

Trois hyper-heuristiques pour le problème d'affectation de fréquence dans un réseau cellulaire

Yasmine Lahsinat¹ Belaid Benhamou² Dalila Boughaci¹

¹ LRIA-FEI/USTHB Alger 16111, Algérie

² Université Aix-Marseille, LSIS

Domaine universitaire de Saint Jérôme, Avenue Escadrille Normandie Niemen
13397 MARSEILLE Cedex 20

{ylahsinat, dboughaci}@usthb.dz belaid.benhamou@univ-amu.fr

Résumé

Dans cet article, nous nous intéressons aux approches hyper-heuristiques pour résoudre le problème de Minimisation d'Interférences lors de l'Affectation de Fréquence noté (MI-FAP) dans un réseau cellulaire. Nous proposons trois variantes d'hyper-heuristiques. Ces hyper-heuristiques se distinguent entre elles de par la stratégie utilisée pour sélectionner l'heuristique de bas niveau à appliquer. La première se base sur une fonction de choix (HFC) pour sélectionner l'heuristique la plus appropriée. La seconde choisit aléatoirement l'heuristique de bas niveau et est notée (HHA). La troisième, nommée stochastique hyper-heuristique (SHH) combine les deux stratégies de sélection : la fonction de choix et la stratégie aléatoire pour sélectionner l'heuristique de bas niveau. La sélection de l'heuristique se fait selon l'une des deux stratégies en utilisant une probabilité fixe. Nous proposons aussi une adaptation d'une méta-heuristique récente pour résoudre le problème MI-FAP. A cet effet, nous avons retenu l'algorithme d'Optimisation Basée sur la Biogéographie noté (OBB). Des expérimentations sont réalisées sur des benchmarks publics de diverses tailles avec pour objectif de prouver l'efficacité de nos approches hyper-heuristiques. Les résultats trouvés sont comparés à ceux obtenus par la méthode d'Optimisation basée sur la Biogéographie (OBB). Les résultats obtenus par les approches hyper-heuristiques sont nettement meilleures que ceux obtenus par la méta-heuristique(OBB). Les résultats expérimentaux montrent l'efficacité des deux méthodes SHH et HHA comparativement à la méthode HFC pour résoudre le MI-FAP.

Abstract

In this paper, we are interested in hyper-heuristic approaches for solving the Minimum Interference Frequency Assignment Problem (MI-FAP) in cellular net-

work. We propose three variants of hyper-heuristics. The Hyper-heuristics vary from each other in the strategy used to select the heuristics. First, we propose a Choice Function based Hyper-Heuristic (CFH) which uses a function to select the appropriate heuristic. The second approach is a Random Hyper-Heuristic (RHH) that selects randomly the heuristic to apply. The third one is a Stochastic Hyper-Heuristic (SHH). SHH combines both the choice function and the randomness strategies. SHH uses a walk probability to decide, among the selection strategies, the heuristic that is going to be applied. All of the algorithms have been tested, evaluated on public available benchmarks and compared with the Biogeography Based Optimization (BBO) meta-heuristic. The experiments show the effectiveness of the proposed Hyper-Heuristics methods compared to BBO. On other hand, the experimental results prove that the performances of both SHH and RHH methods are better than the CFH method in solving the MI-FAP.

1 Introduction

Un réseau radio mobile passe plusieurs phases lors de son design telles que le placement des antennes et l'affectation de fréquences. Cette dernière est une étape cruciale dans le design d'un réseau de télécommunication. C'est un problème d'optimisation combinatoire bien connu qui revient à trouver un plan de fréquences optimal de façon à satisfaire la demande en trafic et à assurer la qualité de service tout en minimisant les interférences co-canal et canaux adjacents. Le problème de minimisation des interférences lors de l'affectation de fréquence en anglais *Minimum Interference Frequency Assignment Problem* et noté (MI-FAP) a été introduit dans les années soixante par Met-

zeger [21], il a depuis suscité l'intérêt de beaucoup de chercheurs [1, 11, 17].

En pratique les opérateurs de téléphonies mobile disposent d'un nombre limité de fréquences radio représentant une ressource rare et couteuse de l'opérateur. Ils disposent aussi d'un nombre d'émetteurs / récepteurs (*TRXs*) auxquels il faut affecter des fréquences pour assurer le service. Afin de gérer ces deux faits, les opérateurs font appel au principe de réutilisation de fréquences. Ce dernier n'a pas que des avantages, il a pour effet d'engendrer des interférences particulièrement lorsque des (*TRXs*) proches géographiquement utilisent la même fréquence ou des fréquences adjacentes. En théorie le problème d'affectation de fréquences est un problème NP-Difficile [13].

D'un point de vue mathématique le problème MI-FAP est une généralisation du problème de coloration de graphe. Le problème de minimisation d'interférences lors de l'affectation des fréquences est un problème d'optimisation combinatoire dont l'issue est la découverte d'une affectation de fréquences pour chaque *TRXs* du réseau avec le minimum possible d'interférences et le respect des contraintes ci-dessous nommées :

1. Les contraintes de cellules :

- Les contraintes de séparation : les émetteurs / récepteurs d'une même cellule ne doivent pas utiliser la même fréquence.
- Les canaux bloqués : certaines fréquences ne sont pas opérationnelles dans certaines cellules.

2. Les contraintes des interférences inter-cellules :

- Les interférences Co-channel : ces dernières sont engendrées dans le cas où deux *TRXs* proches géographiquement utilisent la même fréquence (f).
- Les interférences Canal-adjacent : ces dernières sont engendrées dans le cas où deux *TRXs* proches géographiquement utilisent des fréquences adjacentes(f), ($f+1$ / $f-1$).

