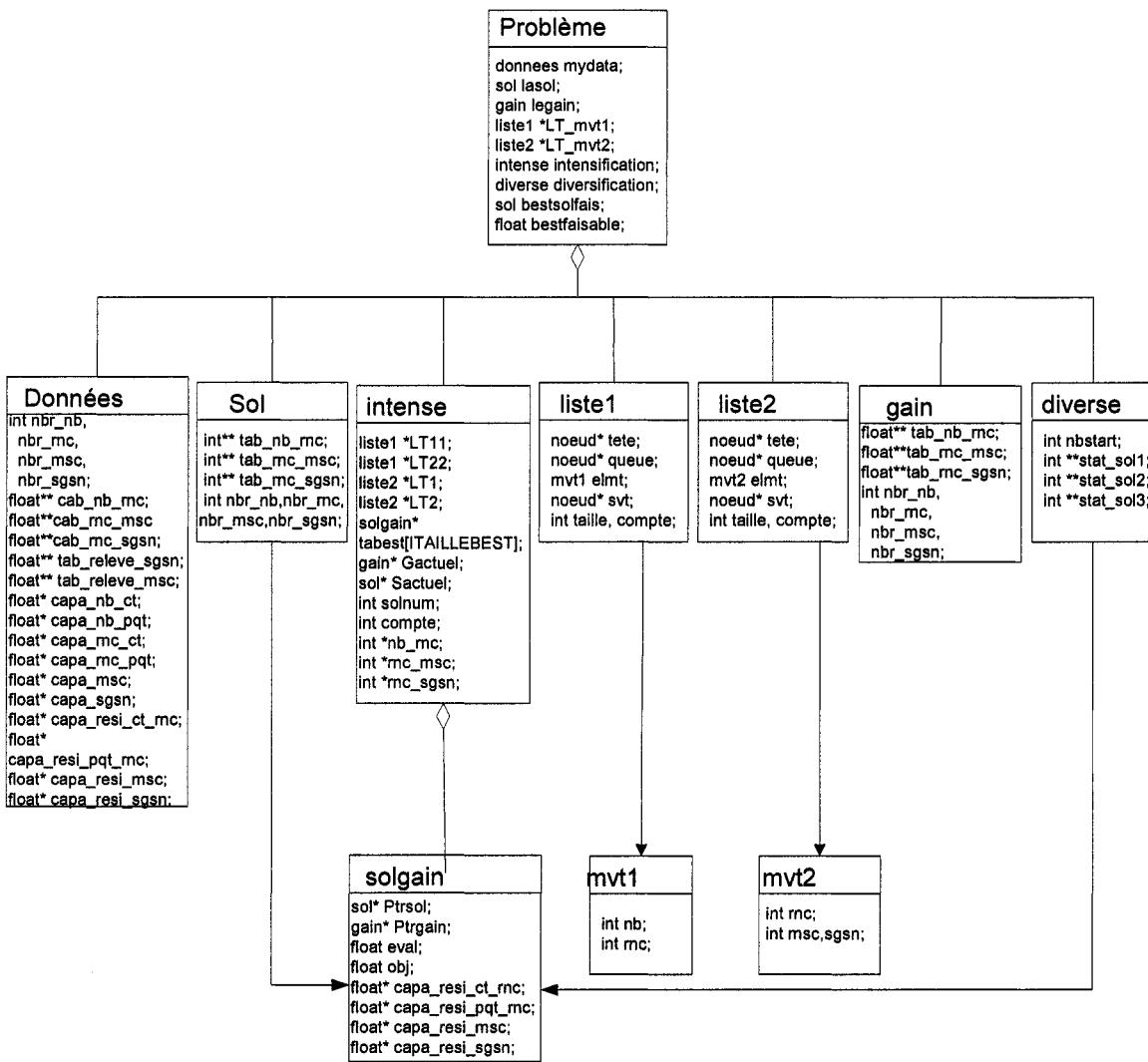


Figure 4.4 Diagramme des classes principales

La classe « *Intense* »

C'est dans cette classe qu'est implémenté le code du mécanisme de mémoire à moyen terme. Elle contient les principales méthodes suivantes :

- *mvt_permut_NB ()* et *mvt_permut_RNC ()* qui choisissent respectivement les mouvements de permutation possibles de type $m_1(a_1, a_2)$ et $m_2(b_1, b_2)$;
- *mvt_deplace_NB ()* et *mvt_deplace_RNC ()* qui choisissent respectivement les mouvements de déplacement de type $m_3(a, b)$ et $m_4(b, c, d)$;
- *applique_mvt_NB ()* et *applique_mvt_RNC ()* qui permettent d'appliquer les mouvements choisis par les méthodes précédentes, d'évaluer la solution obtenue et de mettre à jour les tables des gains ;
- *intensifie ()* qui est la méthode principale qui mène le processus d'intensification.

La classe « *Diverse* »

C'est elle qui permet de stocker l'information sur le nombre de redémarrages à l'aide de la variable *nbstart* et emmagasine, grâce à la méthode *compte ()*, les statistiques sur les solutions obtenues tout au long de l'application des mécanismes de mémoire à court et moyen termes. Ces statistiques serviront à construire les futures solutions de redémarrage.

La classe « *Sol* »

Elle permet de manipuler la solution du problème en stockant dans trois tableaux les patrons d'affectation des Nœuds B aux RNC (variable *tab_nb_rnc[][]*), des RNC aux MSC (variable *tab_rnc_msc[][]*) et des RNC aux SGSN (variable *tab_rnc_msc[][]*). Elle permet aussi de générer la solution initiale du problème, d'évaluer la fonction objectif de la solution courante et de vérifier le respect des contraintes de capacité grâce aux méthodes suivantes :

- *initiale (donnees mydata)* qui permet de générer la solution initiale à fournir à l'algorithme à partir des données contenues dans les fichiers d'entrée ;
- *initiale (const diverse diversdata, donnees mydata)* qui permet de générer la solution initiale lors du redémarrage dans le mécanisme de mémoire à long terme. La solution de départ est générée à partir des statistiques sur l'ensemble des solutions explorées lors de l'application des mécanismes de mémoire à court et moyen terme ;
- *objectif ()* qui permet d'évaluer la fonction objectif de la solution courante ;
- *contraintes ()* qui permet de vérifier si une solution respecte les contraintes de capacité au niveau des RNC, MSC et SGSN.

La classe « *donnees* »

Elle contient les données sur la taille du réseau à simuler (variables *nbr_nb*, *nbr_rnc*, *nbr_msc* et *nbr_sgsn*), les coûts de câblage (variables *cab_nb_rnc*, *cab_rnc_msc* et *cab_rnc_sgsn*), les coûts de relève entre Nœud B impliquant des SGSN et ceux impliquant des MSC (variables *tab_releve_sgsn* et *tab_releve_msc*), les capacités des RNC, des MSC et SGSN, le volume de trafic provenant de chaque Nœud B ainsi que les capacités résiduelles des RNC, MSC et SGSN. Ses principales méthodes sont :

- *lirefichier (char [])* qui permet de lire les données sur les coûts de câblage et de relève contenues dans le fichier de données fourni en entrée ;
- *lireficapa (char [])* qui permet de lire les valeurs des capacités des RNC, MSC et SGSN ainsi que le volume de trafic produit par les Nœuds B à partir du fichier des capacités fourni également en entrée ;
- *evaluation ()* qui permet d'évaluer une solution en pénalisant sa valeur objective si la solution ne respecte pas les contraintes de capacité au niveau des RNC, des MSC et des SGSN. La pénalité est appliquée en fonction de la quantité de capacité violée ainsi que le nombre de fois que celle-ci est violée.

La classe « *gain* »

Elle permet de générer les tables des gains des affectations des Nœuds B aux RNC, des RNC aux MSC et des RNC aux SGSN. Elle contient deux méthodes principales :

- *generer1 (const donnees, const sol)* qui permet, à partir des données du problème et d'une solution courante, de construire le tableau des gains de l'affectation des Nœuds B aux RNC. C'est ce tableau qui sera utilisé pour choisir le mouvement de réaffectation d'un Nœud B vers un nouveau RNC en sélectionnant dans ce tableau la plus petite valeur de gain ;
- *generer2 (const donnees, const sol)* qui permet de générer les tableaux des gains de l'affectation des RNC aux MSC et de l'affectation des même RNC aux SGSN. Ces tableaux seront utilisés pour choisir le mouvement de réaffectation d'un RNC vers un nouveau MSC et un nouveau SGSN.

Les classes « *mvt1* » et « *mvt2* »

Ces deux classes permettent de construire respectivement les mouvements de réaffectation de Nœud B à un nouveau RNC et de réaffectation de RNC vers un nouveau MSC et un nouveau SGSN.

Les classes « *liste1* » et « *liste2* »

Elles permettent de construire et d'entretenir les deux sous-listes taboues de la méthode RT. La classe « *liste1* » contient tous les mouvements inverses des mouvements de réaffectation d'un Nœud B à un nouveau RNC afin d'éviter de revenir à des solutions déjà visitées. Le nombre de mouvements qu'on peut emmagasiner est limité par la taille de la liste. Et la classe « *liste2* » contient tous les mouvements inverses des mouvements de réaffectation d'un RNC à de nouveaux MSC et SGSN. Les deux classes possèdent deux méthodes principales :

- *insere ()* qui permet d'insérer un mouvement dans la sous-liste taboue sous forme de liste FIFO (premier arrivé premier servi) ;
- *appartient ()* qui permet de tester si un mouvement est déjà inscrit dans la liste ou non.

La classe « *solgain* »

Elle permet de garder l'état d'exécution de la méthode RT à une solution donnée pour être en mesure de la restaurer plus tard dans le but d'intensifier la recherche autour de cette solution. Parmi les informations stockées, il y a le patron d'affectation de la solution (variable *sol* Ptrsol*), la valeur de la fonction objectif (variable *float obj*), l'évaluation de la solution (variable *float eval*), les tableaux des gains (variable *gain* Ptrgain*) ainsi que les capacités résiduelles des RNC, MSC et SGSN (variables *capa_resi_ct_rnc[]*, *capa_resi_pqt_rnc[]*, *capa_resi_msc[]* et *capa_resi_sgsn[]*). La méthode *set ()* permet d'effectuer le stockage des informations mentionnées ci-dessus.

4.1.5 Environnement matériel et logiciel

Le programme a été implémenté sous *Visual Studio 6.0* en code C++ Orienté Objet. Pour les simulations, nous avons utilisé un PC « *Pentium 4, 2.4 GHz et 524M de RAM* » avec « *Microsoft Windows 2000* » comme système d'exploitation. Pour la génération des fichiers de trafic et de capacité des équipements des différents réseaux simulés, nous nous sommes servis d'un programme *matlab* qui utilise une fonction aléatoire pour modéliser la demande en trafic au niveau des Nœuds B du réseau.

4.2 Plan d'expériences

Pour évaluer notre implémentation, nous la soumettrons à une série de tests afin de déterminer son efficacité et sa sensibilité par rapport à différents paramètres clés. Tout d'abord, nous testerons le comportement des différents mécanismes de mémoire mis en jeu pour la recherche de solutions. Pour ce faire, nous élaborons un plan d'expérience qui

nous permet de tester les principaux paramètres de chaque composante de mémoire soient : la taille de la liste taboue, le délai de déclenchement du mécanisme de « rappel », le poids de la sanction de la violation des contraintes liées au trafic de voix par rapport à celles liées au trafic de données pour le mécanisme de mémoire à court terme, le délai de déclenchement des mouvements de réaffectation en vue de rétablir les contraintes de capacités violées, la taille de la zone d'intensification pour le mécanisme de mémoire à moyen terme et le nombre de cycles de diversification de la recherche pour le mécanisme de mémoire à long terme. Les tests se feront un paramètre à la fois pour chaque mécanisme de mémoire. Pour chaque paramètre à tester, nous choisirons trois valeurs, et pour chaque valeur, nous exécuterons l'algorithme sur un certain nombre de cas tests avec des réseaux constitués de Nœuds B variant entre 6 et 200, de RNC variant entre 3 et 30 et de MSC et SGSN variant entre 2 et 8. Nous testerons par la suite, le comportement de la méthode dans son ensemble. Enfin, nous comparerons nos résultats avec une estimation d'une borne inférieure.

4.2.1 Comportement de la composante de mémoire à court terme

Dans l'implémentation de la mémoire à court terme, nous manipulons deux listes taboues, une pour les mouvements effectués pour l'affectation des Nœuds B aux RNC et l'autre pour les mouvements effectués pour l'affectation des RNC aux MSC et aux SGSN. Ainsi, dans le cadre de nos expériences, nous chercherons à déterminer l'effet de la taille de chacune de ces listes taboues sur la qualité des solutions obtenues. Pour cela, nous simulerons le programme avec deux listes de même taille et deux listes de taille différente afin de déterminer, entre les mouvements de type réaffectation de Nœuds B vers des RNC et ceux de type réaffectation de RNC vers des MSC et des SGSN, lesquelles qu'il faut faire en plus grand nombre pour obtenir une solution de qualité.

Également, notre implémentation de la mémoire à court terme met en jeu un mécanisme de «rappel» qui est appelé lorsqu'il s'écoule un certain temps depuis la dernière solution faisable afin de ramener l'exploration vers les solutions faisables. Ce mécanisme de rappel se met en branle en sanctionnant les gains des futures solutions non

faisables pour les RNC, MSC et SGSN dont la capacité a été violée. La sanction est appliquée séparément pour la violation des capacités en commutation de circuits et en commutation de paquets. Ainsi, pour ce mécanisme de « rappel », nous allons simuler l'effet de son délai de déclenchement, l'effet du poids de la sanction de la violation du trafic de voix par rapport au trafic de données et l'effet de la variation de son intensité maximale d'application (qui est le multiplicateur *SEVERITE* qui est incrémenté à chaque solution non faisable trouvée jusqu'à une valeur maximale *MAXSEV*). L'ensemble de ces tests sera effectué en désactivant toutes les autres composantes de mémoire (moyen et long terme).

4.2.2 Comportement de la composante de mémoire à moyen terme

Lors de l'application de la composante de mémoire à court terme, la méthode retient un nombre de solutions égal à *ITAILLEBEST* parmi les meilleures solutions trouvées au cours de l'exploration de l'espace de recherche. C'est aux alentours de ces solutions que la composante de mémoire à moyen terme intensifie la recherche de meilleures solutions. Pour ce faire, la RT utilise deux types de mouvements qui sont appliqués aux deux niveaux d'affectation (Nœuds B vers RNC et RNC vers MSC/SGSN) : le premier type de mouvement consiste à permuter dans un premier temps deux Nœuds B, ensuite deux RNC (c'est ce type de mouvement qui est appliqué par défaut); le deuxième type de mouvement consiste à réaffecter séquentiellement un Nœud B vers un nouveau RNC et un RNC vers un nouveau MSC et un nouveau SGSN, ceci dans le but de rétablir les contraintes de capacité (c'est un type de mouvement qui est mis en œuvre après un nombre *IDRESPECT* de solutions consécutives non faisables). Dans cette partie de la simulation, nous chercherons à déterminer l'effet du délai de déclenchement des mouvements de réaffectation en utilisant trois valeurs différentes de la variable *IDRESPECT*. Également, nous simulerons l'influence de la taille de la région d'intensification (variable *ITAILLEBEST*) sur la qualité des solutions obtenues et sur le temps de calcul. Les tests seront effectués en activant seulement les mécanismes de mémoire à court et moyen terme.

4.2.3 Comportement de la composante de mémoire à long terme

Tout au long de l'exploration de l'espace des solutions, la RT garde des statistiques sur les solutions déjà visitées. C'est sur la base de ces statistiques que la méthode génère la solution de départ à chaque redémarrage du mécanisme de diversification. Le paramètre le plus important du mécanisme de diversification est le nombre *NBSTART* de redémarrages effectués. Nous analyserons donc son effet sur les solutions obtenues.

