Aix-Marseille Université
M2 IAAA
Modélisation et résolutions pour l'optimisation
Projet

Emma Mendizabal, Joey Skaf et Zélie Vernin

12 novembre 2023

1 Introduction

Dans ce projet, on s'intéresse aux problèmes d'allocation de fréquences entre des stations. On dispose de n stations réparties dans k régions distinctes. Chaque station est composée d'un émetteur et d'un récepteur et peut exploiter certaines fréquences. Pour pouvoir communiquer avec d'autres stations, chaque station doit disposer de deux fréquences : une pour son émetteur et une pour son récepteur. Pour des raisons matérielles, l'écart entre les deux fréquences de la station i doit être égal à δ_i. Deux stations différentes peuvent donc avoir le même écart comme des écarts différents. Si deux stations sont proches l'une de l'autre, les fréquences utilisées par ces stations doivent être suffisamment espacées pour éviter les interférences. On notera Δ_{ij} l'écart minimum à garantir entre les fréquences des stations i et j. Enfin, pour chaque région, on souhaite limiter le nombre de fréquences différentes utilisées. On notera n_i le nombre maximum de fréquences différentes utilisées pour la région i.

2 Problème d'optimisation sous contraintes

2.1 Minimiser le nombre de fréquences utilisées

2.1.1 Modélisation sous forme d'un COP

Variables :

Pour chaque station i, nous avons deux variables, E_i et R_i, qui représentent respectivement les fréquences émettrices et réceptrices de la station i.

Domaines :

Chaque variable E_i et R_i peut prendre les valeurs dans le domaine des fréquences utilisables. On a alors :

$$\forall i \ E_i, F_i \in \mathbb{N}$$

Contraintes :

1. Contrainte d'écart de fréquences au sein d'une même station

   Pour chaque paire de fréquences (émettrice et réceptrice), on souhaite que leur écart soit égal à δ_i. Soit :

$$\forall i \ |E_i - R_i| = \delta_i$$

2. Contrainte d'écart minimal :

   Pour chaque paire de stations (i, j), où $i \neq j$, on souhaite que l'écart entre les fréquences de ces deux stations soit supérieur ou égal à Δ_{ij}.

   Cela peut être formulé comme suit pour l'émetteur (la même règle s'applique au récepteur) :

Pour chaque paire (i, j), $i \neq j$:

$$\forall i \neq j : |E_i - R_j| \geq \Delta_{ij}, |E_i - E_j| \geq \Delta_{ij}, |R_i - R_j| \geq \Delta_{ij}$$

3. Contrainte de limitation du nombre de fréquences par région :

   Pour chaque région i, le nombre de fréquences différentes utilisées ne doit pas dépasser n_i. Cela peut être formulé comme suit :

   Pour chaque région i :

$$\forall i : Card(\{E_j, R_j | E_j, R_j \in \Omega_{ij}\}) \leq n_i$$

   où Ω_{ij} est l'ensemble des stations j de la région i .

Objectif :

L'objectif est de minimiser le nombre total de fréquences utilisées. Soit :

$$C = \min\{E_i, R_i | i = 1, ...n\}$$

Finalement, on peut utiliser un solveur de CSP pour résoudre ce problème en minimisant le nombre de fréquences utilisées tout en respectant les contraintes d'écart minimal et de limitation du nombre de fréquences par région.

2.1.2 Production des instances

On utilise le format XSCP3 pour produire les instances. On commence par charger les données (on prendra par exemple le fichier 'celar_500_30_20_5_0.870000_48.json').

On crée une fonction fqc_par_region prenant en entrée deux listes de fréquences - une d'émettrices et une de réceptrices - ainsi qu'une liste de stations et une liste de régions. Cette fonction renvoie la liste des fréquences utilisées par régions.

On crée les variables de notre modélisation : une liste de fréquences émettrices (E_fq) et une liste de fréquences réceptrices (R_fq) ainsi que le nombre de stations (nb_stations).

On implémente ensuite les contraintes décrites ci-dessus et on écrit la fonction à optimiser, soit ici le nombre de fréquences utilisées.

Les deux solveurs nous renvoient alors chacun deux listes de fréquence ; une liste de fréquences émettrices et une liste de fréquences réceptrices. Ces fréquences respectent donc toutes les contraintes énumérées tout en minimisant le nombre de fréquences utilisées.

Le k^e élément correspond à la fréquence de la k^e station.

2.1.3 Efficacité des différents solveurs

Solveur ACE :
Le temps mis par le solveur pour résoudre ce problème sous contraintes est de 5.8s

Solveur CHOCO :
Le temps mis par le solveur pour résoudre ce problème sous contraintes est de 12min4.2s

On en conclut, que, dans notre cas et pour ce problème de minimisation de nombre de fréquences utilisées, le solveur ACE est bien plus efficace. En effet, les deux solveurs nous proposent une solution optimale mais le solveur ACE est beaucoup plus rapide que le solveur CHOCO.