Diverses méthodes et approches pour résoudre le problème MI-FAP ont été proposées. Nous pouvons trouver des approches évolutionnaires telles que EAs et des algorithmes d'optimisation par colonies de fourmis (ACO / EAs) [18], un algorithme de recherche Tabu [8, 5] et ses améliorations [22]. On peut citer aussi un algorithme de recuit simulé utilisé dans [9], un algorithme Branch and cut proposé dans [12], un algorithme culturel [2], ainsi que des approches de résolution par algorithme hybride utilisant l'algorithme

génétique et la recherche Tabu [20] et d'autre approches hybrides que l'on trouve dans [19]. Une résolution par hyper-heuristique parallèles a été proposé dans [24] sur des données non disponibles.

Dans ce papier, nous proposons trois hyper-heuristiques qui se basent sur différents mécanismes lors de la sélection de l'heuristique de bas niveau à appliquer pour résoudre le MI-FAP. La première hyper-heuristique se base sur une fonction de choix, elle est notée(HFC), la deuxième choisit aléatoirement l'heuristique à appliquer, elle est notée (HHA) et enfin la troisième nommée Stochastique Hyper-Heuristique (SHH) combine les deux précédentes méthodes de sélection pour choisir l'heuristique de bas niveau à utiliser. Des expérimentations numériques sont réalisées sur des différents benchmarks dans le but de tester et de prouver l'efficacité des approches proposées. Nous comparons les résultats obtenus par les approches hyper-heuristiques avec ceux obtenus par l'adaptation de l'algorithme d'optimisation basée sur la Biogéographie pour le problème MI-FAP. Cette dernière est une des méta-heuristiques les plus récentes qui a retenu notre attention. Nous étions alors curieux de voir ses performances sur le problème MI-FAP.

Le reste de ce papier est organisé comme suit : la section 2 présente le problème MI-FAP. La section 3 introduit les approches hyper-heuristique proposées. La section 4 présente brièvement l'algorithme OBB, que nous adaptons pour le problème MI-FAP. Quelques résultats expérimentaux sont donnés dans la section 5. Enfin, la section 6 donne une conclusion et quelques perspectives inhérentes à ce travail.

2 Description et modélisation du problème

Le problème de minimisation des interférences lors de l'affectation des fréquences dans un réseau cellulaire peut être énoncé comme suit : considérons un réseau cellulaire N découpé en nc cellules : $C = \{C_1, C_2 \dots C_i, \dots C_{nc}\}$. Le réseau dispose d'un intervalle $[F_{min}, F_{max}]$ de fréquences opérationnelles et $NF = |F_{min} - F_{max}|$ le nombre de fréquences. Pour chaque cellule, on dispose d'un ensemble de fréquences opérationnelles. Les interférences entre les (*TRXs*) sont représentées par les deux matrices IM_{co} respectivement IM_{adj} définies comme suit :

- La matrice $IM_{co}[nc][nc]$: une matrice de rang $nc \times nc$ où l'élément $IM_{co}[i][j]$ exprime le niveau d'interférence co-channel dans le cas où des émetteurs/ récepteurs de cellules i et j utiliseraient une même fréquence (f).
- La matrice $IM_{adj}[nc][nc]$: une matrice de rang

$nc \times nc$ où l'élément $IM_{adj}[i][j]$ exprime le niveau d'interférence canal-adjacent dans le cas où des émetteurs/ récepteurs de cellules i et j utiliseraient des fréquences adjacentes (f), ($f + 1/f - 1$).

Une solution $S = \{f_1, f_2, f_i \dots f_d\}$ du problème MI-FAP est une affectation de fréquences aux d émetteurs / récepteurs du réseau qui satisfait les contraintes des cellules. Chaque élément f_i de S correspond à la fréquence assignée au i^{me} émetteur/ récepteur du réseau.

Le problème d'affectation de fréquences dans un réseau cellulaire consiste à trouver une allocation de fréquence à chaque TRX du réseau qui minimise les interférences. C'est un problème d'optimisation combinatoire dont la fonction objectif permet d'évaluer le niveau d'interférences généré par toute affectation S de fréquences aux différents ($TRXs$) du réseau. Cette fonction que nous cherchons à minimiser est exprimée par la formule suivante [15] :

$$F(S) = \sum_{i=1}^{nc} \sum_{j=1}^{nc} (Y_{ij} \times IM_{co}[i][j] + Z_{ij} \times IM_{adj}[i][j]) * w_i(1)$$

où :

- w_i : est un poids associé à chaque cellule. Il dépend du trafic de la cellule.
- Y_{ij} : est une variable de décision, qui prend 1 si des ($TRXs$) de cellules i et j utilisent la même fréquence dans S , 0 sinon.
- $IM_{co}[i][j]$: est un élément de la matrice IM_{co} qui exprime le niveau d'interférence provoqué par l'affectation dans S d'une même fréquence à des ($TRXs$) des cellules i et j .
- Z_{ij} : est une variable de décision, qui prend 1 si des ($TRXs$) de cellules i et j utilisent dans S des fréquences adjacentes, 0 sinon.
- $IM_{adj}[i][j]$: est un élément de la matrice IM_{adj} qui exprime le niveau d'interférence provoqué par l'affectation de fréquences adjacentes dans S à des ($TRXs$) des cellules i et j .

Si Sol représente l'ensemble des solutions, la valeur optimale de la fonction objectif est alors $F^* = MIN_{S \in Sol}(F(S))$ et la solution réalisant cette valeur optimale est notée S^* .

3 Des approches Hyper-heuristiques pour le MI-FAP

Les hyper-heuristiques sont des méthodologies de recherche travaillant à un haut niveau d'abstraction et d'indépendance par rapport au problème d'optimisation traité. Elles sont définies dans [4] comme "*heuristics to choose heuristics*", c'est-à-dire des heuristiques permettant de choisir d'autres heuristiques.