4.2.4 Comportement général de la méthode

Pour étudier le comportement général et l'efficacité des différents mécanismes mis en œuvre dans notre adaptation de la RT, nous allons générer des séries de tests en faisant varier séparément, puis simultanément le nombre de Nœuds B, le nombre de RNC et le nombre de commutateurs (MSC/SGSN). Pour chacune de ces séries, nous allons exécuter notre programme respectivement avec la composante de mémoire à court terme seulement, puis avec les combinaisons de mémoire à court et moyen terme, et celles de mémoire à court et long terme, et enfin avec toutes les composantes de mémoire de la méthode. Ceci dans le but de montrer l'amélioration qu'apportent les mécanismes d'intensification et de diversification sur la meilleure solution trouvée après l'application de la composante de mémoire à court terme, et aussi d'estimer le pourcentage de solutions faisables fournies par chaque mécanisme. Enfin, afin de nous donner une idée sur la courbe d'évolution du temps de calcul par rapport à l'augmentation de la taille du problème (taille du réseau), nous ferons une simulation du temps de calcul de la méthode RT pour des tailles de réseaux différentes.

4.2.5 Comparaison avec une borne inférieure

La solution obtenue après l'application complète de la RT est souvent un optimum local qui peut, dans certains cas, s'avérer être l'optimum global. Mais, à défaut de connaître d'avance l'optimum global, il est difficile de se prononcer sur la qualité de la

solution obtenue. Dans ce contexte, nous nous contenterons de définir une borne inférieure à notre problème et de comparer nos résultats avec celle-ci. Pour définir la borne inférieure, nous relâcherons certaines contraintes telles les limitations de capacités au niveau des équipements comme les RNC, MSC et SGSN et pour les comparaisons, nous déterminerons, pour chaque série de tests, l'écart entre la borne inférieure du problème simulé et la solution finale obtenue par la RT.

4.3 Simulations et analyse des résultats

Pour effectuer les tests de performance, nous avons conduit des séries de tests, en suivant le plan d'expérience décrit dans la Section 4.2. Pour ce faire, nous avons divisé notre simulation en trois volets : le premier consiste à simuler indépendamment le comportement de chaque composante de mémoire; le deuxième consiste à simuler le comportement général de la méthode en appliquant séquentiellement les trois composantes de mémoire; le dernier volet consiste, quant à lui, à comparer les résultats obtenus lors du deuxième volet à une estimation de la borne inférieure de chacune des séries de tests utilisées. Le Tableau 4.1 illustre les cas tests utilisés pour le premier volet de la simulation.

Tableau 4.1 Cas tests utilisés pour le comportement des composantes de mémoire

Nombre de Nœuds B	Nombre de RNC	Nombre de MSC	Nombre de SGSN	Nombre de cas tests
6	3	2	2	20
15	8	3	3	20
20	11	4	4	20
50	15	5	5	20
100	20	6	6	10
150	25	7	7	10
200	30	8	8	10

4.3.1 Comportement de la composante de mémoire à court terme

Dans l'implémentation de la mémoire à court terme, les principaux paramètres sont : les deux sous-listes taboues, une pour les mouvements effectués pour l'affectation des Nœuds B aux RNC et l'autre pour les mouvements effectués pour l'affectation des RNC aux MSC et aux SGSN, le poids de la sanction de la violation des contraintes liées au trafic de voix par rapport à celles liées au trafic de données et le mécanisme de «rappel» qui permet de ramener l'exploration vers les solutions faisables. Pour tester les effets de ces paramètres, nous avons effectué des simulations en considérant seulement le mécanisme de mémoire à court terme. Les deux autres mécanismes (mémoire à moyen terme et mémoire à long terme) ont été désactivés.

Effet de la taille des deux sous-listes taboues

Dans notre adaptation, la taille de la liste taboue indique le nombre de mouvements inverses qu'on peut garder en mémoire afin d'éviter de les refaire et d'aboutir à des solutions déjà visitées. Mais puisqu'on manipule deux sous-listes taboues, nous allons dans cette partie de la simulation, considérer que les deux sous-listes taboues ont la même taille ($LT1=LT2$). Ceci aura pour effet de ne favoriser aucune des deux sous-affectations (Nœud B vers RNC et RNC vers MSC/SGSN). Nous avons donc exécuté notre programme avec des sous-listes taboues de même taille égale à 5, 10 et 20. Ensuite, nous avons effectué une moyenne des coûts des solutions des cas-tests de chaque taille de réseau. La Figure 4.5 illustre la moyenne d'évaluation du coût des solutions obtenues en fonction du nombre de cellules (Nœud B), de RNC, de MSC et de SGSN, et cela pour chacune des valeurs choisies pour la taille de LT1 et LT2.

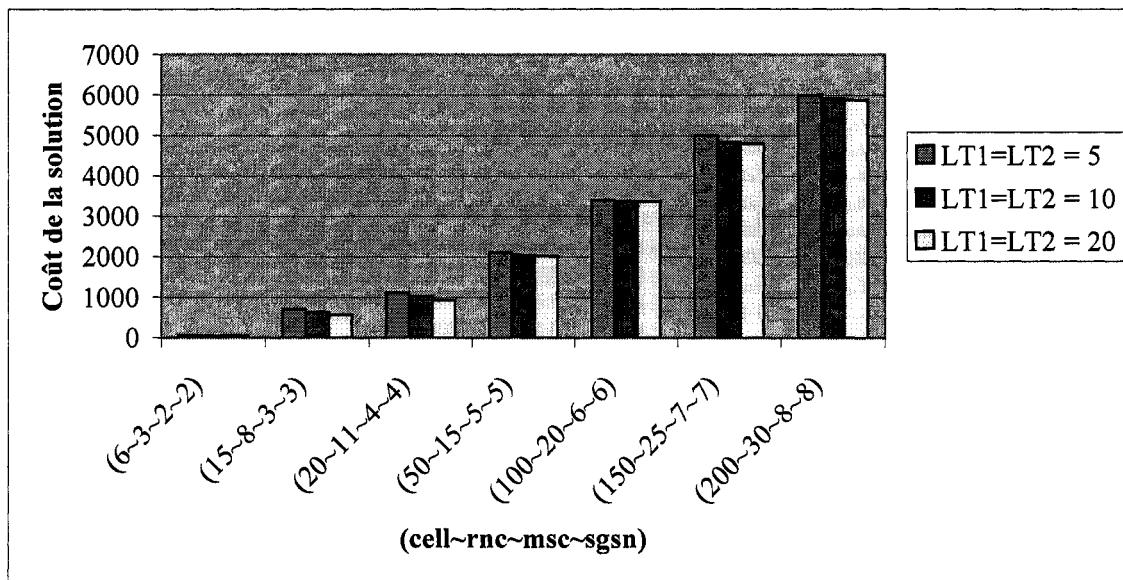


Figure 4.5 Effet de la taille de la LT sur les solutions obtenues

On remarque que des sous-listes taboues de même taille égale à 20 donnent en moyenne de meilleurs résultats, c'est à dire des solutions de coûts moindres. En effet, par rapport au cas où $LT1=LT2=5$, on peut affirmer que l'algorithme explore plus efficacement l'espace de recherche en ne revenant pas sur un grand nombre de solutions déjà explorées car les mouvements qui y mènent sont mémorisés dans $LT1$ ou $LT2$. Ainsi, à moins que ces mouvements n'améliorent la solution courante, ils sont exclus de l'ensemble des mouvements à effectuer. Avec des sous-listes taboues de taille réduite, la recherche a peut être plus de chance de quitter les zones de solutions non faisables en retournant à des solutions déjà visitées et faisables. Mais, elle explore moins efficacement l'espace des solutions et offre moins de chance d'aboutir à de bonnes solutions.

Rapport de taille entre les deux sous-listes taboues

Dans cette partie de la simulation, nous avons testé l'effet d'une taille différente pour les deux sous-listes taboues. Si la sous-liste taboue de l'affectation des Nœud B aux

RNC est plus grande que celle de l'affectation des RNC aux MSC/SGSN, cela signifie qu'on pourra mémoriser plus de mouvements interdits de réaffectation de Nœuds B aux RNC. Ainsi, cela va rendre la contribution des mouvements de réaffectation de Nœuds B aux RNC plus effective à l'exploration de l'espace des solutions. L'effet inverse se produit lorsque la sous-liste taboue de l'affectation des RNC aux MSC/SGSN est plus grande que celle de l'affectation des Nœuds B aux RNC. En effet dans ce cas là, ce seront les mouvements de réaffectation de RNC vers les MSC/SGSN qui seront les plus efficaces. Pour effectuer les tests, nous avons pris trois paires de valeurs. Dans le premier cas, la sous-liste taboue de l'affectation des Nœuds B aux RNC (notée LT1) est égale à 10 et celle de l'affectation des RNC aux MSC/SGSN (notée LT2) est égale à 5. Dans le deuxième cas, nous avons pris LT1 égale à 5 et LT2 égale à 10, et afin de comparer les résultats avec le cas où les deux listes ont la même taille, nous avons simulé le programme pour LT1=LT2=5. La Figure 4.6 illustre la moyenne d'évaluation du coût des solutions obtenues en fonction du nombre de cellules (Nœud B), de RNC, de MSC et de SGSN, et cela pour chacune des paires de valeurs choisies pour la taille des deux sous-listes taboues.

Les résultats obtenus nous indiquent que, pour des tailles de réseau dont le nombre de cellules est inférieur ou égal à 100, avoir LT1 supérieure à LT2 permet en général d'améliorer les solutions obtenues par rapport au cas où on a la même taille pour les deux sous-listes taboues égale à 5. Ceci signifie que les mouvements effectués au niveau de l'affectation des Nœuds B aux RNC contribuent plus efficacement à la recherche de meilleures solutions que ceux effectués au niveau de l'affectation des RNC aux MSC/SGSN. En effet, à cause du nombre de Nœuds B qui est plus grand que le nombre de RNC, on a plus de chance d'effectuer des mouvements de réaffectation de Nœuds B à un RNC et que cela mène vers une solution non encore explorée qu'un mouvement de réaffectation de RNC à un MSC et à un SGSN. Par contre, pour les réseaux de plus grande taille, nous obtenons de meilleures solutions pour LT2 supérieure à LT1 car, nous avons plus de possibilités de mouvements de réaffectation des RNC vers les commutateurs (MSC/SGSN) que dans le cas où LT1 est supérieure à LT2. À cause du

fait que, les commutateurs (MSC/SGSN) sont les derniers équipements à prendre en charge le trafic provenant des Nœuds B avant qu'il soit routé vers sa destination, les mouvements de réaffectation des RNC vers les commutateurs (MSC/SGSN) sont par conséquent les plus efficaces en terme de découverte de nouvelles solutions faisables à coût moindre. En effet, ces mouvements prennent en compte non seulement les capacités des commutateurs (MSC/SGSN), mais aussi l'état de saturation des RNC.

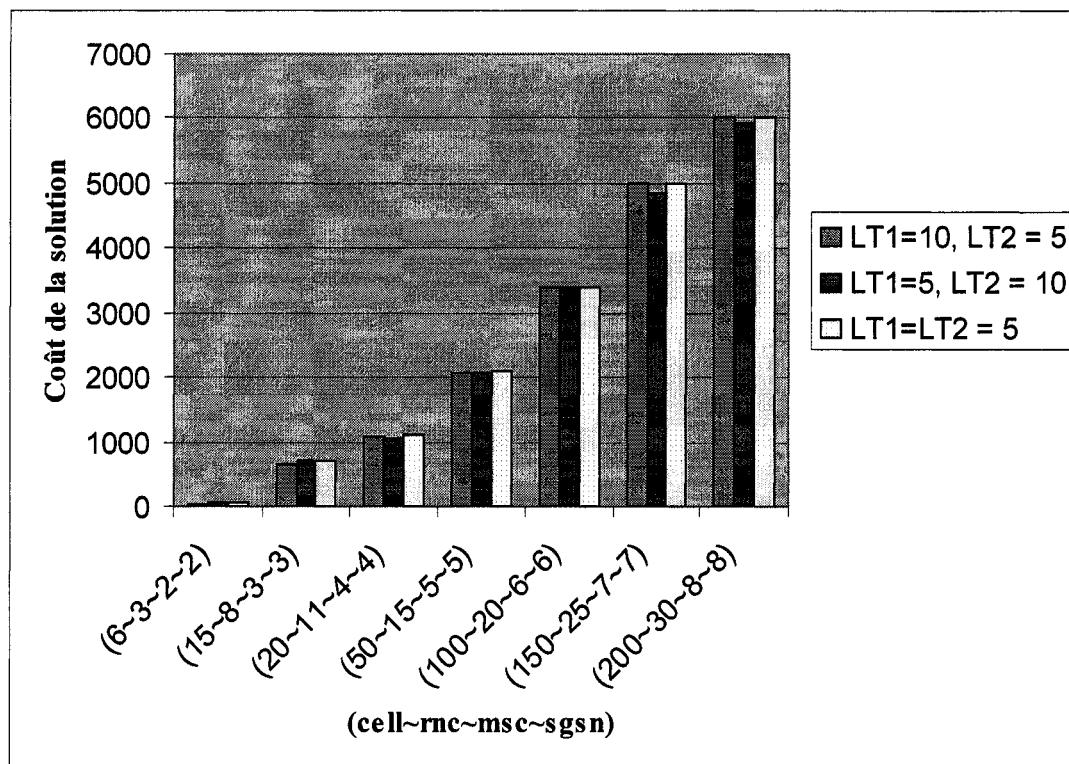


Figure 4.6 Rapport de taille entre les deux sous-listes taboues

Effet du délai de déclenchement du mécanisme de « rappel »

La mémoire à court terme met en jeu un mécanisme de « rappel » qui est appellé lorsqu'il s'écoule un certain temps depuis la dernière solution faisable afin de ramener l'exploration vers les solutions faisables. Dans notre programme, ce délai est représenté par la variable *DRESPECT*. Sa valeur ne devrait être a priori ni trop grande sinon la

recherche s'éloignera des zones prometteuses, ni trop petite sinon la recherche est constamment ramenée vers des zones de solutions qui respectent les contraintes de capacités mais qui ne sont pas nécessairement bonnes en terme de coûts, omettant ainsi certaines solutions qui sont bonnes. Pour simuler le phénomène, nous avons pris trois valeurs pour la variable *DRESPPECT* (2, 5 et 8) pour que le mécanisme de « rappel » soit déclenché après respectivement 2, 5 et 8 solutions consécutives non faisables. La Figure 4.7 illustre la moyenne des coûts des solutions obtenues en fonction du nombre de cellules (Nœud B), de RNC, de MSC et de SGSN, et cela pour chacune des valeurs choisies pour *DRESPPECT*.

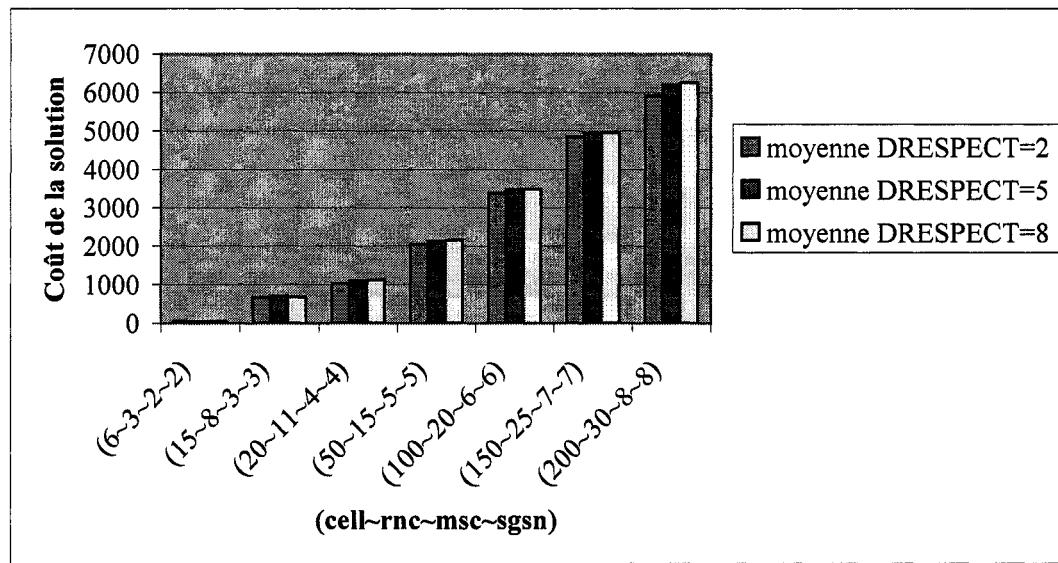


Figure 4.7 Effet du délai de déclenchement du mécanisme de « rappel »

Selon les résultats obtenus, nous constatons qu'un délai de déclenchement de 2 solutions consécutives non faisables permet d'obtenir un meilleur coût. Cela confirme le fait que plus on tarde le déclenchement du mécanisme de « rappel » (cas de *DRESPPECT*=8), plus l'on risque de s'éloigner des zones prometteuses où on a le plus de chance de trouver de meilleures solutions.

