

Algoritmi di Ottimizzazione:
Iterated Local Search + Simulated Annealing per il TSP

Preparato da:
Luca Di Leo

29/12/2024
Università di Modena & Reggio emilia

Contents

| | | |
|----------|--|-----------|
| 1 | Introduzione | 2 |
| 2 | Panoramica del Problema | 2 |
| 3 | Dettaglio sulle Metaeuristiche Implementate | 2 |
| 3.1 | Iterated Local Search (ILS) | 2 |
| 3.2 | Simulated Annealing (SA) | 3 |
| 3.3 | Metaeuristica Ibrida (ILS + SA) | 4 |
| 4 | Implementazione Tecnica | 4 |
| 5 | Analisi Sperimentale | 5 |
| 5.1 | EUC_2D | 6 |
| 5.2 | EXPLICIT | 8 |
| 5.3 | GEO | 9 |
| 6 | Conclusioni | 10 |

List of Figures

| | | |
|---|---|---|
| 1 | Plots di EUC 2D con differenti numeri di nodi | 7 |
| 2 | Plots di EXPLICIT con differenti numeri di nodi | 8 |
| 3 | Plots di GEO con differenti numeri di nodi | 9 |

1 Introduzione

Il presente progetto si propone di affrontare il problema del commesso viaggiatore (TSP) utilizzando un approccio algoritmico basato su tecniche metaeuristiche. Gli obiettivi principali sono:

- Studiare e implementare algoritmi metaeuristici per risolvere il TSP.
- Sviluppare una metaeuristica ibrida personalizzata combinando Iterated Local Search (ILS) e Simulated Annealing (SA).
- Analizzare le performance degli algoritmi implementati su istanze reali della libreria TSPLIB.

2 Panoramica del Problema

Il Traveling Salesman Problem (TSP) ha una lunga storia che risale al XVIII secolo, ma è stato formalmente introdotto nella sua versione moderna nei primi anni del XX secolo. Originariamente, era utilizzato come modello per problemi logistici, come l'organizzazione di percorsi per venditori itineranti. Nel corso degli anni, il TSP è diventato un riferimento per lo studio dei problemi di ottimizzazione.

Il problema può essere descritto come segue: dato un insieme di città e le distanze tra ciascuna coppia di esse, trovare il percorso chiuso più breve che visiti ciascuna città una sola volta.

Nella vita reale, il TSP trova applicazioni in numerosi settori, tra cui:

- *Logistica e trasporti*: ottimizzazione dei percorsi di consegna.
- *Produzione industriale*: minimizzazione dei tempi di lavorazione.
- *Robotica*: pianificazione dei movimenti.
- *Bioinformatica*: allineamento di sequenze genetiche. Le istanze possono variare notevolmente in dimensioni, da pochi nodi fino a migliaia, e in struttura, con grafi completi o incompleti.

3 Dettaglio sulle Metaeuristiche Implementate

Le metaeuristiche sono algoritmi di ottimizzazione che guidano la ricerca di soluzioni in spazi di ricerca complessi. Sono progettate per trovare soluzioni di buona qualità in tempi ragionevoli, sacrificando la garanzia di trovare la soluzione ottima.

3.1 Iterated Local Search (ILS)

L'Iterated Local Search è un algoritmo che esegue più ricerche locali (Local Search) partendo da soluzioni iniziali generate da perturbazioni della soluzione corrente. L'obiettivo è migliorare continuamente la soluzione con un processo iterativo che include una componente di "perturbazione" per esplorare nuove aree dello spazio delle soluzioni. Il funzionamento implementato è il seguente:

1. **Soluzione iniziale**: viene generata una soluzione iniziale, tramite l'algoritmo greedy del nearest neighbour.
2. **Local search**: si applica un algoritmo di ricerca locale per migliorare la soluzione corrente, e riportare la soluzione al proprio minimo locale. Il vicinato utilizzato è il 2-opt.

3. **Perturbazione:** la soluzione corrente viene modificata tramite una perturbazione per esplorare nuove aree dello spazio delle soluzioni. Viene applicato un multi-swap di k nodi, dove $k = n // 25$, e n è il numero di nodi.
4. Si ripete la ricerca locale e la perturbazione, accettando ogni volta una soluzione migliore o una "buona" soluzione secondo criteri predefiniti.

I parametri dell'ILS sono:

- Iterazioni massime: 100
- Early stopping: 20 iterazioni

La scelta di questi parametri mira a garantire un compromesso tra esplorazione e tempi di calcolo. La perturbazione tramite multi-swap consente di modificare la soluzione in modo significativo, aumentando le probabilità di sfuggire ai minimi locali.

3.2 Simulated Annealing (SA)

Il SA si ispira al processo di raffreddamento dei materiali in metallurgia, dove il graduale raffreddamento permette agli atomi di raggiungere una configurazione stabile. L'algoritmo segue questi passi:

1. **Soluzione iniziale:** viene generata una soluzione iniziale, tramite l'algoritmo greedy del nearest neighbour.
2. **Modifica della soluzione:** viene generata una nuova soluzione vicina alla corrente, utilizzando l'algoritmo del 2-opt.
3. **Accettazione probabilistica:** la nuova soluzione è accettata con una probabilità che dipende dalla differenza di lunghezza e da un parametro di temperatura.
4. **Riduzione della temperatura:** il parametro di temperatura viene gradualmente ridotto, diminuendo la probabilità di accettare soluzioni peggiori. Il raffreddamento è di tipo *geometrico*.