Les hyper-heuristiques sont des (méta) heuristiques de haut niveau qui manipulent un ensemble d'heuristiques de bas niveau. L'objectif est de gérer la sélection des heuristiques de bas niveau à appliquer parmi un ensemble d'heuristiques. La stratégie de sélection de l'heuristique de bas niveau est très importante pour l'hyper-heuristique. Elle doit avoir l'habilité de choisir l'heuristique la plus appropriée à chaque itération durant le processus de recherche de solution et assurer la bonne collaboration entre les différentes heuristiques manipulées.

Dans cet article, nous proposons trois approches fondées sur l'hyper-heuristique pour résoudre le problème MI-FAP. La principale différence entre ces trois approches se situe dans la stratégie utilisée pour la sélection de l'heuristique de bas niveau. Les approches hyper-heuristiques proposées dans ce travail manipulent cinq heuristiques de bas niveau. Nous présentons dans ce qui suit les cinq heuristiques de bas niveau que nous proposons pour le problème MI-FAP :

L'heuristique $h1$: $h1$ implémente une recherche locale de base et simple à mettre en oeuvre. C'est une procédure itérative qui démarre d'une solution initiale générée aléatoirement X . A chaque itération, une solution voisine est obtenue en appliquant un mouvement. Pour le problème MI-FAP, le mouvement consiste à modifier la fréquence d'un (TRX). Le choix du (TRX) se fait aléatoirement parmi l'ensemble des ($TRXs$). Le choix de la nouvelle fréquence à affecter se fait avec le respect des contraintes de cellules. Le processus est répété un nombre de fois avec pour objectif l'amélioration de la solution.

L'heuristique $h2$: $h2$ est une recherche locale stochastique. Elle prend en entrée la meilleure solution trouvée, puis génère un voisin en appliquant un mouvement. Comme pour l'heuristique $h1$ le mouvement consiste à modifier la fréquence d'un (TRX). Pour $h2$ le mouvement s'effectue en utilisant une probabilité fixe $wp > 0$. La probabilité $wp > 0$ intervient lors du choix du (TRX) dont la fréquence sera modifiée. Le choix du (TRX) se fait selon les deux critères suivants :

- Le premier critère consiste à sélectionner un

(*TRX*) parmi l'ensemble des (*TRXs*) de manière aléatoire avec une probabilité $wp > 0$. La règle du mouvement ici est similaire à celle appliquée dans *h1*.

- Le deuxième critère consiste à choisir, avec une probabilité $1 - wp$, un (*TRX*) appartenant à la cellule dont l'affectation des fréquences - à ses différents (*TRXs*) - a engendré le plus d'interférences. On remplace la fréquence actuelle du (*TRX*) par la fréquence la moins assignée aux (*TRXs*) des cellules proches géographiquement de la cellule trouvée.

L'heuristique *h3* : *h3* reprend le principe d'un opérateur de mutation qui selon un paramètre de mutation noté (*mrte*) appliquera ou non, pour chaque (*TRX*) d'une solution *X*, une perturbation. C'est à dire que l'affectation de fréquence actuelle de chaque (*TRX*) sera modifiée ou non selon le paramètre *mrte*. La solution obtenue est ensuite passée comme paramètre d'entrée (solution de départ) à une recherche locale pour l'améliorer. La recherche locale appliquée est implémentée par l'heuristique (*h1*).

L'heuristique *h4* : *h4* s'inspire de l'opérateur de croisement d'un algorithme génétique afin de créer une nouvelle solution. Pour la mettre en oeuvre nous proposons de combiner la meilleure solution avec la moins bonne solution obtenues à l'appel de cette heuristique. Ensuite on applique un opérateur de mutation sur la nouvelle solution obtenue. L'opérateur de mutation est celui proposé dans l'heuristique *h3*. Suite à la phase de mutation, nous appliquons la recherche local stochastique implémentée par l'heuristique *h2* afin d'améliorer la solution obtenue.

L'heuristique *h5* : *h5* s'inspire tout comme l'heuristique *h4* de l'opérateur de croisement d'un algorithme génétique dans le but de créer une nouvelle solution. Pour la mettre en oeuvre nous proposons de combiner la meilleure solution avec une nouvelle solution générée aléatoirement. Ensuite on applique l'heuristique *h3* à la solution obtenue. Cette solution est ensuite améliorée en utilisant la recherche locale stochastique implémentée par *h2*.

Nous sommes maintenant en mesure de décrire les différentes hyper-heuristiques que nous proposons pour résoudre le problème MI-FAP.

3.1 L'hyper-heuristique fonction de choix (HFC)

Dans un premier temps, nous proposons une approche hyper-heuristique basée sur une fonction de choix

pour sélectionner l'heuristique de bas niveau à appeler. Cette fonction de choix est une adaptation au problème MI-FAP de la fonction utilisée dans [6, 7] pour le problème de scheduling. Elle associe à chaque heuristique de bas niveau un score qui correspond à son efficacité. Ce score permettra de sélectionner l'heuristique la plus adéquate à chaque étape du processus de recherche.

Algorithm 1 Hyper-heuristique Fonction Choix (HFC)

Require: Une instance du MI-FAP, un ensemble d'heuristiques de bas niveau, la fonction de choix, les paramètres α , β et γ .

Ensure: meilleure solution S^* .

- 1: Générer une solution initiale S ;
 - 2: Evaluer la qualité $F(S)$ de la solution S donnée par la fonction F de l'équation (1);
 - 3: $S^* = S$; $F^* = F(S)$;
 - 4: **while** (le critère d'arrêt non vérifié) **do**
 - 5: Pour chaque heuristique h_i , calculer son score en utilisant la fonction de choix $score(h_i)$.
 - 6: Sélection de l'heuristique h_i ayant le score le plus élevé.
 - 7: Appliquer h_i à S afin d'obtenir une nouvelle solution S' de qualité $F(S')$.
 - 8: **if** $F(S') < F^*$ **then**
 - 9: $S^* = S'$.
 - 10: $F^* = F(S')$.
 - 11: **end if**
 - 12: **end while**
-

Afin d'attribuer un score à chaque heuristique de bas niveau, la fonction de choix se base sur les éléments suivants :

1. La performance récente de chaque heuristique (représentée par g_1).
2. La performance récente de paires d'heuristiques (représentée par g_2).
3. La récence du dernier appel de l'heuristique (représentée par g_3).

