

# Approches heuristiques pour l'affectation de cellules aux commutateurs dans les réseaux mobiles

## 1.0 Introduction

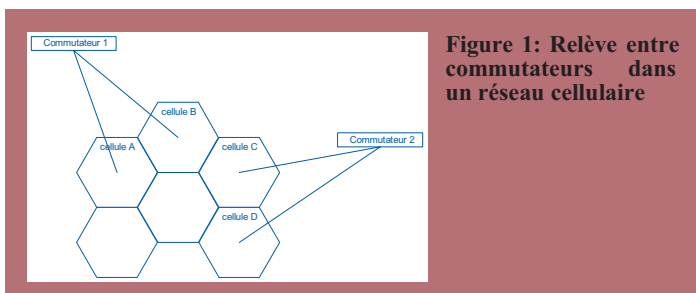
**T**ypiquement, un réseau cellulaire est constitué d'un ensemble de stations de base desservant les cellules et de plusieurs commutateurs connus sous le nom de Mobile Switching Centers (MSC). Dans de tels systèmes, la communication entre les usagers présents à l'intérieur d'une même couverture géographique se fait grâce aux liaisons câblées ou micro-ondes entre les stations de base et les MSC. Ainsi, les MSC gèrent la communication entre les usagers d'un même réseau et permettent également l'accès au réseau téléphonique commuté public. Les commutateurs ont la possibilité de se transférer la prise en charge d'un usager passant d'une cellule à une autre. Lorsque celui-ci passe d'une cellule à une autre, une mise à jour est effectuée au niveau des commutateurs concernés. Cette opération est connue sous le nom de relève (handoff). Si la relève a lieu entre deux cellules reliées au même commutateur, on parle de relève simple, car les mises à jour à effectuer sont peu nombreuses. En revanche, si cette relève s'effectue entre deux cellules reliées à des commutateurs distincts, on parle de relève complexe, les mises à jour impliquant plus de ressources dans ce cas.

Lorsqu'un usager est en mouvement dans une cellule, le signal analogique émis est aussitôt pris en charge par la cellule la plus proche. Il existe un seuil de sensibilité (ou seuil de filtrage) au-delà duquel le signal émis par l'utilisateur est suffisamment puissant pour être pris en compte par la cellule.

Comme l'illustre la Figure 1, les cellules A et B relèvent d'un même commutateur 1, alors que les cellules C et D dépendent du commutateur 2. Un utilisateur présent en C pourrait émettre un signal suffisamment puissant pour être perçu simultanément par les cellules B et C compte tenu de la proximité des cellules, ce qui pose problème. Il faudrait se procurer un réseau de signalisation capable de juger quelle cellule reçoit le signal avec le plus de clarté et de déterminer ainsi le commutateur qui sera responsable de la communication. Dans le cas où l'utilisateur quitte la cellule C et se retrouve dans la cellule D, on est en présence d'une relève simple. Seul le commutateur 2 est sollicité au cours de cette opération. En revanche, si l'utilisateur passe de la cellule C à la cellule B, il y a une relève complexe. Les commutateurs 1 et 2 s'échangent l'information relative à l'utilisateur et une mise à jour de la base de données du réseau est effectuée. Le coût d'une relève complexe est donc beaucoup plus élevé que celui d'une relève simple.

L'affectation des cellules aux commutateurs consiste essentiellement à trouver la configuration des liaisons commutateur-cellule qui minimise le coût total du réseau, en tenant compte d'un certain nombre de contraintes dont la fréquence des relèves entre cellules et la capacité de chaque commutateur. Parmi les facteurs qui doivent être pris en compte, on trouve la configuration du réseau, la capacité des commutateurs et le trafic acheminé par l'intermédiaire du réseau.

Considérons l'aspect de la configuration du réseau. Le problème



par Samuel Pierre, Alejandro Quintero & Agnès de Montgolfier, Département de génie informatique, École Polytechnique de Montréal, Montréal, QC.

### Abstract

For a set of cells and switches, the problem of cell assignment to switches in cellular mobile networks consists of determining a cell assignment pattern, which minimizes a certain cost function, while respecting certain constraints, especially those related to limited switch's capacity. The cost function integrates a link and a handoff cost component. Assigning cells to switches in cellular mobile networks being an NP-hard problem, enumerative search methods are practically inappropriate to solve large-sized instances of this problem. This paper presents the interaction between tabu search, simulated annealing and standard genetic algorithm to solve this problem. The implementation of these algorithms has been subject to extensive tests in order to measure the quality of solutions. The results obtained show that the simulated annealing and tabu search improve the individuals representing solutions provided by standard genetic algorithm.

