

VRP – VEHICLE ROUTING PROBLEM

- Réalisé par CISERANE Marius & BOULLOT Matthias



Sommaire



Rappel du projet et des objectifs



Résultats



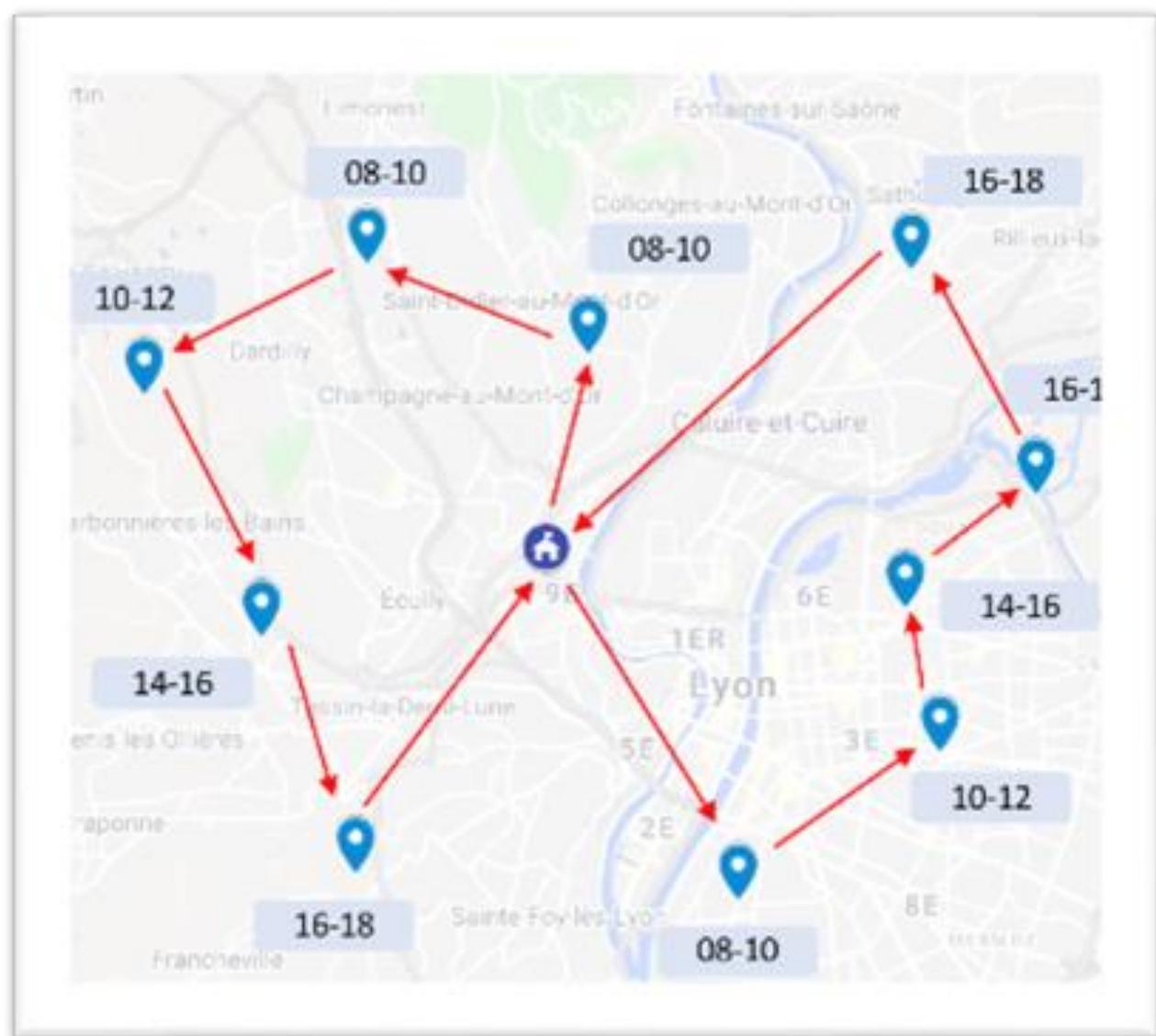
Bilan



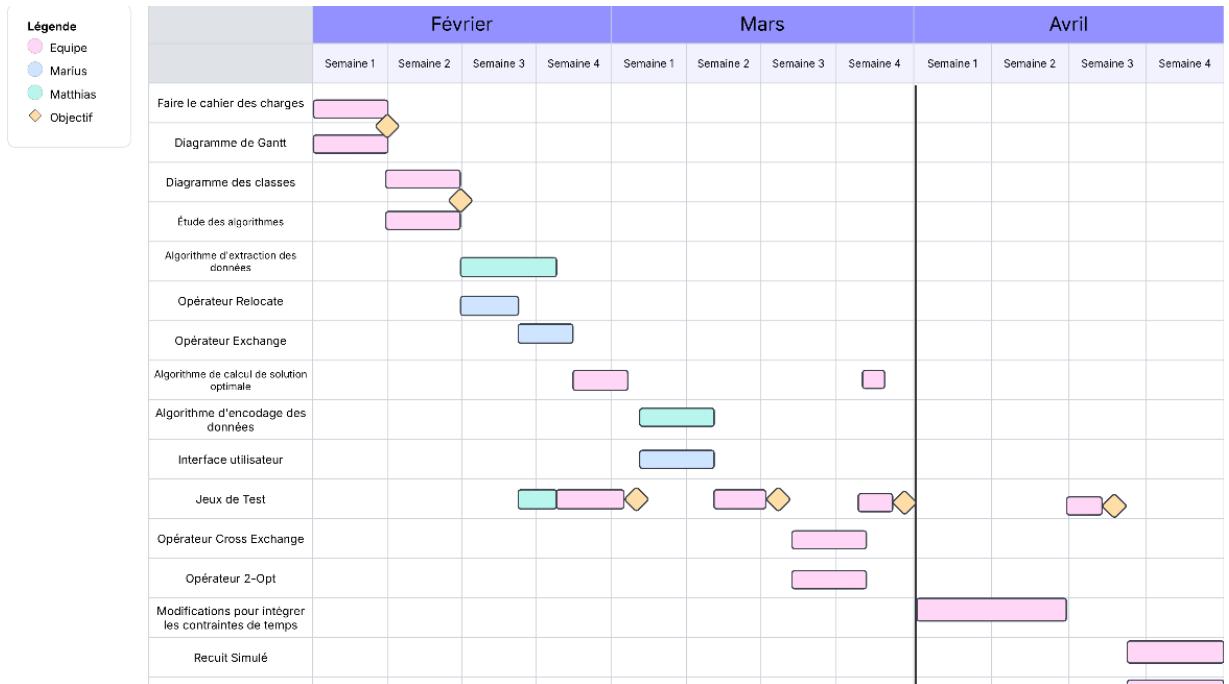
Apports du projet

Rappel du projet

- Définitions :
 - VRP : Vehicle Routing Problem
 - CVRPTW : Capacitated Vehicle Routing Problem with Time Window



Objectifs du projet



- Quoi ? -> Un logiciel utilisable par un particulier
- Comment ? -> Avec l'utilisation de mét heuristicques

Présentation du code

```
1 def intra_relocate(trajet :Trajet) -> tuple[float, tuple[int, int]] :
2     """
3         Calcule et renvoie un tuple avec des informations sur le trajet avec la plus courte longueur
4         après une itération de relocate.
5
6         Paramètres
7         -----
8         trajet : Trajet
9             Trajet sur lequelle est appliquée l'opérateur relocate.
10
11        Renvoie
12        -----
13        La différence de longueur entre le nouveau trajet et l'ancien, et
14        un tuple contenant la position originale et la nouvelle position du client relocalisé.
15        """
16        assert isinstance(trajet, Trajet)
17
18        nb = trajet.nb_clients
19        mini = 0
20        ind = None
21
22        for i in range(nb) :
23            cli = trajet.clients[i]
24            dist_tmp = trajet.dist_retirer_client(i)
25            for j in range(nb+1) :
26                if j == i or j == i+1 : continue
27
28                dist_tmp2 = dist_tmp + trajet.dist_ajouter_client(j, cli)
29
30                if dist_tmp2 < mini :
31                    mini = dist_tmp
32                    ind = (i, j if j < i else j-1)
33
34        return (mini, ind)
```

```
1 for i in range(nb) :
2     cli = trajet.clients[i]
3     dist_tmp = trajet.dist_retirer_client(i)
4     for j in range(nb+1) :
