

Problème: Couplage Maximum dans les Réseaux Sans Interférence

Groupe : *Prénom Nom de chaque membre*

December 11, 2025

1 Introduction

L'objectif de ce projet est de modéliser et résoudre un problème d'affectation de connexions dans un réseau sans fil en utilisant la programmation linéaire en nombres entiers (PLNE). Chaque utilisateur peut être connecté à un point d'accès (AP) unique, tout en respectant les contraintes de capacité, d'exclusivité et d'interférence entre les APs.

Ce modèle s'inscrit dans un contexte réaliste : il permet aux chercheurs en informatique et télécommunications d'optimiser la qualité de service, la distribution des charges, et la consommation énergétique dans des réseaux IoT ou WiFi denses.

2 Modélisation mathématique

2.1 Notations et paramètres

- \mathcal{U} : ensemble des utilisateurs
- \mathcal{A} : ensemble des points d'accès (AP)
- $d_{u,a}$: distance entre l'utilisateur u et l'AP a
- D_{\max} : portée maximale admissible d'un point d'accès
- $\mathcal{E} = \{(u, a) \in \mathcal{U} \times \mathcal{A} \mid d_{u,a} \leq D_{\max}\}$: ensemble des connexions possibles
- t_u : type de l'utilisateur u (IoT, smartphone, laptop, etc.)
- $c_{u,a} \geq 0$: coût énergétique pour la liaison (u, a)
- $\lambda \in \{0, 1\}$: coefficient binaire pour inclure ($\lambda = 1$) ou ignorer ($\lambda = 0$) le coût énergétique
- $k_a \in \mathbb{N}$: capacité maximale de l'AP a
- $\mathcal{I} \subseteq \mathcal{A} \times \mathcal{A}$: ensemble des paires d'AP susceptibles d'interférer
- $M_{a,b} \in \mathbb{N}$: seuil maximal d'utilisateurs combinés sur la paire (a, b) , calculé automatiquement

- $x_{u,a} \in \{0, 1\}$: variable de décision, $x_{u,a} = 1$ si l'utilisateur u est assigné à l'AP a , 0 sinon
- $w_u \in \mathbb{Z}^+$: poids représentant la priorité de l'utilisateur u (ex. High = 3, Medium = 2, Low = 1)

2.2 Objectif

Fonction objectif : gestion des priorités et optimisation énergétique combinées

L'objectif global du modèle est de garantir un service de qualité aux utilisateurs en respectant leur priorité, tout en réduisant la consommation énergétique lorsque cela est pertinent. La fonction objectif pondérée est :

$$\max \sum_{(u,a) \in \mathcal{E}} (w_u - \lambda c_{u,a}) x_{u,a}$$

où :

$$c_{u,a} = 0.1 \cdot \frac{P_{\text{base}}^{(t_u)} \cdot d_{u,a}^\alpha}{D_{\text{max}}^\alpha}$$

Justification métier : Cette formulation permet de gérer simultanément les priorités des utilisateurs via le poids w_u et d'intégrer l'optimisation énergétique directement dans le même objectif. Les utilisateurs plus critiques obtiennent un poids plus élevé, ce qui reflète la hiérarchie réelle des priorités dans les réseaux Wi-Fi professionnels. La normalisation par D_{max}^α garantit que le coût énergétique reste proportionné et comparable au poids des utilisateurs, tout en excluant naturellement les utilisateurs hors portée (non inclus dans \mathcal{E}). Le facteur 0.1 a été introduit pour limiter l'impact de la consommation énergétique par rapport à la priorité des utilisateurs, reflétant la pratique générale en réseaux où la qualité de service prime sur l'optimisation énergétique.

Définition du coût énergétique Le coût énergétique $c_{u,a}$ pour chaque liaison (u, a) est défini comme :

$$c_{u,a} = 0.1 \cdot \frac{P_{\text{base}}^{(t_u)} \cdot d_{u,a}^\alpha}{D_{\text{max}}^\alpha}$$

où t_u est le type d'appareil, $P_{\text{base}}^{(t_u)}$ sa puissance nominale, $d_{u,a}$ la distance à l'AP, et α un exposant de perte de propagation (typiquement $\alpha = 3$ en environnement intérieur, $\alpha = 3.5$ en environnement urbain, $\alpha = 2.7$ en environnement extérieur).