### Sommaire

Le problème de l'affectation des cellules aux commutateurs dans les réseaux cellulaires consiste, étant donné un ensemble de cellules et de commutateurs dont les emplacements sont connus, à affecter les cellules aux commutateurs de façon à minimiser une fonction de coût qui comprend une composante de coût de liaison cellule-commutateur et une composante de coût de relève. Cette affectation doit tenir compte de la contrainte de capacité des commutateurs qui ne peuvent supporter qu'un nombre limité d'appels. Un tel problème d'optimisation combinatoire est reconnu NP-difficile, ce qui justifie le recours à des méthodes heuristiques de résolution pour conjurer le risque d'explosion combinatoire. Cet article examine l'efficacité de différentes interactions entre trois méta-heuristiques bien connus jusqu'ici utilisés isolément pour résoudre ce problème: l'algorithme génétique, l'algorithme de recuit simulé et l'algorithme de recherche taboue. Les résultats obtenus de ces interactions montrent que les algorithmes de recuit simulé et de recherche taboue permettent d'améliorer les résultats obtenus par l'algorithme génétique.

revient, à partir de  $n$  cellules et de  $m$  commutateurs, à trouver un schéma d'affectation des  $n$  cellules aux  $m$  commutateurs qui minimise le coût total du réseau tout en respectant certaines contraintes. Si nous devons explorer de façon exhaustive tous les schémas d'affectation possibles ( $m^n$  dans notre cas) pour en choisir le meilleur, nous déboucherions rapidement sur un problème d'explosion combinatoire que même les ordinateurs les plus performants mettraient un temps excessivement long à résoudre.

Considérons maintenant l'aspect de la capacité des commutateurs. Il s'agit alors de réaliser les affectations en tenant compte du fait qu'un commutateur a une capacité limitée, c'est-à-dire que seul un nombre limité de cellules peut lui être raccordé. Cette capacité s'exprime en

volumes d'appels par unité de temps.

Considérons enfin l'aspect du trafic acheminé par l'intermédiaire du réseau. Ce trafic peut varier considérablement à deux moments de la journée. Ainsi, un schéma d'affectation efficace à un moment de la journée peut s'avérer inefficace à un autre moment. Il s'agit donc d'implanter deux affectations correspondant chacune à un moment de la journée. Il peut alors arriver qu'une cellule soit reliée à deux commutateurs différents, et cela doit être pris en compte dans le calcul du coût global du réseau.

Tout compte fait, la nature du problème demande une approche heuristique de résolution, ce que certains chercheurs ont déjà préconisé [4] [5]. Cet article examine l'efficacité de différentes interactions entre ces trois heuristiques. La section 2 expose brièvement le principe de fonctionnement de ces heuristiques. La section 3 définit les interactions entre l'algorithme génétique, le recuit simulé et la recherche taboue. La section 4 présente et analyse les résultats de simulation. La section 5, en guise de conclusion, résume nos principales observations.

## 2.0 Synthèse des méthodes heuristiques de résolution

Dans cette section, nous allons présenter brièvement les trois heuristiques utilisées: l'algorithme génétique, la recherche taboue et le recuit simulé.

### 2.1 L'algorithme génétique

Les algorithmes génétiques (AG) introduits par John Holland [2] sont fondés sur la théorie de la survie des espèces de Charles Darwin. Comme dans la nature où les êtres se reproduisent, dans le modèle des AG, les spécimens se reproduiront aussi; en particulier, ceux jugés les plus forts se reproduiront à un rythme plus rapide. Des opérateurs génétiques seront appliqués sur des candidats en espérant engendrer ainsi de nouveaux candidats plus performants [6] [7].

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 ainsi d'une solution optimale.

#### 2.1.1 Principe des algorithmes génétiques

Les mécanismes de base usuels sur lesquels repose la méthode des algorithmes génétiques sont principalement la représentation des chromosomes et les opérateurs génétiques. Un bon choix des paramètres de ces chromosomes assure la convergence vers une bonne solution.

Dans un AG, les chromosomes ont souvent une représentation binaire. Ce choix le rend intuitivement applicable à tous les problèmes dont les solutions sont transposables en binaire [6]. Les chromosomes sont alors représentés par une chaîne de bits. Cette représentation est indépendante du problème posé et rend l'algorithme génétique d'autant plus robuste.

Les opérateurs génétiques de base sont au nombre de trois: la **sélection**, le croisement et la mutation. La sélection est le processus selon lequel des chaînes de la population sont choisies pour une nouvelle génération d'après leur valeur par la fonction objectif. Plus la valeur de la fonction objectif est élevée, plus cette chaîne a de chances d'être sélectionnée. Notons qu'une des techniques les plus utilisées pour réaliser la sélection est celle de la roulette de casino: d'abord, on calcule la valeur d'aptitude de chaque chromosome, puis on calcule l'aptitude totale en faisant la somme des valeurs d'aptitude de chaque individu de la population; enfin, on calcule le pourcentage de chaque chromosome par rapport à l'aptitude totale.

Le **croisement** est le processus selon lequel les bits de deux chaînes sélectionnées au hasard sont interchangés: dans le langage génétique, on

dira que ces chaînes sont croisées. 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 donc être totalement différentes de leurs parents. Le croisement décrit se produit en un lieu, mais on peut aussi retrouver des croisements avec plusieurs lieux dans certains AG.

La **mutation** est le processus selon lequel la valeur d'un gène choisi au hasard dans un chromosome est régénérée. Ce processus ne survient qu'occasionnellement dans un algorithme génétique. En modifiant aléatoirement la valeur d'un bit dans une chaîne, la mutation est utile pour ramener du matériel génétique qui aurait été oublié par les opérateurs de sélection et de croisement.

De façon générale, un AG fonctionne de la façon suivante:

**Étape 1:** On génère une population initiale de taille  $n$  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:** Le cycle de génération des populations commence alors, chaque nouvelle population remplaçant la précédente. Le nombre  $x$  de générations est déterminé au départ. Dans chaque génération, on choisit  $n$  chromosomes auxquels on va appliquer les différents opérateurs génétiques. Après chaque génération, les  $n$  nouveaux chromosomes créés remplacent la génération précédente.

