ROAR II

Ricerca Operativa: Applicazione Reali

Alessandro Gobbi Alice Raffaele Gabriella Colajanni Eugenia Taranto

IIS Antonietti, Iseo (BS)

21 marzo 2022

Introduzione

Chi siamo e i nostri contatti

Alessandro Gobbi (UniBS)
alessandro.gobbi@unibs.it

Alice Raffaele (UniVR)
alice.raffaele@univr.it

Gabriella Colajanni (UniCT)
colajanni@dmi.unict.it

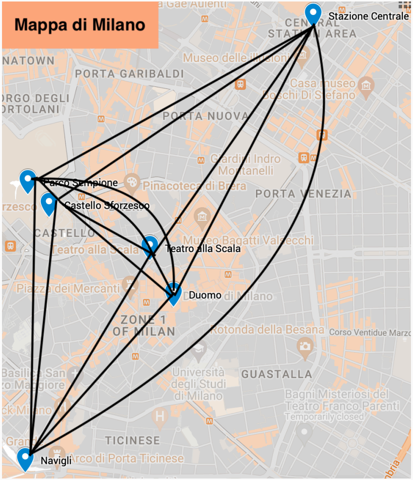
Eugenia Taranto (UniCT)
eugenia.taranto@unict.it

Sondaggio iniziale

`www.menti.com` – Codice: **5517 5531**

Lavoro di gruppo
(30 minuti)

In gita a Milano



Fra qualche settimana andrete a fare una gita giornaliera a Milano, dove state pianificando un tour tra monumenti e piazze da visitare. In particolare, avete già deciso, una volta arrivati in Stazione Centrale, di visitare il Duomo, il Teatro alla Scala, il Castello Sforzesco, il Parco Sempione e i Navigli.

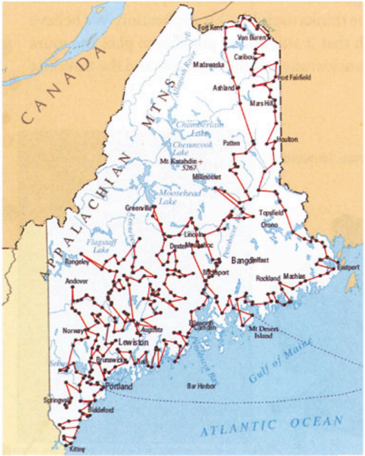
Nella tabella sottostante, sono riportate le distanze a piedi (in km) tra tutti i punti di attrazione che avete scelto. Per esempio, il Duomo e il Teatro alla Scala sono distanti 0,45 km.

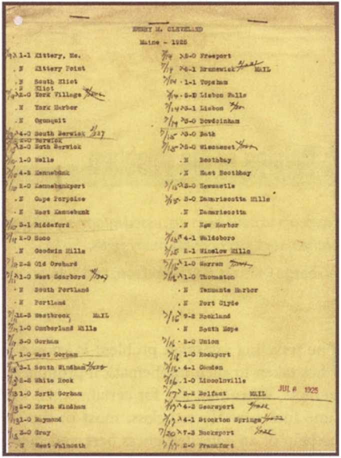
	Stazione Centrale	Duomo	Teatro alla Scala	Castello Sforzesco	Parco Sempione	Navigli
Stazione Centrale	–	2,9	2,5	2,8	3,2	5,7
Duomo	2,9	–	0,45	1,9	1,6	2,9
Teatro alla Scala	2,5	0,45	–	1	1,5	3,1
Castello Sforzesco	2,8	1,9	1	–	0,28	3
Parco Sempione	3,2	1,6	1,5	0,28	–	3,4
Navigli	5,7	2,9	3,1	3	3,4	–

Determinare un tour che parta e ritorni in Stazione Centrale e che vi consenta di visitare tutti i punti di attrazione **una e una sola volta**, camminando il meno possibile.

