

# Rapport de projet - Affectation des fréquences dans les réseaux mobiles

---

Thomas Gloriod, Paul Locatelli et Pierre Rognon



Printemps 2013

# Table des matières

<b>1</b>	<b>Présentation du sujet</b>	<b>3</b>
1.1	Détails du problème . . . . .	3
1.2	Objectifs . . . . .	4
<b>2</b>	<b>Principe de résolution</b>	<b>5</b>
2.1	Méthode approchées . . . . .	5
2.2	La recherche tabou . . . . .	5
<b>3</b>	<b>Algorithme développé</b>	<b>6</b>
3.1	Génération de la solution initiale . . . . .	6
3.2	Implémentation de la recherche tabou . . . . .	6
<b>4</b>	<b>Résultats observés</b>	<b>7</b>
<b>5</b>	<b>Conclusion</b>	<b>8</b>
<b>A</b>	<b>Génération de la solution initiale</b>	<b>9</b>
<b>B</b>	<b>Implémentation de la recherche tabou</b>	<b>10</b>

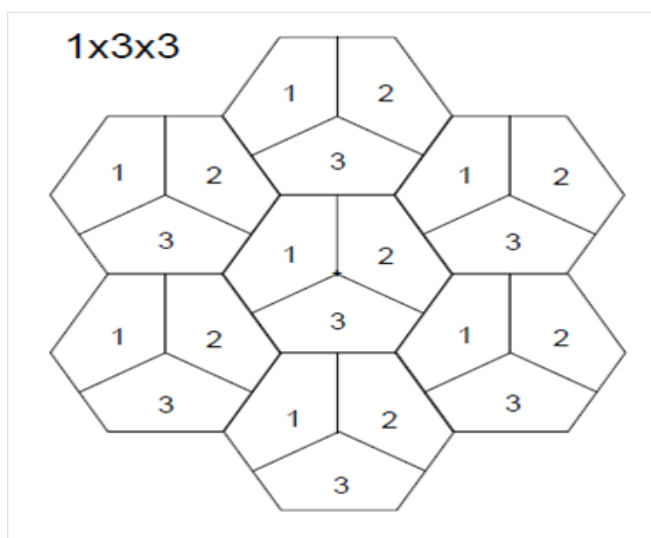
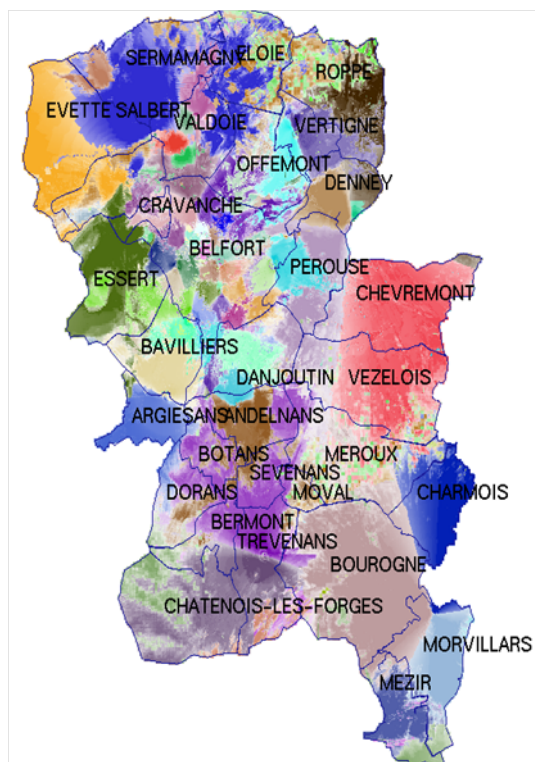
# Chapitre 1

## Présentation du sujet

Le problème d'affectation de fréquences est un problème type en recherche opérationnelle. Ce problème s'apparente à un problème de coloration de graph. Il s'agit ici d'attribuer des fréquences à différents secteurs d'une même zones. Deux secteurs limitrophes ne pouvant pas émettre sur la même fréquence.

### 1.1 Détails du problème

Le département étudié, ici le Territoire de Belfort comporte 36, chacun étant divisé en maximum trois secteurs pour un total de 88 secteurs. Chaque secteur d'un même site émet sur la sous-bande fréquence 1, 2 ou 3.