Après la  $x^{\text{ième}}$  génération, les chromosomes auront évolué de telle façon que cette dernière génération contienne des chromosomes meilleurs que ceux des générations précédentes.

#### 2.1.2 L'algorithme génétique d'affectation de cellules

L'adaptation des algorithmes génétiques pour résoudre notre problème d'affectation de cellules aux commutateurs vise à 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 sur la capacité des commutateurs et celle d'affectation unique des cellules aux commutateurs. Dans un premier temps, il faut coder les solutions du problème afin de pouvoir le résoudre à l'aide d'un AG. La forme la plus répandue du codage est la représentation binaire où les gènes ne prennent que des valeurs 0 et 1. La représentation utilisée dans ce problème est cependant non-binaire. Chaque chromosome représente un schéma d'affectation spécifique. Les éléments de la chaîne représentant un chromosome sont des entiers, qui représentent les différents commutateurs numérotés de 1 à  $m$ . La Figure 2 donne un exemple d'un chromosome représentant un schéma d'affectation de 8 cellules à trois commutateurs.

La longueur des chaînes est égale au nombre de cellules et reste inchangée, car toutes les cellules du réseau doivent être affectées. De même, la valeur maximale d'un gène de ces chromosomes est égale au nombre maximal de commutateurs du réseau. La lecture des chromosomes se fait de gauche à droite; ainsi, le premier bit du chromosome contient le numéro du commutateur auquel la cellule numéro 1 est reliée. Le codage adopté permet de satisfaire une contrainte: celle de l'affectation unique des cellules aux commutateurs, car un gène d'un chromosome ne peut pas prendre simultanément plus d'une valeur. La seule contrainte à satisfaire reste alors celle sur la capacité des commutateurs.

1	2	2	3	1	3	2	3
---	---	---	---	---	---	---	---

Figure 2: Représentation non-binaire d'un chromosome

## 2.2 L'algorithme de recuit simulé

La méthode du recuit simulé (RS) a l'originalité de pouvoir s'appliquer à une grande variété de domaines et en particulier aux télécommunications. Le recuit simulé est une heuristique d'optimisation qui consiste en une recherche locale par perturbations. Ce processus donne la possibilité de s'éloigner occasionnellement d'un minimum local pour permettre ainsi un élargissement du champ de recherche de la solution idéale [3].

### 2.2.1 Principe de la méthode

Le recuit simulé est une procédure de recherche selon laquelle la topologie courante, retenue momentanément comme meilleure solution, est continuellement comparée à d'autres topologies qui lui sont très proches. Ces topologies voisines sont obtenues à la suite de petites perturbations sur la topologie courante. Lorsqu'une perturbation aboutit à une topologie meilleure que la solution courante, elle est sauvegardée comme solution courante. Cependant, il peut arriver que, suite à une perturbation, la topologie voisine obtenue soit conservée comme solution courante, même si elle n'est pas meilleure que la solution courante, à condition qu'elle respecte une certaine probabilité d'acceptation. Le fait d'accepter de temps à autre une solution dégradée permet d'éviter de s'enfermer trop tôt dans un minimum local. D'autre part, la probabilité d'acceptation doit être suffisamment faible, de telle sorte que l'algorithme puisse s'approcher le plus possible de l'optimum global.

En fin de compte, l'algorithme se termine lorsque le critère d'arrêt est satisfait. À cette étape, la recherche locale devrait avoir abouti à un minimum local ou à un optimum global. Il s'ensuit que la solution idéale trouvée est, soit localement optimale vu le nombre élevé de minima locaux, soit globalement optimale dans le meilleur des cas.

### 2.2.2 Affectation de cellules aux commutateurs avec l'algorithme de recuit simulé

Dans l'algorithme du recuit simulé, les paramètres en présence sont: la température  $\theta$ , et le facteur de recuit  $\alpha$ . Cet algorithme commence par générer une topologie initiale, chaque cellule étant affectée à un commutateur de façon aléatoire, sans se soucier des contraintes de capacité des commutateurs. En attendant, elle est sauvegardée comme première solution courante. Le paramètre  $\theta$  est initialisé à une valeur  $\theta_0$  relativement élevée. On débute alors la boucle des perturbations. Cette boucle sera répétée tant que le critère d'arrêt n'aura pas été atteint. Notons que le critère d'arrêt doit être défini de manière à permettre la réalisation d'un maximum de répétitions de la boucle. En outre, la valeur du facteur de recuit  $\alpha$  doit être choisie adéquatement pour éviter de décroître  $\theta$  trop rapidement, ce qui rendrait moins poussée la recherche de la meilleure solution.

## 2.3 L'algorithme de recherche taboue

La méthode de recherche taboue (RT) est une technique adaptative introduite dans les années 70 en optimisation combinatoire pour résoudre les problèmes difficiles. Elle est considérée comme une méta-heuristique, qui peut être utilisée pour résoudre différents types de problème [1].

