

# Sujet du cours Algorithmes et complexité

## Projet “Taxi autonomes” (2022)

**Note :** Le projet doit être rendu pour le 31 Mai 2022 **inclu**.

**Note :** Vous trouvez dans le moodle les instances pour tester vos algorithmes. Pour des doutes et/ou des questions sur la partie algorithmes n’hésitez pas à m’adresser un email

**Le problème** Ces derniers temps on parle beaucoup des véhicules autonomes. Le problème que nous voulons résoudre concerne la gestion d’une flotte de taxis électriques autonomes. Les clients appellent le centre de service et décrivent le déplacement qui les intéresse, le système évalue s’il existe un véhicule approprié pour répondre à la demande; s’il existe il est affecté à la demande et son itinéraire est défini afin qu’il parvienne au client à l’heure souhaitée où, si cela n’est pas possible, le plus rapidement possible. Le système gère également le mécanisme de recharge et évalue le bénéfice fait avec le service.

Les véhicules se déplacent dans une ville représentée par une liste de rues, pour plus de simplicité toutes les rues sont à double sens. Dans le système de gestion, chaque rue est décrite par une paire des indices numériques, qui correspondent aux intersections (ou places) situées aux extrémités de la rue, et un nombre entier positif, qui indique le temps de parcours mesuré en secondes. Ces informations sont suffisantes pour déterminer la topologie du réseau routier et pour calculer le temps nécessaire pour se déplacer d’un point à un autre du réseau le long d’un chemin donné, ainsi que pour calculer les chemins les plus appropriés.

Le système connaît le nombre de véhicules disponibles et leur autonomie, exprimée par le nombre total maximum de secondes pendant lesquelles ils peuvent se déplacer avant que la batterie se décharge complètement. On connaît également la durée en secondes de la recharge, ce qui ne peut arriver que au siège de la société de gestion. Celui-ci est situé au point index 1 du réseau routier.

Le système dispose de la liste des appels par ordre croissant de réception (dans le cas d’horaires égaux l’ordre donné est respecté). Chaque appel est caractérisé par :

1. l’heure à laquelle il a été reçu ;
2. le nom de famille du client ;
3. les indices des points d’origine et de destination de l’itinéraire demandé ;
4. l’heure minimum avant laquelle le client ne peut pas quitter l’origine ;
5. l’heure maximale à laquelle le client doit arriver à destination.

Pour commencer, un traitement des données est nécessaire : il faut ordonner

1. la liste des clients par ordre alphabétique ;
2. la liste des appels par ordre décroissant de durée du trajet.

La *durée* d’un trajet fait référence au temps minimum qu’il faut à un véhicule pour se déplacer sur le réseau routier du point d’origine au point de destination.

Le système doit alors définir les positions initiales des véhicules sur le réseau routier. Supposez, pour simplifier, que les positions autorisées sont toutes et seulement les extrémités des routes (intersections et carrés). Pour couvrir efficacement

la ville, le système recherche l'ensemble des emplacements les plus éloignés les uns des autres, c'est-à-dire tels que la somme des distances relatives est maximale. Malheureusement ce problème est complexe en termes de calcul, donc vous pouvez le résoudre seulement avec l'heuristique *gloutonne*, décrite ci-dessous.

