

OPTIMISATION OPERATIONNELLE



Planification du raccordement électrique de bâtiments

By
Jordan NGUEKO
Koceila LEMDANI
Romarc TCHAPPI
Paulin DAPA

Mastère 2 DATA & IA - HETIC Paris

Enseignant
M. Hakim HORAIKY



SOMMAIRE

I.	INTRODUCTION	3
1.	<i>Contexte et problématique</i>	3
2.	<i>Objectifs du projet</i>	4
II.	METHODOLOGIE ET APPROCHE	4
1.	Approche orientée objet	4
a.	<i>Utilisation des méthodes magiques Python</i>	4
b.	<i>Modélisation des entités (bâtiments, infrastructures)</i>	5
2.	Stratégie de résolution	6
a.	<i>Analyse des relations spatiales bâtiments-infrastructures</i>	6
b.	<i>Développement d'une métrique de difficulté</i>	6
c.	<i>Algorithme de planification itératif</i>	7
III.	ANALYSE ET COMPRÉHENSION DES DONNÉES	8
1.	Structure des données spatiales	8
2.	Contraintes techniques	9
a.	<i>Coûts par type d'infrastructure (500-900 €/m)</i>	9
IV.	MODÉLISATION ORIENTÉE OBJET ET IMPLÉMENTATION ALGORITHMIQUE	
	IMPLÉMENTATION ALGORITHMIQUE	10
1.	Architecture des classes Python	10
a.	<i>Classe Infrastructure avec logique métier</i>	10
b.	<i>Classe Bâtiment avec méthodes magiques</i>	11
2.	Analyse de l'état post-intempérie	11
3.	Intégration des données et calcul des contraintes	12
a.	<i>Fusion des sources de données</i>	12
b.	<i>Calcul dynamique des coûts et durées</i>	13
4.	Algorithme de planification itératif	13
a.	<i>Implémentation du processus principal</i>	13
b.	<i>Logique de l'algorithme</i>	15
c.	<i>Structure du fichier de sortie</i>	15
V.	CONCLUSION	17
VI.	ANNEXES	18

I. INTRODUCTION

1. Contexte et problématique

Les événements climatiques extrêmes causent des dommages considérables aux réseaux électriques, privant de nombreux bâtiments d'alimentation. La **remise en service rapide et efficace** devient un enjeu critique pour les gestionnaires de distribution face à la diversité des dommages : infrastructures totalement détruites nécessitant un remplacement complet versus infrastructures intactes remises en service immédiatement.

Les gestionnaires font face à des **contraintes opérationnelles majeures** : équipes de réparation limitées, matériel de remplacement à mobiliser, et pression sociale pour rétablir rapidement l'électricité. Cette situation d'urgence nécessite une méthodologie de planification rigoureuse pour optimiser l'allocation des ressources disponibles.

Dans ce contexte, la planification des réparations électriques post-intempérie se heurte à des contraintes spécifiques :

- **Contraintes économiques** : Les coûts variables selon les types d'infrastructure impactent directement les choix stratégiques - réparations aériennes (500 €/m), semi-aériennes (750 €/m) et souterraines (900 €/m).
- **Contraintes temporelles** : Les durées de réparation varient considérablement (2h/m pour l'aérien, 4h/m pour le semi-aérien, 5h/m pour le souterrain), influençant directement la rapidité de rétablissement du service.
- **Gestion des priorités** : Comment déterminer l'ordre optimal des réparations ? Cette question **stratégique** nécessite une méthodologie objective pour éviter les décisions arbitraires et optimiser l'impact social des interventions.

2. Objectifs du projet

L'objectif principal consiste à développer un plan de raccordement qui priorise les bâtiments les plus simples à raccorder en minimisant les coûts tout en maximisant le nombre de prises raccordées. Cette approche permet une remise en service progressive et optimisée du réseau électrique endommagé.

Les bâtiments les plus simples à raccorder sont ceux qui nécessitent un minimum de réparations au niveau des infrastructures électriques qui permettent leur raccordement.

La méthodologie vise à identifier les infrastructures les plus importantes à raccorder tout en suivant des phases de raccordement.