Ainsi nous avons :

$$\begin{aligned} \forall_i, g_1(h_i) &= \sum_n \alpha^{n-1} \frac{I_n(h_i)}{T_n(h_i)} \\ \forall_i, g_2(h_{RL}, h_i) &= \sum_n \beta^{n-1} \frac{I_n(h_{LL}, h_i)}{T_n(h_{RL}, h_i)} \\ \forall_i, g_3(h_i) &= elapsedTime(h_i) \\ \alpha, \beta &\in [0, 1], \gamma \in R \end{aligned}$$

Avec :

- h_i : la i^{me} heuristique et h_{RL} la dernière heuristique lancée.

- α , β et γ sont des poids reflétant l'importance de chaque terme. Ils ont été fixés empiriquement.
- Les valeurs I_n et T_n donnent l'information sur la qualité de la solution et le temps CPU de la $i^{ième}$ heuristique.

Le score de chaque heuristique de bas niveau est défini comme suit :

$$\forall_i, score(h_i) = \alpha g_1(h_i) + \beta g_2(h_{RL}, h_i) + \gamma g_3(h_i)$$

- g_1 est la performance récente de l'heuristique (h_i).
- g_2 est la performance récente de la paire d'heuristiques : l'heuristique (h_i) et celle la précédant (h_{RL}).
- g_3 mesure la récence du dernier appel de l'heuristique (h_i).

Les termes g_1 et g_2 sont les éléments qui permettent d'intensifier la recherche tandis que l'élément g_3 apporte de la diversification et cela en favorisant les heuristiques qui n'ont pas été choisies depuis un long moment.

L'hyper-heuristique basée sur cette fonction de choix est présentée dans l'algorithme 1.

3.2 L'hyper-heuristique aléatoire (HHA)

Algorithm 2 Hyper-Heuristique Aléatoire (HHA).

Require: Une instance du MI-FAP et un ensemble d'heuristiques de bas niveau.

Ensure: meilleure solution S^* .

- 1: générer une solution initiale S ;
 - 2: Evaluer la qualité $F(S)$ de la solution S donnée par la fonction objective de l'équation (1);
 - 3: $S^* = S$; $F^* = F(S)$;
 - 4: **while** (le critère d'arrêt non rencontré) **do**
 - 5: Sélection aléatoire de l'heuristique h_i .
 - 6: Appliquer h_i à S afin d'obtenir une nouvelle solution S' de qualité $F(S')$.
 - 7: **if** $F(S) < F^*$ **then**
 - 8: $S^* = S'$.
 - 9: $F^* = F(S')$.
 - 10: **end if**
 - 11: **end while**
-

La seconde hyper-heuristique que nous proposons pour le problème MI-FAP sélectionne l'heuristique à appeler de façon aléatoire, ce qui donne une chance à chaque heuristique d'être appelée. Comme il a été souvent mentionné, la force de l'aléatoire réside dans sa capacité à apporter de la diversité. L'hyper-heuristique utilisant cette sélection aléatoire est illustré dans l'algorithme 2.

3.3 L'hyper-heuristique stochastique (SHH)

La troisième hyper-heuristique que nous proposons combine les deux stratégies de sélections introduites précédemment : la sélection basée sur la fonction de choix et la sélection aléatoire de l'heuristique à appeler. L'idée est inspirée de la recherche locale stochastique proposée dans [3] pour résoudre le problème de détermination du gagnant dans une enchère combinatoire. A chaque étape, le processus de recherche sélectionne selon une probabilité fixe $wp_1 > 0$ une des deux méthodes de sélection suivantes :

- Sélectionner aléatoirement l'heuristique à appliquer avec une probabilité fixe $wp_1 > 0$.
- Sélectionner l'heuristique en utilisant la fonction de choix avec une probabilité $1 - wp_1$.

L'hyper-heuristique stochastique est illustrée par l'algorithme 3.

Algorithm 3 Stochastic Hyper-heuristique (SHH).

Require: Une instance du MI-FAP, un ensemble d'heuristiques, la probabilité wp_1 , la fonction de choix, α , β , γ .

Ensure: meilleure solution S^*

- 1: générer une solution initiale S ;
 - 2: Evaluer la qualité $F(S)$ de la solution S donnée par la fonction objective de l'équation (1);
 - 3: $S^* = S$; $F^* = F$; F , F^* qualité des solutions S respec S^* obtenues grâce à la formule (1).
 - 4: **while** (le critère d'arrêt non rencontré) **do**
 - 5: $r \leftarrow$ un nombre aléatoire entre 0 et 1;
 - 6: **if** ($r < wp_1$) **then**
 - 7: $h_i =$ Une heuristique choisie aléatoirement;
 - 8: **else**
 - 9: $h_i =$ Une heuristique choisie selon la fonction de choix;
 - 10: **end if**
 - 11: Appliquer h_i sur S afin d'obtenir une nouvelle solution S' de qualité F' .
 - 12: **if** $F' < F^*$ **then**
 - 13: $S^* = S'$.
 - 14: $F^* = F'$.
 - 15: **end if**
 - 16: **end while**
-

4 Adaptation de l'algorithme d'optimisation basée sur la Biogéographie pour le MI-FAP

Pour compléter notre étude nous voulons utiliser une méthode à population pour résoudre le problème MI-

