UNIVERSITÀ DEGLI STUDI DI TRIESTE

Dipartimento di Ingegneria e Architettura
Corso di Laurea Triennale in Ingegneria Elettronica e Informatica
Anno Accademico 2021/2022

Extended Summary of "DYNAMIC TRAVELING SALESMAN PROBLEM WITH STOCHASTIC RELEASE DATES"

Laureando:
Giovanni Zanin

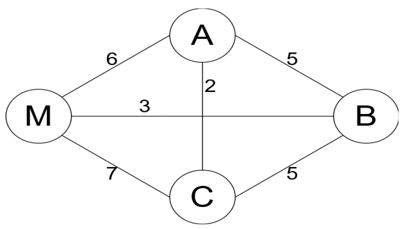
Relatore:
Prof. Lorenzo Castelli

Il presente Extended Summary discute il problema del commesso viaggiatore (traveling salesman problem, TSP) in particolare nella sua versione dinamica con date di rilascio stocastiche (dTSP-srd) e di una possibile euristica di risoluzione per tale problema secondo quanto illustrato nell'articolo “Dynamic traveling salesman problem with stochastic release dates” [1].

1 Introduzione

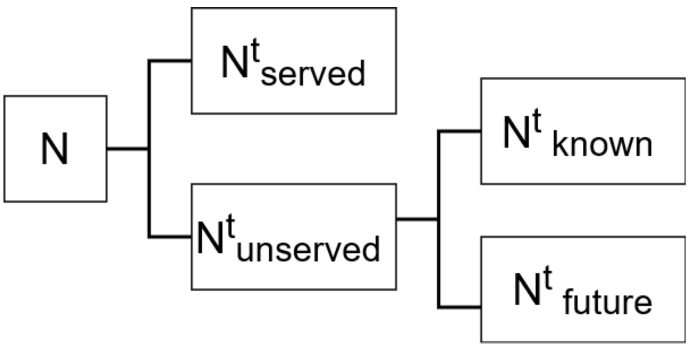
L'applicazione più immediata del problema del commesso viaggiatore è quella di minimizzare il tempo di lavoro di un commesso il cui compito è quello di consegnare dei pacchi a un certo numero N di clienti geograficamente distribuiti. Il problema può essere illustrato prendendo in considerazione un grafo [2] con le seguenti caratteristiche:

- un nodo rappresenta il magazzino, dal quale il commesso inizia le sue consegne e al quale ritorna al termine di ogni viaggio;
- i rimanenti N nodi simboleggiano i clienti;
- ogni arco rappresenta un tragitto al quale è associato un numero naturale, corrispondente al tempo che il commesso impiega a percorrere tale tragitto;
- il grafo è completo, ovvero da ogni nodo il commesso è in grado di raggiungerne un altro senza dover usare altri nodi “di passaggio”;
- tra gli archi del grafo vale la disuguaglianza triangolare.



➢ Figura 1: esempio di problema del commesso viaggiatore con N=3.

Nell'articolo di Archetti et al. viene proposto un algoritmo per risolvere in modo approssimato una particolare versione di TSP in cui all'istante t=0 non è detto che tutti i pacchi destinati ai clienti siano immediatamente pronti alla consegna. Ogni cliente i è infatti caratterizzato da un istante che chiameremo r_i che indica la data di rilascio del suo pacco (release date) ovvero l'istante di tempo in cui il suo pacco è nel magazzino pronto per essere consegnato. Si faccia riferimento alla figura 2. Qualora in un dato istante t un pacco p non sia ancora stato consegnato al suo destinatario ($p \in N^t_{unserved}$), se p è nel magazzino pronto alla consegna ($p \in N^t_{known}$) la sua r_i sarà un numero che indica un istante nel passato, altrimenti ($p \in N^t_{future}$) r_i sarà descritto da una variabile aleatoria discreta e dalla sua relativa funzione di densità (pdf). In quest'ultimo caso diremo che $p \in N_{dynamic}$ se la sua pdf può essere aggiornata in qualsiasi istante, $p \in N_{static}$ se può essere aggiornata solo alla data di effettivo rilascio [3].



➢ Figura 2: suddivisione dei clienti all'istante t.

La variante dTSP-srd avvicina la versione originale del problema alla realtà, dove tipicamente:

- la merce da consegnare è pronta in magazzino dopo una certa data di rilascio;
- tale data di rilascio non è certa ma è descrivibile da una funzione di densità (continua, ma approssimabile a una discreta);
- la merce viene portata al magazzino (da dove partiranno le consegne) con veicoli che possono essere dotati o meno di GPS (nel primo caso la pdf della data di rilascio sarà aggiornata in base alla posizione del veicolo, nel secondo sarà aggiornata solo alla data di effettivo rilascio).

