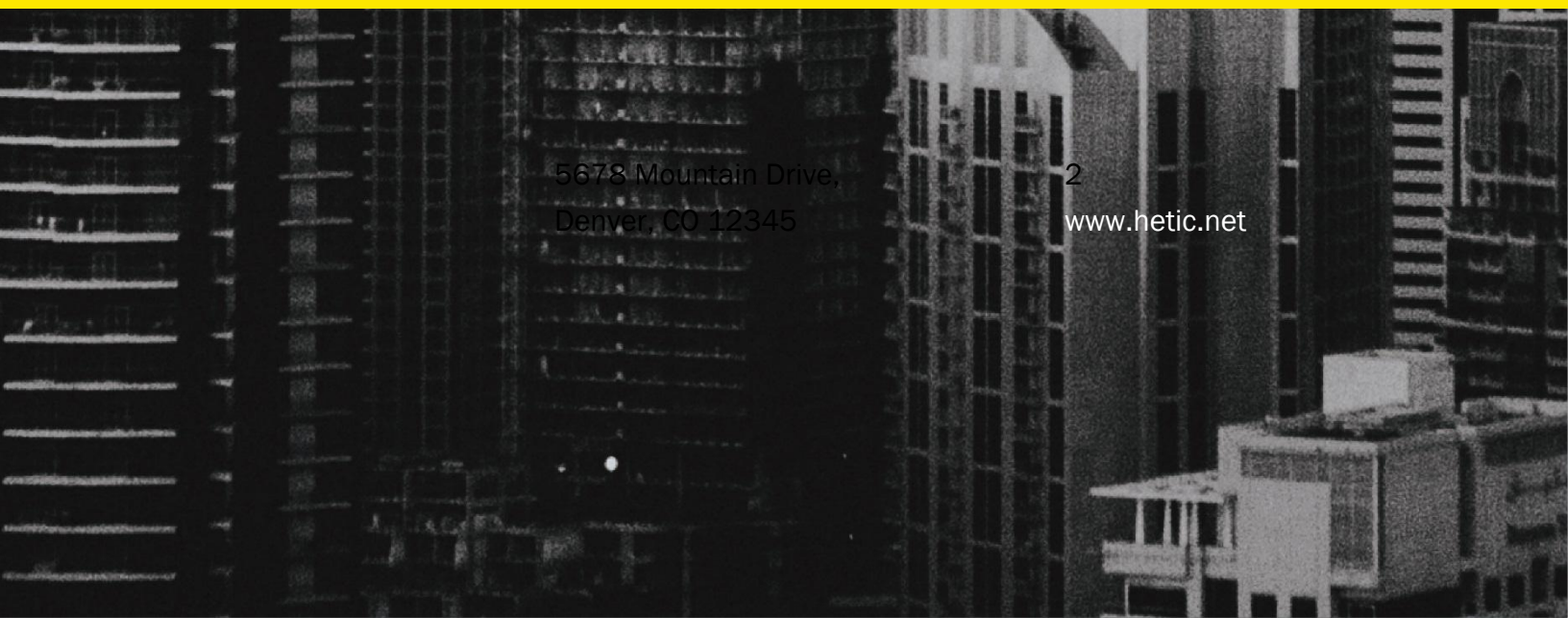




MAG 218 2

RAPPORT DE PLANIFICATION DE RACCORDEMENT DES BATIMENTS



5678 Mountain Drive,
Denver, CO 12345

2
www.hetic.net

MEMBRES DU GROUPE

- DEGBE TEKOU EVANS
- BOUDJEKA ANIK
- NJANKOUO ABDEL AKIM
- NYALEU HERMINE

INTRODUCTION

Une tempête a endommagé une partie électrique, isolant plusieurs bâtiments et habitations. La mairie souhaite rétablir le réseau en priorisant les zones où l'investissement permet de reconnecter le plus de bâtiment possible, au coût minimal.

Nous allons alors planifier la remise en service des infrastructures électriques en optimisant le rapport coût et temps de réparation par bâtiment.

La question que notre groupe s'est posée est : Comment prioriser les travaux pour maximiser le nombre de bâtiment raccorder tout en minimisant les ressources ?

DONNEES ET SOURCES

Nous avons reçu du client qui travaille à la mairie :

- **Un shapefile des bâtiments :**

Qui sert à identifier la position des maisons à raccorder avec deux (02) champs à savoir `id_bat` et `nb_maison`.

- **Un shapefile des infrastructures :**

Qui représente les tronçons d'infrastructure permettant de raccorder les bâtiments avec les champs `id_infra` et `longueur` (l'état des infrastructures : si c'est à remplacer ou pas).