5         if j == i or j == i+1 : continue
6
7         dist_tmp2 = dist_tmp + trajet.dist_ajouter_client(j, cli)
8
9         if dist_tmp2 < mini :
10             mini = dist_tmp
11             ind = (i, j if j < i else j-1)
12
13 return (mini, ind)
```

```
1  def intra_relocate(trajet :Trajet) -> tuple[float, tuple[int, int]] :
2      """
3          Calcule et renvoie un tuple avec des informations sur le trajet avec la plus courte longueur
4          après une itération de relocate.
5
6          Paramètres
7          -----
8          trajet : Trajet
9              Trajet sur lequelle est appliqué l'opérateur relocate.
10
11         Renvoie
12         -----
13         La différence de longueur entre le nouveau trajet et l'ancien, et
14         un tuple contenant la position originale et la nouvelle position du client relocalisé.
15         """
16         assert isinstance(trajet, Trajet)
17
18         nb = trajet.nb_clients
19         mini = 0
20         ind = None
21
22         for i in range(nb) :
23             cli = trajet.clients[i]
24             dist_tmp = trajet.dist_retirer_client(i)
25             for j in range(nb+1) :
26                 if j == i or j == i+1 : continue
27
28                 dist_tmp2 = dist_tmp + trajet.dist_ajouter_client(j, cli)
29
30                 if dist_tmp2 < mini :
31                     mini = dist_tmp
32                     ind = (i, j if j < i else j-1)
33
34         return (mini, ind)
```



```
1  for i in range(nb) :
2      cli = trajet.clients[i]
3      dist_tmp = trajet.dist_retirer_client(i)
4      for j in range(nb+1) :
5          if j == i or j == i+1 : continue
6
7          dist_tmp2 = dist_tmp + trajet.dist_ajouter_client(j, cli)
8
9          if dist_tmp2 < mini :
10             mini = dist_tmp
11             ind = (i, j if j < i else j-1)
12
13     return (mini, ind)
```