Il problema del commesso viaggiatore (*Traveling Salesman Problem*)

Viaggi di lavoro nel 1925





HENRY M. CLEVELAND

Maine - 1925

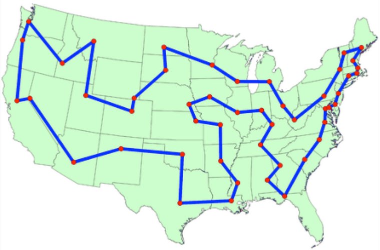
1-1 Kittery, Me.
N Kittery Point
N South Eliot
N Eliot
2-0 York Village
N York Harbor
N Ogunquit
4-0 South Berwick
2-0 Berwick
2-0 North Berwick
1-0 Wells
4-2 Kennebunk
2-0 Kennebunkport
N Cape Porpoise
N West Kennebunk
5-1 Biddeford
2-0 Saco
N Goodwin Mills
2-2 Old Orchard
1-0 West Scarboro
N South Portland
N Portland
11-3 Westbrook MAIL
1-0 Cumberland Mills
3-0 Gorham
1-0 West Gorham
3-1 South Windham
2-2 White Rock
1-0 North Gorham
2-0 North Windham
1-0 Raymond
2-0 Gray
N West Falmouth

2-0 Freeport
6-1 Brunswick MAIL
1-1 Topsham
2-0 Lisbon Falls
3-1 Lisbon
3-0 Bowdoinham
3-0 Bath
5-0 Wiscasset
N Boothbay
N East Boothbay
3-0 Newcastle
3-0 Damariscotta Mills
N Damariscotta
N New Harbor
4-1 Waldoboro
2-1 Winslow Mills
1-0 Warren
1-0 Thomaston
N Tenants Harbor
N Port Clyde
9-2 Rockland
N South Hope
2-0 Union
1-0 Rockport
4-1 Camden
1-0 Lincolnville
2-2 Belfast MAIL
JUL 8 1925
4-2 Searsport
4-1 Stockton Springs
7-3 Bucksport
2-0 Frankfort

“Dear Sirs
My route list is balled up the worst I ever saw. Will take half as long again to work it as last year. [...] I wish you would send me my old list 1924 from Dexter on as it is much better than this. I don’t see how you could break it out as you did especially from Albion to Madison would be jumping all over the map. This section I changed. The river from Bangor down has no bridge and you have those towns down as if I could cross it anywhere. Last year’s list was made out the best of any one and I can’t see the object of changing it over.
I think I have made myself plain.

Yours truly, H. M. Cleveland”

Il problema del commesso viaggiatore



"It is shown that a certain tour of 49 cities, one in each of the 48 states and Washington DC, has the shortest road distance." (Dantzig, Fulkerson and Johnson, 1954)

Un commesso deve visitare n città una e una sola volta, ritornando poi al punto di partenza. La distanza (o il costo) tra ogni coppia di città (i, j) è nota e si indica con $c_{i,j}$. Si vuole determinare il percorso più breve del commesso viaggiatore calcolando un **ciclo hamiltoniano** di lunghezza minima.

Nozioni di teoria dei grafi

Dato un grafo $G = (V, E)$:

- un **cammino hamiltoniano** è un cammino che attraversa tutti i vertici una e una sola volta;
- si ha un **ciclo hamiltoniano** quando in un cammino hamiltoniano si può collegare l'ultimo vertice con il primo, formando un ciclo che torni al punto di partenza;
- G è detto **grafo hamiltoniano** se contiene almeno un ciclo hamiltoniano.

Numero di soluzioni possibili per il commesso viaggiatore

- Visto che ogni città deve essere visitata, la soluzione sarà una *permutazione ciclica* di tutte le città.

  $$\downarrow$$

  Date n città, quante sono le possibili soluzioni in teoria?
  Se le distanze sono simmetriche e il grafo è completo, questo numero è pari a $\frac{(n-1)!}{2}$.

- Non è così semplice controllare e confrontare tutte le soluzioni, adottando un approccio a forza bruta usando un computer. Assumendo che per controllare ogni tour serva 1 μs:

n	**Numero di tour**	**Tempo**
5	12	12 μs
8	2520	2,5 ms
10	181.440	0,18 s
12	19.958.400	20 s
15	87.178.291.200	12,1 h
18	177.843.714.048.000	5,64 y
20	60.822.550.204.416.000	1927 y

