

UNIVERSITÀ DEGLI STUDI DI MILANO  
Facoltà di Scienze e Tecnologie  
*Corso di Laurea in Informatica*

# FURTHEST INSERTION ALGORITHM

**Relatore:** Prof. Giovanni RIGHINI

Tesi di:  
Asaf COHEN  
Matricola: 975599

Anno Accademico 2023-2024



# Ringraziamenti

*dedicato a DA COMPLETARE...*



# Abstract

Il Problema del Commesso Viaggiatore (TSP) rappresenta una delle sfide più interessanti e rilevanti nell'ambito dell'ottimizzazione combinatoria. Originariamente formulato negli anni '30, il TSP richiede di determinare il percorso più breve per visitare un insieme di città esattamente una volta, ritornando infine alla città di partenza. Nonostante la sua apparente semplicità concettuale, il TSP è noto per la sua complessità computazionale e la sua rilevanza pratica in una vasta gamma di settori, inclusi trasporti, logistica, e progettazione di circuiti.

Il TSP è classificato come un problema NP-hard, il che significa che non esiste un algoritmo efficiente in grado di risolvere tutte le istanze del problema in tempo polinomiale. Di conseguenza, sono state sviluppate numerose euristiche e approcci approssimati per trovare soluzioni accettabili in un tempo ragionevole. Le euristiche sono strategie di ricerca che, pur non garantendo la soluzione ottima, sono in grado di produrre risultati soddisfacenti entro limiti temporali praticabili.

In questa tesi, esploreremo una specifica euristica per il TSP (Furthest Insertion), l'obiettivo principale sarà quello di presentare, analizzare e valutare l'efficacia di questa euristica attraverso simulazioni e confronti con altre tecniche note.

La scelta di concentrarsi su un'euristica per il TSP è motivata dalla necessità di affrontare problemi di dimensioni reali in contesti applicativi. Mentre le soluzioni esatte sono desiderabili per la loro precisione, spesso richiedono una potenza di calcolo eccessiva per problemi di grandi dimensioni. Le euristiche offrono un compromesso utile tra precisione e efficienza, consentendo di ottenere soluzioni praticabili che possono guidare decisioni reali.

Questa tesi sarà strutturata nel seguente modo: innanzitutto, forniremo una panoramica del TSP e delle sue varie formulazioni. Successivamente, esamineremo le principali categorie di approcci risolutivi, concentrandoci in particolare sulle euristiche basate su inserzione. Presenteremo quindi la nuova euristica Furthest Insertion discutendo la sua implementazione e le scelte progettuali adottate. Seguirà una sezione dove testeremo e valuteremo le prestazioni dell'euristica su una serie di istanze del TSP. Infine, concluderemo con un'analisi dei risultati ottenuti, identificando i punti di forza e le limitazioni dell'approccio proposto.

# Indice

<b>Ringraziamenti</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Il Problema del commesso viaggiatore</b>	<b>2</b>
2.1 Formulazione del problema . . . . .	2
2.2 Metodi esatti per il TSP . . . . .	3
<b>3 Euristiche per il TSP</b>	<b>5</b>
3.1 Euristiche costruttive . . . . .	5

# Capitolo 1

## Introduzione

Il Problema del Commesso Viaggiatore (TSP) è una delle sfide più emblematiche e studiate nell'ambito della ricerca operativa e dell'ottimizzazione combinatoria. Originariamente formulato negli anni '30 da Karl Menger, il TSP richiede di determinare il percorso più breve per visitare un insieme di città esattamente una volta, tornando infine alla città di partenza. Nonostante la sua semplice descrizione concettuale, il TSP è noto per la sua complessità computazionale e la sua rilevanza pratica in una vasta gamma di settori.

Le applicazioni del TSP sono diffuse e impattano direttamente su molte attività quotidiane. Ad esempio, nel settore della logistica e della gestione delle catene di distribuzione, il TSP è utilizzato per ottimizzare le rotte dei veicoli di consegna, minimizzando i costi di carburante e il tempo impiegato. In ambito produttivo, il TSP viene impiegato per pianificare i percorsi di ispezione delle fabbriche o per ottimizzare il flusso di lavoro all'interno di un'azienda. Anche nei sistemi di navigazione satellitare e nelle applicazioni GPS, il TSP è alla base dell'ottimizzazione dei percorsi per ridurre il tempo di viaggio.