Explications :

- t_u : type de l'utilisateur (IoT sensor, wearable, smartphone, tablet, laptop)
- $P_{\text{base}}^{(t_u)}$: puissance de base attribuée à chaque type d'appareil. Exemple :

Type d'appareil	Puissance de base $P_{\text{base}}^{(t_u)}$
IoT sensor	1
Wearable	1
Smartphone	3
Tablet	4
Laptop	6

- $d_{u,a}$: distance entre l'utilisateur et le point d'accès
- α : exposant de perte de propagation
- $\lambda \in \{0, 1\}$: coefficient binaire pour inclure (1) ou ignorer (0) le coût énergétique

Justification métier : Cette formulation permet d'estimer la consommation énergétique réelle des appareils tout en restant simple et cohérente avec la QoS. La normalisation assure que le coût énergétique est numériquement comparable aux poids des utilisateurs, garantissant une hiérarchie claire entre priorité et optimisation énergétique. Le facteur 0.7 reflète le principe que la qualité de service prime sur la consommation énergétique dans la gestion des réseaux.

2.3 Contraintes

2.3.1 Portée maximale admissible

$$x_{u,a} = 0 \quad \forall (u, a) \text{ tel que } d_{u,a} > D_{\max}$$

Justification métier : D_{\max} est déterminé automatiquement selon la **bande WiFi** et le **type d'environnement**. Les valeurs indiquées ci-dessous sont des valeurs par défaut **adaptées au système de coordonnées utilisé dans la visualisation**, mais elles peuvent être ajustées selon les besoins pour mieux correspondre aux scénarios réels ou à la mise à l'échelle graphique :

- **2.4 GHz** : Intérieur 5, Urbain 7, Open-space 12
- **5 GHz** : Intérieur 3, Urbain 5, Open-space 10

2.3.2 Exclusivité des utilisateurs

$$\forall u \in \mathcal{U} : \sum_{a:(u,a) \in \mathcal{E}} x_{u,a} \leq 1$$

2.3.3 Capacité des AP

$$\forall a \in \mathcal{A} : \sum_{u:(u,a) \in \mathcal{E}} x_{u,a} \leq k_a$$

2.3.4 Limitation de l'interférence

Dans les réseaux sans fil, deux points d'accès (AP) peuvent interférer lorsqu'ils sont proches l'un de l'autre et utilisent des canaux qui se chevauchent. Pour modéliser cette interaction de manière réaliste tout en garantissant la faisabilité du problème, nous définissons :

- $\mathcal{I} \subseteq \mathcal{A} \times \mathcal{A}$: l'ensemble des paires d'AP susceptibles d'interférer. Une paire (a, b) appartient à \mathcal{I} si :

$$d_{a,b} \leq D_{\text{intf}} \quad \text{et} \quad O(a, b) = 1,$$

où $d_{a,b}$ est la distance euclidienne entre a et b , D_{intf} est le rayon d'interférence maximal et $O(a, b)$ est un indicateur valant 1 si les canaux des AP se chevauchent. Pour simplifier la modélisation, nous fixons le rayon d'interférence maximal à $D_{\text{intf}} = 1.5 \times D_{\max}$ pour tous les environnements.

- $M_{a,b}$: le nombre maximal combiné d'utilisateurs pouvant se connecter simultanément aux deux APs a et b .

Pour chaque paire $(a, b) \in \mathcal{I}$, nous calculons $M_{a,b}$ selon une relation décroissante linéaire avec la distance, de façon à réduire la capacité combinée lorsque les APs sont proches et à l'autoriser au maximum lorsque les APs sont éloignés :

$$M_{a,b} = \left\lfloor k_a + k_b - \min(k_a, k_b) \cdot \max\left(0, 1 - \frac{d_{a,b}}{D_{\text{intf}}}\right) \right\rfloor.$$

Explication :

- k_a, k_b : capacités respectives des APs a et b .
- $\min(k_a, k_b)$: nous utilisons la capacité du plus petit AP pour moduler la pénalité, garantissant que l'effet de l'interférence reste proportionnel et réaliste.
- $f(d_{a,b}) = \max(0, 1 - d_{a,b}/D_{\text{intf}})$: facteur d'interférence dépendant de la distance. Plus les APs sont proches, plus la pénalité est forte. Au-delà du rayon D_{intf} , l'interférence disparaît.
- La fonction `floor` permet de garantir un nombre entier d'utilisateurs.

