

Laboratorio di Algoritmi

Progetto “Taxi” (giugno 2019)

Nota: La scadenza del progetto è fissata per giovedì 13 giugno **compreso**.

Nota: Si consiglia di consultare sulla pagina web il documento che riporta le avvertenze utili per lo svolgimento del progetto. Si consiglia anche di verificare di tanto in tanto gli aggiornamenti a questo documento, che potrà essere aggiornato con la correzione di eventuali errori e le risposte ai dubbi degli studenti.

Il problema Si fa un gran parlare, ultimamente, dei veicoli a guida autonoma. Il problema qui considerato riguarda la gestione di una flotta di taxi elettrici a guida autonoma. Bisogna:

- estrarre dai dati un certo numero di informazioni ausiliarie;
- assegnare ai veicoli le chiamate dei clienti in modo da servirli rispettando gli orari indicati da questi ultimi;
- decidere di conseguenza le traiettorie dei veicoli;
- richiamare i veicoli in sede quando è necessario ricaricarne la batteria;
- valutare il guadagno complessivo ottenuto;
- confrontarlo con una opportuna stima per eccesso del guadagno ottenibile.

I veicoli si muovono in una città rappresentata da un elenco di strade, per semplicità tutte a doppio senso. Ogni strada è definita da una coppia di indici numerici, che corrispondono agli incroci o alle piazze poste sugli estremi della via, e da un numero intero positivo, che ne indica il tempo di percorrenza misurato in secondi. Queste informazioni sono sufficienti a determinare la topologia della rete stradale e a calcolare i tempi necessari a muoversi da un punto a un altro della rete lungo un percorso dato, nonché a calcolare i percorsi più opportuni.

È dato il numero dei veicoli, così come la loro autonomia, espressa dal numero massimo totale di secondi durante i quali possono muoversi prima di scaricare completamente la batteria. È anche data la durata in secondi della ricarica, che può avvenire solo nella sede della ditta di gestione, dove è disponibile un solo apparecchio di ricarica. Se più veicoli ne hanno bisogno, aspettano in coda nell'ordine di arrivo. La sede si trova nel punto di indice 1 della rete stradale.

La prima decisione da prendere è dove posizionare i veicoli inizialmente sulla rete stradale. Supponiamo per semplicità che le posizioni lecite siano tutti e soli gli estremi delle strade (incroci e piazze). Per coprire efficacemente la città, si cerca l'insieme di posizioni reciprocamente più distanti tra loro, vale a dire quelle tali che la somma delle relative distanze sia massima. Siccome questo problema è computazionalmente complesso, lo risolveremo con un'euristica *greedy*, descritta nel seguito.

Bisogna quindi trattare l'elenco delle chiamate, ognuna delle quali è caratterizzata da:

- l'ora in cui viene ricevuta;
- il cognome del cliente;
- gli indici dei punti di origine e destinazione del tragitto richiesto;

- l'ora minima prima della quale il cliente non può partire dall'origine;
- l'ora massima entro la quale il cliente deve arrivare alla destinazione.

L'elenco è in ordine di chiamata crescente (in caso di ore coincidenti, si rispetta comunque l'ordine dato). Per cominciare, sono richieste alcune elaborazioni documentali: bisogna generare

- l'elenco dei clienti in ordine alfabetico;
- l'elenco della chiamate in ordine di durata decrescente della corsa.

dove per *durata* di una corsa si intende il tempo minimo necessario a un veicolo a muoversi sulla rete stradale dal punto di origine al punto di destinazione.

La parte principale del progetto consiste nel simulare l'andamento del servizio. Al principio della giornata ogni veicolo è nella posizione assegnatagli, ha la batteria completamente carica ed è libero. Le chiamate arrivano una dopo l'altra, e ogni chiamata provoca la ricerca di un veicolo che sia in grado di servirla, cioè rispetti opportune condizioni elencate più avanti. Se nessun veicolo è in grado di servire la chiamata, questa viene rifiutata. Altrimenti, uno dei veicoli viene scelto (in base al criterio discusso nel seguito) e parte immediatamente per raggiungere il punto di origine dove il cliente lo aspetta in un momento compreso fra l'ora minima di partenza e l'ora massima di arrivo. Se il veicolo arriva al punto di origine prima dell'ora minima, rimane fermo ad aspettare il cliente (per semplicità, si suppone che possa farlo senza intralciare il traffico). Poi porta il cliente a destinazione, dove deve arrivare sempre in un momento compreso fra l'ora minima di partenza e l'ora massima di arrivo. Il cliente abbandona il veicolo e questo rimane fermo in attesa della chiamata successiva. Siccome i veicoli hanno un'autonomia limitata, però, se la carica della batteria è inferiore al 20% dell'autonomia massima, il sistema lo invia alla sede per farlo ricaricare. Ovviamente, il veicolo non è libero mentre sta raggiungendo o servendo un utente, mentre sta rientrando in sede e mentre attende la ricarica o si sta ricaricando. Torna libero immediatamente alla fine della ricarica. Per evitare spostamenti inutili, un veicolo ricaricato non torna alla posizione iniziale, ma rimane in sede ad aspettare la chiamata successiva assegnatagli.

