



Présentation générale, organisation interne, conception et démonstration

SimTaxi

- ⇒ Rappel de l'énoncé 3
- Organisation interne 4
- Optimisations 9
- Adaptation de la moyenne des courses 10
- Génération d'une course 12
- Statistiques 13
- Politiques 19
- Conclusion 25



Rappel de l'énoncé

Simulation visant à optimiser des politiques d'utilisation de l'information à l'intérieur d'une compagnie de taxis.

But : Trouver une politique minimisant le nombre de km parcourus à vide par les taxis.



Chaque membre du groupe est responsables pour un module :

- Du développement (doc. interne de développement)
- De la documentation (doc. interne de rédaction)

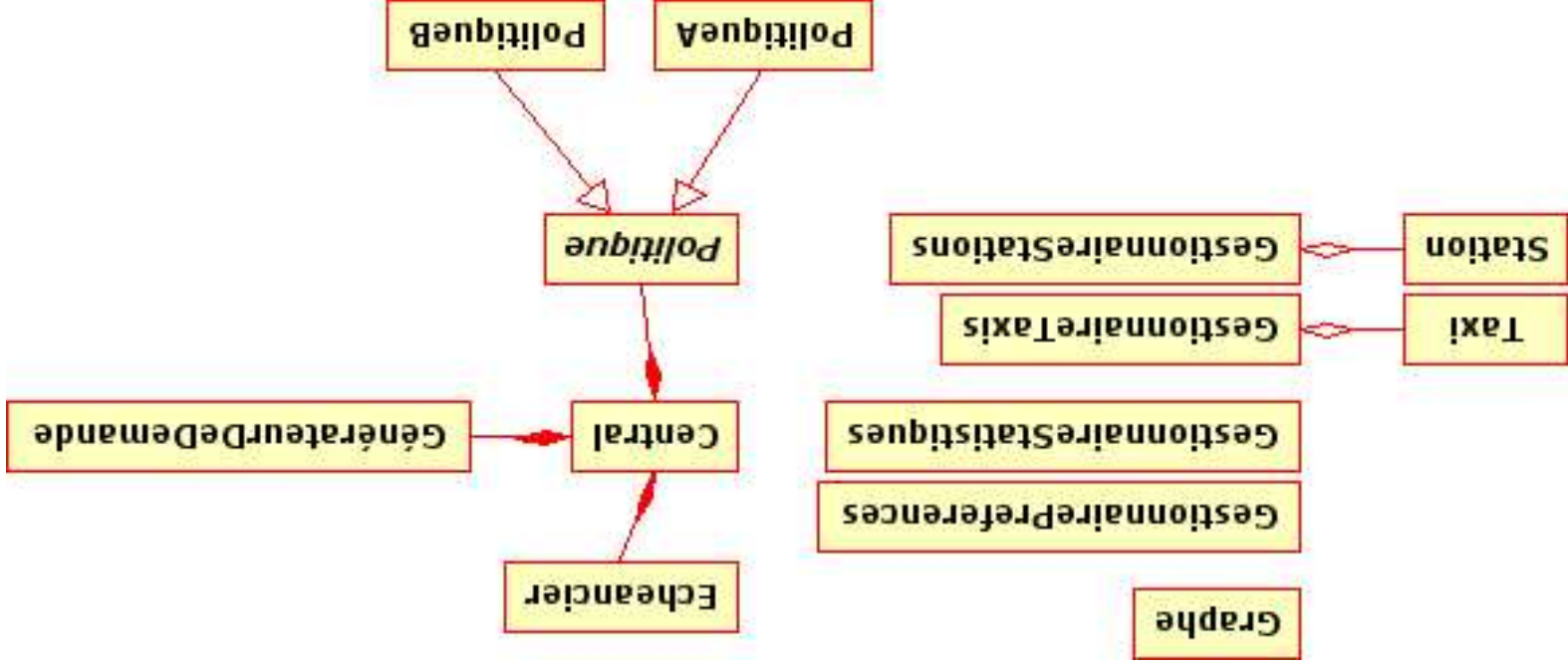
Les fichiers sources (format texte) de la documentation et du code python sont réuni et structuré sur le CVS.