- **Des fichiers Excel et CSV:**

« *Reseau_en_arbres.xlsx* » qui structure l'arborescence du réseau elle contient des champs comme `id_batiments`, `nb_maison`, `type_infra` (l'état des infrastructures : si c'est à remplacer ou pas) et `longueur`.

« *infra.csv* » qui contient les infrastructures et leur format à savoir s'ils sont aériens, semi-aérien ou fourreau.

« *batiments.csv* » qui contient les bâtiments, le type de bâtiment et le vrai nombre de maison.

NB : ces deux derniers documents ont été fournis sous notre demande car nous avons trouvé des incohérences dans le nombre de maison par bâtiments et une insuffisance dans les données.

ANALYSE EXPLORATOIRE

Nous avons visualisé la données dan QGIS sur un fond de carte, ouvert les différents fichiers dans Excel et d'autre dans notre éditeur de code VS code en utilisant la bibliothèque de python Pandas. Nous avons remarqué grâce à la nouvelle donnée fournie qu'il y avait une école et un hôpital dans la liste des bâtiments à raccorder (ce qui a d'ailleurs retenue notre attention et nous a ravivé notre désir à résoudre les problèmes sociaux grâce à la data science) C'est justement à cette étape que nous nous sommes rendu compte de l'incohérence du nombre de maison par bâtiment.