Un veicolo è in grado di servire una chiamata se:

1. è libero, cioè non sta servendo altri clienti, tornando in sede, aspettando in coda la ricarica o ricaricandosi;
2. è in grado di raggiungere l'origine della chiamata, caricare l'utente e raggiungere la destinazione entro l'ora massima di arrivo indicata;
3. ha carica sufficiente a raggiungere l'origine della chiamata, da lì la destinazione e dalla destinazione la sede per ricaricarsi (anche se non torna ogni volta in sede a ricaricarsi, deve poter essere in grado di farlo).

Il prezzo pagato dal cliente è proporzionale (per semplicità, uguale) alla durata della corsa in secondi. A questo prezzo si aggiunge un premio fisso di puntualità nel caso in cui il cliente parta esattamente all'ora minima di partenza indicata, senza dover attendere. Qualunque ritardo in partenza cancella il premio, mentre i ritardi in arrivo non sono permessi, dato che l'intera corsa viene cancellata senza nemmeno cominciare.

La simulazione si baserà sul concetto di *evento discreto*, che viene descritto in dettaglio nella sezione seguente. In sintesi, si tratta di gestire da un lato lo stato del sistema, dall'altro un insieme di eventi, ciascuno caratterizzato da un istante di occorrenza. All'inizio, il sistema è in un dato stato e un insieme di eventi futuri sono già noti. Quindi, il meccanismo della simulazione a eventi discreti procede iterativamente in questo modo:

1. considera il primo evento dell'insieme (quello cronologicamente minimo) e in base a regole che dipendono dallo stato corrente del sistema, modifica lo stato stesso (per esempio, se arriva una chiamata, il sistema gli assegna un veicolo, ne definisce la traiettoria e marca il veicolo come impegnato);
2. sempre in base all'evento corrente, "innesca" altri eventi discreti (cioè li genera e li aggiunge all'insieme) in istanti futuri determinati;
3. infine, rimuove l'evento elaborato dall'insieme e torna al punto 1.

La simulazione termina quando l'insieme degli eventi è vuoto.

Al termine della simulazione, si richiede un rapporto dettagliato degli eventi, e il valore di alcuni indicatori di prestazione riassuntivi:

- il numero di chiamate rifiutate,
- il numero di ricariche effettuate,
- il tempo totale di movimento dei veicoli,
- il guadagno totale del servizio.

Per concludere, siccome la scelta del veicolo a cui assegnare ogni chiamata e del momento in cui ricaricare i veicoli è del tutto euristica, è utile cercare di valutare quanto potenzialmente il servizio stesso sia redditizio. Calcolare l'ottimo è computazionalmente complesso¹. Ci limitiamo quindi a calcolare una stima per eccesso del guadagno realizzabile, basata sul tempo complessivo messo a disposizione dai veicoli e del tempo minimo richiesto per servire ciascuna chiamata.

Il progetto Il programma da realizzare carica i dati da tre file di testo il cui nome va fornito nella linea di comando. Nell'ordine:

1. il primo file rappresenta la rete stradale,
2. il secondo file rappresenta le caratteristiche dei veicoli,
3. il terzo file rappresenta le chiamate degli utenti.

Il file di testo che rappresenta la rete stradale riporta nella prima riga il numero di punti (incroci e piazze) nei quali possono stazionare i veicoli ed essere caricati e scaricati i clienti e il numero di strade. Per esempio,

10 20

indica che la rete ha 10 punti e 20 strade. Segue l'elenco delle strade, una per riga, rappresentate dai due indici dei punti estremi e dal tempo di percorrenza in secondi. Per esempio:

3 10 89

2 8 132

...

indica che i punti 3 e 10 sono collegati da una strada che si percorre in 89 secondi, i punti 2 e 8 da una strada che si percorre in 132 secondi, e così via. Le strade sono tutte a doppio senso di marcia e con tempi di percorrenza identici nei due versi. Ignoriamo per semplicità semafori, tempi di svolta, ecc...