Enfin, la contrainte d'interférence s'écrit :

$$\forall (a, b) \in \mathcal{I} : \sum_{u:(u,a) \in \mathcal{E}} x_{u,a} + \sum_{u:(u,b) \in \mathcal{E}} x_{u,b} \leq M_{a,b}.$$

Interprétation : Cette formulation garantit que le nombre combiné d'utilisateurs connectés à deux APs proches ne dépasse jamais une limite réaliste, qui décroît linéairement avec la proximité. Elle est entièrement déterministe et automatique, ce qui permet de maintenir la faisabilité du modèle même pour des réseaux denses.

2.3.5 Binarité des variables

$$\forall (u, a) \in \mathcal{E} : x_{u,a} \in \{0, 1\}$$

3 Formulation complète

Ensembles et variables

$$\mathcal{U} : \text{ensemble des utilisateurs}, \quad \mathcal{A} : \text{ensemble des points d'accès (AP)}$$

$$\mathcal{E} = \{(u, a) \mid d_{u,a} \leq D_{\text{max}}\} \quad (\text{liaisons possibles})$$

$$x_{u,a} \in \{0, 1\} \quad \forall (u, a) \in \mathcal{E} \quad (1 \text{ si } u \text{ est connecté à } a, 0 \text{ sinon})$$

Objectif combiné : priorité et énergie

$$\max \sum_{(u,a) \in \mathcal{E}} (w_u - \lambda c_{u,a}) x_{u,a}$$

où

$$c_{u,a} = 0.1 \cdot \frac{P_{\text{base}}^{(t_u)} \cdot d_{u,a}^{\alpha}}{D_{\max}^{\alpha}}$$

et w_u reflète la priorité de l'utilisateur u , $\lambda \in \{0, 1\}$.

Contraintes

Portée maximale (selon WiFi et environnement)

$$x_{u,a} = 0 \quad \forall (u, a) : d_{u,a} > D_{\max}$$

Exclusivité (un utilisateur connecté à un seul AP)

$$\forall u \in \mathcal{U} : \sum_{a:(u,a) \in \mathcal{E}} x_{u,a} \leq 1$$

Capacité des AP

$$\forall a \in \mathcal{A} : \sum_{u:(u,a) \in \mathcal{E}} x_{u,a} \leq k_a$$

Limitation d'interférence

$$\forall (a, b) \in \mathcal{I} : \sum_{u:(u,a) \in \mathcal{E}} x_{u,a} + \sum_{u:(u,b) \in \mathcal{E}} x_{u,b} \leq M_{a,b}$$

Binarité des variables

$$x_{u,a} \in \{0, 1\} \quad \forall (u, a) \in \mathcal{E}$$

4 Description de l'IHM développée

L'Interface Homme-Machine (IHM) a été développée en Python avec **PyQt5** pour fournir une interface intuitive et réactive permettant la saisie des données, l'exécution de l'optimisation via Gurobi, et la visualisation des résultats.

4.1 Structure générale de l'IHM

L'interface principale (**NetworkGUI**) est composée de trois sections principales :

1. Section Utilisateurs (Users Table)

- Permet à l'utilisateur de saisir et de gérer dynamiquement les utilisateurs du réseau.
- Chaque ligne correspond à un utilisateur et comprend :
 - **Nom** : champ texte (**QLineEdit**),
 - **Priorité** : menu déroulant (**Low**, **Medium**, **High**),

- **Coordonnées X et Y** : champs numériques avec validation (`QLineEdit` + `QIntValidator`),
- **Type de dispositif** : menu déroulant conditionnel (`IoT Sensor`, `Wearable`, `Smartphone`, etc.), activé uniquement si l’option “Include Power Consumption” est cochée.
- Les boutons **Add User** et **Remove Last User** permettent d’ajouter ou supprimer des lignes dynamiquement.