### 2.3.1 Fondements de la méthode de recherche taboue

Présentons dans un premier temps l'algorithme de descente simple. Il part d'une solution initiale qu'il essaie d'améliorer de manière itérative. Pour cela, il génère à chaque étape un sous-ensemble  $V$  du voisinage  $N(s)$  de la solution courantes. Ensuite, il choisit dans cet ensemble  $V$  la meilleure solution, c'est-à-dire celle qui minimise la fonction objectif sur  $V$ . Cette dernière solution devient la solution courante et l'algorithme continue jusqu'au moment où aucun élément de  $V$  ne permet d'avoir une meilleure valeur de la fonction objectif.

La méthode de recherche taboue est une amélioration de l'algorithme général de descente. Elle essaie principalement d'éviter le piège des minima locaux. Pour cela, il est nécessaire d'accepter de temps en temps 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 non forcément meilleures introduit un risque

de cycle, c'est-à-dire un retour vers des solutions déjà explorées. D'où l'idée de conserver une liste taboue  $T$  (tabu list) des solutions déjà visitées. Ainsi, lors de la génération de l'ensemble  $V$  des solutions voisines candidates, on enlève toutes les solutions appartenant à la liste taboue. Notons tout de même que, d'une part, le stockage de toutes les solutions déjà visitées peut nécessiter beaucoup de mémoire et que, d'autre part, il peut s'avérer utile de revenir à une solution déjà visitée pour continuer la recherche dans une autre solution. Un compromis a été adopté en ne gardant dans la liste taboue que les  $k$  dernières solutions. L'algorithme s'arrête quand aucune amélioration n'est intervenue depuis un certain nombre d'itérations ou si toutes les solutions voisines candidates sont taboues.

### 2.3.2 Affectation de cellules aux commutateurs avec la recherche taboue

La démarche adoptée consiste globalement à modifier itérativement une solution initiale en espérant aboutir à une solution finale respectant les contraintes du problème [5]. À cette fin, la méthode RT utilise des mouvements pour passer d'une solution à une autre à l'intérieur d'un espace de recherche prédéfini. Dans l'adaptation de la méthode RT, l'espace de recherche choisi est libre des contraintes de capacité sur les commutateurs, mais respecte la contrainte d'affectation unique des cellules aux commutateurs. La faisabilité de la solution finale n'est donc pas garantie, mais le fait de pouvoir examiner un plus grand nombre de possibilités augmente les chances d'aboutir à de bonnes solutions. RT associe à chaque solution deux valeurs numériques: la première est le coût de la solution, calculé à partir de la fonction objectif; la deuxième est une évaluation de la solution prenant en compte le coût et une sanction, sous forme de pénalité, pour le non-respect des contraintes de capacité, s'il y a lieu. À chaque étape, RT choisit la solution ayant la meilleure évaluation.

Contrairement à la méthode de descente, quand elle arrive à un optimum local, la méthode RT choisit la solution voisine qui dégrade le moins la fonction objectif. Pour éviter les cycles autour de cet optimum, une liste taboue garde les  $k$  dernières solutions et interdit momentanément un retour vers ces solutions. Les solutions sont libérées après  $k$  itérations ou lorsqu'elles satisfont un critère d'aspiration.

## 3.0 Interactions entre les heuristiques

Cette section expose les interactions des heuristiques que nous proposons pour la résolution du problème d'affectation des cellules à des commutateurs dans les réseaux cellulaires.

### 3.1 Interaction entre l'algorithme génétique et le recuit simulé

L'inconvénient principal de l'algorithme de recuit simulé (RS) est qu'il est appliqué à une topologie générée de façon totalement aléatoire. Ainsi, la solution à améliorer peut être une solution de coût très élevé ou une solution qui ne respecte pas les contraintes de capacité. Il serait donc préférable d'appliquer l'algorithme de recuit simulé à une solution que l'on sait faisable et, dans le meilleur des cas, de faible coût.

Nous voulons optimiser l'algorithme génétique en améliorant à chaque génération le meilleur chromosome de la population. Pour cela, à chaque génération, nous prélevons le meilleur chromosome de la population, nous lui appliquons l'algorithme de recuit simulé et nous obtenons une solution qui sera au moins aussi bonne que la précédente, puisque l'algorithme de recuit simulé ne peut produire qu'une solution de coût inférieur ou égal à celui de la topologie de départ. Nous obtenons ainsi un nouveau chromosome que nous insérons dans la population de l'algorithme génétique, à la place du chromosome de coût maximal, c'est-à-dire le moins bon chromosome de la population. L'algorithme général de cette version est présenté à la Figure 3.

### 3.2 Interaction entre l'algorithme génétique et la recherche taboue

Dans le cas de la recherche taboue, la solution de départ est créée en attribuant chaque cellule au commutateur le plus proche, en terme de