¹Senza contare che bisognerebbe anche distinguere il caso in cui le chiamate vengono rivelate una alla volta, come nella simulazione che si sta conducendo in questo caso, da quello in cui le chiamate sono tutte note a priori e questa conoscenza viene sfruttata per servirle meglio.

Il file di testo che descrive i veicoli consiste in una sola riga. Nell'ordine, sono riportati il numero dei veicoli, la durata in secondi dell'intervallo di servizio, l'autonomia in secondi di movimento della batteria, la durata in secondi della ricarica. Per esempio:

```
3 3600 1000 200
```

indica che ci sono 3 veicoli, che l'orizzonte temporale del servizio è di 3 600 secondi, l'autonomia dei veicoli di 1,000 secondi in movimento, e infine la durata della ricarica di 200 secondi.

Il file di testo che conserva le chiamate riporta nella prima riga il loro numero. Ogni successiva riga descrive una chiamata, riportando l'ora (espressa in secondi a partire dal principio del servizio), il cognome del cliente (per semplicità, sempre una singola parola di 25 caratteri al massimo), gli indici dei punti di origine e destinazione del tragitto richiesto, l'ora minima e massima entro le quali il cliente deve effettuare il proprio viaggio (espresse in secondi a partire dal principio del servizio) e il premio guadagnato se il cliente viene caricato esattamente al momento desiderato. Per esempio:

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

indica che ci sono 200 chiamate, la prima delle quali avviene dopo 1 107 secondi dal principio del servizio (e considerata istantanea per semplicità), richiede per il sig. Palmieri uno spostamento dal punto 2 al punto 7 che cominci non prima di 1 765 secondi e termini non oltre 2 994 secondi dal principio del servizio; il premio di puntualità è pari a 1 018, da aggiungere al prezzo della corsa (pari alla durata del percorso dal punto 2 al punto 7) solo se il viaggio inizia effettivamente al secondo 1 765.

Il primo punto del progetto è disporre i clienti in ordine alfabetico. Per semplicità, assumeremo che tutti i clienti facciano una sola chiamata e abbiano cognomi diversi. Quindi il numero dei clienti è uguale a quello delle chiamate. Bisognerà stamparli a video in ordine alfabetico, uno per riga, preceduti da una riga con la parola chiave **C**lienti:. Per esempio:

```
Clienti:
Bartolini
Dolciotti
Fiore
...
```

Quindi, bisogna ordinare i viaggi richiesti, compresi quelli che non sarà possibile servire, per durata decrescente del tragitto da origine a destinazione. A pari durata, si ordineranno per ora di chiamata crescente. Si stamperà a video una riga con la parola chiave **V**iaggi:, seguita dalle chiamate, una per riga, per le quali si riporterà l'ora di chiamata, il cognome del cliente e la durata in secondi, seguita dall'elenco degli indici dei punti attraversati lungo il percorso (dal punto di origine a quello di destinazione). Per esempio:

```
Viaggi:
661 Fiore 699
1107 Hijjoui 675
302 Dolciotti 587
...
```

Prima di eseguire la simulazione, occorre determinare le posizioni iniziali dei veicoli sulla rete. Lo faremo tenendo il primo veicolo in sede (cioè nel punto 1 della rete), ponendo il secondo nel punto più lontano possibile dal primo (cioè nel punto che richiede il massimo tempo di viaggio per essere raggiunto dal primo), il terzo nel punto più lontano dai primi due (cioè con somma massima dei tempi di viaggio dai punti precedenti), e così via sino all'ultimo veicolo. In caso di pari distanza totale, si sceglie il punto di indice minimo. Ogni veicolo deve stare in un punto diverso. Terminata questa fase, si stamperanno le posizioni iniziali dei veicoli su una sola riga, precedute dalla parola chiave **Posizioni:** e separate da spazi. Per esempio:

Posizioni: 1 3 2

Per la simulazione, scandiamo il tempo in secondi a partire dal principio del servizio. I veicoli sono inizialmente tutti liberi, hanno la batteria completamente carica e stanno nelle posizioni calcolate subito sopra. Consideriamo i seguenti possibili tipi di eventi discreti:

1. una chiamata;
2. l'arrivo a destinazione di un cliente;
3. l'arrivo in sede di un veicolo che deve ricaricare la batteria;
4. il termine della ricarica di un veicolo.