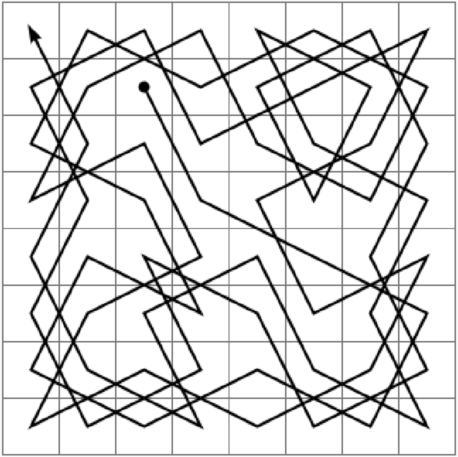
- Nel 1972, il problema fu dimostrato essere NP-Hard.

$$\downarrow$$

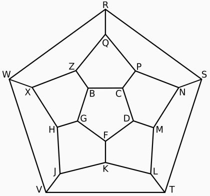
I metodi esatti richiedono nel caso peggiore un tempo esponenziale rispetto alla dimensione dell'istanza in input. Per questo, negli anni ci si è concentrati a sviluppare molte euristiche per ottenere soluzioni di buona qualità in tempi accettabili.

Un po' di storia

- 1766: Eulero studia il problema del **percorso del cavallo**



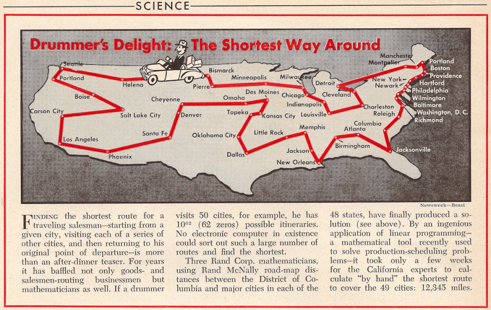
- 1856–1859: Hamilton sviluppa il gioco icosiano:



"I have found that some young persons have been much amused by trying a new mathematial game which the Icosian furnishes, one person sticking five pins in any five consecutive points, such as abcde or abcde′, and the other player then aiming to insert, which by theory in this letter can always be done, fifteen other pins in cyclical succession, so as to cover all the other points, and to end in immediate proximity to the pin wherewith his antagonist has begun."

- 1930: Menger è il primo a parlare di **cammini hamiltoniani** e a introdurre i costi di viaggio, arrivando quindi a una formulazione generale del problema: “*We use the term ‘Messenger problem’ (because this question is faced in practice by every postman, and, by the way, also by many travelers) for the task, given a finite number of points with known pairwise distances, to find the shortest path connecting the points.*”
- 1937: Flood ottiene delle soluzioni quasi ottimali risolvendo un problema di **routing di alcuni scuolabus**, dopo aver sentito parlare del problema a Princeton nel 1934.

- 1949: il nome **Traveling Salesman Problem** appare per la prima volta nell'articolo scientifico di Julia Robinson, "*On the Hamiltonian Game (A Traveling Salesman Problem)*". Tuttavia, è ignoto chi abbia effettivamente coniato il termine.
- 1954: Dantzig, Fulkerson and Johnson propongono la prima formulazione del problema utilizzando la **Programmazione Lineare Intera**, prima su grafi indiretti e poi anche su grafi diretti.

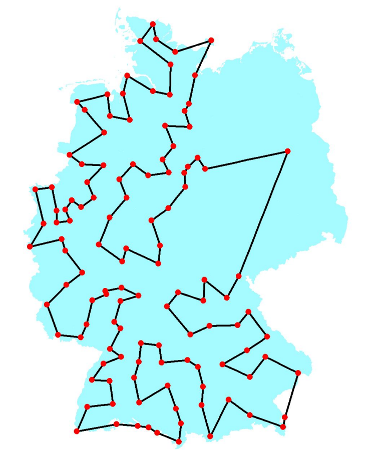


FINDING the shortest route for a traveling salesman—starting from a given city, visiting each of a series of other cities, and then returning to his original point of departure—is more than an after-dinner teaser. For years it has baffled not only goods- and salesmen-routing businessmen but mathematicians as well. If a drummer visits 50 cities, for example, he has 10^{62} (62 zeros) possible itineraries. No electronic computer in existence could sort out such a large number of routes and find the shortest.