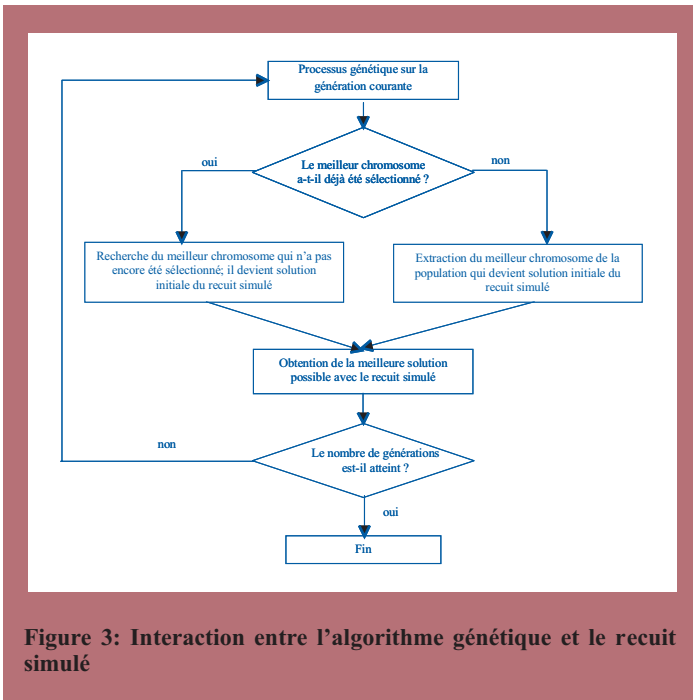


Figure 3: Interaction entre l’algorithme génétique et le recuit simulé

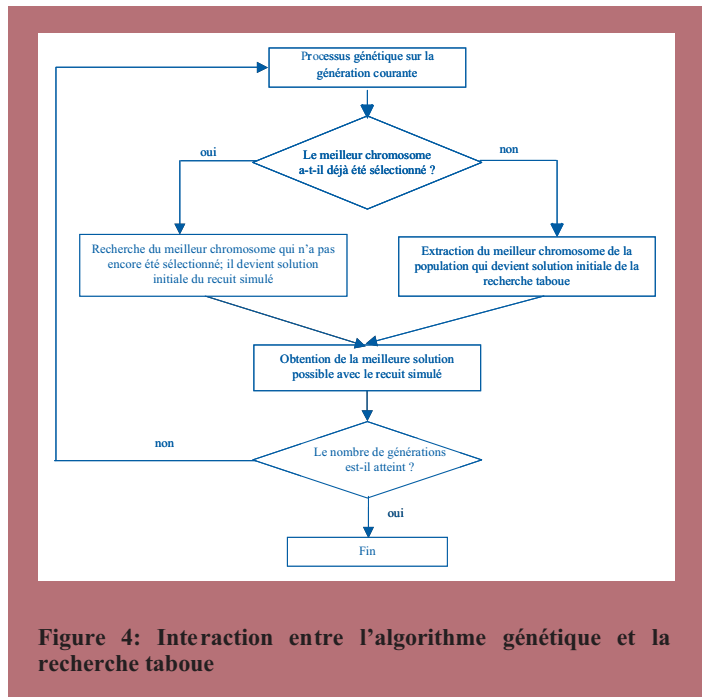


Figure 4: Interaction entre l’algorithme génétique et la recherche taboue

distance. À chaque test, c’est donc la même solution de départ qui est fournie et la recherche taboue aboutira toujours à la même topologie finale. Donc, avec un même fichier de données initial, on ne pourra aboutir qu’à une seule solution finale.

L’idée est donc de générer la solution initiale à l’aide de l’algorithme génétique. Nous aurons ainsi à notre disposition une variété de solutions de départ et on peut espérer que l’algorithme de recherche taboue permettra d’atteindre d’autres minima locaux, voire le minimum global. Donc, l’idée est d’appliquer la recherche taboue à chaque génération de l’algorithme génétique. L’algorithme général est présenté à la Figure 4.

Nous voulons optimiser l’algorithme en améliorant à chaque génération le meilleur chromosome de la population. Pour cela, à chaque génération, on prélève le meilleur chromosome de la population, on lui applique l’algorithme de recherche taboue et on obtient une solution qui sera au moins aussi bonne que la précédente, puisque l’algorithme de recherche taboue ne peut produire qu’une solution de coût inférieur ou égal à celui de la topologie de départ. Nous obtenons ainsi un nouveau chromosome que nous insérons dans la population de l’algorithme génétique, à la place du chromosome de coût maximal, c’est-à-dire le moins bon chromosome de la population.

Dans cette deuxième interaction entre l’algorithme génétique et la recherche taboue, le processus génétique se fait à chaque génération de l’algorithme génétique. Une fois le processus génétique accompli, on s’intéresse au meilleur chromosome de la population courante. Si ce chromosome constitue une topologie qui a déjà été sélectionnée pour être solution initiale de la recherche taboue, on déplace notre choix sur le chromosome suivant, dans la population triée en ordre croissant des coûts des chromosomes, et ainsi de suite jusqu’à trouver le meilleur chromosome de la population qui n’a jamais été choisi pour être topologie initiale de la recherche taboue.

#### 4.0 Implémentation et résultats

Nous présentons dans cette section les résultats obtenus des interactions entre les algorithmes présentées précédemment.

##### 4.1 Algorithmes seuls

Dans un premier temps, on effectue une simulation sur chacun des algorithmes pris séparément afin d’avoir des solutions de référence. Les tests ont été effectués sur un ensemble de 900 topologies avec 3 réseaux différents: 100 cellules et 5 commutateurs, 150 cellules et 6 commutateurs,

et 200 cellules et 7 commutateurs. Pour chacun des algorithmes, nous avons effectué 20 tests pour chaque topologie différents et les valeurs présentées correspondent à une moyenne des 20 valeurs obtenues.