Pour que les clients soient couverts, les secteurs ayant une frontière communes doivent avoir des fréquences différentes afin de minimiser les interférences. Au sein du territoire

12 695 points test permettent d'évaluer la qualité de l'allocation des fréquences. Chaque points test est caractérisé par un nombre de clients et le débit qu'il souhaite avoir. Le nombre de clients n'ayant pas accès au réseau est donc la fitness à minimiser.

## 1.2 Objectifs

Afin d'assurer l'accès au réseau au maximum de clients, il s'agit d'allouer les fréquences à chaque secteur de chaque sites de manière à minimiser le nombre de clients non-couvert par le réseau.

# Chapitre 2

## Principe de résolution

Pour résoudre ce type de problème, l'utilisation de méthode exacte demanderait le calcul de  $6^{36}$  soit environ  $10^{28}$  allocation de fréquences différentes. C'est pourquoi pour ce genre de problème où le nombre de solution est trop importante, les méthodes de résolution approchées sont utilisées.

### 2.1 Méthode approchées

Le principe de cette méthode de résolution est simple. On cherche d'abord une solution au problème, soit de manière aléatoire, soit en en calculant pas trop mauvaise. On définit ensuite une liste de voisin à cette solution. Un voisin d'une solution suit des règles précises, définies au préalable. On calcule la fitness de tous les voisins et on garde la meilleure solution trouvée parmi elles. Il suffit ensuite de répéter l'opération sur la nouvelle solution afin d'améliorer le résultat jusqu'à ce qu'aucun voisin ne soit meilleur que la solution courante.

### 2.2 La recherche tabou

La recherche tabou est une amélioration de l'algorithme énoncé précédemment. Le principe est similaire cependant lorsqu'une solution voisine est étudiée, elle est placée ensuite dans la liste tabou pendant plusieurs itérations et ne pas reconsidérée. Cette optimisation permet de ne pas perdre de temps en recalculant les mêmes solutions à chaque fois.

# Chapitre 3

## Algorithme développé

3.1 Génération de la solution initiale

3.2 Implémentation de la recherche tabou

## Chapitre 4

### Résultats observés

# Chapitre 5

## Conclusion

L'algorithme de recherche tabou est fonctionnel. Il permet d'obtenir une fitness très satisfaisante afin d'offrir aux clients la meilleur couverture réseau.

Des améliorations restent possibles en modifiant la méthode du choix des voisins à chaque itération, ou en modifiant la durée de la liste tabou.

L'exercice est intéressant car il demande de s'immiscer dans un programme déjà existant, de le comprendre afin d'y ajouter les fonctions à développer. C'est un exercice réaliste puisque dans le monde de l'entreprise, les programmes à développer partent rarement de zéro.

Ce projet nous a également permis de nous perfectionner dans le langage C++, de découvrir de nouvelles subtilités et d'améliorer notre manière de programmer. Pour partager les sources du programme, l'utilisation d'un logiciel de gestion de version facilite la tâche.

Enfin ce projet a été une expérience enrichissante car effectué au sein d'un groupe de trois étudiants. La communication est importante ainsi que la répartition des tâches. Le travail en groupe permet le partage des compétences et des idées.



# Annexe A

## Génération de la solution initiale

```
int* site::randomizeTableFreq(){
    int* tableFreq = new int[3];
    int* table = new int[3]; for(int i=0; i<3;i++)table[i]=i+1;
    int* tableTmp = NULL;
    int taille = 3;
    long random;
    bool shift;
    for(int i=0; i<3; i++)
    {
        shift = false;
        random = Random::aleatoire(taille);
        tableFreq[i]=table[random];
        tableTmp = new int[taille-1];
        for(int j=0; j<taille; j++)
        {
            if(j != random){
                if(shift) tableTmp[j-1] = table[j];
                else tableTmp[j] = table[j];
            }
            else{shift=true;}
        }
        delete table;
        table = tableTmp;
        taille--;
    }
    delete tableTmp;
    return tableFreq;
}
```

## **Annexe B**

### **Implémentation de la recherche tabou**