FAP. Notre choix est naturellement porté sur la méthode basée sur la Biogéographie qui représente une des méthodes à population les plus récentes. Nous avons été curieux de tester ces performances sur le problème MI-FAP et voir comment elle se comporte en comparaison par rapport à l’approche hyper-heristique. L’optimisation basée sur la biogéographie (OBB) est une méta-heuristique évolutionnaire proposée par *Dan Simon* en 2008 [25]. La méthode s’inspire de la science de la Biogéographie. Cette dernière étudie la distribution des espèces dans leur environnement naturel. La méthode utilise les résultats obtenus par le modèle mathématique de la Biogéographie introduit en 1960 par Robert MacArthur et Edward Wilson. Ce modèle décrit le processus de migration (émigration et immigration) des espèces d’un habitat (île) vers un autre à la recherche de l’habitat comportant les meilleures conditions de vie. L’optimisation basée sur la biogéographie est une méthode qui manipule une population. Chaque individu de la population correspond à un habitat par analogie à la Biogéographie. La qualité de chaque habitat est donnée par *Habitat Suitability Index* (HSI) qui correspond à la qualité de la solution. Chaque habitat est caractérisé par l’ensemble de ses variables indépendantes appelées en anglais *Suitability Index Variable* (SIV), le nombre d’espèces, le taux d’immigration λ et le taux d’émigration μ . La méthode d’optimisation basée sur la Biogéographie est régie par deux processus : un processus de migration et un processus de mutation. La migration dans OBB est un processus probabiliste où les bonnes solutions partagent l’information avec les mauvaises solutions. Les bonnes solutions dans la méthode d’optimisation basée sur la Biogéographie sont celles ayant un bon HSI, un large nombre d’espèces, un taux d’émigration μ élevé et un faible taux d’immigration λ .

$$\lambda = I(1 - k/n) \quad (2)$$

$$\mu = E(k/n) \quad (3)$$

Avec E , I les taux d’émigration respectivement d’immigration maximales, k le nombre d’espèces présentes sur l’île et n le nombre d’espèces maximal que peut contenir une île.

Le processus de migration est un processus probabiliste où les taux d’immigration λ et d’émigration μ de chaque habitat sont utilisés afin de transmettre les caractéristiques entre les habitats. Durant ce processus les bonnes solutions partagent leurs caractéristiques avec les mauvaises solutions. Ainsi les mauvaises solutions sont améliorées en héritant des caractéristiques des bonnes solutions. Nous proposons dans la figure 1 un schéma qui illustre le processus de migration pour le problème MI-FAP. Considérons les deux solutions

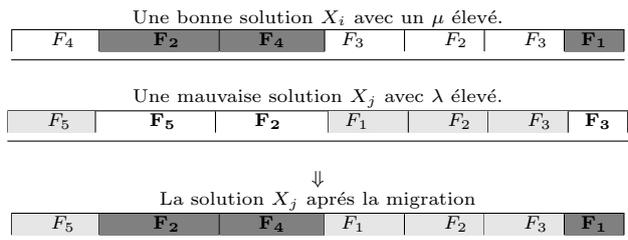


Figure 1: **Schéma illustrant le processus de migration pour le MI-FAP**

X_i et X_j qui représentent des plans de fréquences. Tel que X_i est une bonne solution et X_j une mauvaise solution. A partir de la figure 1, nous pouvons observer qu’au cours du processus de migration, les fréquences assignées aux deuxième, troisième et sixième *TRXs* dans la solution X_j ont été modifiées. Les nouvelles valeurs de ces fréquences sont celles héritées de la bonne solution X_i .

Nous détaillons ci-après l’adaptation de la méthode (OBB) que nous proposons pour le problème MI-FAP. La population initiale est générée aléatoirement. Chaque habitat de la population correspond à un plan de fréquence. Nous définissons *PopSize* comme le nombre d’individus de la population. Ensuite, l’algorithme applique à chaque fois les étapes suivantes : évaluer la qualité de chaque solution en calculant son *HSI*, pour chaque solution mapper le nombre d’espèces à partir du *HSI*, calculer le taux d’émigration μ et le taux d’immigration λ . Suite à ces opérations, certaines solutions sont choisies pour le processus de migration. Pour rappel cette étape dépend des taux μ et λ . L’algorithme se poursuit avec le processus de mutation. La mutation est un processus qui permet d’apporter de la diversité et éviter la convergence rapide de l’algorithme, il est régi par un paramètre de mutation fixe noté (P_m). Nous proposons d’appliquer la mutation sur la partie la moins bonne de la population. Afin de conserver la meilleure solution dans la population manipulée par (OBB), une stratégie d’élitisme est implémentée. A chaque génération (G), la stratégie d’élitisme que nous adoptons permettra de conserver (n_{elit}) solutions de la génération précédente ($G - 1$) si aucun autre individu n’est meilleur qu’eux.

L’algorithme 4 présente le principe de fonctionnement de la méthode OBB pour le problème MI-FAP.

Algorithm 4 La méthode d’optimisation basée sur la Biogéographie pour le MI-FAP

Require: Une instance du MI-FAP, HSI

Ensure: La meilleure solution trouvée S

- 1: Génération aléatoire de la population initiale
 - 2: **while** (le critère d’arrêt non rencontré) **do**
 - 3: Evaluer la qualité de chaque solution.
 - 4: Calculer le nombre d’espèces.
 - 5: Calculer les taux μ et λ .
 - 6: Appliquer la migration.
 - 7: Appliquer la mutation.
 - 8: **end while**
-

5 Résultats expérimentaux

Dans le but de valider nos approches hyper-heuristiques et la méthode d’optimisation basée sur la Biogéographie, des tests ont été effectués sur différentes instances du problèmes MI-FAP. Dans nos expériences, nous avons considéré le benchmark du projet de Coopération européenne dans le domaine de la recherche Scientifique et Technique Action 259 (COST 259). Nous avons travaillé sur les données produites par Siemens AG. Les données sont disponibles dans [10]. Nos algorithmes ont été implémentés en utilisant le langage java sous window 7 et sous la machine intel core i5 4GB de RAM. Nous donnons ci-dessous quelques caractéristiques des instances testées.