Parallèlement, l'approche reconnaît qu'une infrastructure plus longue est nécessairement plus coûteuse et lente à installer, donc plus difficile à réparer, ce qui influence directement les priorités d'intervention.

L'objectif technique consiste à créer un algorithme de planification automatisé basé sur une métrique de difficulté objective, permettant d'éviter les décisions arbitraires et de garantir une reconstruction efficace du réseau électrique sinistré dans un contexte de ressources limitées.

II. METHODOLOGIE ET APPROCHE

1. Approche orientée objet

a. Utilisation des méthodes magiques Python

Le projet adopte une approche orientée objet exploitant les capacités avancées de Python pour créer une modélisation intuitive et expressive des entités du réseau électrique post-intempérie. Cette approche permet de manipuler naturellement les objets représentant les infrastructures et bâtiments tout en maintenant une lisibilité optimale du code algorithmique.

L'utilisation stratégique des méthodes magiques (dunder methods) transforme les opérations complexes en expressions naturelles :

- `__radd__` : permet l'utilisation native de la fonction `sum()` pour calculer la difficulté totale d'un bâtiment en additionnant les difficultés de ses infrastructures
- `__lt__` : autorise le tri automatique des bâtiments par difficulté croissante, facilitant l'identification du prochain bâtiment à traiter
- `__repr__` : fournit une représentation lisible de l'état des bâtiments, distinguant visuellement les bâtiments électrifiés de ceux nécessitant des réparations

Cette approche transforme des calculs complexes en opérations intuitives, où `sum(infrastructures)` calcule automatiquement la difficulté totale et où le tri des bâtiments s'effectue naturellement.

b. Modélisation des entités (bâtiments, infrastructures)

La modélisation repose sur deux classes principales interconnectées, spécifiquement conçues pour capturer les relations complexes du réseau électrique endommagé :

- **Classe Infrastructure** : encapsule les propriétés essentielles des segments électriques avec leur identifiant unique, longueur physique, état post-intempérie ("*infra_intacte*" ou "*a_remplacer*"), et nombre de bâtiments desservis. La logique de calcul de difficulté intégrée applique le principe fondamental : **difficulté = longueur / nombre_de_bâtiments_desservis**, mais uniquement pour les infrastructures nécessitant une réparation.
- **Classe Bâtiment** : modélise les bâtiments avec leurs infrastructures de raccordement associées, intégrant des méthodes pour évaluer l'état électrique global et calculer la difficulté totale de raccordement. La méthode `has_electricity()` vérifie que toutes les infrastructures nécessaires sont intactes, tandis que `get_building_difficulty()` agrège les difficultés individuelles.
- Cette modélisation orientée objet facilite la maintenance du code, améliore sa lisibilité et permet une extension aisée vers de nouveaux types d'infrastructures ou de contraintes réglementaires spécifiques au contexte post-catastrophe.

2. Stratégie de résolution

a. Analyse des relations spatiales bâtiments-infrastructures

L'analyse des relations spatiales constitue le fondement de la stratégie de résolution, nécessitant une compréhension fine des dépendances topologiques entre bâtiments et infrastructures électriques. Cette analyse s'articule autour de plusieurs dimensions critiques :

- **Identification de l'état post-intempérie** : distinction systématique entre les bâtiments directement reconnectables (toutes infrastructures intactes) et ceux nécessitant des réparations préalables. Cette classification initiale permet d'identifier les "quick wins" - raccordements immédiats sans coût de réparation.
- **Cartographie des dépendances** : établissement des relations entre chaque bâtiment et l'ensemble des infrastructures nécessaires à son raccordement. Cette cartographie révèle les infrastructures critiques desservant de nombreux bâtiments et celles spécifiques à un seul raccordement.
- **Analyse des effets de synergie** : identification des infrastructures dont la réparation bénéficie simultanément à plusieurs bâtiments, créant des opportunités d'optimisation où une seule intervention peut résoudre plusieurs problèmes de raccordement.