Organisation interne





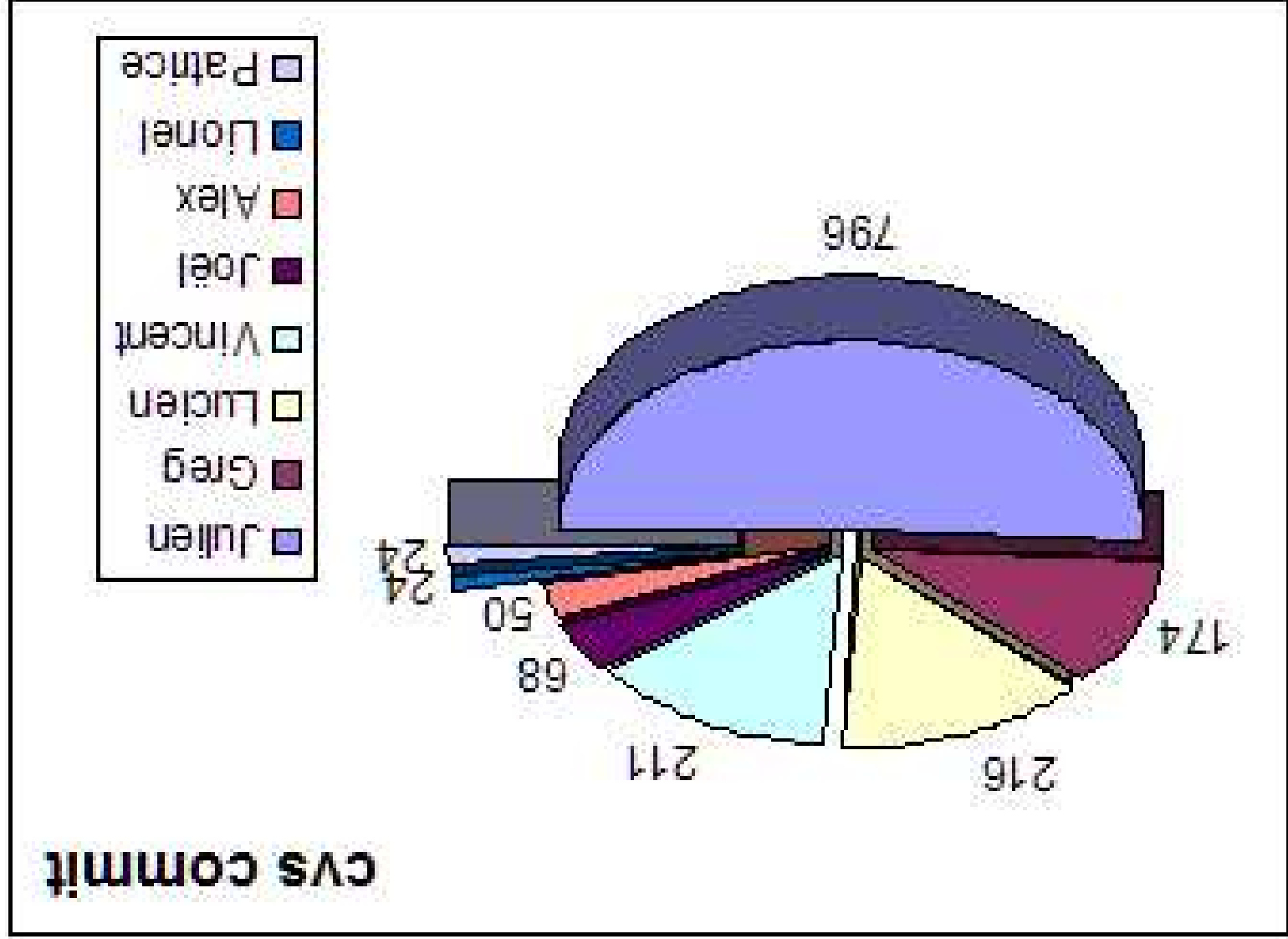
Décomposition

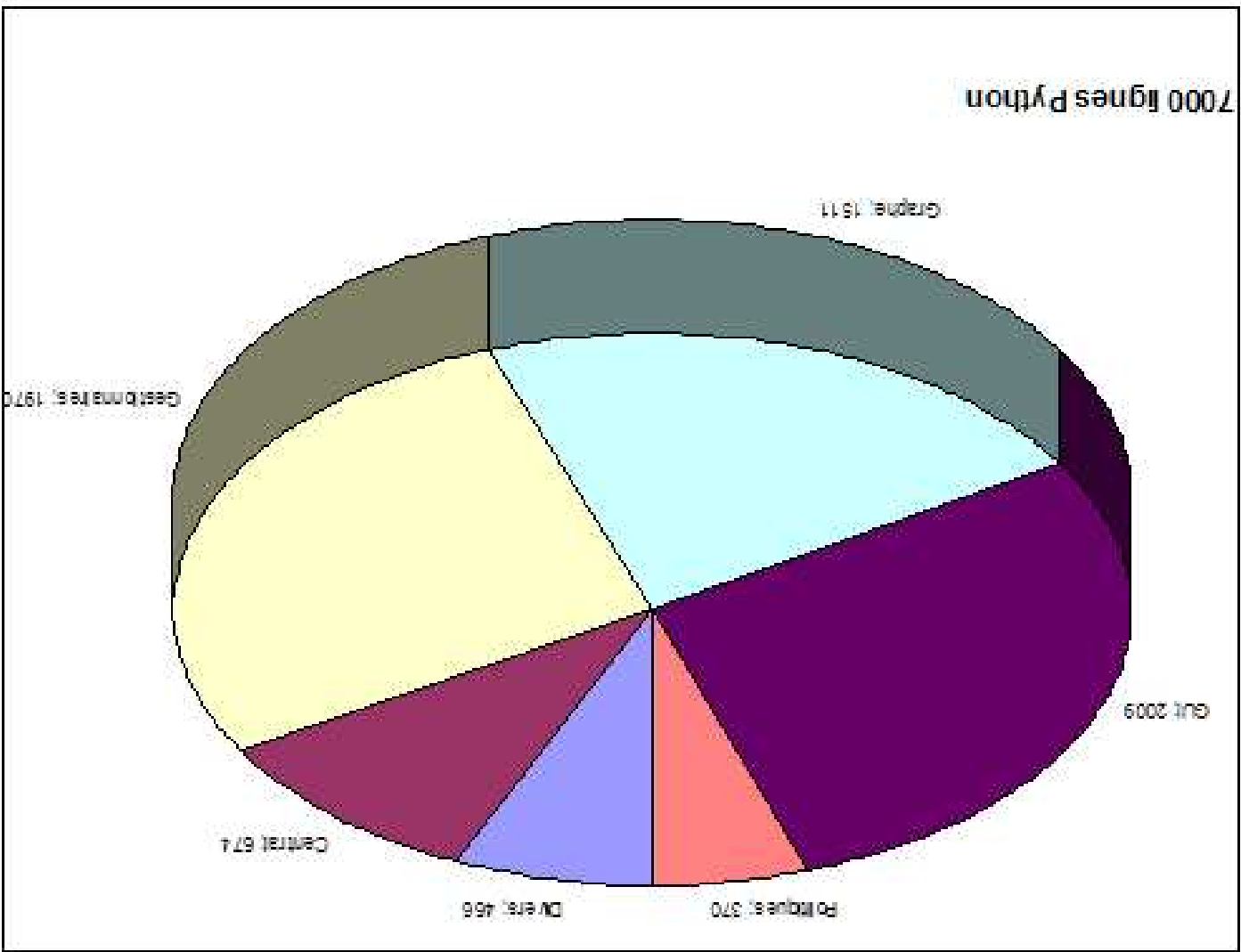




Répartition des tâches

J. BURDY	Chef de groupe
G. BURRI	Interface utilisateur
L. CHABOUDEZ	Comportement des taxis
A. D'AMICO	Module statistiques
V. DECORGES	Politiques et graphiques en cours de simulation
P. FERROT	Nouvelle génération des courses/clients
L. GUÉLAT	Optimisation des algorithmes
J. JAQUEMET	Comparaison des politiques





- ## Optimisations
- PCC : Rechercher le chemin le plus court d'une position à une autre -> Dijkstra. Dijkstra retourne l'arbre des chemins les plus courts pour le sommet de départ.
 - Rechercher la station la plus proche. Un cas fréquent d'utilisation du PCC est pour trouver la station la plus proche après avoir déposé un client.