Effet du poids de la sanction de la violation du trafic de voix versus le trafic de données

Lorsque le mécanisme de « rappel » est mis en branle, on sanctionne les gains des futures solutions dont les capacités au niveau des RNC, des MSC ou des SGSN ont été violées. Cette sanction est appliquée grâce à des valeurs de *PÉNALITÉ* qui sont ajoutées au gain normal ou multipliées à la quantité de capacité violée avant le calcul du nouveau gain. La sanction est appliquée séparément pour la violation des capacités en commutation de circuits et en commutation de paquets. Nous avons simulé trois cas de figure : dans le premier cas (*VIOL_PQT_PLUS_SANCTIONNÉ*), nous avons appliqué une sanction plus sévère pour la violation des capacités en commutation de paquets, dans le deuxième cas, nous avons sanctionné plus sévèrement la violation des capacités en commutation de circuits (*VIOL_CT_PLUS_SANCTIONNÉ*) et enfin, pour avoir une base de référence, nous avons considéré le cas où on a une sanction égale pour les deux types de commutations (*SANCT_PQT=CT*). La Figure 4.8 illustre les résultats d'évaluation du coût des solutions en fonction du nombre de cellules (Nœud B), de RNC, de MSC et de SGSN, et cela pour chacun des cas de sanction appliqués.

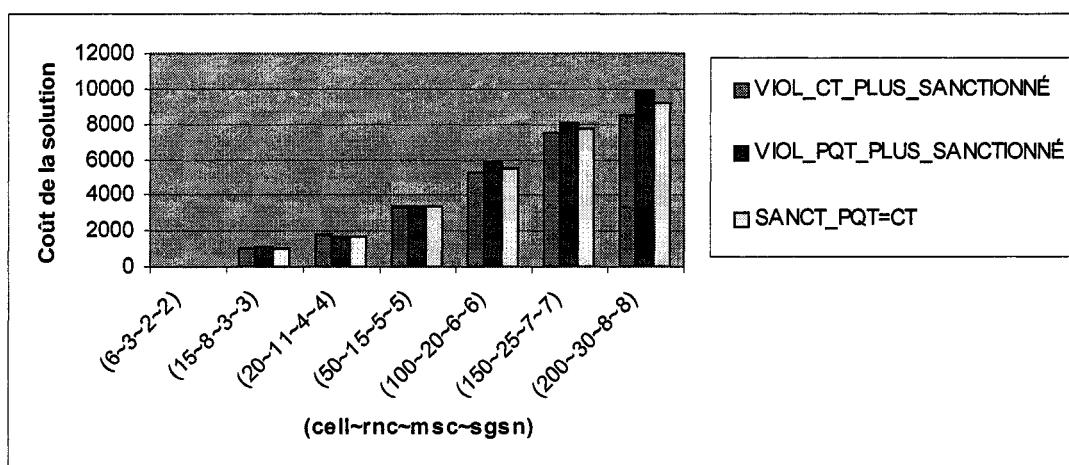


Figure 4.8 Effet du poids de la sanction de la violation du trafic de voix par rapport au trafic de données

Les résultats obtenus nous montrent qu'en sanctionnant plus sévèrement les violations des capacités en commutation de circuits par rapport à celles en commutation de paquets, on arrive à obtenir de meilleures solutions que dans le cas contraire ou dans le cas d'une sanction équitable. Cela s'explique par le fait qu'il survient plus souvent de violation de capacités en commutation de paquets qu'en commutation de circuits.

Effet de l'intensité maximale du mécanisme de « rappel »

L'intensité de la sanction appliquée lorsque le mécanisme de « rappel » se déclenche est modulée progressivement à l'aide d'un multiplicateur *SEVERITE* qui est incrémenté à chaque nouvelle solution qui viole les contraintes de capacités et ce, jusqu'à une valeur maximale *MAXSEV*. Afin d'observer l'impact de ce multiplicateur sur les solutions finales obtenues, nous avons simulé le programme avec trois valeurs de *MAXSEV* de 10, 15 et 20. La Figure 4.9 illustre la moyenne des résultats obtenus pour les trois valeurs de *MAXSEV*.

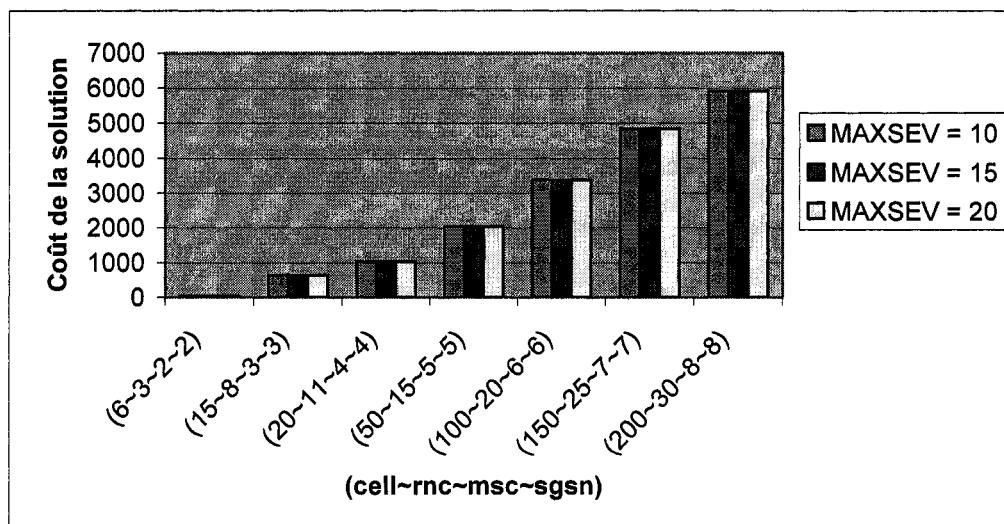


Figure 4.9 Effet de la variation de l'intensité maximale du mécanisme de « rappel »

Les résultats obtenus ne montrent aucune différence pour les trois valeurs de *MAXSEV*. Cela s'explique par le fait que, dans les deux cas de *MAXSEV* égale à 15 et 20, on n'arrive pas à incrémenter la variable *SEVERITE* à une valeur plus grande que 10, si bien qu'il n'y a aucune différence avec les résultats obtenus dans le cas où *MAXSEV* est égale à 10. En effet, pour toutes les tailles de réseaux considérées, la méthode RT ne trouve pas plus de 10 solutions consécutives qui violent les contraintes de capacités.

4.3.2 Comportement de la composante de mémoire à moyen terme

Dans l'implémentation de la mémoire à moyen terme, les principaux paramètres mis en jeu sont : le délai de déclenchement des mouvements de réaffectation, qui ont pour but de rétablir les contraintes de capacités violées, et la taille de la région d'intensification. Pour tester les effets de ces paramètres, nous avons effectué des simulations en activant les mécanismes de mémoire à court et moyen termes car le mécanisme de mémoire à moyen terme a besoin des résultats de l'application du mécanisme de mémoire à court terme pour fonctionner normalement.

Effet du délai de déclenchement des mouvements de réaffectation

Pendant l'exploration de l'espace des solutions aux alentours des meilleures solutions produites et transmises par la composante de mémoire à court terme, le mécanisme de mémoire à moyen terme comptabilise le nombre de solutions consécutives qui violent les contraintes de capacité. Lorsque ce nombre atteint la valeur *IDRESPECT*, fixée au préalable, des mouvements de réaffectation de Nœuds B et de RNC, basés sur le rétablissement des capacités violées, sont déclenchés. La valeur de *IDRESPECT* ne doit pas être a priori ni trop grande sinon la recherche s'éloignera des zones prometteuses, ni trop petite sinon la recherche sera trop contrainte et pourra manquer de bonnes solutions. Afin de simuler l'effet de ce délai de déclenchement des mouvements de réaffectation, nous avons choisi des valeurs de *IDRESPECT* de 2, 5 et 8. La Figure 4.10 illustre la moyenne des résultats obtenus pour les trois valeurs de *IDRESPECT*.

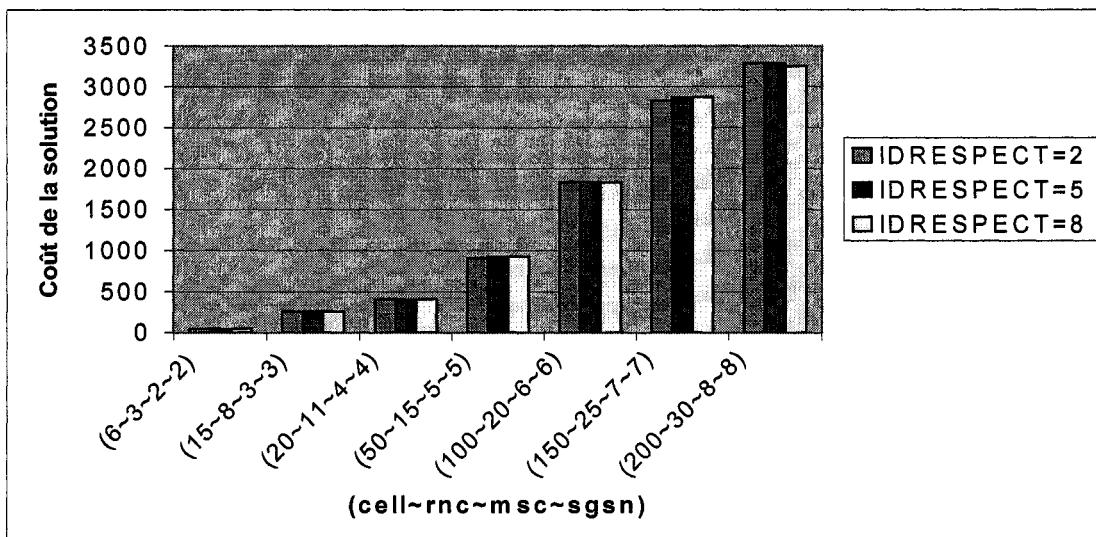


Figure 4.10 Effet du délai de déclenchement des mouvements de réaffectation

Nous obtenons dans l'ensemble de meilleures solutions pour un délai *IDRESPECT*=2. En effet, comme dans le cas de la variable *DRESPPECT* de la composante de mémoire à court terme, un déclenchement rapide mais non prématûre des mouvements servant à rétablir les contraintes de capacités violées permet à la RT de commencer très tôt la recherche de solutions faisables et moins coûteuses. Les résultats obtenus nous ont amené à choisir *IDRESPECT*=2 pour le reste des simulations impliquant la composante de mémoire à moyen terme.

Influence de la taille de la région d'intensification

Lors de son application, la composante de mémoire à court terme garde un nombre *ITAILLEBEST* des dernières meilleures solutions trouvées. Ces solutions sont transmises à la composante de mémoire à moyen terme pour intensifier la recherche autour de celles-ci. Cet ensemble de solutions constitue ce qu'on appelle « *région d'intensification* ». Nous avons donc étudié l'impact de la taille de cette région d'intensification sur la qualité des solutions obtenues au final. Pour ce faire, nous avons

simulé le programme avec des valeurs de *ITAILLEBEST* de 3, 5 et 8. Il est évident qu'une région d'intensification plus grande nous donne plus de chance d'améliorer la meilleure solution produite par le mécanisme de mémoire à court terme, mais au prix d'un temps de calcul beaucoup plus long. Cependant, il peut arriver qu'une taille de la région d'intensification plus grande ne permette pas nécessairement d'avoir une meilleure solution par rapport à une région de taille moins grande, tout simplement parce que l'espace de recherche aux alentours de certaines des solutions de la région d'intensification peut être pauvre en terme de bonnes solutions. La Figure 4.11 illustre la moyenne des résultats obtenus pour les trois valeurs de *ITAILLEBEST*.

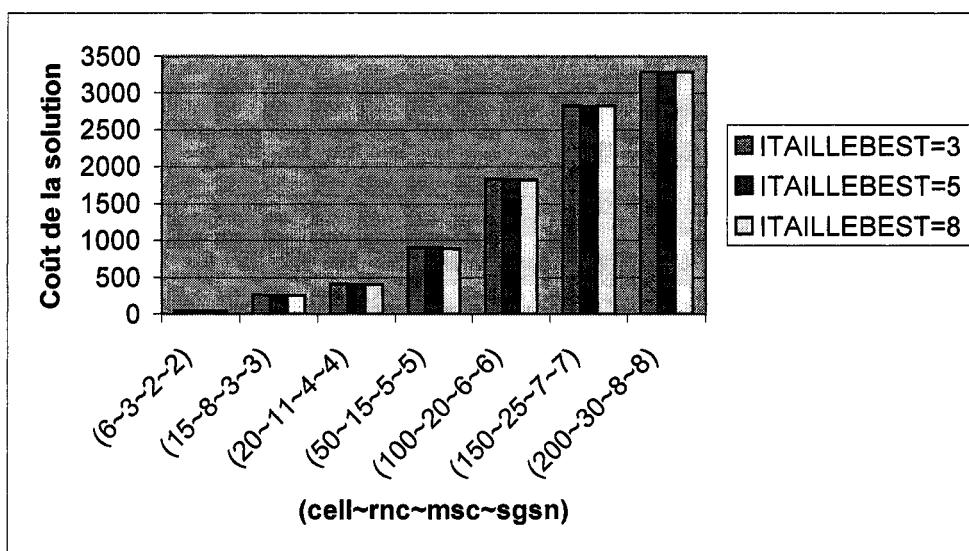


Figure 4.11 Effet de la taille de la région d'intensification

Nous constatons que, pour des problèmes de petite taille (jusqu'à 15 cellules, 8 RNC et 3 commutateurs MSC/SGSN), une région d'intensification de taille 3 convient parfaitement. Par contre, pour les problèmes de taille plus élevée, une valeur de *ITAILLEBEST* égale à 5 donne de meilleurs résultats. En effet, pour un problème de petite taille, à cause des possibilités limitées, l'intensification améliore assez vite la

solution de la composante de mémoire à court terme. Pour un problème de taille plus élevée, il y a plus de possibilités et les améliorations apportées par l'intensification sont meilleures pour une région d'intensification un peu plus grand : dans ce cas-ci, une taille de l'ordre de 5 solutions.

4.3.3 Comportement de la composante de mémoire à long terme

Tout au long de l'exploration de l'espace des solutions, la RT garde des statistiques sur les solutions déjà visitées. C'est sur la base de ces statistiques que la méthode génère la solution de départ à chaque redémarrage du mécanisme de diversification.

Effet du nombre de redémarrages

L'application de la composante de mémoire à long terme n'est rien d'autre qu'un rappel des composantes de mémoire à court et moyen termes mais avec une solution initiale basée sur l'historique de la recherche précédente. Ainsi, nous pouvons effectuer un nombre *NBSTART* de redémarrages successifs pour améliorer la meilleure solution trouvée par la combinaison des composantes de mémoire à court et moyen termes. Nous avons donc simulé l'impact de la valeur *NBSTART* sur la qualité de la solution finale. Un nombre de redémarrages plus grand devrait en principe nous donner plus de chance d'améliorer la meilleure solution des composantes de mémoire à court et moyen termes, mais au prix d'un temps total de recherche plus long. Les valeurs choisies pour la variable *NBSTART* sont : 3, 5 et 8. La Figure 4.12 illustre la moyenne des coûts des solutions obtenues. Les résultats obtenus nous montrent qu'effectivement une valeur de *NBSTART* égale à 8 nous donne de bien meilleures solutions que les cas où *NBSTART* est égale à 3 ou 5. En effet, en redémarrant 8 fois le processus d'application des mémoires à court et moyen termes, nous arrivons à améliorer de beaucoup la solution obtenue par la combinaison des mécanismes de mémoire à court et moyen termes appliquée une seule fois. Ceci s'explique par le fait que la diversification aura emmagasiné un grand nombre de statistiques sur l'espace des solutions déjà explorées et qu'à chaque redémarrage, la

solution initiale construite à partir de ces statistiques est assez différente des solutions déjà explorées. Les zones de recherche sont donc très variées et notre chance de rencontrer une meilleure solution augmente.