b. Développement d'une métrique de difficulté

Le cœur de la méthodologie repose sur une **métrique de difficulté composite** intégrant les dimensions économiques, temporelles et d'impact social qui est la suivante :

$$\text{difficulté}(\text{infrastructure}) = \frac{\text{cout} * \text{durée}}{\text{nombre de prise}}$$

Cette formulation capture plusieurs intuitions fondamentales adaptées au contexte post-intempérie :

- **Effet multiplicatif coût-durée** : pénalise fortement les interventions combinant coût élevé et durée importante, reflétant l'urgence de la situation post-catastrophe

- **Amortissement par l'impact social** : favorise les interventions bénéficiant à de nombreux usagers, maximisant l'efficacité sociale des ressources limitées
- **Optimisation multi-critères** : équilibre automatiquement les aspects économiques, temporels et sociaux sans nécessiter de pondération arbitraire

La métrique se recalcule après chaque intervention, reflétant l'impact des réparations effectuées sur les difficultés restantes et capturant les effets de synergie émergents.

c. Algorithme de planification itératif

La stratégie de résolution s'articule autour d'un algorithme itératif intelligent structuré en phases distinctes :

- **Phase de priorisation** qui est l'intégration d'un système de priorités distinguant les bâtiments *critiques* (*hôpitaux priorité 1, écoles priorité 2, résidentiel priorité 3*) pour assurer que les infrastructures essentielles sont rétablies en premier.
- **Sélection optimale** de chaque itération, identification du bâtiment combinant la plus haute priorité et la plus faible difficulté technique, garantissant un équilibre entre urgence sociale et efficacité opérationnelle.
- **Propagation des réparations** après chaque intervention, suppression automatique des infrastructures réparées de tous les clusters de bâtiments concernés, capturant immédiatement les effets de synergie et réduisant les difficultés des bâtiments bénéficiaires.
- **Recalcul dynamique** pour une mise à jour systématique des coûts, durées et difficultés après chaque modification, assurant que les décisions suivantes s'appuient sur l'état réel et actualisé du réseau.

Cette approche itérative garantit une convergence vers une solution optimisée tout en maintenant une complexité algorithmique compatible avec les contraintes opérationnelles d'urgence des gestionnaires de réseaux électriques.

III. ANALYSE ET COMPRÉHENSION DES DONNÉES

1. Structure des données spatiales

L'architecture de données s'organise autour de trois sources complémentaires qui fournissent une vision complète du réseau électrique post-intempérie :

- **Données de base** : identifiant unique, classification fonctionnelle (hôpital, école, résidentiel), et nombre de maisons desservies
- **Relations topologiques** : clusters d'infrastructures nécessaires au raccordement de chaque bâtiment
- **État post-intempérie** : classification des infrastructures entre éléments intacts et éléments à réparer

Quant au réseau électrique comprend trois typologies d'infrastructures aux caractéristiques distinctes que sont :

- Infrastructure aérienne : lignes supportées par des poteaux, installation rapide mais vulnérable aux intempéries.
- Infrastructure semi-aérienne : solution hybride combinant segments aériens et souterrains, compromis entre coût et résistance.
- Infrastructure souterraine (fourreau) : câbles enterrés, résistance maximale aux conditions climatiques mais coût et durée d'installation élevés.

La complexité topologique nécessite une approche méthodique préservant l'intégrité des relations spatiales lors du traitement des données. L'intégration s'effectue en maintenant la cohérence référentielle entre bâtiments, infrastructures et états post-intempérie.

2. Contraintes techniques

a. Coûts par type d'infrastructure (500-900 €/m)

Les coûts d'installation varient significativement selon le type :

- **Aérien : 500 €/m** - installation standard avec techniques éprouvées
- **Semi-aérien : 750 €/m** - surcoût de 50% lié à la complexité de coordination
- **Fourreau : 900 €/m** - coût maximal incluant terrassement et protection renforcée

Ces écarts de 80% entre solutions extrêmes justifient l'approche d'optimisation économique développée.