- Siemens 01 : 506 cellules, 930 $TRXs$, 74 fréquences, 20417 interférences co-channel, 10344 interférences canal adjacent.
- Siemens 02 : 254 cellules, 977 $TRXs$, 82 fréquences, 30982 interférences co-channel, 13970 interférences canal adjacent.
- Siemens 03 : 894 cellules, 1623 $TRXs$, 55 fréquences, 63893 interférences co-channel, 25105 interférences canal adjacent.
- Siemens 04 : 760 cellules, 2785 $TRXs$, 38 fréquences, 113658 interférences co-channel, 64341 interférences canal adjacent.

Les paramètres utilisés dans notre étude expérimentale ont été obtenus après une étude empirique, leurs valeurs sont donnés ci-dessous :

- Les paramètres de l’algorithme de OBB : $PopSize$ 150, $n_{elit}=2$, $P_m=0.1$.
- Les paramètres de l’hyper-heuristique basée sur la fonction de choix : $\alpha=0.9$, $\beta=0.1$, et $\gamma=1.5$
- Paramètre de l’hyper-heuristique stochastique : $wp_1=0.27$.

- Le nombre d’itération pour l’heuristique $h1$: $Number_{Iter}=50$.
- Le nombre d’itération et la probabilité utilisée dans l’heuristique $h2$: $Nb_{Iter}=50$, $wp=0.25$.
- Le paramètre de mutation dans $h3$: $mrates=0.05$.
- Le nombre d’itération des recherches locale appliquées dans les heuristiques $h3$, $h4$, $h5$ est fixé à : 100.

5.1 Les résultats obtenus

Du fait de la nature non déterministe des algorithmes que nous proposons, des séries d’expériences de 10 tests ont été effectuées pour les différents benchmarks. Le critère d’arrêt des algorithmes est défini en fonction du temps CPU. Ce paramètre est fixé à 1800 secondes. Les résultats présentés dans les tableaux 1 à 4 correspondent aux valeurs moyennes (Mean), la meilleure valeur (Best), la valeur la moins bonne (worst) de la fonction objectif, obtenus sur 10 exécutions pour chacun des algorithmes testés. La valeur (std) correspond à l’écart type entre les différentes exécutions. Les résultats en gras indiquent les meilleurs résultats obtenus en termes de meilleure valeur (Best) et valeur moyenne (Mean). Les résultats présentés dans les tableaux 1 à 4 montrent clairement que les méthodes basée sur l’Hyper-heuristique donnent de meilleurs résultats que OBB.

Statistical Features	HFC Cost	HHA Cost	SHH Cost	OBB Cost
Best	47.70	21.27	42.79	223.62
Mean	88.78	41.60	49.04	250.18
Worst	253.54	98.96	58.95	270.13
Std	60.52	22.93	5.26	13.22

Table 1: Les résultats obtenus sur Siemens 01

Le tableau 1 présente les résultats obtenus par les différentes méthodes sur l’instance Siemens 01. Une colonne est réservée pour présenter les résultats de chaque méthode. Nous pouvons voir que les trois hyper-heuristiques HFC, HHA et SHH surpassent largement la méthode OBB sur cette instance en terme de meilleure (best), moyenne (Mean) et mauvaise solutions trouvées. Nous pouvons constater aussi que l’hyper-heuristique aléatoire (HHA) et l’hyper-heuristique stochastique (SHH) donnent de meilleurs résultats en terme de meilleur, moyenne et moins bonne solutions que l’hyper-heuristique basée sur une fonction de choix. Les résultats entre les deux méthodes HHA et SHH sont comparables en termes de valeur moyenne, HHA est meilleur en terme

de meilleur valeur mais moins bonne en terme moins bonne valeur.

Statistical Features	HFC Cost	HHA Cost	SHH Cost	OBB Cost
Best	4990.70	5187.72	4199.66	5941.29
Mean	5815.79	5422.59	4672.32	6426.40
Worst	6180.81	5672.4	4941.44	7068.98
Std	376.09	137.31	237.64	372.10

Table 2: Les résultats obtenus sur Siemens 02

Le tableau 2 présente les résultats obtenus sur l’instance Siemens 02. On peut voir que les résultats obtenus par l’hyper-heuristique stochastique (SHH) sont meilleurs que ceux de toutes les autres méthodes. Les deux hyper-heuristiques HHA et HFC sont légèrement meilleurs que la méthode OBB. Les résultats obtenus par HHA et HFC sont comparables voir HFC est légèrement meilleure. En effet, HFC est meilleure que HHA en terme de meilleure solution et HHA est légèrement meilleure que HFC en terme de solution moyenne.

Statistical Features	HFC Cost	HHA Cost	SHH Cost	OBB Cost
Best	923,12	884.64	702.48	1460,65
Mean	1409,34	1004.09	787.99	1544,94
Worst	1816.39	1196.19	944.16	1771,72
Std	268.58	90.42	76.63	94.79

Table 3: Les résultats obtenus sur Siemens 03

Le tableau 3 présente les résultats obtenus par les différentes méthodes sur l’instance Siemens 03. Ici aussi, on constate que l’hyper-heuristique SHH présente des résultats meilleurs que ceux de toutes les autres méthodes. Les résultats de HHA sont légèrement meilleurs que ceux de HFC qui est elle-même meilleure que la méthode OBB.