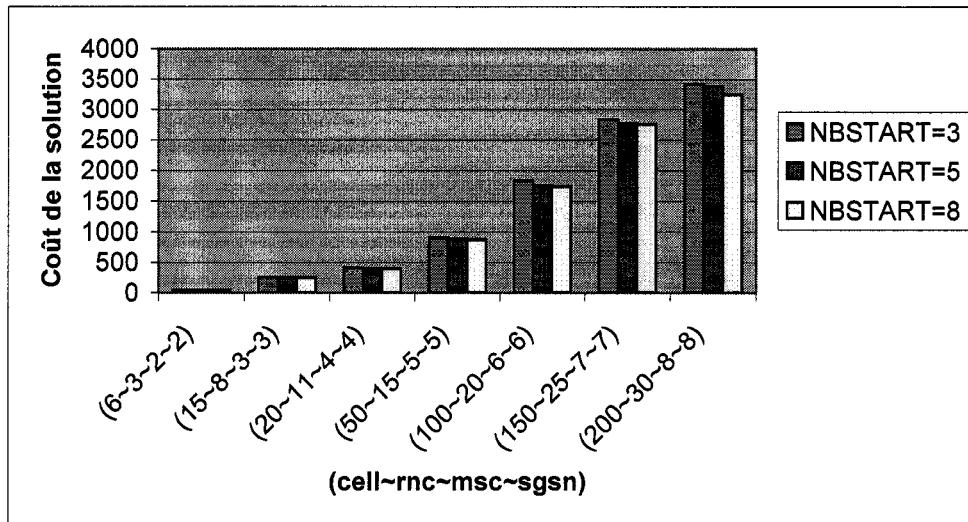


Figure 4.12 Effet du nombre de redémarrage sur les solutions obtenues

4.3.4 Comportement général de la méthode

Dans le deuxième volet de la simulation, nous avons étudié le comportement général et l'efficacité de notre adaptation de la RT pour la résolution de problèmes typiques d'affectation de cellules (Nœuds B) dans les réseaux mobiles 3G. Pour cela, nous avons généré 7 séries de tests dont chaque série contient un nombre de cas tests variant entre 70 et 200. Le Tableau 4.2 illustre le détail des séries de tests exécutés. La composition de chaque série de tests est présentée à l'annexe A. Pour chacune de ces séries, nous avons exécuté notre programme respectivement avec la composante de mémoire à court terme, puis avec les combinaisons de mémoire à court et moyen termes, et celles de mémoire à court et long termes, et enfin avec toutes les composantes de

mémoire de la méthode. Les figures 4.13 à 4.19 et 4.20 à 4.26 décrivent d'une part l'amélioration apportée par l'intensification et la diversification sur la solution trouvée par la composante de mémoire à court terme, et d'autre part le pourcentage de solutions faisables trouvées par chaque mécanisme de mémoire.

Tableau 4.2 Séries de tests utilisées pour le comportement général de la méthode

N° série de tests	Nombre de Nœuds B	Nombre de RNC	Nombre de MSC	Nombre de SGSN	Nombre de cas tests
1	variable (6-200)	3	2	2	180
2	variable (15-200)	8	3	3	180
3	20	variable (6-20)	4	4	200
4	50	variable (6-30)	5	5	200
5	100	20	Variable (2-8)	variable (2-8)	70
6	150	25	variable (2-8)	variable (2-8)	70
7	variable (6-200)	variable (3-30)	variable (2-8)	variable (2-8)	100

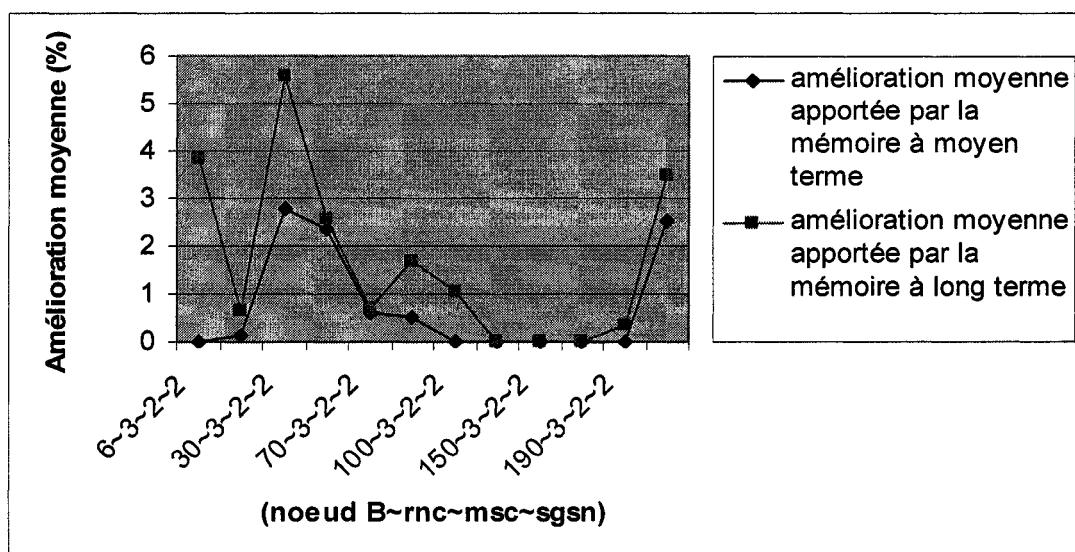


Figure 4.13 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 1)

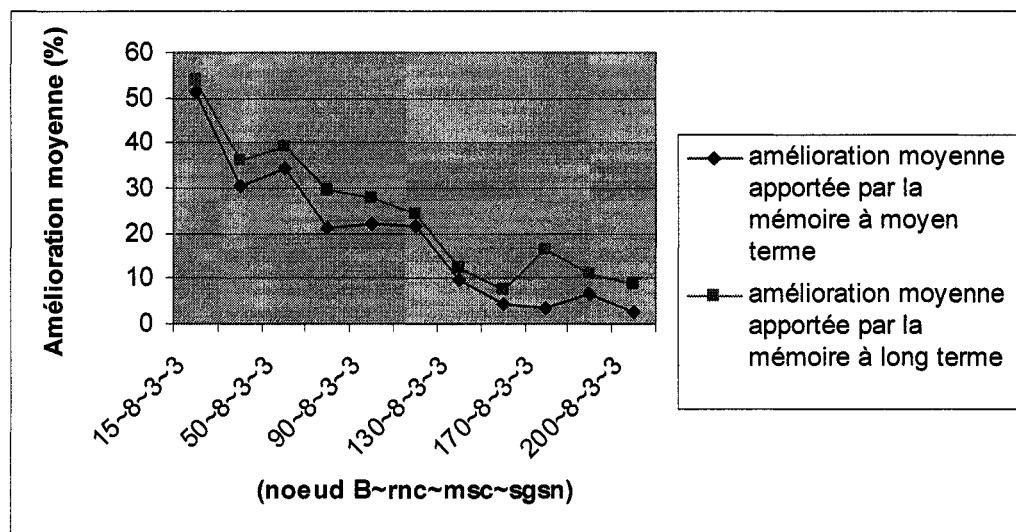


Figure 4.14 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 2)

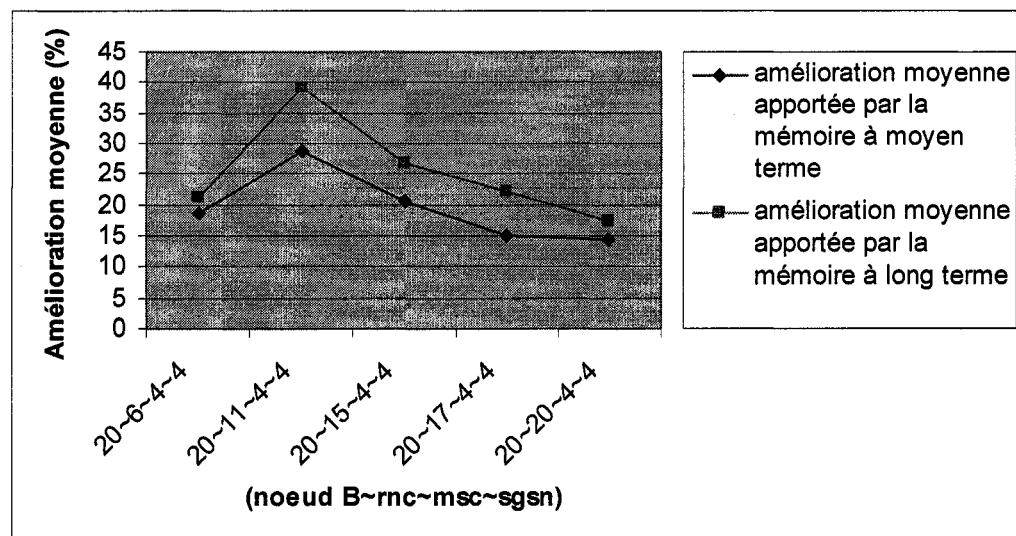


Figure 4.15 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 3)

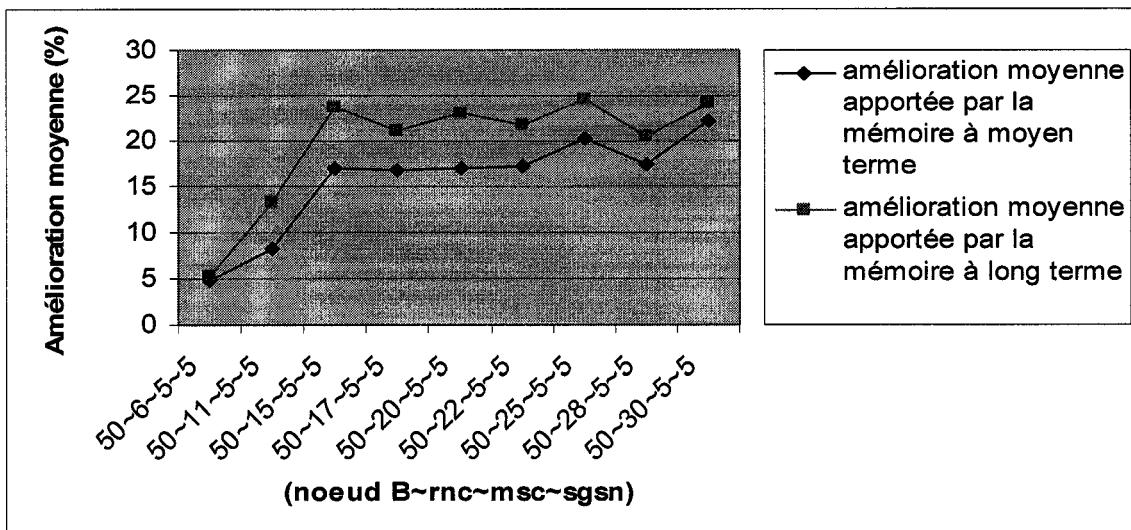


Figure 4.16 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 4)

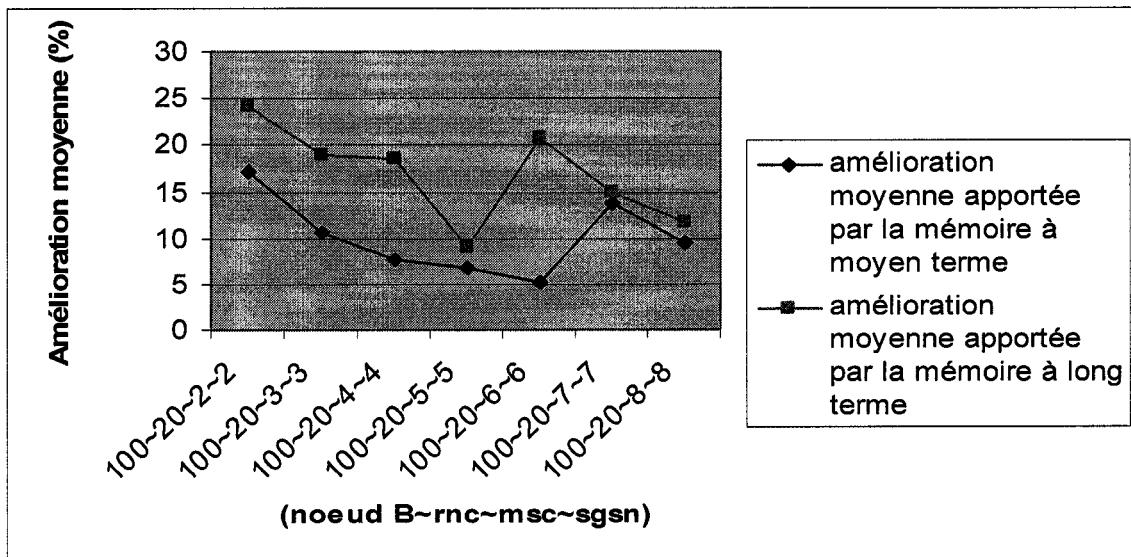


Figure 4.17 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 5)

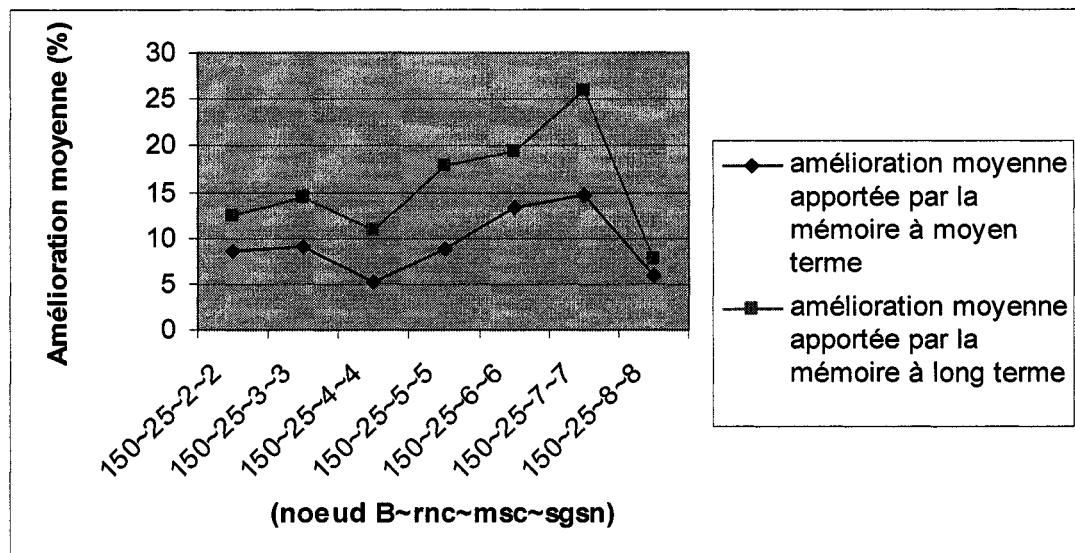


Figure 4.18 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 6)