Les contraintes temporelles constituent un facteur critique en situation post-intempérie :

- **Aérien : 2h/m** - installation rapide avec outillage standard
- **Semi-aérien : 4h/m** - durée doublée par la coordination technique
- **Fourreau : 5h/m** - durée maximale incluant terrassement et pose protégée

Le rapport de 2,5 entre durées extrêmes souligne l'importance de l'optimisation temporelle pour minimiser la privation d'électricité.

Et pour finir, la **gestion différenciée** de l'état des infrastructures nécessite un traitement dynamique des clusters. Chaque bâtiment peut dépendre de multiples infrastructures de types différents, nécessitant une analyse fine pour déterminer les coûts et durées totaux. Le système recalcule automatiquement ces paramètres après chaque intervention, reflétant l'évolution de l'état du réseau.

IV. MODÉLISATION ORIENTÉE OBJET ET IMPLÉMENTATION ALGORITHMIQUE

1. Architecture des classes Python

a. Classe Infrastructure avec logique métier

La classe **Infrastructure** constitue l'entité de base du modèle, encapsulant les propriétés techniques et l'état post-intempérie de chaque segment du réseau électrique :

```
class Infra:
    def __init__(self, infra_id: str, length: float, infra_type: str, nb_houses: int):
        self.infra_id = infra_id
        self.length = length
        self.infra_type = infra_type # "infra_intacte" ou "a_replacer"
        self.nb_houses = nb_houses

    def repair_infra(self):
        self.infra_type = "infra_intacte"
```

```
    def get_infra_difficulty(self) -> float:
        # Principe : longueur / nombre de bâtiments desservis
        base_difficulty = self.length / self.nb_houses
        return base_difficulty if self.infra_type == "a_replacer" else 0

    def __radd__(self, other):
        return other + self.get_infra_difficulty() if isinstance(other, (int, float)) else NotImplemented
```

La méthode *get_infra_difficulty()* implémente le principe économique fondamental : plus une infrastructure est longue, plus elle est coûteuse à réparer, mais plus elle dessert de bâtiments, plus son coût s'amortit. Seules les infrastructures nécessitant une réparation contribuent à la difficulté.

b. Classe Bâtiment avec méthodes magiques

La classe **Bâtiment** agrège les infrastructures nécessaires au raccordement et exploite les méthodes magiques pour simplifier les opérations complexes :

```
class Batiment:
    def __init__(self, id_building: str, list_infras: List[Infra]):
        self.id_building = id_building
        self.list_infras = list_infras

    def get_building_difficulty(self) -> float:
        return sum(self.list_infras) # Utilise __radd__ d'Infra

    def __lt__(self, other) -> bool:
        return self.get_building_difficulty() < other.get_building_difficulty()
```

```
def has_electricity(self) -> bool:
    return all(infra.infra_type == "infra_intacte" for infra in self.list_infras)

def __repr__(self):
    etat = "⚡ Électrifié" if self.has_electricity() else "À réparer"
    return f"Batiment(id={self.id_building}, état={etat}, difficulté={self.get_building_difficulty():.2f})"
```

Les **méthodes magiques** transforment les opérations complexes en expressions naturelles : **sum()** calcule automatiquement la difficulté totale, < permet le tri par difficulté, et **__repr__** fournit un affichage informatif de l'état.

2. Analyse de l'état post-intempérie

Le module **state_batiment.py** effectue l'analyse initiale pour distinguer les bâtiments selon leur état électrique :

```
python

import pandas as pd

network_data = pd.read_excel("reseau_en_arbre.xlsx")
broken_network = network_data[network_data['infra_type'] == 'a_remplacer']

set_id_batiments = set(network_data['id_batiment'].values)
set_id_broken_batiments = set(broken_network['id_batiment'].values)
```

```
list_id_batiments, state_batiment = [], []

for id_batiment in network_data['id_batiment'].values:
    list_id_batiments.append(id_batiment)
    if id_batiment in set_id_broken_batiments:
        state_batiment.append("a_reparer")
    else:
        state_batiment.append("intact")

state_df = pd.DataFrame({'id_batiment': list_id_batiments, 'state_batiment': state_batiment})
state_df.to_excel('etat_batiment.xlsx', index=False)
```

Cette analyse utilise des *structures set* pour optimiser les performances des opérations d'appartenance et distingue immédiatement les bâtiments reconnectables de ceux nécessitant des réparations.