2.2 Utiliser les fréquences les plus basses possibles

2.2.1 Modélisation sous forme d'un COP

Les variables, domaines et contraintes sont les mêmes que ceux définis dans le problème précédent. Seul la fonction à minimiser varie :

Objectif :

L'objectif est de minimiser la fréquence maximale utilisée parmi toutes les stations, ce qui revient à minimiser la plus haute des fréquences utilisées. On peut reformuler ce problème de la manière suivante :

$$\min\ \max(E_i, R_i)$$

2.2.2 Production des instances

On peut utiliser un solveur de CSP pour résoudre ce problème en maximisant la fréquence minimale tout en respectant les contraintes d'écart minimal et de limitation du nombre de fréquences par région.

2.2.3 Efficacité des différents solveurs

Solveur ACE :
Le temps mis par le solveur pour résoudre ce problème sous contraintes est de 5.8s

Solveur CHOCO :
Le temps mis par le solveur pour résoudre ce problème sous contraintes est de 140min40.5s

De la même manière que pour le problème d'optimisation précédent,on en conclut, que, pour trouver les fréquences les plus basses possibles, le solveur ACE est bien plus efficace. En effet, les deux solveurs nous proposent une solution optimale mais le solveur ACE est beaucoup plus rapide que le solveur CHOCO.

2.3 Minimiser la largeur de la bande de fréquences utilisées

2.3.1 Modélisation sous forme d'un COP

Les variables, domaines et contraintes sont les mêmes que ceux définis dans le problème précédent. Seul la fonction à minimiser varie :

Objectif :

L'objectif est de minimiser la largeur de la bande de fréquences totale utilisée par toutes les stations. Vous pouvez formuler cela comme suit :

$$C = \min(\max(E_i, R_i) - \min(E_i, R_i))$$

Cela signifie que l'on cherche à minimiser la différence entre la fréquence maximale et la fréquence minimale utilisée parmi toutes les stations. Cela revient à minimiser la largeur totale de la bande de fréquences utilisée.

Finalement, on peut utiliser un solveur de CSP pour résoudre ce problème en minimisant la largeur de bande totale tout en respectant les contraintes d'écart minimal et de limitation du nombre de fréquences par région.

2.3.2 Production des instances

Le procédé reste le même que pour les deux autres problèmes de minimisation. On prend garde à ne bien lancer que la cellule correspondante.

On obtient alors également deux listes de fréquences pour chaque station.

2.3.3 Efficacité des différents solveurs

Solveur ACE :
Le temps mis par le solveur pour résoudre ce problème sous contraintes est de 29.8s

Solveur CHOCO :
Le temps mis par le solveur pour résoudre ce problème sous contraintes est de 112min45.7s

De la même manière que pour les problèmes d'optimisation précédents, on en conclut, que, pour trouver les fréquences les plus basses possibles, le solveur ACE est bien plus efficace. En effet, les deux solveurs nous proposent une solution optimale mais le solveur ACE est beaucoup plus rapide que le solveur CHOCO.

3 Problèmes de satisfaction des contraintes valuées

3.0.1 Structure de valuation

La structure de valuation du fichier est une structure additive et peut donc être décrite comme suit :

$$V = \langle \bar{\mathbb{N}}, +, <, 0, \infty \rangle$$

Choix des poids des contraintes

Nous avons choisi de définir les contraintes sur l'écart de fréquences au sein d'une même station comme étant une contrainte dure. En effet il semble que d'après l'énoncé, pour des raisons matérielles, la contrainte d'écart des fréquences entre une même station est primordiale. De plus, on fixe aussi les fréquences car elles ne peuvent pas juste prendre n'importe quelles valeurs. Cette contrainte est une contrainte dure. On définit finalement la contrainte qui consiste à dire que deux fréquences doivent être séparés de Δ comme étant une contrainte molle. En effet, cela peut être gênant s'il y a des interférences entre deux stations mais pas forcément rédhibitoire.

3.0.2 Production des instances : Écriture du fichier au format WCSP

Nous avons réalisé le script main_WSCP qui crée un fichier en format WSCP à partir de nos données et des contraintes du problème.

Un fichier WCSP est une représentation textuelle de ces problèmes, et il suit une structure standardisée. Le fichier WCSP offre une représentation formelle et structurée du problème d'optimisation. Il spécifie les variables, les domaines, les contraintes et les coûts associés d'une manière qui peut être interprétée par des solveurs de problèmes WCSP.

On utilise ici le format WCSP pour représenter notre problème d'optimisation. Dans le contexte des WCSP, chaque contrainte entre les variables est associée à un coût, et l'objectif est alors de minimiser la somme totale des coûts.

3.0.3 Résolution des instances :

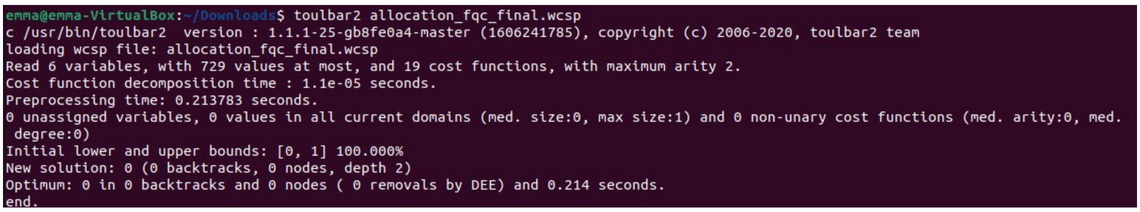
Nous avons lancé une machine virtuelle sur Ubuntu pour pouvoir utiliser l'outil Toolbar. Malheureusemetn nous n'avons pas obtenu de solution.