Afin de déterminer le nombre de générations qui permet d’obtenir le meilleur compromis coût obtenu/temps nécessaire, nous avons effectué des tests avec plusieurs nombres de générations. Les probabilités de croisement et de mutation ont été fixées respectivement à 0.9 et 0.08. La Figure 5 présente l’évolution du coût du meilleur chromosome de la population en fonction du nombre de générations de l’algorithme génétique pour un réseau de 100 cellules et 5 commutateurs.

Cela illustre le fait que le nombre de 500 générations permet d’aboutir à de bons résultats, en un temps acceptable. Avec 800 générations, on obtient des résultats un peu meilleurs, mais le temps nécessaire est plus élevé. Avec seulement 100 générations, les coûts de topologies obtenus sont élevés; le seul point positif est le temps d’exécution qui est faible. Avec 500 générations, le compromis atteint est bon; on atteint des coûts comparables à ceux atteints avec 800 générations, en un temps acceptable. Nous avons choisi donc d’effectuer les tests de référence avec un nombre de références égal à 500. Afin d’avoir en notre possession des valeurs de référence, nous effectuons des tests avec chacun des trois algorithmes pris séparément. Les résultats sont présentés à la Figure 6. Dans chacun des cas, la solution initiale est spécifique à l’algorithme:

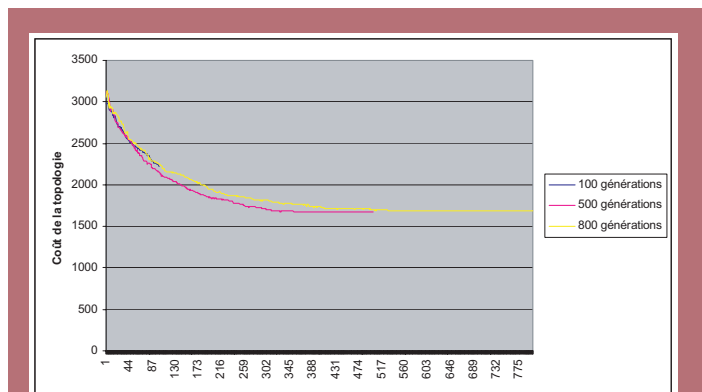
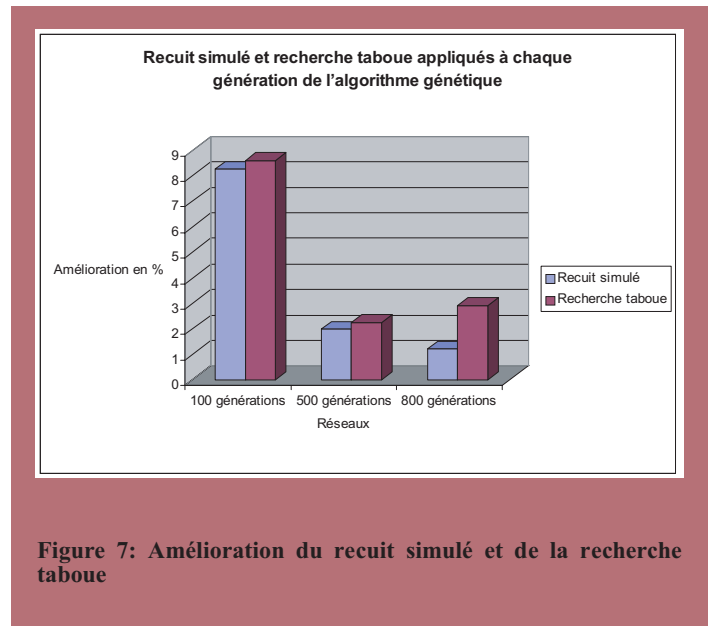
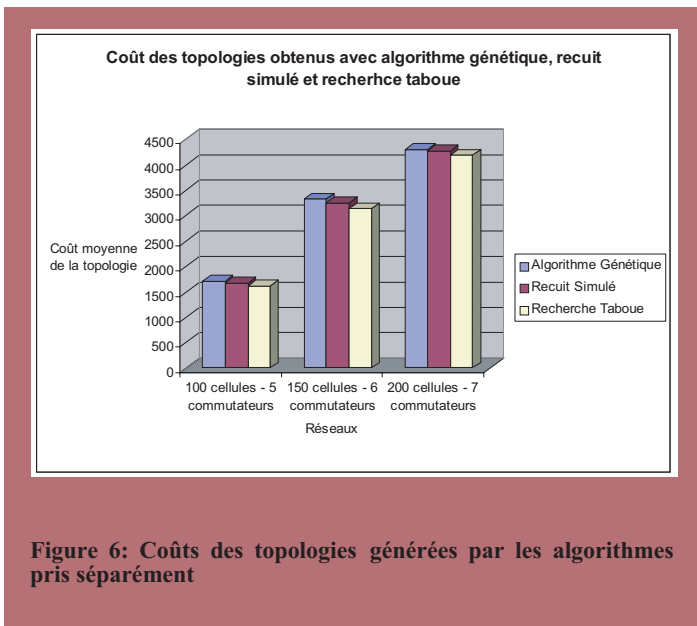


Figure 5: Évolution du coût du meilleur chromosome de la population en fonction du nombre de générations de l’algorithme génétique



elle est générée aléatoirement pour l'AG, aléatoirement également pour le recuit simulé et correspond à une affectation des cellules aux commutateurs les plus proches pour la recherche taboue. Il en résulte que les meilleurs coûts sont obtenus par l'algorithme de recherche taboue.