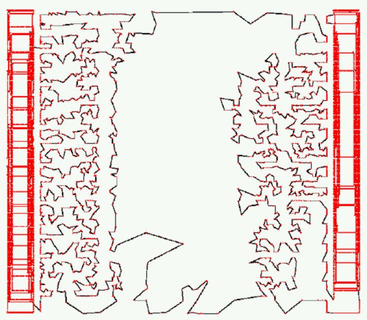
Three Rand Corp. mathematicians, using Rand McNally road-map distances between the District of Columbia and major cities in each of the 48 states, have finally produced a solution (see above). By an ingenious application of linear programming—a mathematical tool recently used to solve production-scheduling problems—it took only a few weeks for the California experts to calculate "by hand" the shortest route to cover the 49 cities: 12,345 miles.

- 1962: Procter & Gamble: campagna pubblicitaria di un **contest** con in palio un premio da 10.000 $.
- Dal regolamento: "*Imagine that Toddy and Muldoon want to drive around the country and visit each of the 33 locations represented by dots on the contest map, and that in doing so, they want to travel the shortest possible route. You should plan a route for them from location to location which will result in the shortest total mileage from Chicago, Illinois back to Chicago, Illinois.*"

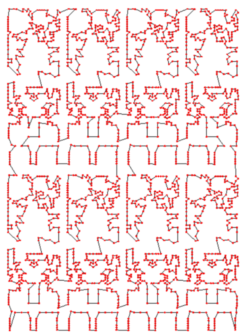
- 1977: Groetschel trova un tour ottimale per **120 città** della Germania dell'Ovest:



- 1987: Padberg and Rinaldi trovano un tour ottimale per **2.392 punti** lavorando a un problema della Tektronics Incorporated

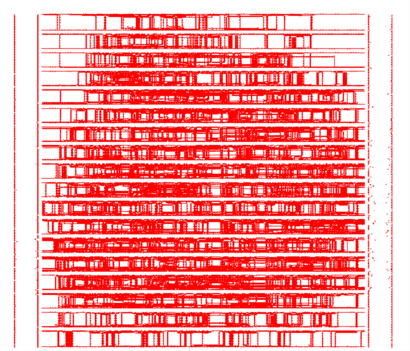


- 1994: Applegate, Bixby, Chvátal e Cook trovano un tour ottimale per **7.397 punti**, in un problema emergente da un’applicazione nei AT&T Bell Laboratories, riguardante le *programmable logic array*, dispositivi logici programmabili usati per implementare circuiti logici combinatori.



- 2004: Applegate, Bixby, Chvátal, Cook e Helsgaun trovano un tour ottimale per **24.978 città** della Svezia

L'istanza più grande risolta all'ottimo



2006: Cook e altri ricercatori trovano un tour ottimale per **85.900 punti**, istanza emersa dal problema di design di un microchip.

Per molte altre istanze contenute nella **TSPLIB** (collezione di istanze del problema del commesso viaggiatore), si trovano soluzioni che sono garantite essere al massimo 2-3% peggio di quelle ottimali.

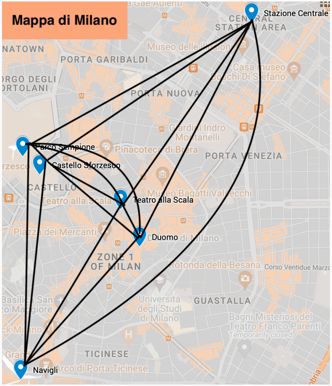
Applicazioni del problema del commesso viaggiatore

- Routing e pianificazione (pickup e delivery, tour e viaggi, trasporti)
- Mappatura dei genomi
- Telescopi di puntamento, raggi X e laser
- Guida di macchine industriali (perforazione e saldatura di circuiti stampati, personalizzazione di chip per computer)
- Lavori di scheduling (produzione di smart card)

Possiamo distinguere tra:

- algoritmi costruttivi, che ricevono in input un’istanza del problema e ritornano in output una soluzione ammissibile:
- algoritmi di ricerca locale, che ricevono in input un’istanza e una soluzione ammissibile del problema, e ritornano in ouput, se possibile, un’altra soluzione ammissibile e migliore:

Algoritmo euristico costruttivo: *Nearest Neighbor Insertion*



Input: un grafo $G = (V, E)$ pesato.