Per semplicità, carichiamo subito tutti gli eventi chiamati nell'insieme, anche se elaboreremo ogni chiamata come se non conoscessimo le successive, dato che stiamo simulando la gestione di un servizio in tempo reale.

Un evento di tipo chiamata modifica lo stato del sistema cercando un veicolo adatto a servire la chiamata. Se lo trova, il veicolo diventa impegnato, parte subito verso il punto di origine, dove si ferma ad aspettare il cliente (se questi non è ancora arrivato), lo carica (istantaneamente), e prosegue verso il punto di destinazione. I due percorsi sono ovviamente i più brevi possibili sulla rete. Per semplicità, i veicoli disponibili per servire una chiamata sono solo quelli liberi al momento della chiamata. Fra questi veicoli si sceglie:

- quello che arriva prima al punto di destinazione;
- in caso di parità, si sceglie quello che arriva più tardi al punto di origine (per ridurre il rischio di tenere il veicolo in attesa per niente);
- in caso di ulteriore parità, si sceglie quello di indice minimo.

Se nessun veicolo è libero, si rifiuta la chiamata². Un evento di tipo chiamata innesca anche un evento successivo: se la chiamata viene servita, infatti, è determinata l'ora in cui il cliente arriva a destinazione: si tratta di valutare la durata dei due percorsi dalla posizione corrente del veicolo al punto di origine della corsa e da questo al punto di destinazione, nonché l'eventuale attesa nel punto di origine. Il nuovo evento di arrivo alla destinazione va aggiunto all'insieme.

Un evento di arrivo a destinazione modifica lo stato del sistema rendendo libero un veicolo. Per risparmiare energia, il veicolo non torna alla posizione iniziale, ma rimane fermo nella nuova posizione. Se però la carica della batteria è strettamente inferiore al 20% dell'autonomia, il veicolo non torna libero, ma va immediatamente in sede a ricaricarsi seguendo il percorso più breve. L'evento innesca quindi un evento di arrivo in sede, all'ora determinata dalla durata del percorso.

²A rigore, questo spreca la possibilità che un veicolo stia servendo un cliente o si stia ricaricando, ma torni libero abbastanza presto da poter servire il cliente.

Un evento di arrivo in sede mette il veicolo in coda per la ricarica. La permanenza in coda dipende dal numero di veicoli in coda e dalla durata della ricarica stessa, ma anche dal tempo residuo di ricarica del veicolo che attualmente è in testa alla coda stessa. In base a questi fattori si può stimare l'ora alla quale il veicolo appena arrivato in sede terminerà la ricarica stessa. Di conseguenza, si può anche innescare quest'altro evento.

Un evento di termine della ricarica, rende nuovamente libero un veicolo, che rimane in sede finché non gli viene assegnata una nuova chiamata.

È ovviamente possibile che si verifichino eventi simultanei. In questo caso, per convenzione, assumeremo il seguente ordine, a parità di orario:

1. termine della ricarica di un veicolo,
2. arrivo in sede di un veicolo,
3. arrivo a destinazione di un cliente,
4. chiamata di un cliente.

Questo serve a rendere disponibili per le chiamate i veicoli che si liberano da una ricarica o da un servizio contemporaneamente alla chiamata stessa. Se due eventi sono simultanei e dello stesso tipo, si considera per primo quello relativo al veicolo di indice minimo. Se vi sono eventi simultanei di chiamata, che non hanno un indice di veicolo, si rispetta l'ordine originale del file dei dati.

La simulazione produce un elenco di eventi ordinati come sopra indicato. Si stamperà a video una prima riga con la parola chiave **Eventi:**, seguita dall'elenco degli eventi, riportando di ciascuno in una riga separata l'ora, il tipo di evento (**CHIAMATA**, **FINE.SERVIZIO**, **RIENTRO.SEDE**, **FINE.RICARICA**), l'indice del veicolo coinvolto (0 per gli eventi di chiamata, che non coinvolgono veicoli) e il cognome del cliente coinvolto (nessuno per gli eventi di rientro in sede e fine ricarica). Per esempio:

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

L'elenco degli eventi termina con alcune informazioni riassuntive, cioè il numero di chiamate rifiutate, il numero di ricariche effettuate, il tempo totale di movimento dei veicoli e il guadagno totale del servizio. Ciascuna informazione occupa una riga ed è preceduta da un'opportuna parola chiave, come nel seguente esempio:

```
Rifiuti:
Ricariche:
Tempo totale:
Guadagno: 8339
```