#### 4.2 Interactions entre les trois algorithmes

Les tests ont été effectués sur un ensemble de 900 topologies avec 3 réseaux différents: 100 cellules et 5 commutateurs, 150 cellules et 6 commutateurs, 200 cellules et 7 commutateurs. Nous travaillons toujours sur une population de 100 chromosomes et nous réalisons les tests en faisant tourner l'algorithme génétique avec 100, 500 et 800 générations. Dans les trois cas, nous appliquons l'algorithme génétique, puis nous extrayons le meilleur chromosome de chaque génération qui devient la solution initiale du recuit simulé ou de la recherche taboue. Les paramètres utilisés sont une probabilité de croisement égale à 0.9 et une probabilité de mutation égale à 0.08. Pour le recuit simulé, nous utilisons un facteur de recuit égal à 0.7. Les résultats sont présentés à la Figure 7.

L'amélioration engendrée par le recuit simulé ou recherche taboue n'est pas significative lorsque l'algorithme génétique est effectué pendant 500 ou 800 générations. En effet, l'AG permet d'aboutir à des solutions acceptables, lorsque le nombre de générations est suffisamment élevé, c'est-à-dire quand il est égal par exemple à 500 ou 800. Deux hypothèses sont envisageables: soit on a atteint, grâce à l'algorithme génétique, un minimum local et l'algorithme de recuit simulé ou de recherche taboue ne permet pas d'en sortir, soit on a réussi à atteindre l'optimum global et, implicitement, l'algorithme de recuit simulé ne peut pas fournir une meilleure solution que celle déjà trouvée. Cependant, c'est la première hypothèse qui se trouve vérifiée. Or, on sait qu'il existe des topologies de meilleurs coûts que ceux-là, en se référant aux coûts trouvés par l'algorithme de recherche taboue. On peut donc en conclure que l'algorithme génétique, appliqué avec un nombre de générations suffisamment grand, permet d'obtenir une solution acceptable qui se situe dans le voisinage d'un minimum local; lorsqu'on applique le recuit simulé sur cette solution, on n'arrive pas à se dégager de ce minimum local.

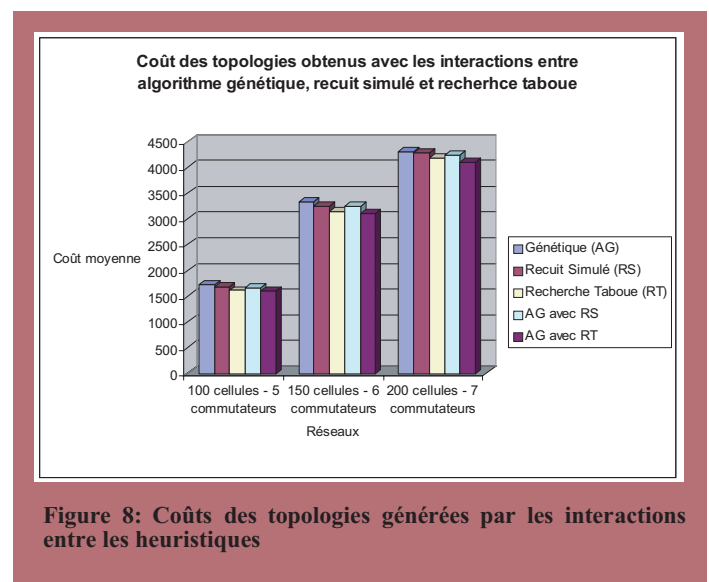
En revanche, l'amélioration fournie par le recuit simulé et aussi pour la recherche taboue sur une solution générée par l'algorithme génétique, avec un nombre de générations faible, par exemple 100 générations, est relativement importante. En effet, en seulement 100 générations, l'AG ne permet pas d'obtenir une solution très satisfaisante; autrement dit, il ne trouve pas une solution dans le voisinage d'un optimum local. Par conséquent, lorsqu'on applique le recuit simulé ou la recherche taboue sur cette solution, il y a plus de chances de trouver une solution bien

meilleure et donc de fournir une amélioration significative (Figure 8).

#### 5.0 Conclusion

Dans cet article, nous avons étudié différents scénarios d'interaction entre trois heuristiques de résolution du problème d'affectation de cellules à des commutateurs dans les réseaux cellulaires mobiles. Nous avons donc conçu, implémenté et testé un algorithme génétique applicable à une population initiale, dans le but d'obtenir la solution qui minimise le coût du réseau, tout en respectant la contrainte d'affectation unique des cellules aux commutateurs et celle sur la capacité des commutateurs. Pour cela, nous avons d'abord conçu une procédure efficace pour la génération d'une population initiale sans doubles. Nous avons ensuite défini, implémenté et appliqué des opérateurs génétiques spécifiques au contexte de notre problème, opérateurs qui ont été appliqués par la suite à cette population initiale.

Tout compte fait, les algorithmes génétique, de recuit simulé et de recherche taboue constituent de bonnes méthodes pour obtenir des schémas d'affectation de cellules aux commutateurs qui sont de bonne qualité même s'ils ne sont pas optimaux. Toutefois, les résultats obtenus de ces interactions montrent que les algorithmes de recuit simulé et de recherche taboue permettent d'améliorer les résultats obtenus par l'algorithme génétique.