Nous avons tout d'abord essayé de produire un fichier WSCP où les contraintes ont été écrites en intension ainsi qu'en extension. Au vu du problème rencontré nous avons alors essayé d'écrire le fichier WSCP avec toutes les contraintes écrites en extension car il semblait que le problème venait de là quand nous testions sur d'autres exemples.

Cependant cela n'a pas résolu le problème et nous obtenons toujours 0 comme seules valeurs de fréquences pour les stations.

Nous pensons donc qu'il y a une erreur dans l'écriture des contraintes au format WSCP mais n'avons pas réussi à la détecter.

Voici la capture d'écran sur le WSCP final :


```
emma@emma-VirtualBox:~/Downloads$ toulbar2 allocation_fqc_final.wcsp
c /usr/bin/toulbar2  version : 1.1.1-25-gb8fe0a4-master (1606241785), copyright (c) 2006-2020, toulbar2 team
loading wcsp file: allocation_fqc_final.wcsp
Read 6 variables, with 729 values at most, and 19 cost functions, with maximum arity 2.
Cost function decomposition time : 1.1e-05 seconds.
Preprocessing time: 0.213783 seconds.
0 unassigned variables, 0 values in all current domains (med. size:0, max size:1) and 0 non-unary cost functions (med. arity:0, med.
 degree:0)
Initial lower and upper bounds: [0, 1] 100.000%
New solution: 0 (0 backtracks, 0 nodes, depth 2)
Optimum: 0 in 0 backtracks and 0 nodes ( 0 removals by DEE) and 0.214 seconds.
end.
```

Pourtant lorsque nous appliquons ce solveur sur le fichier example.wscp, il fonctionne bien :

```
emma@emma-VirtualBox:~/Downloads$ toulbar2 example.wcsp
c /usr/bin/toulbar2  version : 1.1.1-25-gb8fe0a4-master (1606241785), copyright (c) 2006-2020, toulbar2 team
loading wcsp file: example.wcsp
Read 25 variables, with 5 values at most, and 63 cost functions, with maximum arity 2.
Cost function decomposition time : 1e-05 seconds.
Reverse DAC dual bound: 20 (+10.000%)
Preprocessing time: 0.001 seconds.
24 unassigned variables, 116 values in all current domains (med. size:5, max size:5) and 62 non-unary cost functions (med. arity:2,
med. degree:5)
Initial lower and upper bounds: [20, 64] 68.750%
New solution: 28 (0 backtracks, 6 nodes, depth 8)
New solution: 27 (5 backtracks, 15 nodes, depth 5)
Optimality gap: [21, 27] 22.222 % (8 backtracks, 18 nodes)
Optimality gap: [22, 27] 18.519 % (21 backtracks, 55 nodes)
Optimality gap: [23, 27] 14.815 % (49 backtracks, 122 nodes)
Optimality gap: [24, 27] 11.111 % (63 backtracks, 153 nodes)
Optimality gap: [25, 27] 7.407 % (81 backtracks, 217 nodes)
Optimality gap: [27, 27] 0.000 % (89 backtracks, 240 nodes)
Node redundancy during HBFS: 25.417 %
Optimum: 27 in 89 backtracks and 240 nodes ( 460 removals by DEE) and 0.005 seconds.
end.
```

4 Conclusion

En conclusion, l'étude approfondie du problème d'allocation de fréquences entre stations a révélé la complexité inhérente à cette tâche cruciale dans la mise en place de réseaux de communication efficaces. La nécessité de respecter des contraintes strictes, telles que les écarts entre fréquences et les limites régionales, a été clairement identifiée comme des aspects incontournables pour assurer le bon fonctionnement du réseau.

Le modèle de programmation par contraintes (CSP) formulé dans ce rapport offre une représentation précise et formelle du problème, permettant l'utilisation de solveurs dédiés pour explorer l'espace des solutions. Les contraintes "dures", essentielles à la viabilité du système, ont été clairement définies, tandis que les contraintes "molles" ont été introduites pour optimiser la performance globale en tenant compte de critères tels que la minimisation du nombre total de fréquences.

Les résultats obtenus à l'aide de solveurs CSP tels que Choco ou ACE peuvent fournir des solutions efficaces en respectant les contraintes essentielles. Cependant, l'équilibre entre ces contraintes et les objectifs d'optimisation doit être soigneusement évalué pour garantir une allocation de fréquences à la fois robuste et efficiente.

En fin de compte, ce rapport fournit une base solide pour la modélisation et la résolution du problème d'allocation de fréquences, tout en soulignant l'importance d'une approche équilibrée entre contraintes rigides et objectifs d'optimisation pour atteindre une solution optimale dans des environnements dynamiques et exigeants.