2 L'algoritmo

2.1 Specifiche

Come detto l'obbiettivo dell'algoritmo è quello di minimizzare il tempo complessivo, ovvero il tempo che il commesso attende al magazzino sommato a quello che trascorre viaggiando. Per fare ciò dobbiamo tenere conto che:

- tutti i nodi vanno visitati;
- ogni viaggio inizia e termina al magazzino;
- l'istante di inizio di un viaggio dev'essere maggiore o uguale al massimo tra l'istante in cui termina il viaggio precedente e l'istante in cui tutti i pacchi da consegnare in quel viaggio sono stati rilasciati dal magazzino ($N_{viaggio} \subset N_{known}$);
- l'istante in cui un viaggio termina è uguale all'istante in cui inizia sommato alla sua durata (ovvero alla somma dei tempi di percorrenza degli archi che lo compongono).

2.2 Valutazione di una soluzione

Per confrontare quale sia la migliore tra due o più soluzioni è necessario calcolare il tempo complessivo stimato che ognuna delle soluzioni comporta, per svolgere tale calcolo verranno proposti due metodi: uno stocastico e uno deterministico. I due metodi differiscono nel modo in cui vengono gestite le pdf delle variabili aleatorie.

2.2.1 Metodo stocastico

Nella valutazione stocastica delle soluzioni gli orari di inizio e fine dei viaggi sono trattate come delle vere e proprie variabili aleatorie, dipendenti dalle date di rilascio, e le varie pdf vengono combinate tra loro e solo la pdf finale (quella relativa all'orario di termine dell'ultimo viaggio) viene approssimata. Viene qui fornito un esempio di uso di tale metodo che fa riferimento al grafo in figura 1 e che prende in considerazione la soluzione costituita da due viaggi, il primo dei quali visita il cliente B (v_B) e il secondo A e C (v_{AC}).

Notazione: fx indica la pdf della variabile aleatoria x; y^i e y^f sono le variabili aleatorie indicanti l'inizio e la fine del viaggio y; z indica il valore atteso della variabile aleatoria z.

Nell'esempio si è considerato che nell'istante in cui avviene la valutazione le pdf relative alle date di rilascio siano le seguenti:

$$f_{rA}(t)=\begin{cases} 0{,}5 \text{ per } t=5 \\ 0{,}5 \text{ per } t=10 \end{cases} \qquad f_{rB}(t)=\begin{cases} 0{,}5 \text{ per } t=2 \\ 0{,}5 \text{ per } t=3 \end{cases} \qquad f_{rC}(t)=\begin{cases} 0{,}5 \text{ per } t=6 \\ 0{,}5 \text{ per } t=10 \end{cases}$$

Il viaggio v_B può iniziare non appena il pacco destinato a B viene rilasciato, dunque $f_{v_B}^{i} = f_{r_B}$. Il tragitto per andare e tornare da B ha poi durata 6, dunque:

$$f_{v_B}^{f}(t)= \begin{cases} 0{,}5 \text{ per } t=8 \\ 0{,}5 \text{ per } t=9 \end{cases}$$

Ricordando che il viaggio v_{AC} può iniziare solamente dopo che v_B si è concluso e i pacchi destinati ad A e a C sono stati rilasciati, la $f_{v_{AC}}^{i}$ può essere ricavata componendo opportunamente $f_{v_B}^{f}$, f_{r_A} e f_{r_C}, ottenendo:

$$f_{v_{AC}}^{i}(t)= \begin{cases} 0{,}125 \text{ per } t=8 \\ 0{,}125 \text{ per } t=9 \\ 0{,}75 \text{ per } t=10 \end{cases}$$

v_{AC} ha durata 15 dunque:

$$f_{v_{AC}}^{f}(t)= \begin{cases} 0{,}125 \text{ per } t=23 \\ 0{,}125 \text{ per } t=24 \\ 0{,}75 \text{ per } t=25 \end{cases}$$

Quest'ultima pdf corrisponde anche al tempo di lavoro complessivo che la soluzione presa in considerazione comporta, che quindi vale (in media):

$$v_{AC}^{f}=24{,}625.$$

2.2.2 Metodo deterministico

Nel metodo deterministico le varie pdf non vengono composte fra loro, si ha invece un’approssimazione puntuale delle pdf di partenza (quelle relative alle date di rilascio) e una conseguente approssimazione degli istanti di inizio e fine dei viaggi, non più descritti da variabili aleatorie bensì da numeri puri. Per valutare la soluzione dell'esempio precedente con il metodo deterministico vengono presi quindi in considerazione i valori:

$$r_A=7{,}5 \qquad r_B=2{,}5 \qquad r_C=8$$

Il viaggio v_B avrà quindi orario di partenza atteso $v_B^{i}= r_B$ e dato che la sua durata è 6 si considera:

$$v_B^{f}=8{,}5$$

v_{AC} può avere inizio solo dopo che v_B è terminato e i pacchi destinati ad A e C sono pronti alla consegna, vale a dire:

$$v_{AC}^{i}=\max \{v_B^{f}, r_A, r_C\}= v_B^{f}= 8{,}5$$

v_{AC} ha durata 15 ed è l'ultimo viaggio che il commesso deve compiere, dunque il metodo deterministico valuta che il tempo complessivo che la soluzione comporta valga:

$$v_{AC}^{f}=23{,}5$$

2.3 Riottimizzazioni

L'articolo fa riferimento ad un algoritmo per la riottimizzazione di una data soluzione detto Iterated local search (ILR) [4]. Tale algoritmo si compone di due parti (si faccia riferimento alla figura 3):

- destroy and repair (DR): dalla soluzione iniziale s (rappresentata come un vettore) vengono rimossi α nodi (dove α è racchiuso in [α_{min}, α_{max}]) e successivamente riaggiunti ad un viaggio scelto casualmente. Si ottiene così s';
- local search (LS): viene valutato ogni possibile miglioramento ad s' ottenibile spostando un nodo da un viaggio ad un altro, anticipando o ritardando una visita al magazzino, unendo due viaggi oppure scindendo un viaggio in due distinti. Si ottiene in questo modo s".