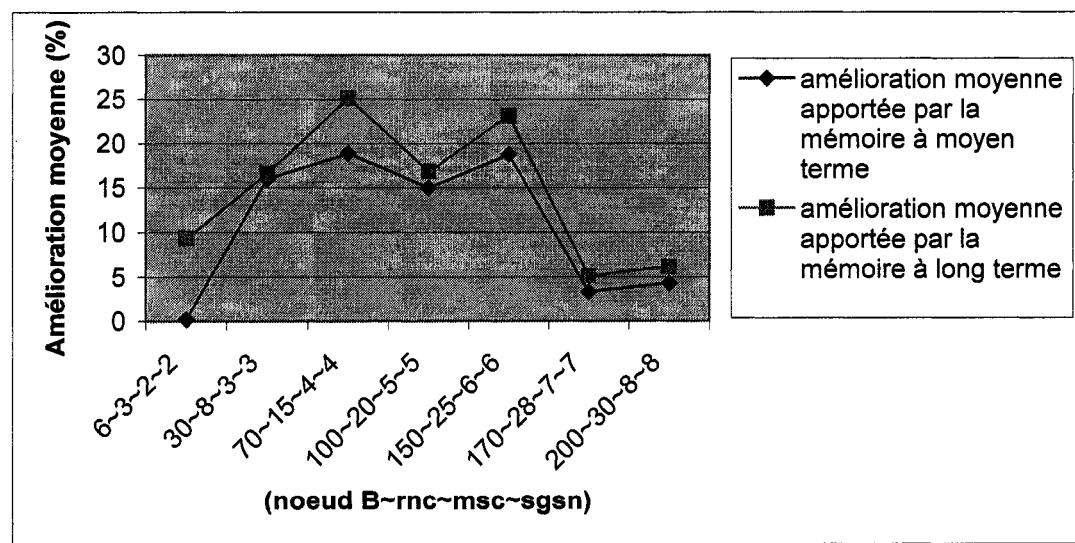


Figure 4.19 Améliorations des solutions de la mémoire à court terme par les mémoires à moyen et long termes (série n° 7)

En comparant les résultats de simulation de la série 1 (Figure 4.13) par rapport aux résultats des autres séries (2 à 7), nous constatons qu'il y a une meilleure contribution des mécanismes de mémoire à moyen et long termes à l'amélioration des solutions, pour un nombre de RNC supérieur à 3. En effet, sur la Figure 4.13, les pourcentages d'amélioration pour les deux mécanismes de mémoire sont tous inférieurs à 6% alors que sur les autres figures, les pourcentages varient en général entre 10 et 50%. Ceci s'explique par le fait que, le RNC étant l'équipement intermédiaire entre les Nœuds B et les commutateurs; pour un réseau avec trois RNC, lors des mouvements de réaffectation des Nœuds B aux RNC et des RNC aux MSC et SGSN, la composante de mémoire à court terme arrive à examiner presque tous les cas possibles, si bien que les composantes de mémoire à moyen et long termes amènent rarement la recherche vers des zones non encore explorées. Par contre, pour un réseau avec plus de trois RNC, la RT a plus de choix de mouvements de réaffectation. Ainsi, l'intensification permet de raffiner les choix faits par la composante de mémoire à court terme et la diversification amène la recherche vers des zones non encore explorées augmentant ainsi les chances d'améliorer la meilleure solution trouvée par la composante de mémoire à court terme. En observant les figures 4.17 et 4.18, nous remarquons que, pour une même variation du nombre de commutateurs (MSC et SGSN), malgré l'augmentation légère du nombre de Nœuds de 100 à 150 et du nombre de RNC de 20 à 25, les pourcentages d'amélioration des mécanismes de mémoire à moyen et long termes demeurent stables. En effet, sur ces deux figures, les pourcentages d'amélioration se situent entre 5 et 25%. Cela prouve que les mécanismes de mémoire à moyen et long terme performent bien malgré une légère variation de la taille du problème au niveau de la sous-affectation des Nœuds B aux RNC. Enfin, en observant les figures 4.13 à 4.19, nous constatons que la courbe d'amélioration moyenne apportée par la mémoire à long terme est toujours au-dessus de celle de l'amélioration apportée par la mémoire à moyen terme. Cela montre que la diversification performe bien dans son rôle d'amener la recherche vers des zones prometteuses et non encore visitées, augmentant ainsi les chances d'améliorer la meilleure solution trouvée par les composantes de mémoire à court et moyen terme.

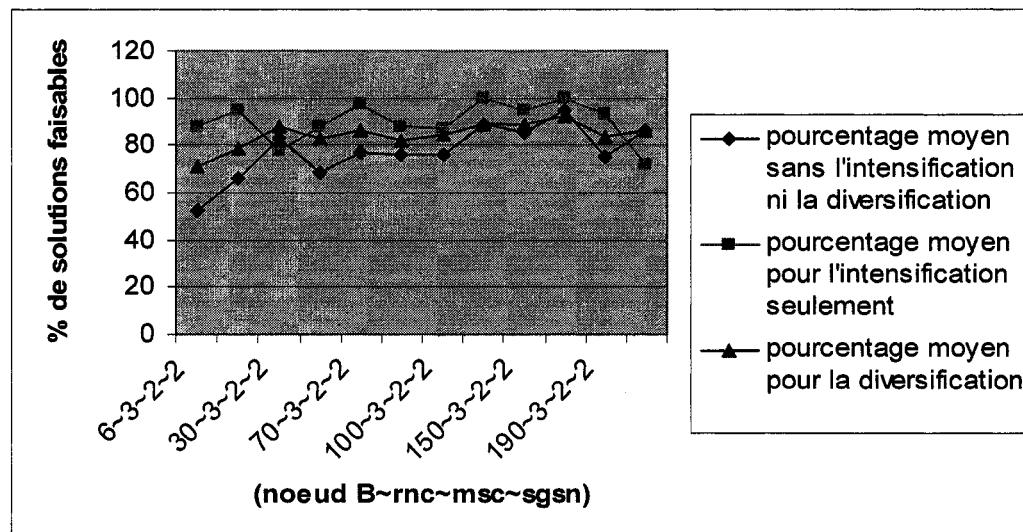


Figure 4.20 Pourcentage de solutions faisables pour chaque mécanisme (série n° 1)

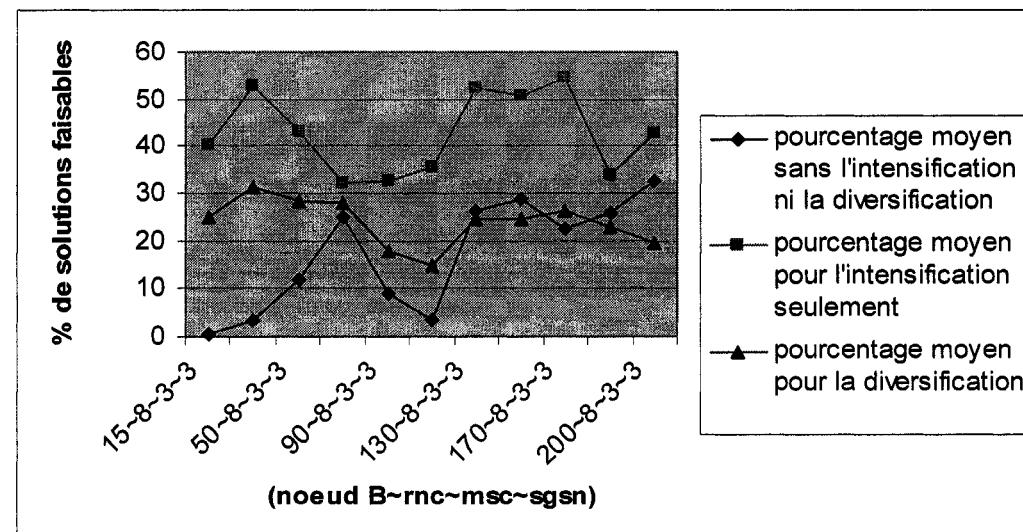


Figure 4.21 Pourcentage de solutions faisables pour chaque mécanisme (série n° 2)

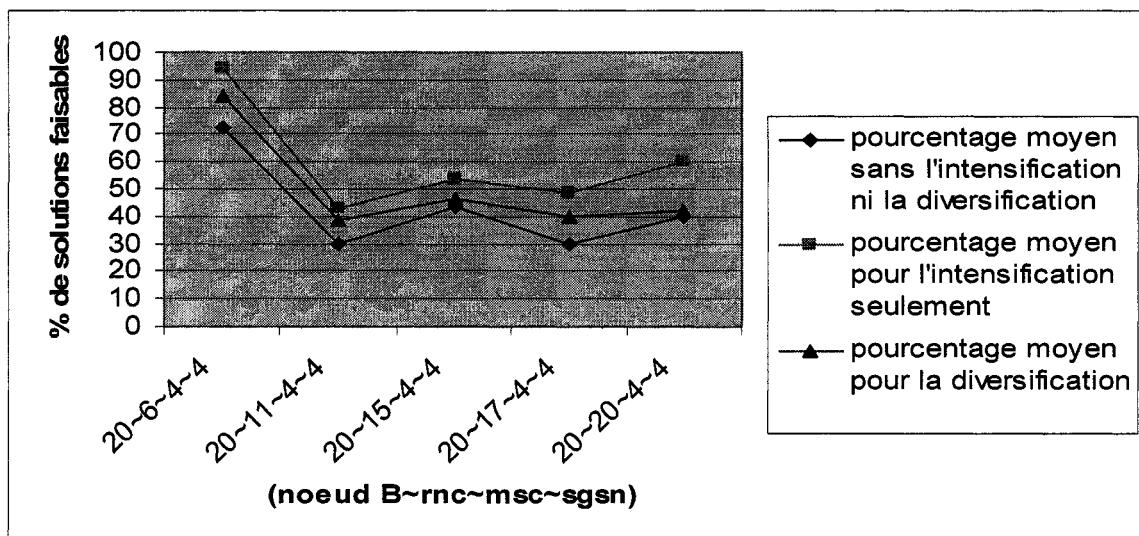


Figure 4.22 Pourcentage de solutions faisables pour chaque mécanisme (série n° 3)

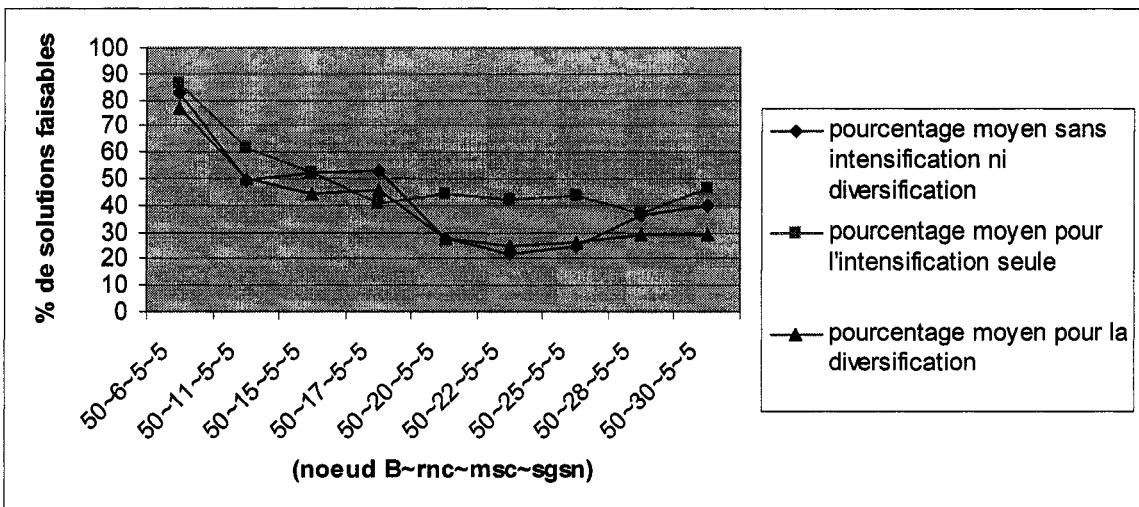


Figure 4.23 Pourcentage de solutions faisables pour chaque mécanisme (série n° 4)

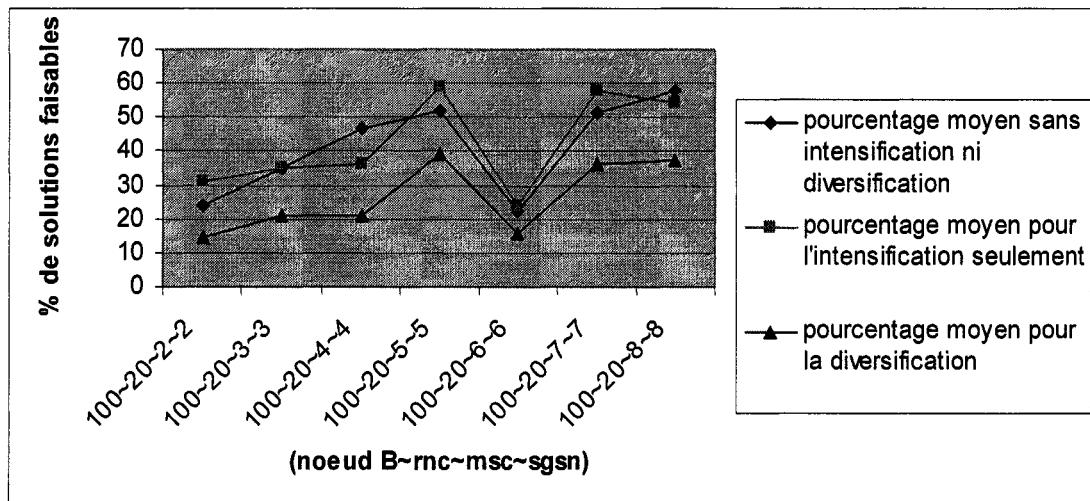


Figure 4.24 Pourcentage de solutions faisables pour chaque mécanisme (série n° 5)

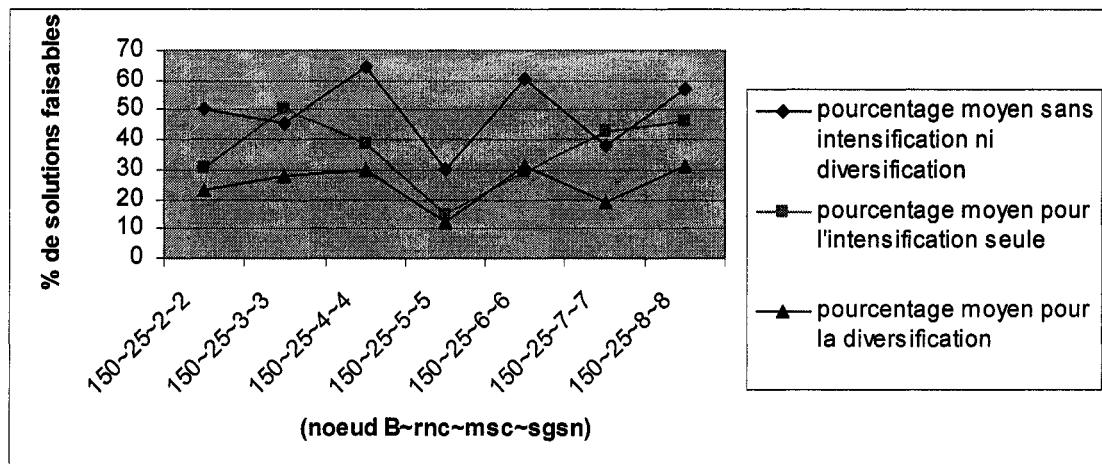


Figure 4.25 Pourcentage de solutions faisables pour chaque mécanisme (série n° 6)

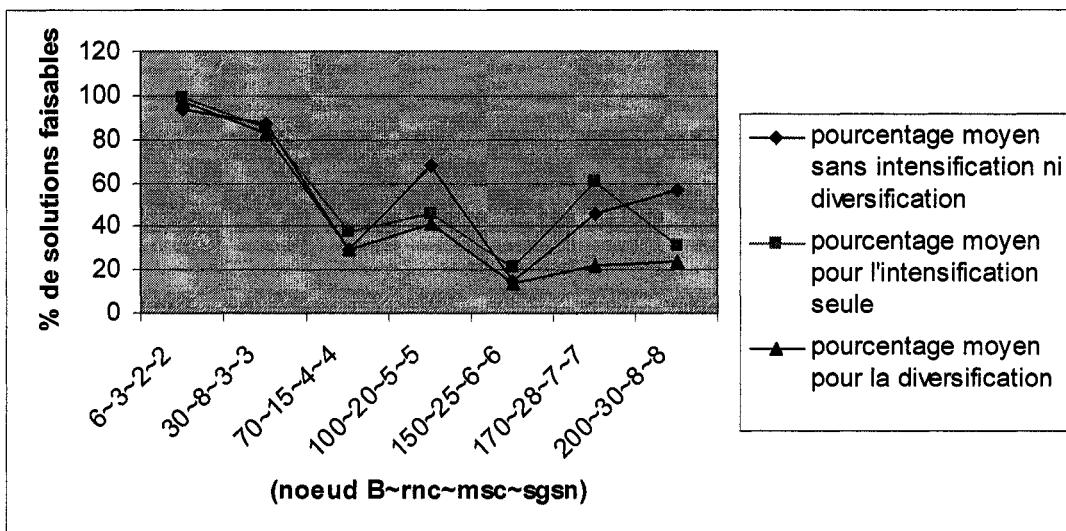


Figure 4.26 Pourcentage de solutions faisables pour chaque mécanisme (série n° 7)