3. Intégration des données et calcul des contraintes

a. Fusion des sources de données

Le module jointure.py intègre les différentes sources d'information en préservant l'intégrité référentielle :

```
import pandas as pd
import ast

network_data = pd.read_csv("batiments_cluster_infra.csv") # 85 lignes
etat_network = pd.read_excel("etat_batiment.xlsx")
batiments = pd.read_csv("batiments.csv")

# Suppression des doublons
etat_network = etat_network.drop_duplicates(subset=['id_batiment'])
batiments = batiments.drop_duplicates(subset=['id_batiment'])
```

```
# Fusion progressive
merged_batiment = pd.merge(network_data, etat_network, on='id_batiment', how='left')
merged_batiment.drop(columns=['nb_maisons'], inplace=True)

final_df = pd.merge(merged_batiment, batiments, on='id_batiment', how='left')
```

b. Calcul dynamique des coûts et durées

La fonction `calculer_prix_et_duree()` traite les clusters d'infrastructures complexes avec gestion d'erreurs robuste :

```
prix_dict = {'aerien': 500, 'semi-aerien': 750, 'fourreau': 900}
duree_dict = {'aerien': 2, 'semi-aerien': 4, 'fourreau': 5}

def calculer_prix_et_duree(cluster_infra_str):
    try:
        cluster = ast.literal_eval(cluster_infra_str)
        if not isinstance(cluster, dict):
            return pd.Series([0, 0])

        cout_total = 0
        duree_totale = 0

        for key, longueur in cluster.items():
            if isinstance(key, tuple):
                _, infra_type = key
            else:
                infra_type = key

            prix_m = prix_dict.get(infra_type, 0)
            duree_h = duree_dict.get(infra_type, 0)

            cout_total += prix_m * longueur
            duree_totale += duree_h * longueur

    except Exception as e:
        print("Erreur sur :", cluster_infra_str)
        return pd.Series([0, 0])

# Application et sauvegarde
final_df[['cout_total', 'duree_totale']] = final_df['cluster_infra'].apply(calculer_prix_et_duree)
final_df.drop(columns=['state_batiment', 'infra_type'], inplace=True)
final_df.to_csv("all_for_one_avec_cout.csv", index=False)
```

4. Algorithme de planification itératif

a. Implémentation du processus principal

L'algorithme principal dans `main.py` implémente la logique itérative avec gestion des priorités et effets de synergie :

python

```
import pandas as pd
import ast
from jointure import calculer_prix_et_duree

# Chargement et préparation des données
data = pd.read_csv("all_for_one_avec_cout.csv")
data["cluster_infra"] = data["cluster_infra"].apply(ast.literal_eval)
```

```
# Système de priorités
priorite_map = {"hôpital": 1, "école": 2}
data["priorite"] = data["type_batiment"].map(priorite_map).fillna(3)

ordre_reparation = []
iteration = 1

while not data.empty:
    # Recalcul dynamique après chaque modification
    data[['cout_total', 'duree_totale']] = data['cluster_infra'].apply(calculer_prix_et_duree)
    data['difficulte'] = data['duree_totale'] * data['cout_total'] / data['nb_maisons']
```

```
# Tri par priorité puis difficulté
data = data.sort_values(by=['priorite', 'difficulte'], ascending=[True, True]).reset_index(drop=True)

# Sélection du bâtiment optimal
batiment_facile = data.iloc[0]
infra_reparees = list(batiment_facile['cluster_infra'].keys())

# Propagation des réparations (effet de synergie)
data['cluster_infra'] = data['cluster_infra'].apply(
    lambda d: {k: v for k, v in d.items() if k not in infra_reparees}
)
```

```
# Identification des bâtiments complètement réparés
batiments_re pares = data.loc[data['cluster_infra'].apply(len) == 0, 'id_batiment'].tolist()

# Enregistrement dans le plan de raccordement
for bid in batiments_re pares:
    ordre_reparation.append((iteration, bid))

# Suppression des bâtiments traités
data = data[data['cluster_infra'].apply(len) > 0]
iteration += 1
```

```
# Génération du plan final
ordre_df = pd.DataFrame(ordre_reparation, columns=['ordre', 'id_batiment'])
ordre_df = ordre_df.drop_duplicates(subset='id_batiment', keep='first').reset_index(drop=True)
ordre_df.to_csv("ordre_reparation_priorisee.csv", index=False)

print(f"✅ {len(ordre_df)} bâtiments réparés au total (sur 85 attendus)")
```