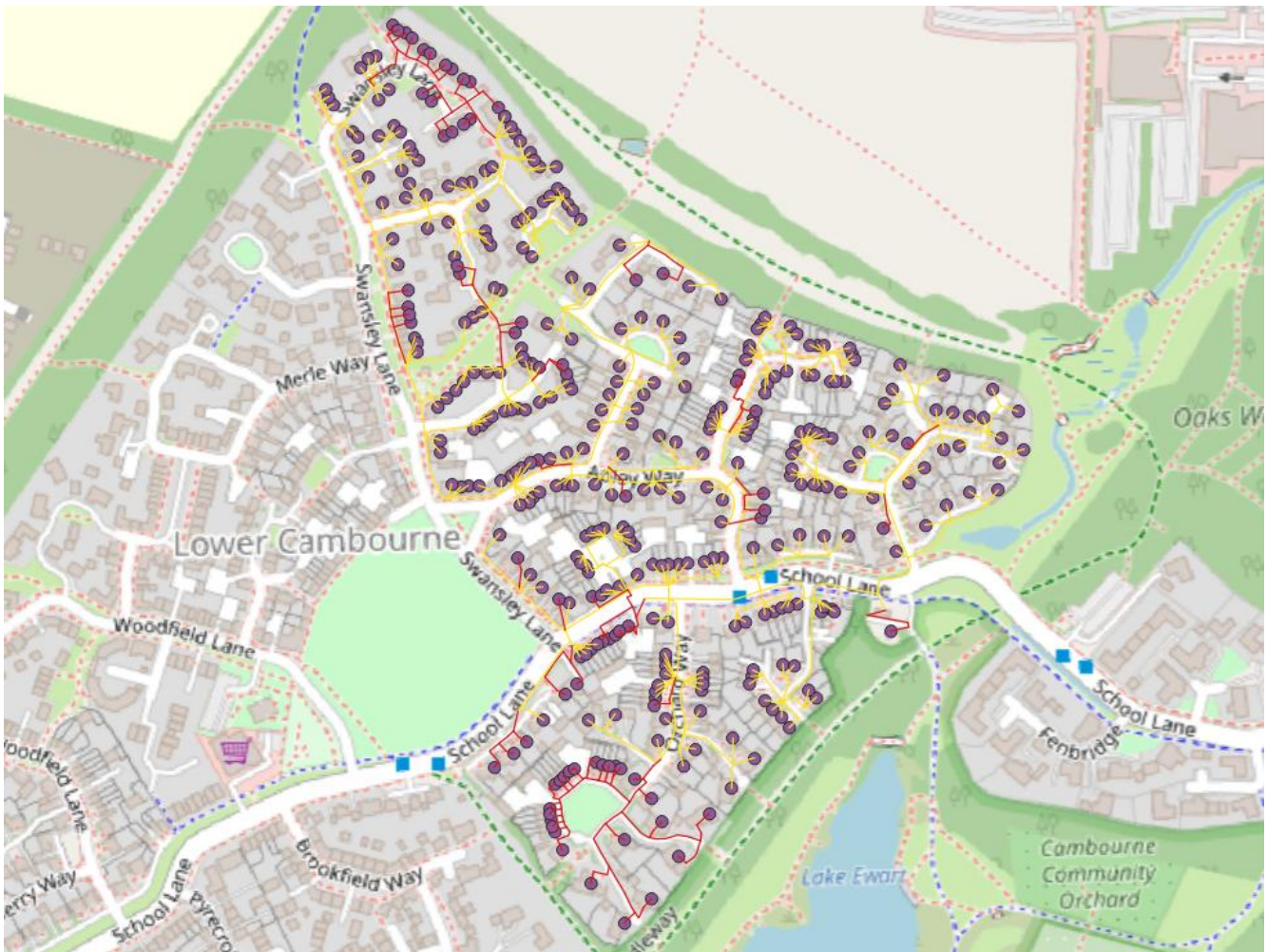


Figure 1

PREPARATION DE LA DONNEE

Nous avons fait une première jointure afin de pouvoir afficher les bâtiments tout en catégorisant les infrastructures à remplacer et ceux qui sont rester intacte après la tempête. (Entre *reseau_en_arbres.xlsx* et le tableau de données contenue dans *infrastructures.shp*

Nous avons ensuite supprimé la colonne qui contenait les faux nombres de maison et avons refait une autre jointure avec *batiment.csv*

Filtrer le réseau à partir des infrastructures à remplacer

Nous avons donc su qu'il y avait **85** bâtiments à raccorder dont **577** infrastructures à réparer.

```
import pandas as pd
from models.building import Building
from models.infra import Infra

network_df = pd.read_excel("reseau_en_arbre.xlsx")
# Réseau des batiments a remplacer
broken_network = network_df[network_df['infra_type'] == "a_replacer"]

# Notre reseau actuels contiens des erreur sur le nombre de maison
# supprimons la colones nb_maisons

broken_network = broken_network.drop(columns=["nb_maisons"])

# faisons une jointure avec la nouvelle données fournie par le clients sur les batiments
new_data_building = pd.read_csv("batiments.csv")
new_broken_network = broken_network.merge(new_data_building, on="id_batiment", how="left")
```

Figure 2

Nous rappelons que dans la nouvelle donnée fournie il y avait une différence de nom de colonne pour *id_infra*. Nous l'avons renommé avant de faire la jointure : voir le code.

```
new_data_infra = pd.read_csv("infra.csv")

# renomons d'abord le nom de la colone infra_id pour corespondre et permettre la jointure
new_broken_network = new_broken_network.rename(columns={"infra_id": "id_infra"})
new_broken_network = new_broken_network.merge(new_data_infra, on="id_infra", how="left")
```

Figure 3

Métrique de Priorisation

Il valait mieux évaluer une métrique par bâtiment, pas par infrastructure, parce qu'un même bâtiment peut être rendu accessible via différentes combinaisons d'infrastructures et que la

réparation d'une infrastructure que nous jugeons prioritaire ne sera pas forcément nécessaire pour ramener l'électricité dans un bâtiments.

Nous sommes donc parties sur la somme des difficultés à réparer toutes les infrastructures qui permettent de raccorder une maison dans un bâtiment à la fois, que nous chercherons à minimiser.

En considérant qu'il y a n maison dans un bâtiments à raccorder, alors la **métrique de difficulté du bâtiment** est **M_{global}** et s'écrit :

$$M_{global} = \sum \frac{C_i \times T_i}{N_i}$$

Interpretation

- C_i Représente le cout de réparation d'une infrastructure.
- T_i Représente le temps réparation d'une infrastructure suivant son type.
- N_i Représente le nombre de maison pouvant être raccordé par les infrastructures.
- **M_{global}** Représente le **score total de difficulté** du plan de raccordement d'un bâtiment.
- Plus **M_{global}** est **faible**, plus notre plan est **efficace** (coût et temps maîtrisés pour un maximum de bâtiments raccordés).
- Cela nous permet de **comparer plusieurs scénarios de planification** entre eux et de partir sur un qui est beaucoup plus optimiser.
- Nous allons donc trier les bâtiments par métrique de difficulté.

```
def get_infra_difficuty(self):
    if self.infra_type == "infra_intacte":
        return 0
    else:
        if self.type_infra == "aerien":
            return ((self.length*500)*2)/self.nb_houses
        if self.type_infra == "semi-aerien":
            return ((self.length*750)*4)/self.nb_houses
        if self.type_infra == "fourreau":
            return ((self.length*900)*5)/self.nb_houses
```

Figure 4

ALGORITHME PROPOSE

Dans un premier temps, nous catégorisons en **phase 0** l'ensemble des bâtiments déjà raccordés ou ne nécessitant aucune réparation au niveau des infrastructures électriques. Ces bâtiments ne font pas partie du processus de priorisation.