Résultats

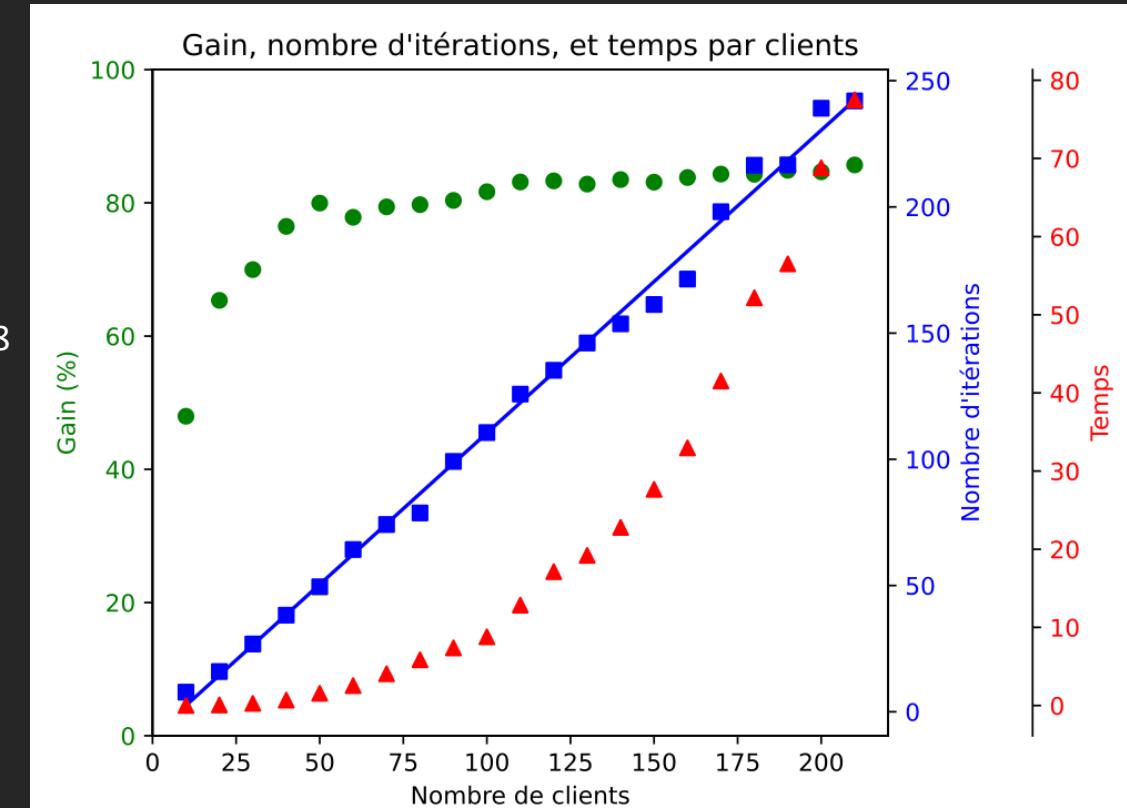


Tableau 1 : Performances de l'algorithme sans contraintes

Situation	Gain (en %)	Nombre d'itérations par clients	Temps d'exécution (en secondes)
1	84,96	0,9789	19,73
2	84,96	0,9805	19,63
3	87,90	1,215	21,89
4	88,16	1,225	21,41

Tableau 3 : Performances avec contrainte de capacité et d'intervalles de temps

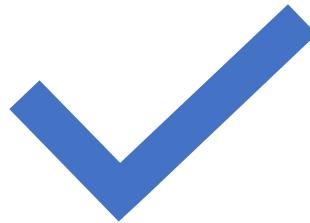
Situation	Gain (en %)	Nombre d'itérations par clients	Temps d'exécution (en secondes)
1	64,90	1,063	24,65
2	65,11	1,062	25,275
3	67,40	1,019	31,30
4	67,47	1,006	29,83

Bilan



Atteinte des objectifs ?