Statistical Features	HFC Cost	HHA Cost	SHH Cost	BBO Cost
Best	108232.4	76968.78	87723.83	110021,78
Mean	110959.6	89047.31	91096.05	111314.60
Worst	114512.9	94984.77	99928.78	112774,69
Std	2516.33	5191.25	3635.05	882.01

Table 4: Les résultats obtenus sur Siemens 04

Le tableau 4 contient les résultats obtenus par les différentes méthodes pour l’instance Siemens 04. On peut voir que les méthodes HHA et SHH surpassent les deux autres méthodes HFC et OBB. Les résultats de HHA et SHH sont comparables avec une légère avance pour la méthode aléatoire. Nous pouvons remarquer aussi que les résultats entre OBB et l’hyper-heuristique basée sur une fonction de choix sont comparables.

Nous remarquons que l’écart type obtenu pour les différents tests est relativement petit.

Les résultats obtenus et résumés dans les quatre tableaux précédents montrent clairement l’efficacité et la pertinence de l’approche hyper-heuristique, comparé à la méthode d’optimisation basée sur la Biogéographie. Ceci peut être expliqué par le fait que la méthode d’optimisation basée sur la Biogéographie converge rapidement et stagne, malgré le processus de mutation qui semble être insuffisant pour permettre à l’algorithme de s’échapper des optimums locaux. Les résultats obtenus par l’hyper-heuristique stochastique et l’hyper-heuristique à sélection aléatoire sont meilleures comparées à ceux obtenus par l’hyper-heuristique basée sur une fonction de choix. En effet, cette dernière souffre du manque de diversité, et stagne souvent dans un optimum local. Quant à l’hyper-heuristique aléatoire de part sa stratégie de sélection aléatoire donne une chance à chaque heuristique d’améliorer la solution, ce qui assure une grande diversité. L’efficacité de l’approche hyper-heuristique stochastique s’explique par le bon équilibre entre diversité et intensification. La sélection aléatoire de l’heuristique apporte l’élément de diversification alors que la sélection selon la fonction de choix permet d’intensifier la recherche. Il est évident que l’hyper-heuristique basée sur une fonction de choix reste sensible aux choix des paramètres utilisés. Nous restons convaincus qu’une étude plus approfondie sur les valeurs et l’influence des paramètres utilisés garantira de bien meilleurs résultats. En conséquence, nous pensons qu’avec de bons réglages sur ces paramètres l’hyper-heuristique stochastique finira par l’emporter largement devant l’hyper-heuristique aléatoire. L’hyper-heuristique stochastique semble plus stable et plus robuste.

Instances	HFC Cost	HHA Cost	SHH Cost	OBB Cost	SA-DP Cost
Siemens 01	12.559	7,33	12,04	47.019	2.20
Siemens 02	51.148	54,39	45,33	65.093	14.27
Siemens 03	81.502	90,95	69	130.387	4.73
Siemens 04	442.302	347,59	370,22	466.566	77.25

Table 5: Comparaison de nos méthodes avec SA-DP

Le problème MI-FAP a été largement étudié sur différents benchmarks autres que ceux que nous avons testés [14, 16]. Nous présentons dans le tableau 5 une comparaison de nos méthodes avec la méthode du recuit simulé combiné à la programmation dynamique [10]. Pour réaliser cette comparaison, nous avons apporté une légère modification à la fonction objectif présentée dans la formule (1). Nous ne prenons plus en compte le poids w_i associé à chaque cellule. La valeur de la fonction objectif correspond alors à la somme des interférences Co-canal et canal adjacent. Ce qui explique la différence entre les résultats présentés dans

le tableau 5 et les résultats présentés dans les tableaux 1-4. Les différentes valeurs reportées dans le tableau 5 correspondent à la valeur moyenne de la fonction objectif obtenue sur dix exécutions. Nos méthodes sont lancées pour un temps CPU fixé à 1800 secondes. La colonne (SA-DP) correspond aux résultats obtenus par la méthode du recuit simulé combiné à la programmation dynamique. Les résultats de la méthode SA-DP ont intégralement été récupérés de [10]. Cette méthode est celle ayant donnée les meilleurs résultats dans la littérature. Néanmoins, nous tenons à signaler qu'il a été reporté dans [10, 23] que ces bons résultats ont été trouvés par la méthode SA-DP après un temps de calcul très important parfois mesuré en semaines. D'autres résultats, moins bons que ceux obtenus par la méthode (SA-DP), sont disponibles sur [10]. On peut trouver aussi dans [22, 23] des résultats obtenus par méthode tabou. Ces derniers restent moins bons que ceux obtenus par (SA-DP).

6 Conclusion

Le travail présenté dans ce papier avait pour objectif d'une part, d'introduire trois approches basées sur l'approche hyper-heuristique pour résoudre le problème d'affectation de fréquence dans un réseau cellulaire. Nous avons d'autre part adapté l'algorithme d'optimisation basée sur la Biogéographie pour le problème MI-FAP. Nous avons présenté, dans un premier temps trois variantes d'hyper-heuristique : une hyper-heuristique basée sur une fonction de choix, une hyper-heuristique à sélection aléatoire et une hyper-heuristique stochastique. Les hyper-heuristiques proposées manipulent un ensemble d'heuristiques de bas niveaux. La différence entre les hyper-heuristiques développées réside dans le mécanisme de sélection de l'heuristique à appeler. Nous avons implémenté puis testé nos méthodes sur différentes instances du problème d'affectation de fréquences. Nous avons pu observer l'efficacité des approches hyper-heuristiques comparées à OBB. Nous avons ensuite comparé nos méthodes à la méthode (SA-DP). Les résultats expérimentaux montrent que les performances des algorithmes proposés restent faibles par rapport à ceux de la méthode SA-DP obtenu après des semaines de temps de calcul. Les résultats que nous avons présentés dans ce papier sont encore en progrès, une analyse expérimentale plus poussée reste à faire. Les heuristiques de bas niveau, les paramètres des algorithmes proposés peuvent être mieux travaillés. Une comparaison avec d'autres algorithmes est planifiée. Ainsi que des tests sur d'autres instances pour lesquels les temps de calculs sont connus est en cours de réalisation.