Adaptation de la moyenne des courses

Déterminer un paramètre L associé à une fonction simple f_L

Par f_L , à chaque distance correspond un nombre de l'intervalle $[0, 1]$

Pour savoir si on planifie une course ou non :

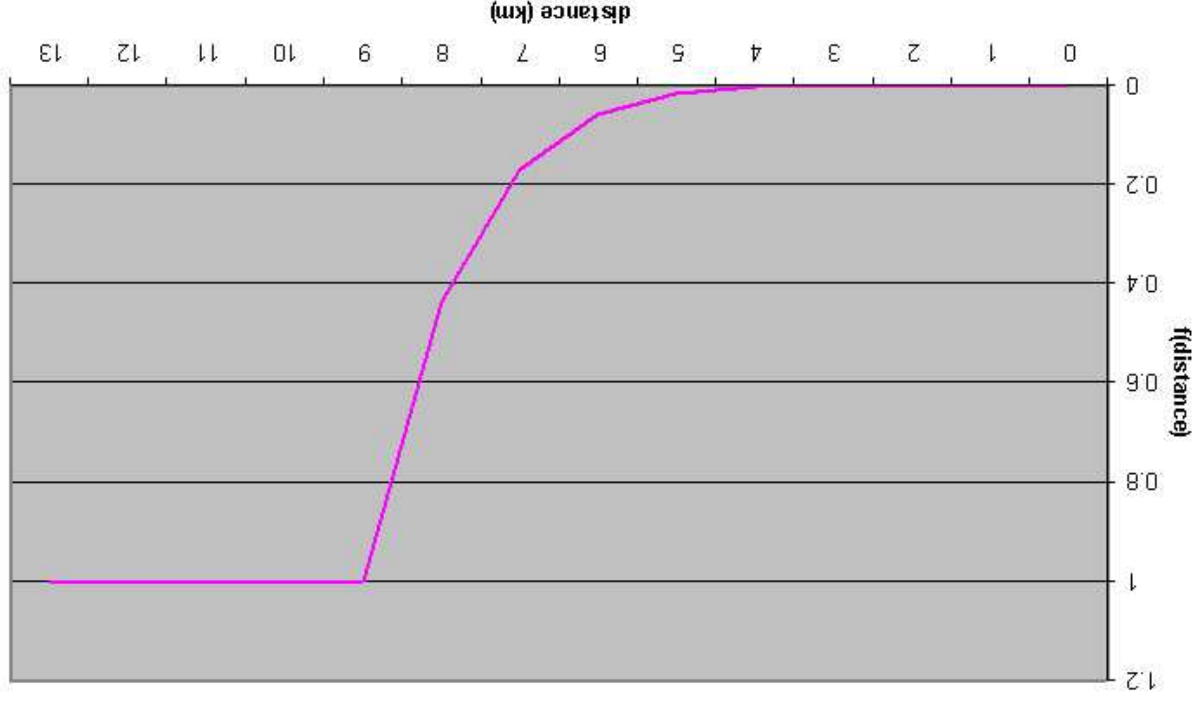
- Générer un point d'arrivée et un point de départ
- Si $f_L(\text{Distance}) > U(0, 1)$, planifier la course
- Sinon, rejeter la course