La partie principale du projet consiste à simuler le déroulement du service. Au début de la journée, chaque véhicule a sa position assignée, dispose d'une batterie entièrement chargée et il est libre (disponible pour le premier passager). Les appels arrivent les uns après les autres, et chaque appel provoque la recherche d'un véhicule capable de la servir. Pouvoir servir un appel signifie que le véhicule respect les conditions (contraintes) énumérées ci-dessous. Si aucun véhicule est en mesure de répondre à l'appel, cela est rejeté. Sinon, un des véhicules est choisi (sur la base du critère discuté ci-dessous) et commence immédiatement à se diriger vers le point d'origine où le client l'attend à une heure comprise entre l'heure minimale de départ et l'heure maximale d'arrivée. Si le véhicule arrive au point d'origine avant l'heure minimum, il reste à l'arrêt à attendre le client (ils sont censés pouvoir le faire sans gêner la circulation). Le véhicule ensuite porte le client à destination, en arrivant avant l'heure d'arrivée maximale (**c'est un des conditions d'acceptation de l'appel**). Le client quitte le véhicule et le système vérifie la charge de la batterie. Si celle-ci est inférieure à 20% de la charge maximale, le véhicule retourne sur le site pour se recharger ; sinon, il attend l'appel suivant <sup>1</sup>. Comme une seule borne de recharge est disponible sur place, il y a le risque que les véhicules doivent attendre leur tour dans la file d'attente dans l'ordre dans lequel ils sont arrivés. Évidemment, le véhicule n'est pas libre de répondre aux appels si il est dans un des conditions suivantes : 1) il est en route vers un client pour aller le prendre; 2) a un passager à bord; 3) il est en train de rentrer au depot pour recharge; 4) il est dans la file d'attente pour recharge; 5) il est en rechargement. Le véhicule est libre et disponible immédiatement après la fin de la charge. Pour éviter les déplacements inutile, un véhicule rechargé ne revient pas à sa position initiale, mais reste en place à attendre le prochain appel qui lui est attribué.

Un véhicule peut servir une appelle si:

1. Il est libre, cet á dire : il n'est pas en route vers un client pour aller le prendre; il n'a pas un passager à bord; il n'est pas en train de rentrer au depot pour recharge; il n'est pas dans la file d'attente pour recharge; il n'est pas en rechargement;
2. Il est capable d'arriver au noeud origine du client, prendre le client á bord et arriver à noeud destination avant l'heure maximale à laquelle le client doit arriver à destination;
3. il a une charge suffisante pour arriver au noeud origine de l'appel, faire le trajet jusqu'á au noeud destination et faire le trajet du noeud destination au point de recharge (même si il ne rentre pas pour recharge á chaque fois, il faut verifier qu'il dois pouvoir le faire).

Le prix payé par le client est proportionnel (pour simplifier, égal) à la durée du trajet calculé en secondes. A ce prix s'ajoute une prime forfaitaire pour la ponctualité cet á dire quand le client part exactement à l'heure minimale du départ indiquée, sans avoir à attendre. Tout retard de départ annule le prime, tandis que les retards d'arrivée ne sont pas autorisés, car le trajet est annulé sans même démarrer.

---

<sup>1</sup>Il est possible que tous les appels ultérieurs nécessitent des déplacements pour lequel la charge restante est insuffisante. Si oui, cette règle heuristique entraînera l'arrêt du véhicule et refusera les clients qu'il pourrait servir après un mise en charge. Les heuristiques par définition ne garantissent pas des solutions optimales.

La simulation sera basée sur le concept de *événements discrets*, qui est décrit en détail dans la section suivante. En résumé, il s'agit de gérer l'état des système d'une part et dans l'autre part gérer un ensemble d'événements caractérisés chacun par l'instant d'occurrence. Au début, le système est dans un état donné et un ensemble d'événements futurs sont déjà connus. Par conséquent, le mécanisme de simulation d'événements discrets procède de manière itérative comme suit :

1. On considère le premier événement (celle qui a le temps minimum) et en fonction des conditions, dépendants de l'état courant du système, modifie l'état lui-même (par exemple, si un appel arrive, le système lui affecte un véhicule, il calcule la trajectoire et définit le véhicule comme indisponible);
2. toujours en fonction de l'événement courant, le système "déclenche" d'autres événements discrets (c'est à dire ils sont générés et misent dans l'ensemble des événements en attente) et à chacun est attaché l'instant du temps;
3. enfin, supprime l'événement traité dans l'ensemble et reviens au point 1.

La simulation s'arrête quand l'ensemble des événements est vide.

À la fin de la simulation il faut rendre un apport détaillé des événements, et la valeur de certains indicateurs de performance liste de suite:

- le nombre d'appels refusés,
- le nombre des recharges effectués,
- le temps total de mouvements des véhicules (en seconds),
- le gain total du service.

Enfin, comme le choix du véhicule auquel attribuer chaque appel et du moment où le rechargement des véhicules est complètement heuristique, il est utile d'essayer d'estimer combien potentiellement le service lui-même est rentable. Calculer l'optimum n'est pas possible car trop complexe <sup>2</sup>. On se borne donc à calculer une surestimation du gain atteignable, qui est basé sur le temps global mis à disposition par les véhicules et le temps minimum requis pour répondre à chaque appel. La manière d'obtenir cette estimation est décrite ci-dessous.