Per concludere, si farà una stima per eccesso del guadagno ottenibile dalle chiamate date. Questa stima si basa sul fatto di ignorare alcuni vincoli fondamentali del problema, come gli intervalli temporali richiesti dai clienti, e i tempi di viaggio necessari a spostare ogni veicolo fino ai punti di origine delle chiamate, nonché dai punti di destinazione alla sede in caso di ricarica, i tempi di attesa in coda per la ricarica e i tempi di attesa del cliente quando il veicolo arriva in anticipo. A questo punto, si può vedere il servizio come il tentativo di riempire il più possibile uno "zaino virtuale":

- la capacità corrisponde al tempo totale disponibile, pari al numero dei veicoli moltiplicato per l'ampiezza dell'orizzonte temporale del servizio³;
- gli oggetti corrispondono alle singole chiamate, dove:
 - il volume corrisponde al solo tempo di spostamento da origine a destinazione;
 - il valore corrisponde al guadagno fornito dalla corsa più il premio di puntualità, che si ipotizza di poter sempre ottenere.

La soluzione del problema di zaino complessivo fornisce un valore che è certamente non inferiore al guadagno ottimo realizzabile. Probabilmente, la stima è molto superiore all'ottimo, ma quanto meno può dare un'idea del suo ordine di grandezza (oltre che dei metodi tipicamente utilizzati per affrontare problemi di Ottimizzazione Combinatoria). Questa stima deve essere stampata a video preceduta dalla parola chiave UB: . Per esempio:

UB: 18794

Chiarimenti

26 maggio Rispetto alla prima versione pubblicata sono state fatte le seguenti modifiche:

1. si sono pubblicati alcuni dati con le relative soluzioni^{4 5};
2. si è corretta e completata la descrizione del formato di stampa richiesto per i dati e i risultati;
3. si è aggiunto un criterio secondario di ordinamento fra eventi: se due eventi sono simultanei e dello stesso tipo, viene gestito per primo quello relativo al veicolo di indice minimo;
4. si è chiarito meglio il meccanismo di gestione della ricarica: per poter servire una chiamata un veicolo deve non solo essere libero e in grado di portare l'utente a destinazione entro la finestra temporale, ma anche avere una carica residua sufficiente a tornare in sede al termine del servizio. Il veicolo torna in sede solo se la carica al termine è inferiore al 20% dell'autonomia, ma deve sempre avere la possibilità di tornare in sede;
5. dalla regola aggiunta deriva che un veicolo potrebbe avere almeno il 20% di carica, ma non averne abbastanza per servire nessuna delle chiamate successive; in questo caso, per semplicità rimane fermo e inutilizzato per il resto della simulazione, anche se sarebbe più ragionevole richiamarlo in sede; per ridurre questo effetto indesiderato (purtroppo frequente), si è aumentata al 20% dell'autonomia la soglia che inizialmente era del 10%.

Alcuni studenti mi segnalano che non è necessario rappresentare esplicitamente la coda dei veicoli che attendono la ricarica in sede. Confermo: su questo punto, considererò equivalenti i progetti che fanno uso di una coda esplicita e quelli che non la usano.

³A rigore, ogni veicolo è uno zaino a sé, ma per evitare di complicare il problema ignoriamo anche la distinzione tra i veicoli, come se si potessero prestare tempo a vicenda.

⁴Le soluzioni non sono state verificate: avendo risolto il problema il più in fretta possibile, si prega di segnalare eventuali errori, che verranno corretti al più presto.

⁵I risultati mostrano che la regola secondo cui si assegna una chiamata a un veicolo e da quel momento considera il veicolo impegnato è molto più stupida di quanto paresse a prima vista, ma la manteniamo valida per evitare di creare confusione.

28 maggio

- si è aggiunto alla descrizione del progetto il formato in cui stampare le informazioni riassuntive sul servizio (numero di chiamate rifiutate, numero di ricariche e tempo totale di movimento dei veicoli);
- si sono aggiunte tali informazioni alle soluzioni;
- si è corretta l'indicazione del veicolo per gli eventi di **FINE_RICARICA**, che era sbagliata.

5 giugno

- alcuni clienti facevano chiamate multiple a causa di duplicati nella lista di cognomi usata per generarle: sono state corrette distinguendo i cognomi;
- nel fissare le posizioni iniziali dei veicoli, ognuno deve stare in un punto diverso (questo era dato per implicito, ora è esplicitamente richiesto nel testo);
- per errore, nelle ultime due soluzioni alcuni veicoli, terminata la ricarica, rimanevano in sede invece di andare a servire ulteriori chiamate: l'errore è stato corretto e le soluzioni aggiornate.