Entre la Figure 4.20 et la Figure 4.21, pour une même variation du nombre de Nœuds B, on constate en général, une diminution des pourcentages de solutions faisables pour les trois mécanismes de mémoire, lorsque le nombre de RNC et de commutateurs augmente. En effet les pourcentages se situent entre 50 et 100% dans le cas où les nombres de RNC et de commutateurs sont plus petits (Figure 4.20) et lorsqu'on augmente ces nombres, les pourcentages chutent considérablement pour se retrouver entre 5 et 55% (Figure 4.21). Ceci s'explique par le fait qu'avec un nombre de RNC et de commutateurs plus grand, on a plus de mouvements possibles et du coup lors de l'exploration de l'espace des solutions, chaque composante de mémoire va générer beaucoup de solutions dont un grand nombre ne seront pas faisables. Cela contribue ainsi à diminuer les pourcentages de solutions faisables pour chaque composante de mémoire. Les figures 4.22 et 4.23 illustrent bien la courbe de décroissance du pourcentage de solutions faisables trouvées par chaque mécanisme en fonction de l'augmentation du nombre de RNC. Pour un nombre élevé de Nœuds B et de RNC (voir les figures 4.24, 4.25 et 4.26), les pourcentages de solutions faisables obtenues en général par la composante de

mémoire à long terme sont moins élevés que ceux des deux autres composantes de mémoire. Cela s'explique par le fait que, si les solutions de redémarrage basées sur les statistiques des solutions déjà explorées sont loin d'être faisables, le nombre élevé de Nœuds B et de RNC fait que la méthode doit explorer plusieurs cas possibles de mouvements (donc de solutions) avant d'arriver à améliorer la solution de redémarrage. Cela contribue à diminuer le ratio entre le nombre de solutions faisables et le nombre total de solutions explorées.

Nous constatons également, qu'en général, la composante de mémoire à moyen terme se révèle plus efficace que celle de mémoire à long terme pour la génération de solutions faisables. Ceci s'explique par le fait que la composante de mémoire à moyen terme applique des mouvements ayant pour but explicite le rétablissement de la faisabilité des solutions, alors que la composante de mémoire à long terme essaie d'améliorer globalement la solution sans chercher spécifiquement à rétablir sa faisabilité.

Globalement, la performance de chaque mécanisme de mémoire ainsi que la RT dans son ensemble donnent des résultats qui répondent parfaitement à nos attentes. De plus, une exécution du programme requiert un temps raisonnable et digne d'une heuristique. Les résultats des temps d'exécution sont illustrés à l'annexe B.

4.3.5 Comparaison des résultats avec une borne inférieure

Pour estimer la qualité des solutions obtenues, nous allons définir pour chaque problème, une borne inférieure par rapport à laquelle nous allons calculer l'écart avec la solution obtenue par l'application de la RT.

Définition de la borne inférieure

Soit un réseau de n Nœuds B (cellules) à affecter à r contrôleurs RNC et ces r RNC à affecter à m commutateurs MSC et à s commutateurs SGSN. Les variables définies dans la Section 3.2.2 nous ont conduit à formuler le problème d'affectation des cellules sous la forme d'une minimisation de la fonction de coût suivante :

$$\begin{aligned}
f = & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in K} c'_{jk} x'_{jk} + \sum_{j \in J} \sum_{l \in L} c''_{jl} x''_{jl} + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{msc} (1 - y_{ii'}) (1 - Y_{jj'}^{msc}) \\
& + \sum_{i \in I} \sum_{i' \in I} \sum_{j \in J} \sum_{j' \in J} h_{ii'}^{sgsn} (1 - y_{ii'}) (1 - Y_{jj'}^{sgsn})
\end{aligned}$$

sujet aux contraintes formulées sur le Tableau 3.1.

Les trois premiers termes de la fonction de coût constituent la sommation des coûts de liaison respectivement des Nœuds B aux RNC, des RNC aux MSC et des MSC aux SGSN. Tandis que les deux derniers termes constituent respectivement la sommation des coûts de relève entre les Nœuds B impliquant des MSC et ceux impliquant des SGSN. Ainsi, l'expression de la borne inférieure du problème, notée LB , peut être divisée en deux termes : le premier terme, appelé $LB1$, va contenir l'évaluation totale des coûts de liaisons et le deuxième terme, appelé $LB2$, contiendra l'évaluation totale des coûts de relève. Pour $LB1$, nous allons sommer les valeurs minimales des coûts des liaisons entre les Nœuds B et les RNC et des liaisons entre les RNC et les MSC et SGSN. Ainsi, par exemple pour un Nœud B i nous allons prendre la valeur minimale parmi les coûts de liaison entre ce Nœud B i et tous les RNC du réseau. Le même principe sera appliqué pour les coûts de liaison des RNC aux MSC et aux SGSN. On obtient donc l'expression suivante de $LB1$:

$$LB1 = \sum_{i \in I} \min_j(c_{ij}) + \sum_{j \in J} \min_k(c'_{jk}) + \sum_{j \in J} \min_l(c''_{jl}) \quad (4.1)$$

Pour l'expression de $LB2$, nous pouvons supposer sans perte de généralité qu'un seul RNC ne peut prendre en charge tous les Nœuds B. On a donc au moins une bipartition de l'ensemble des Nœuds B. Dans le cas d'une bipartition (P, Q) d'un ensemble de n Nœuds B, où P et Q représentent le nombre de Nœuds B dans chaque partition, le nombre total de coûts de relève à prendre en compte est $C_t = (2PQ)_{msc} +$

$(2PQ)_{sgsn}$. L'expression de C_t contient une double sommation de la même expression mais celle-ci s'applique sur deux ensembles différents, le premier est l'ensemble des coûts de relève entre Nœuds B impliquant des MSC et le deuxième est l'ensemble des coûts de relève entre Nœuds B impliquant des SGSN. Le facteur 2 dans l'expression de C_t est dû au fait qu'entre deux Nœuds B i et j , la relève est comptabilisée dans les deux sens ($i \rightarrow j$ et $j \rightarrow i$) et que le coût de relève est le même dans chaque sens. Ainsi, le nombre minimal de relèves à considérer pour une bipartition est obtenu en résolvant le problème :

$$\min (2PQ)_{msc} + (2PQ)_{sgsn}$$

$$\text{sujet à : } P + Q = n, \quad P \geq 1, \quad Q \geq 1$$

Le problème ci-dessus a pour solution les bipartitions $(1, n-1)$ et $(n-1, 1)$. Ces solutions sont celles qui, de toutes les partitions possibles, engendrent le moins de relèves. Ainsi, une borne inférieure pour le nombre de relèves sera $2(n-1)$ relèves.

Soit h_T^{msc} la partie triangulaire supérieure de la matrice $(H^{msc} + H^{Tmsc})$, où H^{msc} , désigne la matrice des coûts de relève impliquant des MSC et H^{Tmsc} , sa transposée. La sommation de la matrice H^{msc} avec sa transposée s'explique par le fait que les coûts de relève entre deux Nœuds B i et j sont comptabilisés dans les deux sens $i \rightarrow j$ et $j \rightarrow i$.

Soit h_T^{sgsn} la partie triangulaire supérieure de la matrice $(H^{sgsn} + H^{Tsgsn})$, où H^{sgsn} désigne la matrice des coûts de relève impliquant des SGSN et H^{Tsgsn} sa transposée.

L'expression de $LB2$ doit contenir au moins $(n-1)$ coûts de relève de la matrice h_T^{msc} et $(n-1)$ coûts de relève de la matrice h_T^{sgsn} . On a donc l'expression suivante pour $LB2$:

$$LB2 = \sum_{p=1}^{n-1} \sum_{q=p+1}^n I_{N^{msc}} \{h_T^{msc}(p,q)\} \bullet h_T^{msc}(p,q) + \sum_{p=1}^{n-1} \sum_{q=p+1}^n I_{N^{sgsn}} \{h_T^{sgsn}(p,q)\} \bullet h_T^{sgsn}(p,q) \quad (4.2)$$

où N_{-}^{msc} désigne l'ensemble des $(n-1)$ valeurs minimales de la matrice triangulaire h_T^{msc} , N_{-}^{sgsn} désigne l'ensemble des $(n-1)$ valeurs minimales de la matrice triangulaire h_T^{sgsn} , $I_{N_{-}^{msc}}$ et $I_{N_{-}^{sgsn}}$ désignent respectivement la fonction indicatrice des ensembles N_{-}^{msc} et N_{-}^{sgsn} avec :

$$I_{N_{-}^{msc}}\{x\} = \begin{cases} 1 & \text{si } x \in N_{-}^{msc} \\ 0 & \text{sinon} \end{cases} \quad (4.3)$$

$$I_{N_{-}^{sgsn}}\{x\} = \begin{cases} 1 & \text{si } x \in N_{-}^{sgsn} \\ 0 & \text{sinon} \end{cases} \quad (4.4)$$

$h_T^{msc}(p, q)$ et $h_T^{sgsn}(p, q)$ désignent respectivement l'élément à la ligne p et à la colonne q des matrices h_T^{msc} et h_T^{sgsn} . En combinant (4.1) et (4.2), nous pouvons estimer la borne inférieure LB par :

$$LB = LB1 + LB2 \quad (4.5)$$

Comparaison avec la borne inférieure

Afin de comparer nos résultats avec l'estimation de la borne inférieure du problème, nous avons implémenté la relation (4.5) dans un programme *matlab*. Ainsi, pour chaque série du Tableau 4.2, le programme se sert des matrices de coûts de liaison entre les Nœuds B et les RNC, entre les RNC et les commutateurs (MSC/SGSN) et des matrices de coûts de relève entre les Nœuds B impliquant des MSC et ceux impliquant des SGSN pour estimer la borne inférieure de chaque taille de réseau. Ensuite, il compare la borne inférieure avec le résultat produit par la RT en calculant le pourcentage de l'écart moyen entre les deux valeurs. Les tableaux 4.3 à 4.9 illustrent les résultats de cette comparaison pour chaque série de test du Tableau 4.2.

Tableau 4.3 Écart par rapport à la borne inférieure pour la série n° 1

#Nœuds B	6	15	30	50	70	90	100	130	150	170	190	200
Écart (%)	4.06	0.97	0.14	1.43	1.53	0.47	1.04	0.95	1.13	0.88	0.64	0.37

Tableau 4.4 Écart par rapport à la borne inférieure pour la série n° 2

#Nœuds B	15	30	50	70	90	100	130	150	170	190	200
Écart (%)	20.6	12.5	4.6	6.02	4.56	3.57	1.90	1.46	5.67	1.81	1.50

Tableau 4.5 Écart par rapport à la borne inférieure pour la série n° 3

#RNC	6	11	15	17	20
Écart (%)	0.94	14.50	15.21	15.12	25.64

Tableau 4.6 Écart par rapport à la borne inférieure pour la série n° 4

#RNC	6	11	15	17	20	22	25	28	30
Écart (%)	1.96	6.79	12.57	16.89	14.83	24.64	21.51	18.39	19.36

Tableau 4.7 Écart par rapport à la borne inférieure pour la série n° 5

#MSC/SGSN	2	3	4	5	6	7	8
Écart (%)	9.86	9.02	16.91	7.43	17.61	6.65	9.50

Tableau 4.8 Écart par rapport à la borne inférieure pour la série n° 6

#MSC/SGSN	2	3	4	5	6	7	8
Écart (%)	3.50	8.64	8.76	12.55	15.69	11.49	7.09

Tableau 4.9 Écart par rapport à la borne inférieure pour la série n° 7

Réseau	6~3~2~2	30~8~3~3	70~15~4~4	100~20~5~5	150~25~6~6	170~28~7~7	200~30~8~8
Écart (%)	7.66	2.20	8.13	11.36	13.77	12.33	7.25

Les résultats de comparaison montrent qu'en général, les solutions obtenues par la RT sont assez proches de la borne inférieure avec des écarts ne dépassant pas 30% et dans certains cas, les écarts tombent en dessous de 1%. Entre autre, les tableaux 4.3 et 4.4 montrent que, pour les séries 1 et 2 où l'on fait varier seulement le nombre de Nœuds B avec un nombre faible de RNC, de MSC et de SGSN, les écarts sont très faibles par rapport aux autres séries. Ceci s'explique par le fait qu'à cause du nombre faible de RNC et de commutateurs (MSC/SGSN), notre méthode arrive à explorer la plupart sinon toutes les possibilités d'affectation de RNC vers les MSC et les SGSN. Ce qui donne des solutions proches de la borne inférieure.

CHAPITRE 5

CONCLUSION

Dans l'ensemble du processus de planification des réseaux mobiles de troisième génération, le problème d'affectation des cellules aux commutateurs occupe une place non moins importante. Dans ce mémoire, nous nous étions fixés comme objectif de tenter de résoudre ce problème en partant du développement d'un modèle mathématique pour formuler le problème jusqu'à l'implémentation d'une méthode pour résoudre le problème. Dans ce dernier chapitre, nous allons présenter tout d'abord une synthèse générale des travaux accomplis pour ensuite discuter des limitations de notre modèle ainsi que de l'adaptation faite de la RT. Nous terminerons par des suggestions de travaux de recherches futures.

5.1 Synthèse des travaux

Notre objectif principal dans ce mémoire a été d'étudier toute la problématique entourant l'affectation des cellules (Nœuds B) aux commutateurs (MSC/SGSN) dans les réseaux mobiles de troisième génération. Pour ce faire nous avons organisé notre recherche en trois volets : le premier consistant à nous intéresser aux modèles existants dans la littérature pour résoudre le problème dans le cadre des réseaux 2G, ceci pour voir comment s'en servir ou s'en inspirer le plus possible pour formuler notre modèle pour le 3G. Également dans ce premier volet, nous nous sommes intéressés aux propositions actuelles d'architecture pour les réseaux 3G afin de voir comment trouver un terrain commun de similitude entre les architectures en regard aux équipements qu'il faut tenir compte dans notre modèle mathématique et lesquels il faut exclure. Dans le deuxième volet de notre recherche, nous avons cherché à dégager un modèle mathématique formulant l'affectation des cellules aux commutateurs en tenant compte de toutes les contraintes liées à l'architecture du réseau 3G et des caractéristiques des différents

équipements. Enfin, dans le dernier volet, dans le but d'implémenter et de simuler notre modèle, nous avons effectué l'adaptation de l'heuristique Recherche Taboue.

En parcourant les travaux réalisés jusque là pour résoudre le problème d'affectation des cellules aux commutateurs dans les réseaux mobiles 2G, nous nous sommes rendu compte que le modèle existant de formulation du problème permet d'affecter un ensemble n de cellules à un ensemble m de commutateurs MSC en tenant compte des contraintes de limitation de capacité au niveau des MSC. Dans notre cas, après avoir choisi d'étudier le problème pour les réseaux UMTS, notre objectif a été de trouver un modèle de formulation permettant d'affecter un ensemble n de Nœuds B (cellules) à un ensemble r de contrôleurs RNC et d'affecter ces r RNC en même temps à m commutateurs MSC et à s commutateurs SGSN. Pour mieux cerner le problème, nous avons donc divisé l'affectation globale des Nœuds B aux commutateurs en deux niveaux d'affectation : le « *niveau 1* » qui est l'affectation des Nœuds B aux RNC et le « *niveau 2* » qui est l'affectation des RNC aux MSC et aux SGSN. Nous sommes arrivé à la conclusion que la seule similitude entre notre problème et la formulation proposée pour les réseaux 2G est au « *niveau 1* ».