Output: un ciclo hamiltoniano.

Step 1: sia S la sequenza di vertici visitati, inizialmente vuota;

Step 2: scegliere $v_{corr} \in V$ come punto di partenza e aggiungere v_{corr} in coda a S;

Step 3: finché non si visitano tutti i vertici in V:

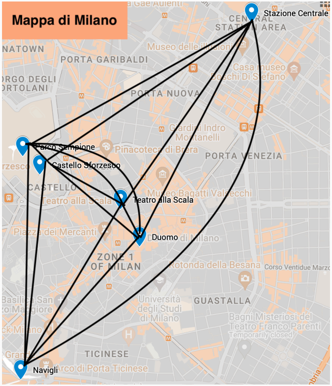
1. selezionare $v \in V$ non ancora visitato e **più vicino** a v_{corr};
2. aggiornare $v_{corr} = v$;
3. aggiungere v_{corr} in coda a S.

Step 4: aggiungere s in coda a S e ritornare S.

Soluzione euristica: Stazione Centrale $\rightarrow$ Teatro alla Scala $\rightarrow$ Duomo $\rightarrow$ Parco Sempione $\rightarrow$ Castello Sforzesco $\rightarrow$ Navigli $\rightarrow$ Stazione Centrale

Costo della soluzione: $2,5 + 0,45 + 1,6 + 0,28 + 3,4 + 5,7 = 13,93$ km.

Algoritmo euristico costruttivo: *Farthest Neighbor Insertion*



Input: un grafo $G = (V, E)$ pesato.

Output: un ciclo hamiltoniano.

Step 1: sia S la sequenza di vertici visitati, inizialmente vuota;

Step 2: scegliere $s = v_{corr} \in V$ come punto di partenza e aggiungere v_{corr} in coda a S;

Step 3: finché non si visitano tutti i vertici in V:

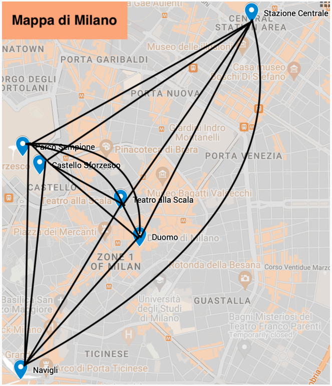
1. selezionare $v \in V$ non ancora visitato e **più lontano** a v_{corr};
2. aggiornare $v_{corr} = v$;
3. aggiungere v_{corr} in coda a S.

Step 4: aggiungere s in coda a S e ritornare S.

Soluzione euristica: Stazione Centrale $\rightarrow$ Navigli $\rightarrow$ Parco Sempione $\rightarrow$ Duomo $\rightarrow$ Castello Sforzesco $\rightarrow$ Teatro alla Scala $\rightarrow$ Stazione Centrale

Costo della soluzione: $5,7 + 3,4 + 1,6 + 1,9 + 1 + 2,5 = 16,1$ km.

Algoritmo euristico costruttivo: *Random Neighbor Insertion*



Input: un grafo $G = (V, E)$ pesato.
Output: un ciclo hamiltoniano.
Step 1: sia S la sequenza di vertici visitati, inizialmente vuota;
Step 2: scegliere $s = v_{corr} \in V$ come punto di partenza e aggiungere v_{corr} in coda a S;
Step 3: finché non si visitano tutti i vertici in V:

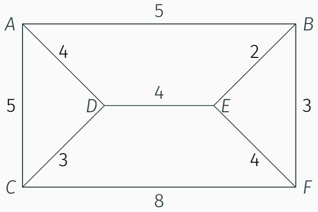
1. selezionare **a caso** un $v \in V$ non ancora visitato;
2. aggiornare $v_{corr} = v$;
3. aggiungere v_{corr} in coda a S.

Step 4: aggiungere s in coda a S e ritornare S.

Soluzione euristica: Stazione Centrale $\rightarrow$ Parco Sempione $\rightarrow$ Duomo $\rightarrow$ Navigli $\rightarrow$ Teatro alla Scala $\rightarrow$ Castello Sforzesco $\rightarrow$ Stazione Centrale
Costo della soluzione: $3,2 + 1,6 + 2,9 + 3,1 + 1 + 2,8 = 14,6$ km.

