

Chapter 1

La rete viaria urbana

Formalizzazione, tutta la roba dal libro di Gaetano.

Chapter 2

Semafori Intelligenti

2.1 Semaforizzazione adattiva

Perché fare i semafori che si adattano in tempo reale?

Reattività alle condizioni inaspettate nel breve periodo

Reattività agli incidenti

2.1.1 Il dilemma dell'ottimizzazione

In fondo è un modo di migliorare l'offerta per il traffico veicolare privato, causandone l'aumento.

Inoltre incidenti, traffico smodato e imprevedibile sono tutte prerogative del trasporto privato. Si farebbe prima a cercare di curare il bisogno di trasporto privato che ad arginarne gli effetti dannosi.

L'unica ragione per fare ricerca nel campo dell'ottimizzazione semaforica è quella di vedere se attraverso i metodi euristici si riescono ad individuare soluzioni efficaci nel medio e lungo termine

2.2 Tipi di ottimizzazione

Tipi di ottimizzazione, roba dal libro.

2.2.1 BALANCE

- Il modello è meglio o peggio di TRE ?
- Perché usare TRE che è molto più lento?
- come funziona BAL e cosa può fare?

Balance ha un modellino *mesoscopico* tipo gltm (n realtà è proprio come GLTM al secondo) e fa un taglio delle reti intorno alle junction che vuole controllare. Sugli ingressi alle sottoreti (una per ogni junction) usa profili di flusso costanti, ma se può usa i flussi uscenti di una junction per determinare i flussi entranti in una a valle (propagazione).

Il modello viene usato per ricavare le funzioni di costo FERMATE, LUNGHEZZA CODE (in realtà numero di veh in coda) e PERDITEMPO. Lui vede le code come F-E perché i suoi archi sono in realtà le corsie di svolta, ed usa una lunghezza MASSIMA per le code.

Punti di forza di Balance:

- Fa tante intersezioni
- E' veloce
- Aggiusta anche le durate degli stage

Punti deboli:

- Non è detto che le intersezioni si parlino tanto bene
- Non guarda avanti
- non vede l'arco ma solo l'approccio: forse una volta che la coda ha raggiunto il sensore per lui tutte le situazioni sono uguali, e sotto carico non gli cambia più niente
- Probabilmente tende a massimizzare la capacità dove è più richiesta, favorendo lo scorrimento ma provocando un comportamento "ingordo" che crea problemi a valle.

Requisiti

Balance lavora stage based, cambiando gli istanti di inizio degli interstage. Per forzare il comportamento "solo offset" bisogna bloccare la durata degli stage, e lasciare libertà totale a tutti gli interstage.

Serve un detector con channel unico per ogni corsia interessante di approccio all'intersezione.

Chapter 3

Obiettivi Intelligenti

Definizione degli obiettivi dell'ottimizzazione classica e di quella sostenibile.

Formalizzazione delle funzioni di costo: l'indice di coda (integrale tempo in coda per veicoli)/tempo passato nel corridoio fa cacare

$$\frac{\sum_{a \in C} \sum_{t \in T} queu_{at}}{\sum_{a \in C} \sum_{t \in T} n_{at}}$$

non è reattivo soprattutto se gli archi sono lunghi perché il tempo in coda è comunque poco rispetto al totale

ci vorrebbe tempo per km percorso, ma come faccio a calcolare lo spazio percorso?

Numero di fermate non è male, calcolato come integrale dell'aumento della QUEUE in ogni intervallo, corretto aggiungendo l'eventuale deflusso se la coda alla fine dell'intervallo non si è azzerata. Questo comporta degli errori (da quantificare esattamente?) negli intervalli in cui la coda comincia o finisce.

$$\sum_{a \in C} \sum_{t \in T} QUEUE_{a,t} - QUEUE_{a,t-1} + OFLW_{a,t} \Delta t$$

Il tutto è da normalizzare rispetto al numero di veicoli entrati su qualsiasi arco.