Un logiciel utilisable par un particulier ?



Perspectives d'amélioration

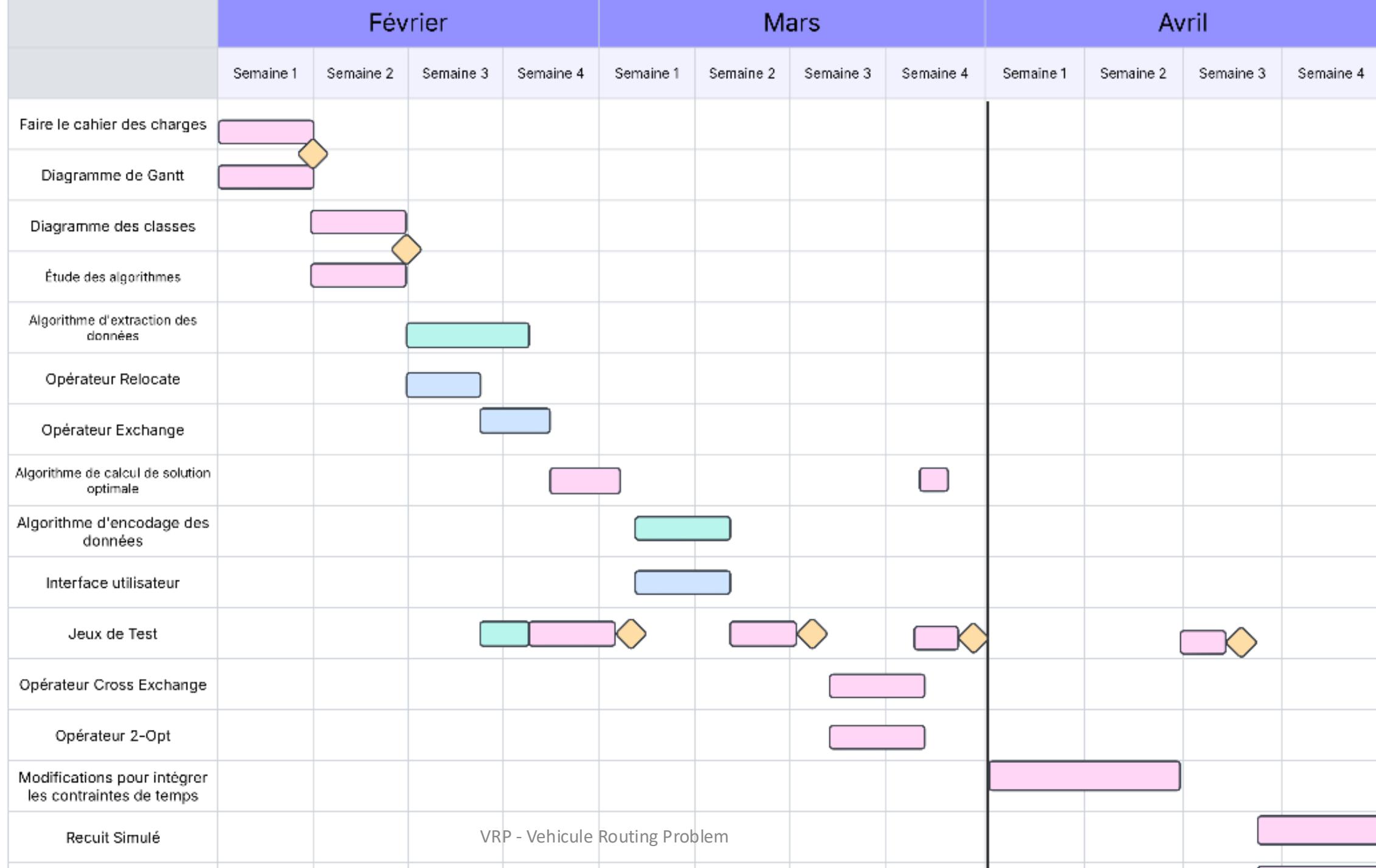
Reçut Simulé

Méthode Tabou

Amélioration des contraintes

Légende

- Équipe
- Marius
- Matthias
- ◆ Objectif



Apports du projet



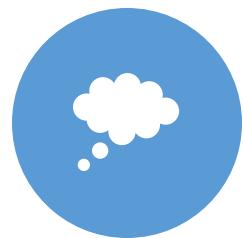
COMPÉTENCES EN
GESTION DE PROJET



TRAVAIL D'ÉQUIPE



MODULES PYTHON
(MATPLOTLIB/JINJA2)



MÉTAHEURISTIQUES



DÉVELOPPEMENT
WEB (ARCHITECTURE
SITE + SERVEUR)