Lavoro di gruppo (45 minuti)

Si consideri il seguente grafo.



Quesiti:

1. Trovare tre soluzioni ammissibili applicando gli algoritmi euristici costruttivi visti prima, scegliendo ogni volta un vertice diverso come punto di partenza. (**Nota**: questo grafo non è completo, potreste non trovare una soluzione ammissibile applicando alcune euristiche. Provate lo stesso).
2. Formulare il problema come problema di Programmazione Lineare Intera e implementare il modello con Excel Solver.

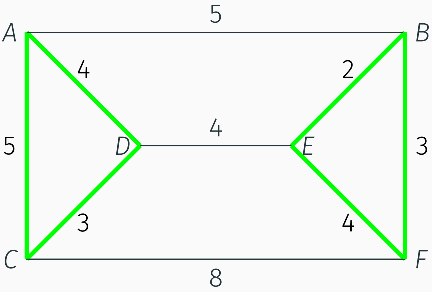
Formulazione del modello per grafi indiretti (caso simmetrico)

Sia $G = (V, E)$ un grafo indiretto pesato, in cui ogni lato $e \in E$ è dotato di un costo $c_e \in \mathbb{R}^+$. Per ogni vertice $v \in V$, sia $\delta(v)$ l'insieme dei lati incidenti a v.

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ & \sum_{e \in \delta(v)} x_e = 2, \quad \forall v \in V \\ & x_e \in \{0, 1\}, \quad \forall e \in E. \end{aligned}$$

Come facciamo a verificare che sia corretta questa formulazione?

Con questa formulazione, per l'istanza di esempio di prima otteniamo la seguente soluzione ottima:



La soluzione ottenuta **non è un ciclo hamiltoniano**!

↓

Mancano dei vincoli per evitare che si formino dei sottocicli.

Sia $S \subset V$ e sia $\delta(S)$ l'insieme dei lati incidenti a S, ovvero aventi un vertice in S e l'altro in $V \setminus S$.
La formulazione completa è la seguente:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
& \sum_{e \in \delta(v)} x_e = 2, \quad \forall v \in V \\
& \sum_{e \in \delta(S)} x_e \geq 2, \quad \forall S \subset V, 2 \leq |S| \leq |V| - 2 \\
& x_e \in \{0, 1\}, \quad \forall e \in E.
\end{aligned}
$$

In questo modo, la formulazione ottenuta ha purtroppo un numero di vincoli esponenziali. Infatti, il numero di possibili sottoinsiemi propri S di V vertici, con $|V| = n$, è $2^n - 2$ (non contiamo l'insieme vuoto e V stesso).

Conclusione

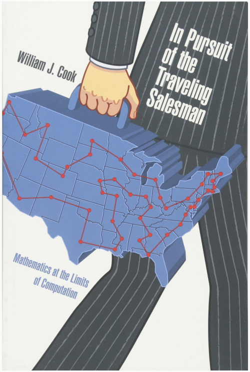
Riepilogo problemi su grafi affrontati

Problema	Tipi di grafi	Input	Output	Algoritmi studiati
Albero di supporto di costo minimo	Indiretti	$G = (V, E)$ pesato	Sottografo di G, aciclico, con costo minimo, che raggiunge tutti i nodi	Kruskal
Cammino minimo tra due nodi	Diretti	$G = (V, A)$ pesato, $s, d \in V$	Sequenza di archi che collegano s a d con costo minimo	Dijkstra
Postino rurale	Sia indiretti che diretti	$G = (V, A)$ pesato, $A' \subseteq A$, $s \in V$	Ciclo con costo minimo che parte e termina in s, contenente tutti gli archi di A'	–
Commesso viaggiatore	Sia indiretti che diretti	$G = (V, A)$ pesato	Ciclo hamiltoniano di costo minimo	Euristici costruttivi

Sondaggio finale

Mentimeter

`www.menti.com` – Codice: 4221 9670



William J. Cook, "*In Pursuit of the Traveling Salesman. Mathematics at the Limits of Computation*"
https://maddmaths.simai.eu/divulgazione/letture-matematiche-recensione-in-pursuit-traveling-salesman-cook/