Nous créons ensuite une **liste initiale** contenant tous les bâtiments impactés par les pannes d'infrastructure, ainsi qu'une **liste vide** destinée à stocker les bâtiments une fois réparés.

L'algorithme procède alors de manière **itérative** selon les étapes suivantes :

1. Sélectionner, dans la liste des bâtiments impactés, celui présentant la **plus faible valeur de la métrique M_i** (bâtiment le moins difficile à raccorder).
2. Réparer ou remettre en service **toutes les infrastructures associées** à ce bâtiment.
3. Ajouter le bâtiment réparé à la **nouvelle liste**.
4. **Mettre à jour la métrique de difficulté** des autres bâtiments encore non raccordés, puisque la réparation d'une infrastructure commune peut réduire leur coût ou leur temps de raccordement.
5. Supprimer le bâtiment réparé de la **liste initiale**.

Ce processus est répété jusqu'à ce que **tous les bâtiments soient raccordés** (c'est-à-dire jusqu'à ce que la liste initiale soit vide)

```
# On regroupe par id_batiment
for id_building, group in new_broken_network.groupby("id_batiment"):
    list_infras = []
    for _, row in group.iterrows():
        infra = Infra(
            id_infra=row["id_infra"],
            length=row["longueur"],
            infra_type=row["infra_type"],
            nb_houses=row["nb_maisons"],
            type_infra=row["type_infra"],
        )
        list_infras.append(infra)
    building = Building(id_building, list_infras, row["type_batiment"])

    list_object_of_building_broken.append(building)
```

Figure 5

```
#le nombre de tout les batiments a reparer
print(len(list_object_of_building_broken))

buildings = list_object_of_building_broken[:]
building_by_priority = []
while buildings :
    low_difficulty_building = min(buildings)
    for infra in low_difficulty_building.list_infras:
        infra.repair_infra()
    building_by_priority.append(low_difficulty_building)
    buildings.remove(low_difficulty_building)
# voici la liste des batiment par ordre de priorité
print(building_by_priority)

# creérons un colone priorité
building_by_priority_df = pd.DataFrame({"batiments":building_by_priority, "priority":range(len(building_by_priority))})
building_by_priority_df.to_excel('final_building_by_priority.xlsx', index=False, columns=["batiments", "priority"])
```

Figure 6

PLAN DE RACCORDEMENT PROPOSE

L'analyse de la liste des bâtiments a permis d'identifier plusieurs structures d'importance critique, dont **Un hôpital** et **une école**. Étant donné son rôle essentiel dans la prise en charge des urgences et le maintien des services de santé, il a été décidé de **le prioriser explicitement dans le plan de raccordement**, indépendamment de sa métrique initiale.

Concrètement, nous avons **ajusté la métrique de priorisation** $M_i = (C_i \times T_i)/N_i$ associée à l'hôpital et l'école en réduisant artificiellement sa valeur afin de les faire apparaître parmi les premiers bâtiments à raccorder. Cette modification traduit une **pondération manuelle du critère de criticité** : les infrastructures vitales, telles que les hôpitaux, les postes de secours ou les écoles, bénéficient d'une **réaffectation prioritaire des ressources** humaines et matérielles.

RESULTAT ET VISUALISATION

La carte produite sous QGIS illustre la **priorisation des bâtiments** en fonction de leur métrique de difficulté de raccordement. Les bâtiments ont été classés par tranches de 20 unités selon la valeur de la métrique M_i , permettant de visualiser clairement les niveaux de priorité.

Les **bâtiments en bleu** correspondent aux plus simples et rapides à raccorder : ils constituent la première phase d'intervention. Les **bâtiments violets** et **pourpres** représentent des niveaux de difficulté intermédiaires, à planifier dans les phases suivantes. Enfin, les **bâtiments rouges** et **rouge foncé** sont les plus coûteux ou complexes à raccorder et seront traités en dernier, une fois les infrastructures principales rétablies.