b. Logique de l'algorithme

L'algorithme suit une **séquence optimisée** qui garantit une progression efficace :

- **Recalcul dynamique** : mise à jour des coûts, durées et difficultés après chaque modification du réseau
- **Sélection intelligente** : choix du bâtiment combinant priorité maximale (hôpitaux, écoles) et difficulté minimale
- **Réparation globale** : intervention sur toutes les infrastructures du bâtiment sélectionné
- **Propagation automatique** : suppression des infrastructures réparées de tous les clusters concernés
- **Identification des synergies** : détection automatique des bâtiments devenus complètement réparés
- **Mise à jour itérative** : préparation de l'itération suivante avec l'état actualisé du réseau

Cette approche garantit une **convergence optimale** vers la solution tout en capturant automatiquement les effets de synergie entre réparations, où une seule intervention peut bénéficier à plusieurs bâtiments simultanément. Le système produit un plan de raccordement séquencé sauvegardé dans "ordre_reparation_priorisee.csv", fournissant aux équipes d'intervention un guide opérationnel précis pour la reconstruction du réseau électrique.

c. Structure du fichier de sortie

Le fichier de sortie est nommé « *ordre_reparation_priorisee.csv* ». Il donne les numéros d'ordre de réparation et raccordement dans la planification optimisée :

- **Valeurs** : 1, 2, 3, ..., 85 (pour les 85 bâtiments)

- **Signification** : Plus le numéro est faible, plus le bâtiment est prioritaire dans le plan d'intervention

Le fichier reflète l'application de l'algorithme d'optimisation qui intègre :

Systeme de priorités par type

- **Ordre 1-X** : Hôpitaux (priorité 1) traités en premier
- **Ordre X+1-Y** : Écoles (priorité 2) traitées ensuite
- **Ordre Y+1-85** : Bâtiments résidentiels (priorité 3) traités en dernier

Optimisation par difficulté

Au sein de chaque niveau de priorité, les bâtiments sont ordonnés par **difficulté croissante**

L'ordre tient compte des **interdépendances** : quand un bâtiment est réparé, les infrastructures communes bénéficient automatiquement aux autres bâtiments, modifiant leur position dans la séquence.

V. CONCLUSION

Le projet a développé avec succès un système d'optimisation automatisé pour la planification des raccordements électriques post-intempérie. L'algorithme traite efficacement les 85 bâtiments du réseau en générant un plan de raccordement séquencé qui optimise l'allocation des ressources limitées.

La métrique de difficulté *(Durée_totale × Coût_total) / Nombre_de_maisons* s'avère particulièrement pertinente, intégrant simultanément les contraintes économiques, temporelles et sociales tout en capturant automatiquement les effets de synergie entre réparations.

En somme, le système fournit aux gestionnaires de réseaux un outil d'aide à la décision objectif éliminant les choix arbitraires. Dans un contexte post-catastrophe où les ressources sont critiques, l'algorithme maximise leur efficacité en identifiant systématiquement les interventions offrant le meilleur rapport effort/impact.

De plus, la méthodologie développée présente un potentiel d'application étendu aux réseaux de télécommunications, d'eau potable et de transport public. Des enrichissements futurs pourraient intégrer les contraintes géographiques, la gestion des ressources humaines et l'adaptation à l'incertitude des conditions post-catastrophe.