2. Section Points d'accès (AP Table)

- Permet de définir les points d'accès avec : **Nom**, **Capacité**, **Canal**, et **Coordonnées (X, Y)**.
- Les entrées numériques utilisent des `QSpinBox` ou `QLineEdit` avec validation.
- Les boutons **Add AP** et **Remove Last AP** permettent d’ajouter ou retirer des APs dynamiquement.
- La table est entièrement stylisée et les colonnes sont redimensionnées automatiquement pour l’ergonomie.

3. Section Paramètres globaux et calcul

- Permet de configurer le **WiFi Band** (2.4 GHz ou 5 GHz) et le **Type d'environnement** (Indoor, Urban, Outdoor).
- La **consommation d'énergie** peut être incluse dans l'optimisation via une checkbox.
- Les boutons **Calculate** et **Test predefined examples** déclenchent respectivement l'exécution du solveur ou le chargement de scénarios prédéfinis.
- L'exécution du solveur est effectuée dans un thread séparé (`SolverThread`) pour maintenir l'interface réactive.

4.2 Visualisation des résultats

Après le calcul, une fenêtre de résultats (`OutputWindow`) s’ouvre :

1. Table d'affectation des utilisateurs aux APs

- Chaque AP est affiché avec la liste des utilisateurs qui lui sont affectés.
- Les tables sont stylisées pour améliorer la lisibilité (background color, bordures, texte centré).

2. Messages synthétiques

- Indiquent l'état de la résolution :
 - Nombre total d'utilisateurs connectés,
 - Moyenne des priorités des utilisateurs connectés,
 - Statut du solveur.
- Les messages peuvent être mis à jour dynamiquement via les fonctions `set_solver_messages` et `append_solver_message`.

3. Calculs intermédiaires (CalculationsWindow)

- Affiche les variables intermédiaires utilisées par le solveur, telles que :
 - Distances entre utilisateurs et APs,
 - Coûts énergétiques,
 - Paires d’interférences,
 - Capacité maximale combinée des APs.
- Les données sont présentées dans plusieurs sous-tables regroupées pour faciliter la compréhension.

4. Topologie du réseau (TopologyWindow)

- Représentation graphique de la topologie des utilisateurs et APs :
 - APs : carrés colorés selon la couleur assignée, avec icône et étiquette de canal,
 - Utilisateurs : carrés colorés correspondant à l’AP assigné, icône utilisateur et étiquette de priorité,
 - Zones de couverture des APs représentées par des cercles transparents avec rayon réel (D_{max}).
- La visualisation est zoomable et panoramique grâce à un `QGraphicsView` personnalisé (`ZoomableView`).

4.3 Gestion des scénarios prédéfinis

La fenêtre `TestCasesWindow` permet de charger des scénarios enregistrés au format JSON. Chaque scénario propose deux actions : **Voir la source** et **Tester le scénario**. Les résultats s’affichent dans la même logique que la fenêtre de sortie principale.

4.4 Fonctionnalités techniques avancées

- **Multithreading** : le solveur est exécuté dans un thread séparé pour ne pas bloquer l’interface.
- **Validation des entrées** : les champs numériques et menus déroulants assurent la cohérence des données.
- **Tables dynamiques** : ajout/suppression de lignes, activation conditionnelle des colonnes.
- **Stylisation et ergonomie** : codes couleurs cohérents, boutons avec gradient et bordures arrondies, polices Roboto.
- **Robustesse** : toutes les fenêtres gèrent les exceptions et affichent des messages d’erreur clairs.

4.5 Workflow utilisateur

1. L'utilisateur saisit les utilisateurs et APs dans les tables.
2. Les paramètres globaux sont sélectionnés (WiFi, environnement, consommation).
3. Le calcul est lancé via le bouton **Calculate**.
4. Les résultats s'affichent dans la fenêtre de sortie : affectations, messages et options pour visualiser les calculs intermédiaires et la topologie.
5. Les scénarios prédéfinis peuvent être chargés et testés directement pour valider l'application.