Adaptation de la moyenne des courses

Si $distance > L$ $f_L(distance) = \frac{distance^T}{L^T}$

Sinon $f_L(distance) = 1$



Pour déterminer la valeur de L : quête dichotomique



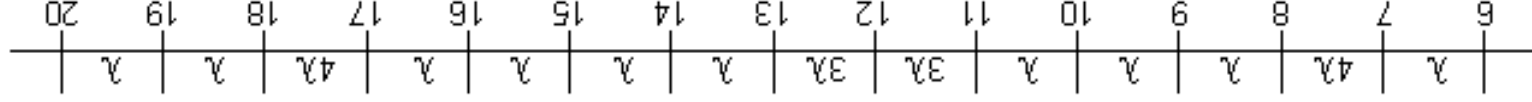
Génération d'une course

Processus de Poisson

2'700 courses en 24 heures

$$\implies \lambda = 2'700 / 86'400 = 1/32$$

Heures de pointe





Statistiques

Le gestionnaire de statistiques stocke toutes les informations liées à chaque événement.

Structure principale

Dictionnaire avec comme clés, les heures des événements. heure :
[nbCourses, [noTaxi, ta, [tsClient, tsAvClient, tsRestation], na,
[nsClient, nsAvClient, nsRestation], [iClient, lAvClient,
lRestation]], [taClient, tsClient, naClient, nsClient], no station : [nb
taxis, nb demandes, capacité], ...], ...



Traitement des événements

Pour tous les événements, lors du traitement les données suivantes sont stockées :

- l'heure de l'évènement.
- le nombre de courses.
- le numéro du taxi affecté au client.
- le nombre de taxis en attente.
- le nombre de taxis en service allant chercher un client.
- le nombre de taxis en service avec un client.
- le nombre de taxis en service retournant en station.
- le nombre de clients en attente.
- le nombre de clients en service.
- l'occupation et la charge des stations.

- le nombre de courses.
- le nombre de taxis en service allant chercher un client, le nombre de taxis en attente ou le nombre de taxis retournant en stations.
- le temps d'attente ou le temps de service retour station suivant l'état du taxi
- la distance retour station dans le cas où le taxi a été intercepté lors ce qu'il retournait en station.
- le nombre de clients en attente.

Événement client



- le nombre de taxis en service allant chercher un client.
- le nombre de taxis en service avec un client.
- le nombre de clients en attente.
- le nombre de clients en service.
- le temps de service du taxi pour aller chercher le client.
- la distance parcourue par le taxi pour aller chercher le client.
- le temps d'attente du client.

Événement charger client



- le nombre de taxis en service avec un client.
- le nombre de taxis en service retournant en station.
- le nombre de clients en service.
- le temps de service du taxi avec le client.
- la distance parcourue par le taxi avec le client.
- le temps de service du client.

Événement poser client



- le nombre de taxis en service retournant en station.
- le nombre de taxis en attente.
- le temps de service du taxi retournant en station.
- la distance parcourue par le taxi pour rentrer en station.

Événement arriver station



Politiques



Politique du plus près

Prendre le taxi se trouvant dans la station la plus proche du client.

Le taxi retourne dans la station la plus proche.

Pas efficace car la station peut être pleine quand il y arrive.





Politique du plus près 2

Prendre le taxi se trouvant dans la station la plus proche du client.

Le taxi retourne dans la station la plus proche sauf si une station

avec un taux d'occupation $> 20\%$ se trouve à moins de $3km$.

Permet d'éviter que le taux d'occupation de certaines stations chute.

Première solution pour le problème des stations pleines.



Politique du plus près avec réservation

Prendre le taxi se trouvant le plus près du client (détournement possible).

Le taxi fait une réservation dans la station la plus proche. Si acceptée s'y rend, autrement fait une réservation dans la 2ème plus proche... Résout le problème d'efficacité rencontré dans les deux politiques précédentes.

Prendre le taxi se trouvant le plus près du client (détournement possible).
Tirage d'un taux de demande aléatoire. Le taxi se rend dans une station où le taux de demande est \geq à celui-ci (Réservation).
Se base sur les statistiques obtenues en cours de simulation.

Politique de la demande



- GUI
- Graphiques
- Comparaisons

Démonstration



- Motivations en baisse
- Complexité croissante du projet
- Problèmes récurrents de performance

Conclusion