VI. ANNEXES

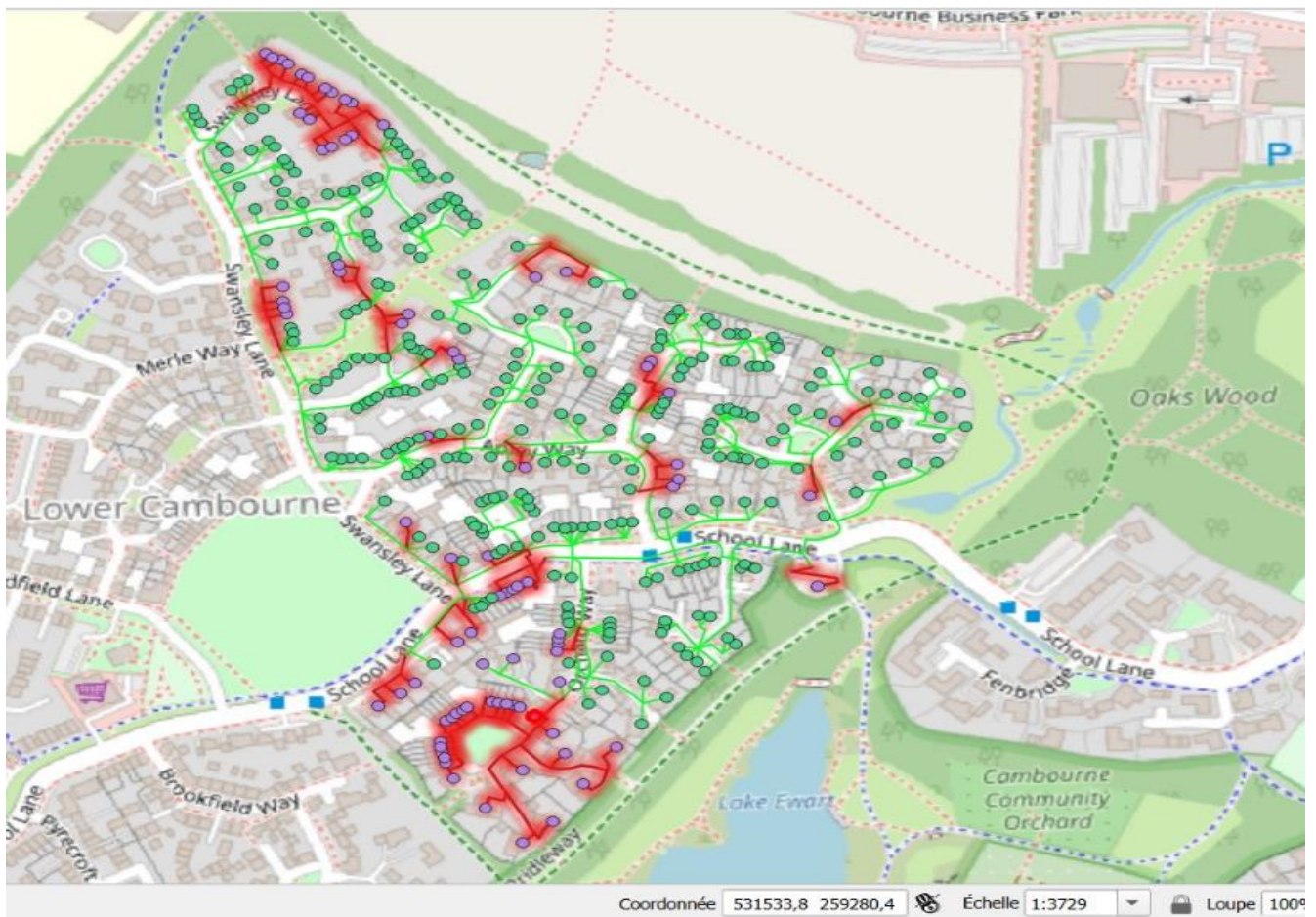


Image 1: infrastructure à réparer

- ☒ **drive-download-20**
- ☒ ● a_reparer
- ☒ ● intact
- ☒ ●
- ☒ **batiments_electricite —**
- ☒ **drive-download-20**
- ☒ — a_remplacer
- ☒ — infra_intacte
- ☒ —