References

- [1] K.I. Aardal, S.P.M. van Hoesel, A.M.C.A. Koster, , C. Mannino, A. Sassano, "Models and solution techniques for frequency assignment problems", in *Annals of Operations Research* 153(1), pp : 79-129, 2007.
- [2] J. Alami and El Imrani, "Using Cultural Algorithm for the Fixed-Spectrum Frequency Assignment Problem", in *the Journal of Mobile Communication* 2 (1), pp : 1-9 , 2008.
- [3] D. Boughaci, B. Benhamou, H. Drias, "A memetic algorithm for the optimal winner determination problem", in *Soft Comput*, 13, 905-917, 2009.
- [4] E. K. Burke and G. Kenda, "Search Methodologies : Introduction tutorials in optimization and decision support techniques.", Springer, 2005.
- [5] D. Castelino , S. Hurley , And N. Stephens, "A Tabu Search Algorithm for frequency assignment", in *Annals of Operations Research* 63, pp : 301-319, 1996.
- [6] P. Cowling, G. Kendall, and E. Soubeiga. Hyperheuristics : A robust optimisation method applied to nurse scheduling. In *Parallel Problem Solving from Nature -PPSN VII : 7th International Conference*, volume Volume 2439/2002 of *Lecture Notes in Computer Science*, pages 851860, Granada, Spain, 2002. Springer Berlin/ Heidelberg.
- [7] P. Cowling, G. Kendall, and E. Soubeiga. Hyperheuristics : A tool for rapid prototyping in scheduling and optimisation. In *Proceedings of the second workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP2002)*, volume Volume 2279/2002 of *Lecture Notes in Computer Science* . Springer LNCS, 2002. ISSN0302-9743 (Print) 1611-3349 (Online).
- [8] R. Dorne, J.K. Hao, "Tabu search for graph coloring, T-colorings and set T-colorings", In S. Voss, S. Martello, I.H. Osman, C. Roucairol (Eds.), *Meta-heuristics : Advances and Trends in Local Search Paradigms for Optimization*. Chapter 6, pp : 77-92, Kluwer.
- [9] M. Duque-Anton, D. Kunz, And B. Ruber, "Channel Assignment for Cellular Radio Using Simulated Annealing", in *the IEEE Transactions on Vehicular Technology* 42,1, 1993.
- [10] A. Eisenbltter and A. Koster, COST 259. *Wireless Flexible Personalized Communications*. Available

online at [http : //fap.zib.de/problems/COST259/](http://fap.zib.de/problems/COST259/).

- [11] A. Eisenblatter, M. Grotschel, A.M .C.A. Koster, " Frequency planning and ramifications of coloring", in *Discussions Mathematicae Graph Theory 22 (1)*, pp : 51-88, 2002.
- [12] M. Fschetti, C. Lepshy, G. Minerva, G.R. Jacur, AND E. Toto, " Frequency assignment in mobile radio systems using branch-and-cut techniques", in *Tech.rep.*, Dipartimento di matematica e informatica, Universit di Udine, Italy, 1997.
- [13] W.K. Hale, "Frequency assignment : Theory and applications", in *Proceedings of the IEEE 68(12)*, pp : 1497-1514, 1980.
- [14] J.K. Hao, R. Dorne, P. Galinier, "Tabu search for frequency assignment in mobile radio networks ", *Journal of Heuristics* 4(1), 47-62, 1998.
- [15] A. Kuurne, "Mobile Measurement based Frequency Planning in GSM Network. *PhD Thesis*", Helsinki University of Technology, 2001.
- [16] X. Lai, J.K. Hao, "Path Relinking for the Fixed Spectrum Frequency", *Expert Systems with Applications*, 42(10) : 4755-4767, 2015.
- [17] R. Leese, Hurley, S, "Methods and Algorithms for Radio Channel Assignment", in *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2002.
- [18] F. Luna, C. Blum, E. Alba, A.J. Nebro, "ACO vs EAs for solving a real-world frequency assignment problem in GSM networks". in *the Proceedings of the genetic and Evolutionary Computation Conference (GECCO 2007)*, pp : 94-101, 2007.
- [19] F. Luna, C. Estebanez, L. Coromoto, J.M Chaves-Gonzalez, A.J. Nebro, R. Aler, C. Segura, M.A. Vega-Rodriguez, E. Alba, J.M. Valls, Miranda, G, Juan Antonio Gomez Pulido, "Optimization algorithms for large-scale real-world instances of the frequency assignment problem", it in *Soft Computing*. 15(5),pp : 975-990, 2011.
- [20] H. Mabed, A. Caminada, J.K. Hao, "Genetic tabu search for robust fixed channel assignment under dynamic traffic data", in *Coput Optim Appl* 50, pp : 483-506 , 2011.
- [21] B.H. Metzger, "Spectrum management technique", in *the 38th National ORSA Meeting, Detroit*, pp : 1497-1514, 1970.
- [22] R. Montemanni, JNJ. Moon, D.H. Smith, "An Improved Tabu Search Algorithm for the fixed Spectrum Frequency Assignment Problem", in *the IEEE Transactions on Vehicular Technology*,, vol 52,issue 4, pp : 891- 901, 2003.
- [23] R. Montemanni, D.H. Smith, "Heuristic manipulation, tabu search and frequency assignment.", *Computers and Operations Research*, vol 37,issue 3, pp : 543-551, 2010.
- [24] C. Segura, G. Miranda, C. Leon, "Parallel hyper-heuristics for the frequency assignment problem", in *Memetic Computing* 3, pp : 33-49 , 2011.
- [25] D. Simon, "Biogeography Based Optimization", in the *IEEE Transactions on Evolutionary Computation*,vol 12, pp : 702 - 713, 2008.