

Projet de Recherche Operationnelle - Rapport

1. Introduction

Ce projet presente une application permettant de modeliser et resoudre un probleme de tournées de techniciens de maintenance avec contraintes de competences (Technician Routing and Scheduling Problem - TRSP). L'objectif est de determiner les tournées optimales pour une flotte de techniciens en respectant les competences requises pour chaque intervention. L'application est developpee en Python (PySide6) et utilise Gurobi pour resoudre un modele PLNE.

2. Objectifs du Projet

- Modeliser correctement un probleme de tournées avec contraintes de competences.
- Optimiser les trajets et minimiser les couts operationnels.
- Integrer contraintes de competences, fenetres de temps et duree de travail.
- Fournir une interface simple pour generer et resoudre des instances.

3. Modelisation du Probleme

Ensembles :

- K : ensemble des techniciens ($k = 1, \dots, m$)
- V : ensemble des noeuds (0=depot, 1...n=interventions)
- S : ensemble des competences (ex: electricite, gaz, reseau)

Variables de decision :

- x_{ijk} (element de) {0,1} : 1 si le technicien k va du site i vers j.
- t_{ik} : heure d'arrivee du technicien k au site i.

Parametres :

- c_{ij} : cout/distancie de trajet entre le site i et j (km)
- d_i : duree de l'intervention i (min)
- req_i : ensemble des competences requises pour l'intervention i
- $skill_k$: ensemble des competences possedees par le technicien k
- $[e_i, l_i]$: fenetre de temps pour le debut du service
- T_{max} : duree maximale de travail par technicien
- M : grande constante (big M)

Fonction objectif :

$$\text{Min } Z = \sum_k \sum_i \sum_j c_{ij} * x_{ijk}$$

Contraintes :

1. Affectation unique : $\sum_k \sum_j x_{ijk} = 1$ pour tout i
2. Depart du depot : $\sum_j x_{0jk} = 1$ pour tout k

Projet de Recherche Operationnelle - Rapport

3. Conservation du flux : $\text{Sum}_i x_{ihk} - \text{Sum}_j x_{hjk} = 0$ pour tout h,k
4. Contrainte de compétences : $x_{ijk} = 0$ si req_j n'est pas inclus dans skill_k
5. Contraintes temporelles : $t_{ik} + d_i + c_{ij} \leq t_{jk} + M(1 - x_{ijk})$
6. Fenêtres de temps : $e_i \leq t_{ik} \leq l_i$ pour tout i,k
7. Durée maximale : $t_{ik} \leq T_{\max}$ pour tout i,k
8. Domaines : x_{ijk} (élément de) $\{0,1\}$, $t_{ik} \geq 0$

Projet de Recherche Operationnelle - Rapport

4. Developpement de l'IHM

L'interface PySide6 permet :

- La generation d'instances (techniciens, interventions, competences).
- La configuration : temps limite, MIP gap, nombre de techniciens.
- La definition des competences par technicien et par intervention.
- L'execution du solveur Gurobi (QThread).
- L'affichage automatique des tournées optimales.

5. Resolution et Tests

Des tests ont été effectués sur 5-20 interventions et 2-5 techniciens.

- Respect des contraintes : compétences, fenêtres de temps, durée maximale.
- Affectations correctes : chaque intervention réalisée par un technicien qualifié.
- Temps de calcul acceptable (< 5 secondes pour instances moyennes).

6. Analyse des Resultats

- L'affectation dépend fortement de la disponibilité des compétences.
- Les interventions proches sont naturellement regroupées dans une même tournée.
- Les fenêtres de temps et compétences rares influencent la structure des tournées.
- L'équilibrage de charge entre techniciens est observé dans les solutions.

7. Conclusion

Le projet combine modélisation PLNE, implémentation Python, interface graphique et optimisation Gurobi. Il constitue une base solide pour des extensions futures (multi-période, préférences clients, optimisation multi-objectif, intégration de données réelles d'entreprises de maintenance).