Ainsi, pour résoudre notre problème, nous avons été obligés de fournir un modèle mathématique qui permet de formuler en même temps les deux niveaux d'affectation tout en tenant en compte des contraintes de capacité non seulement au niveau des RNC mais aussi au niveau des commutateurs MSC et SGSN. Une fois le problème formulé, compte tenu de sa complexité, nous nous sommes vite rendu compte qu'il serait laborieux de vouloir le résoudre avec des méthodes exactes, surtout pour des instanciations de taille élevée. Nous avons alors eu recours à une méta-heuristique en particulier la Recherche Taboue jugée efficace pour la résolution de tels problèmes complexes. C'est une méthode qui utilise des techniques d'exploration sophistiquées pour éviter les pièges des minima locaux.

L'implémentation de la méthode a consisté à définir une série de mouvements pour ses mécanismes de mémoire à court et moyen termes. Les mécanismes de mémoire

à court et moyen termes sont appliqués dans l'ordre suivant : la mémoire à court terme puis la mémoire à moyen terme. Un troisième mécanisme appelé mémoire à long terme permet de redémarrer le processus dans le même ordre et ce pour diversifier la recherche de solutions. Les mouvements visent soit à améliorer le coût de la solution courante ou alors à rétablir sa faisabilité en effectuant par exemple une réaffectation des Nœuds B ou une réaffectation des RNC. Puisque nous avons une multitude de choix pour chaque catégorie de mouvements, nous avons défini une structure de gains permettant de discriminer les mouvements et de choisir ainsi ceux fournissant un meilleur gain sur le coût de la solution courante.

Afin d'évaluer les performances de l'adaptation de la RT, nous avons ciblé des paramètres clés dans chaque mécanisme de mémoire pour lesquels nous avons simulé l'effet sur la qualité des solutions obtenues en faisant varier leurs valeurs numériques. Ainsi, nous avons étudié, entre autres, l'influence de la taille de la liste taboue, l'effet du poids de la sanction de la violation des contraintes liées au trafic de voix par rapport à celles liées au trafic de données, l'effet de la taille de la zone d'intensification utilisée par le mécanisme de mémoire à moyen terme et l'effet du nombre de cycles de diversification effectués par le mécanisme de mémoire à long terme. Également, pour estimer l'efficacité de notre méthode en général à trouver, pour un problème d'affectation de cellules donné, une solution faisable à moindre coût, nous avons défini un modèle d'estimation de la borne inférieure pour chaque problème. L'estimation de cette borne nous a permis de juger de la qualité des solutions obtenues en faisant un calcul d'écart. Ainsi, de manière générale, les résultats obtenus sont convaincants et montrent que nos solutions sont assez proches de la borne.

5.2 Limitations des travaux

D'abord au niveau de la formulation mathématique du problème, nous avons eu recours à des simplifications sur l'architecture du réseau UMTS utilisé. Nous nous sommes concentrés sur des équipements jugés essentiels et représentatifs du réseau, en

écartant d'autres équipements tels les GGSN, les MGW. Il se peut que certains de ces équipements écartés jouent un rôle essentiel qui leur mériterait une considération dans le traitement et l'acheminement du trafic dans le réseau. Dans ce cas, leurs contraintes devraient être intégrées dans la formulation du problème.

Au niveau de l'implémentation et de la simulation, en dépit des résultats satisfaisants obtenus, la performance de notre approche dépend fortement du choix des paramètres de la méthode. Ceux-ci ne sont pas faciles à choisir, et les valeurs utilisées dans le programme donnent globalement de bons résultats, mais peuvent ne pas être forcément les meilleures pour un type de problème donné. Également, la méthode ne garantit pas toujours la faisabilité de la solution finale. En effet, dans quelques cas, nous obtenons une solution finale non faisable qui, malgré les pénalités appliquées demeure moins coûteuse que les solutions faisables trouvées. Dans ces cas là, nous avons opté pour la considération de cette solution non faisable là, quitte à ce que le concepteur modifie légèrement le problème pour le rendre faisable si cela n'engendre pas de trop grands coûts. Notons aussi que les données d'entrée utilisées pour effectuer nos simulations ont été générées par un programme donc ne sont pas réelles.

5.3 Indications de recherche future

Les pistes d'amélioration des travaux réalisés dans ce mémoire sont nombreuses. Tout d'abord, la formulation mathématique que nous avons proposée ne considère qu'une domiciliation simple pour les Nœuds B et les RNC. Ainsi, il serait intéressant d'étudier le problème pour une domiciliation double pour les deux équipements (Nœuds B et RNC) ou alors une domiciliation mixe (double pour les Nœuds B et simple pour les RNC ou l'inverse). Ensuite, pour la résolution du problème, il faudrait adapter d'autres méthodes comme les algorithmes memétiques, la programmation par contrainte ou l'algorithme génétique pour pouvoir comparer nos résultats avec ceux qu'on obtiendrait avec ces méthodes. Enfin, pour l'évaluation de la qualité des solutions finales, les futures

recherches pourraient essayer de déterminer de meilleures estimations de l'optimum global en utilisant des modèles mathématiques plus complexes.

BIBLIOGRAPHIE

- [1] E. Aarts, J.K. Lenstra, Local search in combinatorial optimization, Chichester, UK: John Wiley & Sons, Ltd, 1997.
- [2] G.E. Amoussou, G. Pesant, S. Pierre, « Affectation de cellules à des commutateurs par programmation par contraintes », *IEEE CCECE'2001*, May 13-16, 2001, Toronto, Canada.
- [3] M. André, *Utilisation de la programmation par contraintes et de la recherche taboue pour le problème d'affectation de cellules aux commutateurs*, Mémoire de Maîtrise, École Polytechnique de Montréal, Janvier 2002.
- [4] R. Beaubrun, *Gestion de la Mobilité et Ingénierie de trafic en conception de réseaux mobiles de troisième génération*, Thèse de doctorat, École Polytechnique de Montréal, Septembre 2002.
- [5] D.R. Din, S.S. Tseng, « A genetic algorithm for solving dual-homing cell assignment problem of the two-level wireless ATM network, *Computer Communications*, 2002, pp. 1536-1547.
- [6] C.M. Fuduccia, R.M. Mattheyses, « A Linear-time Heuristic for Improving Network partition », *Proceedings 19th Design Automat. Conf.*, 1982, pp. 175-181.
- [7] M.R. Garey, D.S. Johnson, *Computers and intractability*, San Francisco, CA, Freeman, 1979.
- [8] F. Glover « Tabu Search - Part I », *ORSA Journal on Computing*, vol. 1, No. 3, 1989, pp. 190-206.

- [9] F. Glover « Tabu Search - Part II », *ORSA Journal on Computing*, vol. 2, 1990b, pp. 4-32.
- [10] F. Glover, E. Taillard, D. Werra, « A user's guide to tabu search », *Annals of Operations Research*, vol. 41, No. 3, 1993, pp. 3-28.
- [11] F. Glover « Tabu Search – A Tutorial », *INTERFACES*, vol. 20, No. 4, 1990a, pp. 74-94.
- [12] C. Hédible, S. Pierre, « A genetic algorithm for Assigning Cells to Switches in Personal Communication Networks », *IEEE Transactions on Systems, Man, and Cybernetics*, 2001.
- [13] C. Hédible, S. Pierre, « Algorithme génétique pour l'affectation des cellules à des commutateurs dans les réseaux de télécommunications personnelles », *IEEE CCECE'2000*, May 6-10, 2000, Halifax, Canada, pp. 1077-1081.
- [14] E. Horowitz, S. Sahni, *Fundamentals of computer algorithms*, Computer science Press, Inc., 1990.
- [15] F. Houéto, S. Pierre, « Affectation heuristique de cellules à des commutateurs dans les réseaux cellulaires mobiles », *Annales des télécommunications*, Vol. 56, Nos. 3-4, 2001, pp. 184-197.
- [16] F. Houéto, *Affectation des Cellules à des Commutateurs dans les Réseaux de Communications Personnelles*, Mémoire de Maîtrise, École Polytechnique de Montréal, Août 1999.

- [17] B.W. Kernighan, S. Lin, « An Efficient Heuristic Procedure for Partitioning Graphs », *The Bell System Technical Journal*, vol. 49, 1970, pp. 291-307.
- [18] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, « Optimization by Simulated Annealing », *Science*, vol. 220, 1983, pp. 671-680.
- [19] A.A. Kuehn, M.J. Hamburger, « A heuristic program for locating warehouses », *Management Science*, vol. 9, 1963, pp. 643-666.
- [20] A. Merchant, B. Sengupta, « Assignment of cells to Switches in PCS Networks », *IEEE/ACM transactions on Networking*, vol. 3, No. 5, 1985, pp. 521-526.
- [21] P. Merz, B. Freisleben, « A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem » in *Proc. 1999 Int. Congress of Evolutionary Computation (CEC'99)*, 1999, pp.2063–2070.
- [22] G. Pesant, M. Gendreau, « A view of Local search in constraint programming », *Constraint Programming*, 1997, pp. 353-366.
- [23] A. Quintero, S. Pierre, « A Memetic Algorithm for Assigning Cells to Switches in Cellular Mobile Networks », *IEEE Communications Letters*, Vol. 6, No. 11, November 2002, pp. 484-486.
- [24] J. Sanchez, M. Thioune, UMTS: services, architectures et WCDMA, 2001
- [25] L.A. Sanchis, « Multiple-Way Network Partitioning », *IEEE transactions on Computers*, vol. 38, No 1, 1989, pp. 62-81.

- [26] D. Skorin-Kapov, J. Skorin-Kapov, « On tabu search for the location of interacting hub facilities », *European Journal of Operational Research*, vol. 73, 1994, pp. 502-509.
- [27] C. Smith, D. Collins, 3G wireless networks, 2002
- [28] J. Sohn, S. Park, « Efficient solution procedure and reduced size formulations for p-hub location problem », *European Journal of Operational Research*, vol. 108, 1998, pp. 118-126.
- [29] M. Wallace, Survey: Practical Applications of Constraint Programming, William Penney Laboratory, Imperial College, London, September 1995.
- [30] Y. Wu, S. Pierre, « Optimization of Access Network Design in 3G Networks », *IEEE CCECE'2003 – CCGEI'2003*, May 2003, Montréal, Canada, pp. 781-784.
- [31] A. Quintero, S. Pierre, « Sequential and multi-population memetic algorithms for assigning cells to switches in mobile networks », *Computer Networks 43*, 2003, pp. 247-261.

ANNEXE A

COMPOSITION DES SÉRIES DE TESTS

Tableau A.1 Série de tests n° 1

Nœuds B	RNC	MSC	SGSN
6	3	2	2
15	3	2	2
30	3	2	2
50	3	2	2
70	3	2	2
90	3	2	2
100	3	2	2
130	3	2	2
150	3	2	2
170	3	2	2
190	3	2	2
200	3	2	2

Tableau A.2 Série de tests n° 2

Nœuds B	RNC	MSC	SGSN
15	8	3	3
30	8	3	3
50	8	3	3
70	8	3	3
90	8	3	3
100	8	3	3
130	8	3	3
150	8	3	3
170	8	3	3
190	8	3	3
200	8	3	3

Tableau A.3 Série de tests n° 3

Nœuds B	RNC	MSC	SGSN
20	6	4	4
20	11	4	4
20	15	4	4
20	17	4	4
20	20	4	4

Tableau A.4 Série de tests n° 4

Nœuds B	RNC	MSC	SGSN
50	6	5	5
50	11	5	5
50	15	5	5
50	17	5	5
50	20	5	5
50	22	5	5
50	25	5	5
50	28	5	5
50	30	5	5

Tableau A.5 Série de tests n° 5

Nœuds B	RNC	MSC	SGSN
100	20	2	2
100	20	3	3
100	20	4	4
100	20	5	5
100	20	6	6
100	20	7	7
100	20	8	8

Tableau A.6 Série de tests n° 6

Nœuds B	RNC	MSC	SGSN
150	25	2	2
150	25	3	3
150	25	4	4
150	25	5	5
150	25	6	6
150	25	7	7
150	25	8	8

Tableau A.7 Série de tests n° 7

Nœuds B	RNC	MSC	SGSN
6	3	2	2
30	8	3	3
70	15	4	4
100	20	5	5
150	25	6	6
170	28	7	7
200	30	8	8

ANNEXE B

GRAPHES DU TEMPS MOYEN D'EXÉCUTION

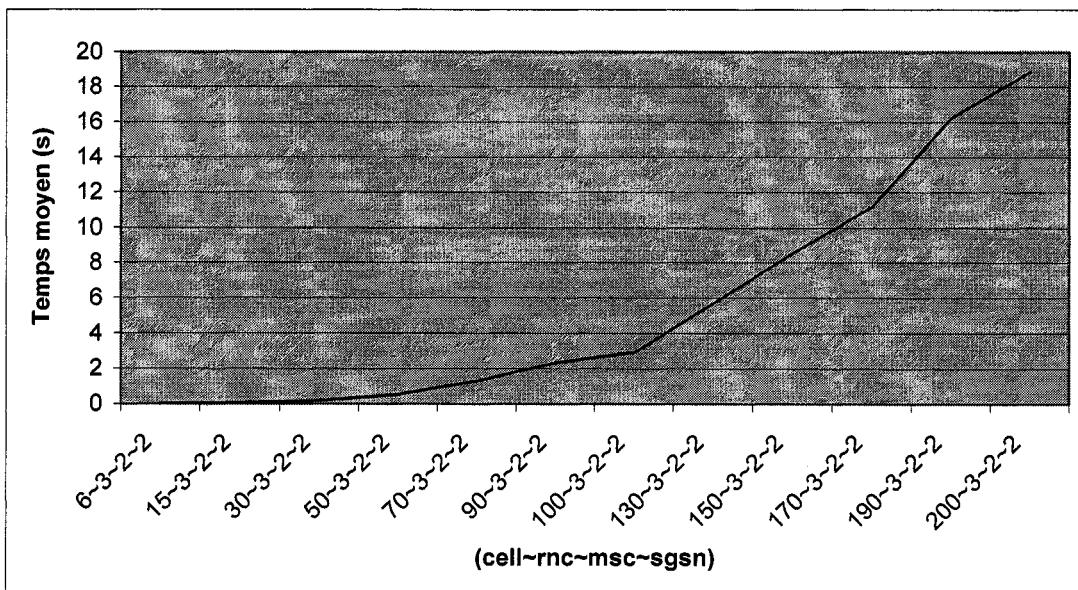


Figure B.1 Temps moyen d'exécution (série 1)

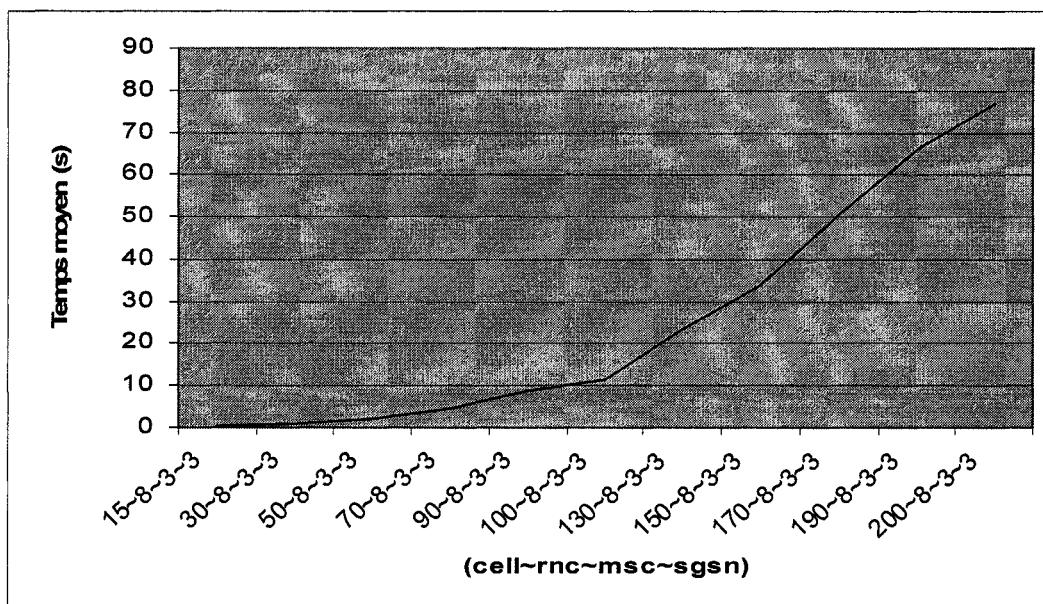


Figure B.2 Temps moyen d'exécution (série 2)