Altro obiettivo utile sarebbe guardare la congestione degli archi. Prendendo la congestione media si auto-pesano di più gli archi corti, il che avrebbe senso. Come indicatore della congestione, dal GLTM prendiamo la lunghezza della coda peggiore nell'intervallo rispetto all'arco $QUEL \in [0, 1]$ quindi come funzione di costo

$$Q_a^{MAX} = \sup\{QUEL_{a,t} | t \in T\} \quad (3.1)$$

Ottimizzazione con GLTM Genetico

Condizioni dei test di TRE:

1. TRE fa un equilibrio e produce TPRB per tutto il giorno: gli intervalli sono ben più lunghi di un ciclo di semaforo quindi le svolte sono mediate e dipendono solo da domanda e green share
2. TRE salva l'istantanea dei flussi per cominciare le simulazioni di ottimizzazione a rete carica
3. Gli offset vengono modificati e per ogni individuo si fa un caricamento al secondo su una finestra temporale ristretta, partendo dall'istantanea salvata

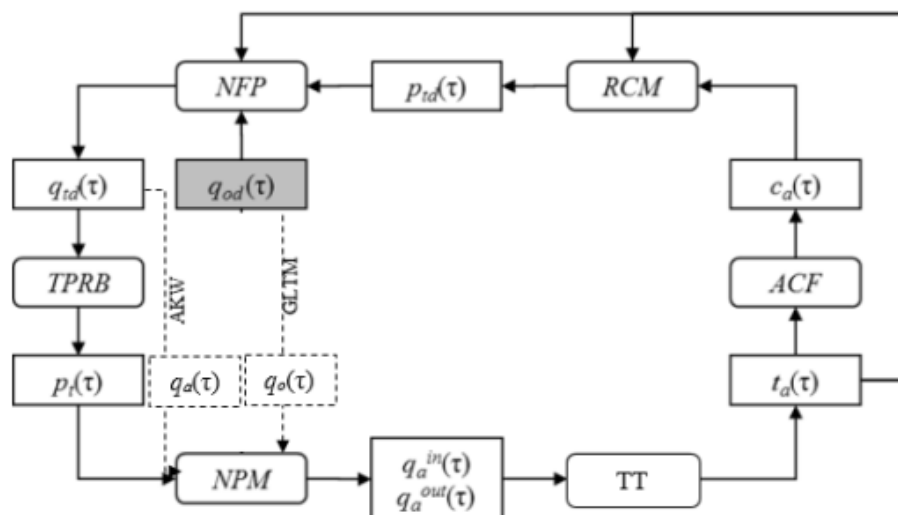


FIGURE 4.1 – L'assegnazione di TRE

Data una finestra temporale $T = \{t | t_0 \leq t \leq t_{end}\}$

4.1 General Link Transmission Model

Funzionamento generale, pregi e limitazioni rispetto all'applicazione come ottimizzatore.

Quale intervallo di risultati si può usare? Data l'incertezza e le approssimazioni, ha senso usare i risultati al secondo? Non varrebbe la pena guadagnare tempo usando intervalli di 6s ?

4.2 Il modello in tempo reale

4.2.1 Corridoio di ottimizzazione

Definito come una serie di link. I link attraversano un certo numero di intersezioni, alcune delle quali sono amministrate da un semaforo. Fra l'inizio e il primo semaforo, fra i semafori, e dopo l'ultimo, i link sono accorpati in sezioni. I KPI vengono calcolati su di essi.

4.3 Tuning del modello

4.4 Tuning del Genetico

Effetti dimensioni popolazione, aggiustamenti mutazione

Chapter 5

Risultati

Test da fare:

Domanda stabile subcritica

Domanda stabile supercritica

Riduzione di capacità alla fine del corridoio per vedere se TreGen tiene le code a monte o fa andare tutti a morire in fondo.

5.1 La qualità in termini assoluti

5.2 Il confronto con microsimulazione

5.3 Performance