Storicamente, il TSP ha attratto l'attenzione di numerosi matematici e informatici in quanto è un problema semplice da formulare ma complesso da "risolvere". Il problema è stato formalizzato e reso noto grazie al lavoro di Hassler Whitney nel 1952 e successivamente nel 1954 da Merrill Flood. La dimostrazione della sua appartenenza alla classe di complessità NP-hard è stata fondamentale per stimolare lo sviluppo di tecniche approssimate e euristiche.

Le sfide legate al TSP sono principalmente dovute alla sua natura combinatoria: per  $n$  città, il numero di possibili percorsi da valutare cresce in modo esponenziale con  $n$ , rendendo impraticabile un'analisi esaustiva per istanze di grandi dimensioni. Questa complessità ha spinto alla ricerca di approcci efficienti, come le euristiche, che non garantiscono la soluzione ottimale ma forniscono soluzioni accettabili in tempi ragionevoli.

## Capitolo 2

# Il Problema del commesso viaggiatore

Il problema del commesso viaggiatore (TSP) può essere sintetizzato molto semplicemente con la seguente domanda: "Date  $n$  città, qual è il percorso più breve che inizia e termina con la stessa città?". Il problema quindi presenta le caratteristiche tipiche di un problema su un grafo, dove il grafo è composto da  $n$  vertici (le città) e dove gli archi indicano le distanze euclidee tra le città. La formulazione classica del TSP può essere descritta matematicamente attraverso la programmazione intera lineare. La seguente formulazione fa riferimento alla formulazione MTZ (Miller-Tucker-Zemlin) e alla formulazione DFJ (Dantzig-Fulkerson-Johnson) [1].

### 2.1 Formulazione del problema

**Dati:** Consideriamo un insieme di  $n$  città, ogni città  $i$  (per  $i = 1, 2, \dots, n$ ) rappresenta un punto nello spazio euclideo, ogni città quindi ha coordinate  $(x_i, y_i)$ . Definiamo la matrice  $c$ , dove  $c_{ij}$  indica la distanza euclidea tra le due città  $i$  e  $j$ .

**Variabili:** La variabile  $x_{ij}$  è una variabile binaria, quindi:

$$x_{ij} \in \{0, 1\} \quad \forall i, j = 1, 2, 3, \dots, n$$

La variabile  $x$  assume valore 0 se l'arco che collega la città  $i$  e  $j$  non fa parte del path, 1 se ne fa parte.

**Vincoli:** I vincoli sono i seguenti:

1. Per ogni città  $i$  deve essere presente un solo arco uscente corrispondente nel



tour, quindi la somma delle variabili  $x_{ij}$  deve essere uguale a 1.

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1, 2, 3, \dots, n$$

2. Per ogni città  $j$  deve essere presente un solo arco entrante corrispondente nel tour, quindi la somma delle variabili  $x_{ij}$  deve essere uguale a 1.

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j = 1, 2, 3, \dots, n$$

3. Non devono essere presenti cicli all'interno del tour, quindi deve valere:

$$\sum_{i \in Q} \sum_{j \in Q} x_{ij} \leq |Q| - 1 \quad \forall Q \subsetneq \{2, 3, \dots, n\}$$

**Funzione Obiettivo:** Si vuole minimizzare il costo totale del tour, quindi:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

La seguente formulazione permette quindi di identificare la soluzione ottima.

È stato dimostrato che TSP è un problema NP-difficile [4], questo implica che attualmente non sono noti algoritmi con complessità polinomiale che risolvono il problema. Se esistesse un algoritmo con complessità polinomiale che risolve il TSP allora si potrebbe dimostrare che vale  $P = NP$  e quindi si risolverebbe uno dei più grandi problemi aperti nella teoria della complessità computazionale.