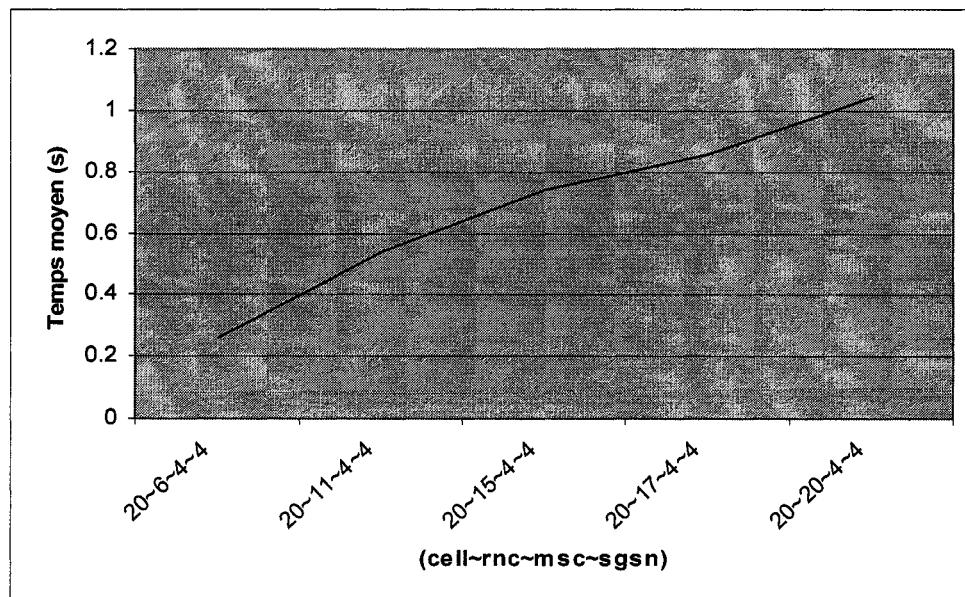


Figure B.3 Temps moyen d'exécution (série 3)

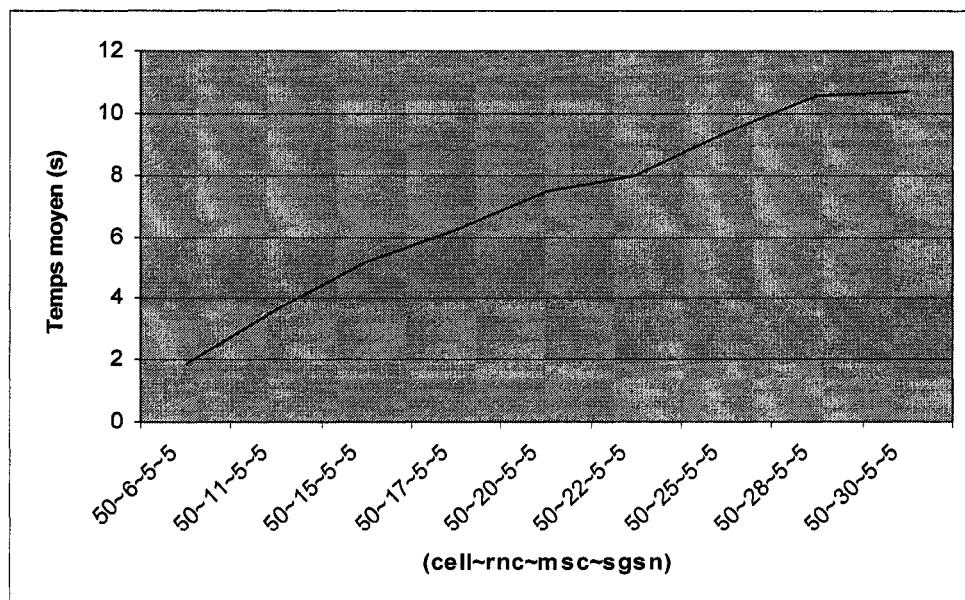


Figure B.4 Temps moyen d'exécution (série 4)

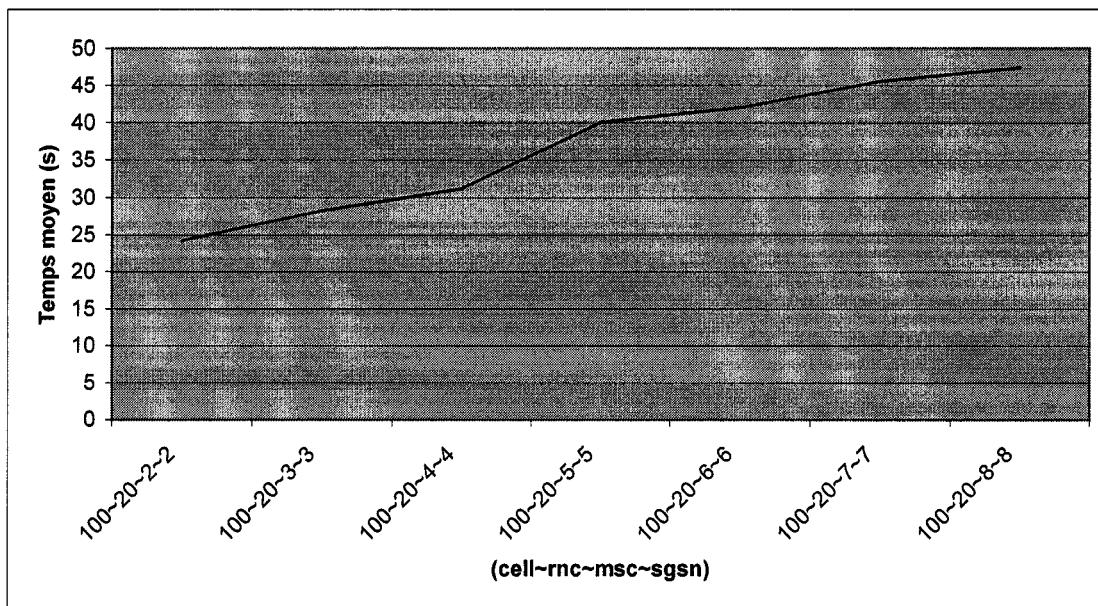


Figure B.5 Temps moyen d'exécution (série 5)

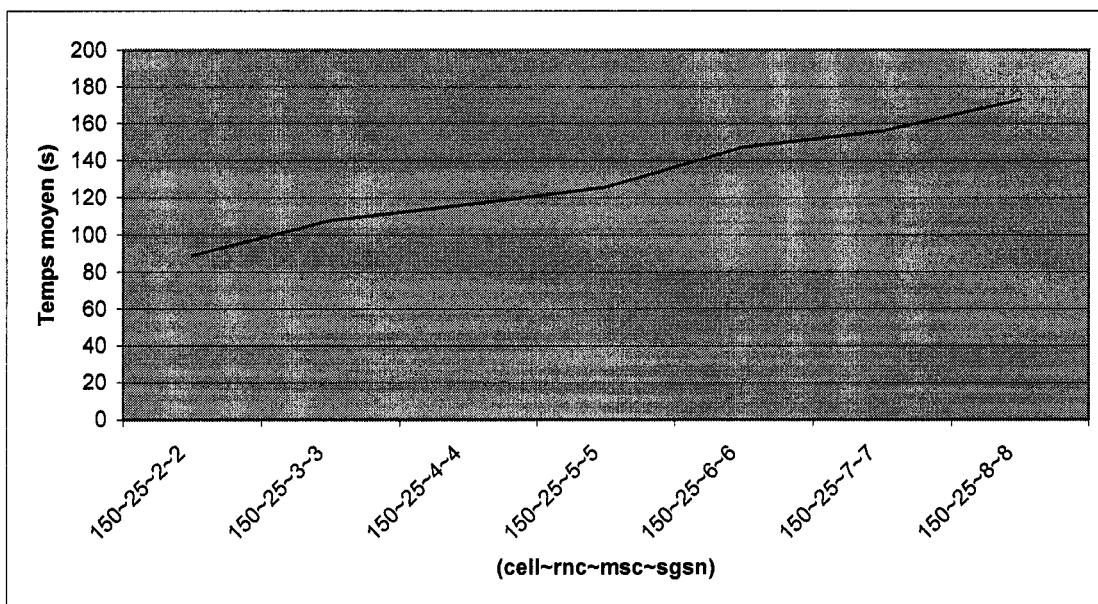


Figure B.6 Temps moyen d'exécution (série 6)

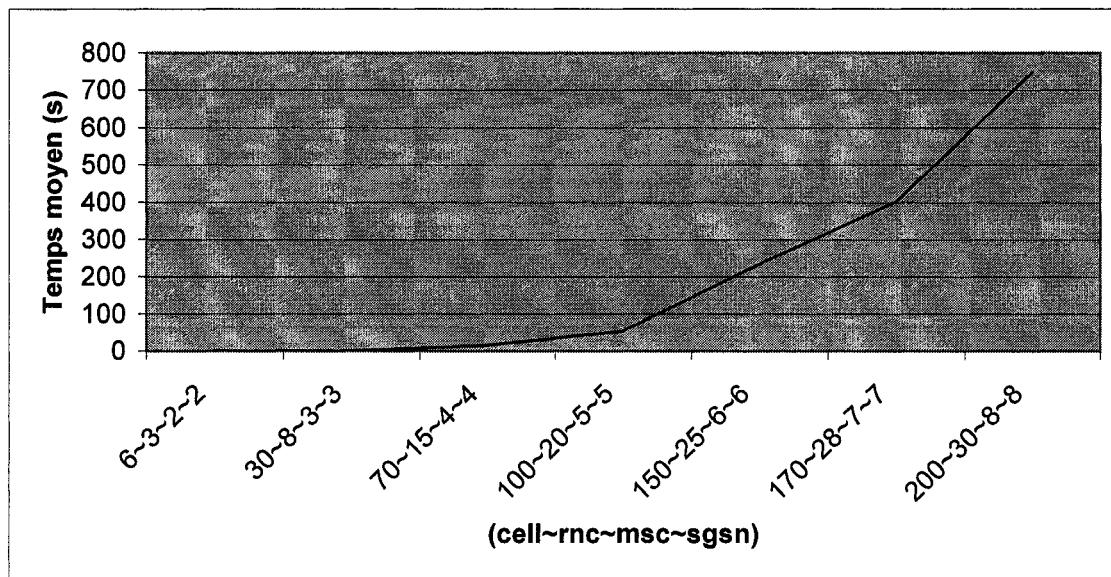


Figure B.7 Temps moyen d'exécution (série 7)

ANNEXE C

FICHIERS TYPIQUES D'ENTRÉE

```
6 3 2 2

12.0000 6.0000 10.3923
10.3923 6.0000 6.0000
6.0000 0.0000 6.0000
6.0000 6.0000 0.0000
0.0000 6.0000 6.0000
6.0000 10.3923 6.0000

6.0000 6.0000
6.0000 0.0000
0.0000 6.0000

10.0000 10.0000
10.0000 0.0000
0.0000 10.0000

0.0000 1.6080 0.3561 0.0000 0.0000 0.0000
0.7129 0.0000 1.1720 2.0477 0.0000 0.0000
0.0174 0.4645 0.0000 0.8115 0.2652 0.0000
0.0000 2.4327 0.0910 0.0000 1.3887 4.2503
0.0000 0.0000 0.2905 0.4776 0.0000 0.8344
0.0000 0.0000 0.0000 3.1311 0.6933 0.0000

0.0000 0.9456 1.1627 0.0000 0.0000 0.0000
1.9662 0.0000 1.5990 0.2816 0.0000 0.0000
0.1783 1.9610 0.0000 0.3453 2.2330 0.0000
0.0000 1.2005 0.1091 0.0000 1.3670 0.2114
0.0000 0.0000 2.1002 1.7080 0.0000 0.3008
0.0000 0.0000 0.0000 0.5924 0.7285 0.0000
```

Figure C.1 Exemple de fichier « .DON »

```

0.8661 8.0681 3.2017 9.6742 1.8516 1.6686
6.7787 2.0321 7.6555 1.4252 1.1055 0.7147
10.4072 10.4072 10.4072
13.0131 13.0131 13.0131
20.6247 20.6247
30.2979 30.2979

```

Figure C.2 Exemple de fichier « .CAP »

```

Réseau de simulation: 6 Noeuds B, 3 RNCs, 2 MSCs, 2 SGSNs
Durée totale de la simulation: 0.611

La meilleure évaluation de la fonction de coût: 48.7846
Respecte contraintes?(1='OUI', 0='NON'): 1
Format de l'affichage de la solution:
NB -> RNC : vertical Noeuds B, horizontal RNC
RNC -> MSC : vertical RNC, horizontal MSC
RNC -> SGSN: vertical RNC, horizontal SGSN

Affectation des Noeuds B aux RNC
1 0 0
1 0 0
0 0 1
1 0 0
0 0 1
0 0 1

Affectation des RNC aux MSC
1 0
0 1
1 0

Affectation des RNC aux SGSN
1 0
0 1
1 0

```

Figure C.3 Exemple de fichier «Result.ras »

```

Reseau de simulation: 6 Noeuds B, 3 RNCs, 2 MSCs, 2 SGSNs

Solution initiale
Respecte les contraintes?(1='OUI', 0='NON') : 0
*****DÉBUT DE LA MÉMOIRE À COURT TERME*****
Iteration: 1
    Réaffectation du NB 2 -> RNC 1
    Réaffectation du RNC 0 -> MSC 1 -> SGSN 1
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 40
L'evaluation courante de la fonction de cout: 40
La solution courante Respecte les contraintes?(1='OUI', 0='NON'): 1

Iteration: 2
    Réaffectation du NB 2 -> RNC 0
    Réaffectation du RNC 0 -> MSC 0 -> SGSN 0
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 40
L'evaluation courante de la fonction de cout: 48.7689
La solution courante Respecte les contraintes?(1='OUI', 0='NON'): 1

Iteration: 3
    Réaffectation du NB 2 -> RNC 2
    Réaffectation du RNC 0 -> MSC 1 -> SGSN 1
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 40
L'evaluation courante de la fonction de cout: 53.9902
La solution courante Respecte les contraintes?(1='OUI', 0='NON'): 1
*****FIN DE LA MÉMOIRE À COURT TERME*****

Apres application de la memoire a court terme
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 40
Respecte les contraintes?(1='OUI', 0='NON'): 1

*****DÉBUT DE L'INTENSIFICATION*****
Niveau: 0
Évaluation de départ de la fonction de coût: 42.7689
Solution de départ respecte les contraintes?(1='OUI', 0='NON'): 1

Iteration: 1
    Permutation des Noeuds B: 2 <-> 4
    Permutation des RNC: 0 <-> 1
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 42.7689
L'evaluation courante de la fonction de cout: 70.7163
La solution courante respecte les contraintes?(1='OUI', 0='NON'): 1

Iteration: 2
    Permutation des Noeuds B: 0 <-> 2
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 42.7689
L'evaluation courante de la fonction de cout: 85.9472
La solution courante respecte les contraintes?(1='OUI', 0='NON'): 1
*****FIN DE L'INTENSIFICATION*****

Apres application de la memoire a moyen terme
La meilleure solution faisable est evaluee a: 40
La meilleure evaluation de la fonction de cout: 40
Respecte les contraintes?(1='OUI', 0='NON'): 1
#####DÉBUT DE LA DIVERSIFICATION#####
Diversification: 0
Solution initiale de la diversification No: 0
Respecte les contraintes?(1='OUI', 0='NON'): 0

```

Figure C.4 Exemple de fichier «TracePrg.tra»