**Le projet** Le programme à réaliser reçoit en entrée (i.e., par ligne de commande) les noms de trois fichiers texte, qui contiennent les données.

Dans l'ordre:

1. le premier fichier contient le réseau routier,
2. le deuxième fichier contient les caractéristiques des véhicules,
3. le troisième fichier contient les appels des clients.

Le fichier texte représentant le réseau routier affiche le numéro en première ligne de nœuds (carrefours et places où les véhicules peuvent stationner et prendre ou déposer les clients) et le nombre de rues (arcs). Par exemple,

10 20

---

<sup>2</sup>Sans oublier qu'il faut aussi distinguer le cas où les appels sont révélés un par un, comme dans la simulation que nous menons dans le cas, où les appels sont tous connus a priori et cette connaissance est exploitée pour mieux les servir.

Indique que le réseau a 10 points/noeuds et 20 rues/arcs. Ci-dessous un exemple de la liste des rues/arcs, une par ligne, représenté par les deux indices des points/noeuds extrêmes et le temps de parcours en secondes.

```
3 10 89
2 8 132
...
```

La première ligne indique que les points 3 et 10 sont reliés par une route avec un temps de parcours de 89 secondes, la seconde ligne indique que les points de 2 et 8 sont reliés par une route avec un temps de parcours de 132 secondes, et ainsi de suite. Les routes sont toutes à double sens et avec des temps de parcours identiques dans les deux sens. Pour plus de simplicité, nous ignorons les feux de circulation, les temps de virage, etc....

Le fichier texte décrivant les véhicules est constitué d'une seule ligne. L'ordre indique le nombre de véhicules, la durée en secondes de l'intervalle de service, la durée de vie de la batterie en secondes de mouvement et la durée en secondes du temps de recharge. Par exemple:

```
3 3600 1000 200
```

indique qu'il y a 3 véhicules, que l'horizon temporel du service est de 3 600 secondes, l'autonomie des véhicules de 1000 secondes en mouvement, et enfin la durée de la recharge est de 200 secondes.

Le fichier texte qui stocke les appels affiche leur numéro sur la première ligne. Chaque ligne suivante décrit un appel, rapportant le temps (exprimé en secondes depuis le début du service), le nom du client (pour simplifier est, toujours défini avec un seul mot de 25 caractères maximum), les indices des points d'origine et de destination du trajet nécessaire, les délais minimum et maximum dans lesquels le client souhaite effectuer son voyage (exprimé en secondes à partir du début de service) et le prix gagné si le client est chargé exactement à l'heure souhaitée.

Par exemple:

```
200
1107 Palmieri 2 7 1765 2994 1018
1335 Hijjoui 4 5 1674 3388 2545
...
```

indique qu'il y a 200 appels, dont le premier se produit après 1 107 secondes depuis le début du service (et considéré comme instantané pour simplifier), nécessite pour M. Palmieri un trajet du point 2 au point 7 commençant au plus tôt 1 765 secondes et ne se terminant pas plus de 2 994 secondes après le début du service ; la prime de ponctualité est égale à 1,018, à ajouter au tarif (égale à la durée du trajet du point 2 au point 7) seulement si le voyage commence effectivement exactement à 1 765.

La première étape du projet consiste à classer les clients par ordre alphabétique. Pour simplicité, nous supposons que tous les clients effectuent un seul appel et ont des noms de famille différents. Donc le nombre de clients est égal au nombre d'appels. Vous devrez les imprimer à l'écran par ordre alphabétique, un par ligne, précédé d'une ligne avec le mot clé **Clients:**.

Par exemple:

```
Clients:
Bartolini
Dolciotti
```

Fiore

...

Ensuite, vous devez commander les trajets nécessaires, y compris ceux qui ne seront pas desservi, pour une durée décroissante du trajet de l'origine à la destination. A la même durée, ils seront triés par heure d'appel croissante. Nous allons imprimer une ligne avec le mot-clé **Voyage** : à l'écran, suivi des appels, un par ligne, pour lesquels l'heure de l'appel, le nom du client et la durée en secondes, suivi de la liste des indices des points visité le long de l'itinéraire (du point d'origine au point de destination).