## 6.0 Bibliographie

- [1]. F. Glover, E. Taillard, D. de Werra, "A user's guide to tabu search", Annals of Operations Research, vol. 41, No. 3, 1993, pp. 3-28.
- [2]. J. H. Holland, "Adaptation in Natural and Artificial System", Computing, vol. 2, no 2, 1973, pp. 88-105.
- [3]. S. Kirpatrick, C.-D. Gelatt Jr., M.-P. Vecchi, "Optimization by Simulated Annealing", Science, vol. 220, 1983, pp. 671-680.
- [4]. A. Merchant, B. Sengupta, "Assignment of Cells to Switches in PCS Networks", IEEE/ACM Transactions on Networking, vol. 3, No. 5, 1995, pp. 521-526.
- [5]. S. Pierre, F. Houeto, "Assigning Cells to Switches in Cellular Mobile Networks Using Tabu Search", IEEE Transactions on Systems, Man, and Cybernetics: Part B (Cybernetics), vol. 32: Part B, no 3, juin 2002, pp. 229-239.
- [6]. S. Pierre, G. Legault, "A Genetic Algorithm for Designing Distributed Computer Network Topologies", IEEE Transactions on Man, Systems, and Cybernetics, vol. 28, no 2, 1998, pp. 249-258.
- [7]. A. Quintero, S. Pierre, "A Multi-population Memetic Algorithm for Assigning Cells to Switches in Cellular Mobile Networks", IEEE Communications Letters, vol. 6, no 11, novembre 2002, pp. 484-486.

### A propos des auteurs

**Samuel Pierre** a obtenu un doctorat en génie électrique de l'École Polytechnique de Montréal en 1991. Il est actuellement professeur titulaire au département de génie informatique de cette dernière institution, directeur du Laboratoire de recherche en réseautique et informatique mobile (LARIM), et titulaire de la Chaire de Recherche Industrielle GRSNG/Ericsson en Systèmes Réseautiques Mobiles de Prochaines Générations. De 1987 à 1998, il a été professeur régulier à l'Université du Québec à Trois-Rivières et à la Télé-université du Québec, professeur associé à l'Université Laval, professeur invité à l'École Polytechnique Fédérale de Lausanne (EPFL, Suisse) et à l'Université Paris 7. Ses domaines de spécialisation incluent les réseaux de communications, l'informatique mobile, l'intelligence artificielle répartie et apprentissage électronique (e-learning). Il a à son actif plus de 200 publications scientifiques dont 6 livres, 6 chapitres de livres, 3 ouvrages collectifs édités. Il est membre sénior de IEEE, membre de ACM et Fellow de l'Institut Canadien des Ingénieurs (ICI). Il est éditeur associé IEEE Canadian Review.



**Alejandro Quintero** est Docteur en Informatique de l'Université Joseph Fourier (Grenoble, France 1993), D.E.A. en Informatique de l'INPG (Institut National Polytechnique de Grenoble, 1989) et Ingénieur Informaticien de l'Université de Los Andes (Bogotá, Colombie 1983). M. Quintero est professeur adjoint au département de génie informatique de l'École Polytechnique de Montréal. Ses activités de recherche portent principalement sur les réseaux de prochaines générations et l'informatique mobile. Au chapitre des publications, M. Quintero est l'auteur principal ou co-auteur de plus de 30 publications scientifiques.



**Agnes de Montgolfier** est ingénieure de l'École de Chimie, Physique et électronique de Lyon, France. Ses activités de recherche portent principalement sur l'informatique mobile et les algorithmes génétiques.



### Informal Meeting of Four National Engineering Organizations in Kingston



This self-defined group meets informally a few times per year to share information and to ensure cooperation between the four national engineering organizations. The most recent meeting took place at the EIC offices in Kingston, Ontario on the 6th Nov. 2003.



From left to right: **Claude Paul Boivin**, CEO, The Association of Consulting Engineers of Canada, **John Plant**, Executive Director, Engineering Institute of Canada **Marie Lemay**, CEO, Canadian Council of Professional Engineers and **Phillop Cockshutt**, Executive Director, Canadian Academy of Engineering.

**Claude Paul Boivin**, CEO, ACEC and **John Plant** Executive Director, EIC close a cooperative agreement to share the EIC Career Site.



### 2003 IEEE Canada Vehicular Technology Conference Grant and Bursary

The IEEE Canada Student Activities Committee is very pleased to announce the recipients of the 2003 IEEE Canada Vehicular Technology Conference Grant and Bursary as follows:

- The travel bursary was awarded to Hamidreza Saligheh Rad, of Queens University, to attend the IEEE Semi-Annual Vehicular Technology Conference, VTC'04, May 2004.
- The research grant of \$1500 was awarded to the RF and Microwave Electronics Research Group, Queens University, to support research on the Binary Phase Shift Keying (BPSK) Demodulator.

In supplement to the monetary award, the recipients will receive a letter of congratulations from IEEE Canada.

On behalf of the IEEE Canada Student Activities Committee, I thank all the Canadian IEEE Student Branch leaders and counsellors for promoting these awards in their schools.

**Jorge Aguirre**  
SAC Awards Chair, IEEE Canada (Region 7)