## 2.2 Metodi esatti per il TSP

Esistono vari approcci nella ricerca della soluzione ottima [2]. La soluzione più semplice e intuitiva per il problema è l'approccio brute-force: consiste nel enumerare tutti i possibili percorsi e successivamente selezionare il migliore. In questo caso è necessario analizzare  $n!$  possibili percorsi (nel peggiore dei casi), per questo motivo tentare di risolvere una istanza TSP con un approccio brute-force implica una complessità computazionale pari a  $O(n!)$  e quindi un tempo che risulta rapidamente inaccettabile. Di seguito una tabella che illustra il numero di percorsi da valutare con un approccio basato su ricerca esaustiva al variare del numero  $n$ .

Numero città	Numero percorsi possibili
4	24
5	120
6	720
7	5,040
8	40,320
9	362,880
10	3,628,800
11	39,916,800
12	479,001,600
13	6,227,020,800
14	87,178,291,200
15	1,307,674,368,000
16	20,922,789,888,000
17	355,687,428,096,000
18	6,402,373,705,728,000
19	121,645,100,408,832,000
20	2,432,902,008,176,640,000
21	51,090,942,171,709,440,000
22	1,124,000,727,777,607,680,000
23	25,852,016,738,884,976,640,000
24	620,448,401,733,239,439,360,000
25	15,511,210,043,330,985,984,000,000

Si può notare come il numero cresce molto rapidamente, anche con istanze relativamente piccole (ad esempio 20 città). Questo approccio risulta quindi inutilizzabile nei problemi reali dove può essere necessario analizzare istanze con migliaia di città.

Esistono altri algoritmi esatti i quali consentono di ridurre il numero di possibili soluzioni (percorsi) da analizzare, ad esempio approcci Branch and Bound [3]. La complessità computazionale nel caso peggiore resta esponenziale in quanto può essere necessario analizzare un numero esponenziale di percorsi e quindi non applicabile su problemi reali.

## Capitolo 3

# Euristiche per il TSP

Come discusso nel capitolo 2, i metodi esatti consentono di ottenere soluzioni ottime per il problema del TSP, ma il tempo per ottenere queste soluzioni aumenta esponenzialmente all'aumentare del numero di città presenti nell'istanza. Per affrontare il TSP, sono stati sviluppati numerosi approcci euristici, ovvero metodi che, pur non garantendo la soluzione ottimale, offrono soluzioni di buona qualità in tempi ragionevoli. Le euristiche sono fondamentali per applicazioni pratiche dove la rapidità di calcolo è essenziale. In questo capitolo esploreremo diverse tecniche euristiche, come le euristiche costruttive, che costruiscono una soluzione passo dopo passo e le euristiche basate su inserzione, che partono da un ciclo iniziale al quale vengono aggiunte via via le città fino ad ottenere un ciclo hamiltoniano. Per completezza discuteremo anche di algoritmi genetici, algoritmi di annealing simulato e metodi basati su intelligenza artificiale. Ogni approccio sarà analizzato nel dettaglio, considerando vantaggi, svantaggi e ambiti di applicazione. L'obiettivo è fornire una panoramica completa delle strategie più efficaci per affrontare il TSP, evidenziando come le diverse tecniche possano essere utilizzate in base alle specifiche esigenze del problema da risolvere.

### 3.1 Euristiche costruttive

# Bibliografia

- [1] M. Velednitsky, Short Combinatorial Proof that the DFJ Polytope is contained in the MTZ Polytope for the Asymmetric Traveling Salesman Problem, UC Berkeley, 2018.
- [2] Analysis of Brute Force and Branch & Bound Algorithms to solve the Traveling Salesperson Problem (TSP), Informatics Department, Engineering Faculty, Widyatama University, 2021.
- [3] T. G. Crainic, B. Le Cun, C. Roucairol, Parallel Branch-and-Bound Algorithms, Département de management et technologie École des Sciences de la Gestion Université du Québec à Montréal and CIRRELT, Canada
- [4] R. Karp, Complexity of the Traveling Salesman Problem
- [5] G. Reinelt, The Traveling Salesman: Computational Solution for TSP Applications, Heidelberg, Springer-Verlag
- [6] M. Gotti, I linguaggi specialistici, Firenze, La Nuova Italia, 1991.