Par exemple:

**Voyages:**

661 Fiore 699

1107 Hijjoui 675

302 Dolciotti 587

...

Avant de lancer la simulation il faut définir les positions initiales des véhicules sur le reseau. Nous procedons comme decrit desuite. Le premier véhicule est placé (c'est-à-dire dans le point 1 du réseau). Le seconde est placé au point le plus éloigné possible du début (c'est-à-dire au point qui nécessite le maximum de temps de trajet pour être atteint par le premier), le troisième point est choisi comme le plus éloigné des deux premiers (c'est-à-dire avec la somme de temps de parcours maximum depuis les deux points précédents), et ainsi de suite jusqu'au dernier véhicule. En cas de distance totale égale, le point d'index minimum est choisi. Chaque véhicule doit se tenir dans un endroit différent. Après cette phase, les positions initiales des véhicules seront imprimées sur une seule ligne, précédées du mot **Positions** : clé et séparés par des espaces. Par exemple:

**Positions:** 1 3 2

Pour la simulation, nous marquons le temps en secondes à partir du principe de service. Les véhicules sont initialement tous libres, ont une batterie complètement chargée et sont dans les positions calculées immédiatement ci-dessus. Considérons les types d'événements possibles suivants discret:

1. une appel;
2. l'arrivé d'un client à destination;
3. le retour au dépôt pour recharge d'un vehicule;
4. la fin de la recharge d'un vehicule.

Par souci de simplicité, chargeons en memoire simplement tous les événements d'appel, même si nous traiterons chaque appel comme si nous ne connaissions pas les suivants, puisque nous simulons la gestion d'un service en temps réel.

Un événement de type "appel" modifie l'état du système parceque il declenche la recherche d'un véhicule approprié à desservir l'appel. S'il le trouve, le véhicule s'engage, il repart immédiatement vers le point d'origine, où il s'arrête pour attendre le client (s'il n'est pas encore arrivé), il le charge (instantanément) et continue jusqu'au point de destination. Les deux trajets sont évidemment le plus court chemin possible sur le réseau. Pour plus de simplicité, les véhicules disponibles pour desservir un appel ne sont que ceux libres au moment de l'appel. Parmi ces véhicules, vous pouvez choisir :

- celui qui arrive le premier au point de destination;

- en cas de parité, celui qui arrive le plus tard au point d'origine est choisi (pour réduire le risque de faire attendre le véhicule pour rien);
- en cas de parité supplémentaire, celui avec l'indice minimum est choisi.

Si aucun véhicule n'est libre, l'appel est rejeté.<sup>3</sup>. Un événement de type appel déclenche également un événement ultérieur : si l'appel est servi, en effet, le temps dans lequel le client arrive à destination est déterminé : il s'agit d'évaluer la durée des deux trajets depuis la position actuelle du véhicule jusqu'au point de départ du trajet et de celui-ci jusqu'au point de destination, ainsi que l'éventuelle attente au point d'origine. Le nouvel événement d'arrivée à destination est ajouté à la collection.

Un événement d'arrivée à destination modifie l'état du système en libérant le véhicule portant le client. Pour économiser de l'énergie, le véhicule ne revient pas à sa position de départ, mais reste immobile dans la nouvel emplacement. Si cependant la charge de la batterie est strictement inférieure à 20% d'autonomie, le véhicule ne se libère pas, mais il se rend immédiatement au dépôt pour se recharger et bien évidemment il choisi le plus court chemin. L'événement déclenche alors un événement de retour au dépôt pour recharge à l'heure déterminée par la durée du voyage.

Un événement de retour au dépôt met le véhicule en file d'attente pour être rechargé. Le passé dans la file d'attente dépend du nombre de véhicules dans la file d'attente et de la durée de la recharge elle-même, mais aussi du temps de recharge résiduelle du véhicule qui se trouve actuellement en tête de la file d'attente. Tous ces facteurs vous permettent de prédire l'heure à laquelle le véhicule qui vient d'arriver au dépôt mettra fin au rechargement lui-même, et déclenchera ainsi l'événement de fin de rechargement.