Se s ha una durata prevista superiore a s" quest'ultima diventa la soluzione di partenza per l'iterazione successiva, altrimenti viene incrementato α. Le condizioni di terminazione del ciclo di ILS sono tipicamente un tempo di computazione massimo e un massimo numero di iterazioni senza miglioramenti.

```
while(condizioni di terminazione non verificate){

   Itereted local search (s,α){

            Destroy and repair (s,α){
               ...
               s'=...;
            }

            Local search (s'){
               ...
               s''=...;
            }

            if (evaluate(s)>evaluate(s'')){
               s=s'';
               α=αmin;
            }
            else{
               α++;
            }

            if (α>αmax){
               α=αmin;
            }

      }

}

return s;
```

➢ Figura 3: ILS in forma di pseudocodice.

La cadenza con cui vengono effettuate le riottimizzazioni può seguire tre diversi schemi:

- R: al termine di ogni viaggio;
- RD: all'inizio e al termine di ogni viaggio;
- RDW: all'inizio e al termine di ogni viaggio e a metà dell'attesa stimata al deposito tra due viaggi consecutivi.

3 Risultati

Come prevedibile risultati migliori si ottengono con più riottimizzazioni (quindi preferendo lo schema RDW, che prevede però più tempo di computazione complessivo) e valutando le soluzioni con il metodo stocastico, il quale ha tuttavia una complessità computazionale maggiore del metodo deterministico. Simulazioni dell'algoritmo su istanze di dTSP-srd aventi fino a 50 nodi (figura 4) evidenziano come il metodo deterministico sia maggiormente conveniente laddove:

- la porzione δ di nodi aventi data di rilascio dinamica è alta;
- il rapporto β tra il lasso di tempo in cui si ha il rilascio di tutti i pacchi e il tempo di TSP del grafo (risoluzione del problema senza date di rilascio) è 1.

Quest'ultima situazione è infatti intuitivamente quella in cui è più difficile la scelta tra attendere l'arrivo di tanti pacchi prima di partire (vantaggiosa per β<<1) e iniziare i viaggi non appena vengono rilasciati i pacchi (consigliata laddove β >>1).

Average percentage gap from the best solution.

	Avg. Deterministic	Avg. Stochastic	Avg. R	Avg. RD	Avg. RDW
$\delta = 0$	2.69	0.72	1.77	1.68	1.67
$\delta = 0.5$	2.96	1.40	2.35	2.17	2.02
$\delta = 1$	3.10	1.34	3.03	2.40	2.25
$\beta = 0.5$	1.91	1.28	1.67	1.61	1.51
$\beta = 1$	3.79	0.90	2.37	2.56	2.11
$\beta = 1.5$	3.05	1.29	2.40	2.09	2.02
Avg.	2.92	1.16	2.14	2.09	1.88

➢ Figura 4: Risultati degli esperimenti di Archetti et al.

L'articolo propone dunque un valido algoritmo per la risoluzione del problema ma non chiude le porte a possibili miglioramenti futuri, dati per esempio da riottimizzazioni anche quando il veicolo si trova presso il cliente [5] o alla valutazione delle soluzioni con veicoli multipli, eventualmente aventi una capacità massima.

Riferimenti bibliografici

[1] C. Archetti, D. Feillet, A. Mor, M. G. Speranza, *Dynamic traveling salesman problem with stochastic release dates*, European Journal of Operational Research 280.3 (2020), 832-844. https://www.sciencedirect.com/science/article/abs/pii/S0377221719306411?via%3Dihub

[2] pages.di.unipi.it, Appunti di ricerca operativa, capitolo 2, Grafi e reti di flusso.

[3] W.B Powell, H. P. Simao, B. Bouzaiene-Ayari, *Approximate dynamic programming in transportation and logistics: A unified framework. EURO Journal on Transportation and Logistics*, 1 (2012), 237–284.
https://www.sciencedirect.com/science/article/pii/S2192437620600048

[4] C. Archetti, D. Feillet, A. Mor, M. G. Speranza, *An iterated local search for the traveling salesman problem with release dates and completion time minimization,* Computers & Operations Research 98 (2018), 24-37.
https://www.sciencedirect.com/science/article/abs/pii/S030505481830114X

[5] M. A. Klapp, A. L. Erera, A. Toriello, *The dynamic dispatch waves problem for same-day delivery. European Journal of Operational Research*, 271 (2018), 519–534.
https://www.sciencedirect.com/science/article/abs/pii/S0377221718304326