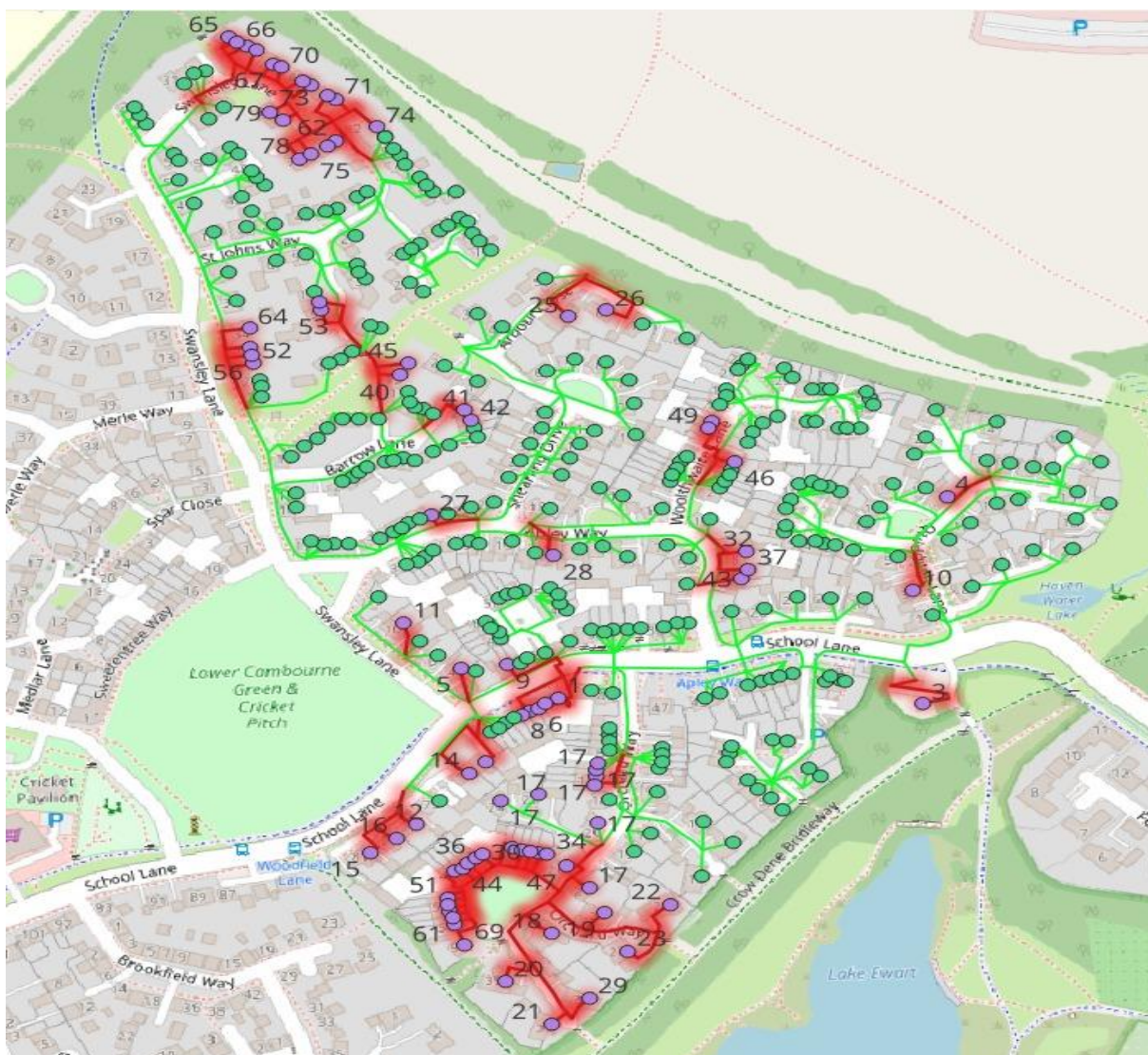


Image 2: ordre de priorité des bâtiments à réparer