Un événement de fin de recharge libère à nouveau le véhicule chargé, qui reste au dépôt jusqu'à ce que il reçoive un nouvel appel.

Il est évidemment possible que des événements simultanés se produisent. Dans ce cas, par convention, nous prendrons l'ordre suivant, à parité de temps :

1. fin de la recharge d'un véhicule,
2. l'arrivée à destination d'un client,
3. rentrée d'un véhicule,
4. appel d'un client.

L'objectif c'est de rendre tout de suite disponibles pour les appels le véhicule qui termine la recharge au même temps que l'arrivée de l'appel lui-même ou également le véhicule qui termine un service au même temps que l'arrivée de l'appel lui-même<sup>4</sup>. Si deux événements sont simultanés et de même nature, celle relative au véhicule est considérée en premier indice minimal. Pour les événements d'appels simultanés, qui n'ont pas d'index de véhicule, l'ordre initial des appels est suivi.

La simulation produit une liste d'événements ordonnés comme indiqué ci-dessus. Il imprimera sur l'écran une première ligne avec le mot-clé **Events** :, suivi de la liste des événements, reportant chacun sur une ligne séparée l'heure, le type d'événement (**CALL**, **END\_SERVIZIO**, **RETURN\_SEDE**, **END\_RELOAD**), l'indice du véhicule impliqué (0

<sup>3</sup>La possibilité qu'un véhicule serve un client ou soit en train de recharger, mais qu'il soit libre suffisamment tôt pour pouvoir servir le client est exclu.

<sup>4</sup>Et aussi pour s'assurer que, si la destination d'un client est le siège social et que le véhicule s'arrête pour se recharger, le client descend du véhicule avant qu'il ne fasse la queue à la colonne

pour les événements d'appel, n'impliquant pas de véhicules) et le nom de famille du client concernées (aucune pour les événements de retour au dépôt et de fin de rechargement).

Par exemple:

```
Eventi:
302 CHIAMATA 0 Dolciotti
432 CHIAMATA 0 Rainesi
661 CHIAMATA 0 Fiore
...
```

La liste des événements se termine par quelques informations récapitulatives, c'est-à-dire le nombre d'appels refusé, le nombre de recharges effectuées, le temps total de déplacement des véhicules et le bénéfice total du service. Chaque information occupe une ligne et est précédée de un mot-clé approprié, comme dans l'exemple suivant :

```
Appels non desservis:
Recharges:
Temp total:
Gain: 8339
```

Enfin, nous surestimerons le gain que l'on peut tirer des appels donnés. Cette estimation est basée sur l'ignorance de certaines contraintes fondamentales du problème, telles que les intervalles de temps demandés par les clients, et les temps de trajet nécessaires pour se déplacer chaque véhicule vers les points d'origine des appels, ainsi que depuis les points de destination au dépôt en cas de recharge, temps d'attente dans la file d'attente pour la recharge et temps d'attente du client lorsque le véhicule arrive en avance. Avec toutes ces simplifications, vous pouvez voir le service comme essayer de remplir un "sac à dos virtuel" autant que possible :

- la capacité correspond au temps total disponible, égal au nombre de véhicules multiplié par la durée de l'horizon temporel de service <sup>5</sup> ;
- les objets sont les appels sachant que:
  - le volume c'est le simple temps pour aller de l'origine à la destination;
  - le valeur des objets c'est le gain obtenu pour chaque trajet plus le prime pour la ponctualité, que on hypothèse de pouvoir toujours obtenir.

La solution du problème global du sac à dos fournit une valeur qui n'est certainement pas inférieure au gain optimal réalisable. Probablement, l'estimation est beaucoup plus élevée à l'optimum, mais au moins il peut donner une idée de son ordre de grandeur (ainsi que du méthodes généralement utilisées pour résoudre les problèmes d'optimisation combinatoire). Ce devis doit être imprimé à l'écran précédé du mot-clé UB:. Par exemple:

```
UB: 18794
```

---

<sup>5</sup> Strictement parlant, chaque véhicule est un sac à dos en soi, mais pour éviter de compliquer le problème, nous ignorons aussi la distinction entre les véhicules, comme s'ils pouvaient se prêter du temps.