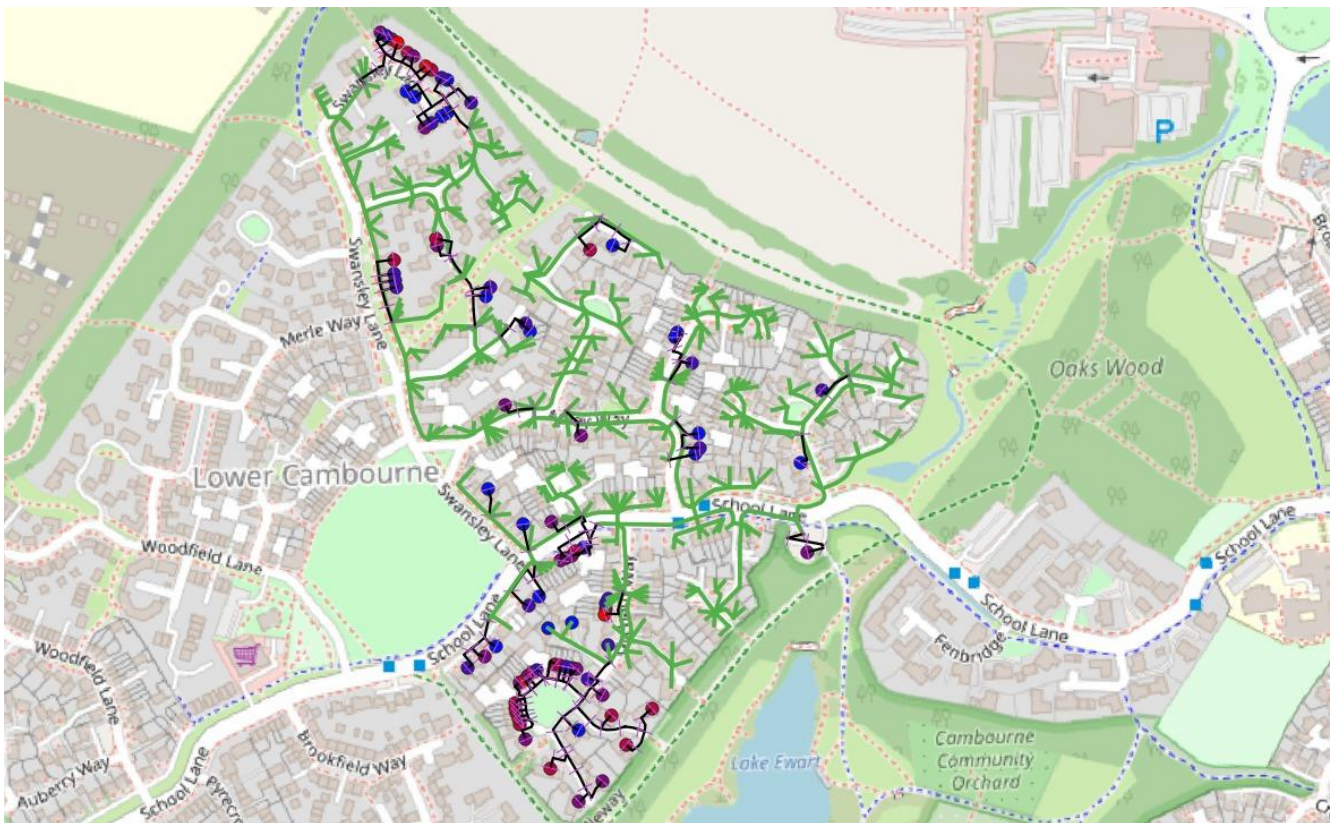


Figure 7

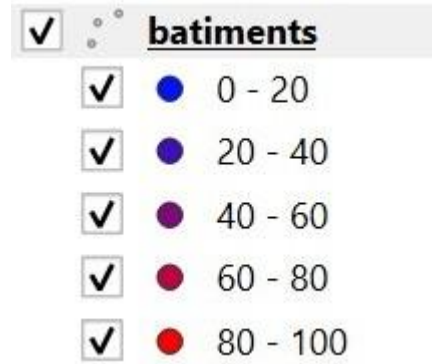


Figure 8

DIFFICULTE ET SOLUTIONS

La principale contrainte a concerné le manque d'informations précises et de clarté de la part du client au lancement du projet, les besoins exprimés étaient souvent incomplets ou évolutifs, Il a donc été nécessaire de multiplier les échanges avec le client afin de clarifier les objectifs, d'ajuster la solution en fonction des besoins

CONCLUSION

Ce projet a permis d'élaborer un plan de raccordement électrique optimisé, fondé sur une métrique de priorisation intégrant le coût, le temps et le nombre de prises raccordées.

L'approche en graphe et l'algorithme itératif ont facilité une planification dynamique et adaptable aux contraintes du terrain.

La prise en compte de la mutualisation des infrastructures a contribué à réduire significativement les coûts globaux.

Nous avons également intégré une logique de criticité, en plaçant l'hôpital en tête des priorités.

Cette stratégie garantit une remise en service rapide des infrastructures essentielles.

Le modèle proposé allie rigueur mathématique et flexibilité décisionnelle.

Il constitue une base solide pour une gestion intelligente et durable des réseaux électriques..