I parametri del SA sono:

- Temperatura iniziale (T_0): 1000, abbastanza alta così da avere un'esplorazione iniziale vasta, ma non troppo;
- Fattore di raffreddamento (α): 0.95, così da convergere lentamente;
- Iterazioni massime: 10000
- Iterazioni con temperatura costante: 50
- Temperatura minima (T_{\min}): 0.0001

I parametri sono stati scelti per garantire un equilibrio tra esplorazione dello spazio delle soluzioni e tempi computazionali. La temperatura iniziale elevata consente un'ampia esplorazione iniziale, mentre il fattore di raffreddamento progressivo bilancia l'intensificazione della ricerca.

3.3 Metaeuristica Ibrida (ILS + SA)

La metaeuristica ibrida combina i punti di forza di ILS e SA. La fusione tra ILS e SA è efficace perché unisce l'abilità dell'ILS di esplorare ampie regioni dello spazio delle soluzioni, grazie a perturbazioni significative, con la capacità del SA di ottimizzare localmente accettando temporaneamente soluzioni peggiori. Il funzionamento dell'algoritmo è il seguente:

1. Generazione di una soluzione iniziale con algoritmo greedy (nearest neighbour).
2. Applicazione del SA per affinare la soluzione iniziale.
3. Perturbazione della soluzione basata su una funzione adattiva
4. Applicazione del SA per riottimizzare la soluzione perturbata.
5. Questi ultimi due passi vengono ripetuti iterativamente.

La funzione di perturbazione si adatta alla fase di esplorazione. La perturbazione può essere:

- **Aggressive**, usata nel primo 50% dell'esplorazione. La perturbazione è la Double bridge move, così da modificare porzioni significative del percorso.
- **Medium**, usata nel successivo 30% dell'esplorazione. La perturbazione è il Multi-swap, così da alterare gruppi moderati di nodi.
- **Soft**, usata nell'ultimo 20% dell'esplorazione. La perturbazione è il 2-opt randomizzato, così da variare lievemente la soluzione

L'uso di una funzione adattiva per le perturbazioni consente di regolare l'intensità delle modifiche man mano che l'algoritmo converge, garantendo una maggiore esplorazione nelle prime fasi e una raffinazione più mirata nelle fasi finali. Questa combinazione di strategie permette di ottenere soluzioni di alta qualità, sfruttando la complementarità delle due metaeuristiche. I parametri sono:

- Temperatura iniziale (T_0): 1000, abbastanza alta così da avere un'esplorazione iniziale vasta, ma non troppo;
- Fattore di raffreddamento (α): 0.95, così da convergere lentamente;
- Iterazioni massime dell'SA: 10000
- Iterazioni con temperatura costante: 50
- Temperatura minima (T_{\min}): 0.0001
- Numero massimo di iterazioni totali: 100

4 Implementazione Tecnica

Il progetto è organizzato in diverse directory modulari per garantire chiarezza e riutilizzabilità del codice:

- *analysis*: script per analizzare i risultati delle simulazioni.
- *utils*: funzioni di supporto per la gestione dei dati e degli algoritmi.

- *plotting*: script per la generazione di grafici e visualizzazioni.
- *data*: raccolta delle istanze di test (TSPLIB e dataset generati).
- *outputs*: risultati e grafici generati durante l'analisi.

L'intero progetto è sviluppato in Python, con l'utilizzo di librerie standard per il calcolo numerico e la visualizzazione (es. NumPy, Matplotlib).

5 Analisi Sperimentale

I grafici mostrati rappresentano la distanza delle soluzioni trovate dalle varie metaeuristiche e dalla metaeuristica ibrida sulle istanze della TSPLIB in confronto con l'ottimo della singola istanza. Questi sono organizzati in tre macro-sezioni in base alla tipologia delle istanze:

- **Istanze Euclidee**: Basate su coordinate in uno spazio bidimensionale.
- **Istanze Geografiche**: Basate su coordinate geografiche, per quanto poche e talvolta affette da file danneggiati.
- **Istanze con Matrice Esplicita**: Basate su una matrice delle distanze, anch'esse in numero limitato e con alcuni file corrotti.

Ogni macro-sezione è ulteriormente suddivisa in base alla dimensione delle istanze, con le seguenti categorie:

- *Da 0 a 50 nodi* (laddove presenti).
- *Da 51 a 100 nodi* (laddove presenti).
- *Da 101 a 200 nodi* (laddove presenti).
- *Da 201 a 500 nodi* (laddove presenti).
- *Da 501 a 1000 nodi* (laddove presenti).

La distanza trovata è stata normalizzata rispetto alla distanza ottima, presente nei file della TSPLIB, per facilitare il confronto tra i risultati.

Nei grafici, ogni metaeuristica è rappresentata come segue:

- Iterated Local Search (ILS): Barra blu.
- Simulated Annealing (SA): Barra verde.
- Metaeuristica Ibrida (ILS + SA): Barra rossa.

Un puntino nero, corrispondente al valore "1", rappresenta la distanza ottima per ciascuna istanza. Questo consente di visualizzare immediatamente il livello di approssimazione di ciascun metodo rispetto alla soluzione ottima.

5.1 EUC_2D

L'analisi dei risultati ottenuti per le istanze euclidee del TSP evidenzia diverse tendenze a seconda delle dimensioni dei grafi analizzati:

- *Grafi con meno di 100 nodi:*

La metaeuristica ibrida mostra prestazioni modeste, riuscendo talvolta ad avvicinarsi all'ottimo ma senza consistenza. Il Simulated Annealing (SA) si comporta peggio, trovando soluzioni con un costo mediamente superiore del 20% rispetto a quello ottimale. L'Iterated Local Search (ILS) risulta essere la tecnica più efficace, ottenendo in media soluzioni migliori.

- *Grafi con meno di 200 nodi:*

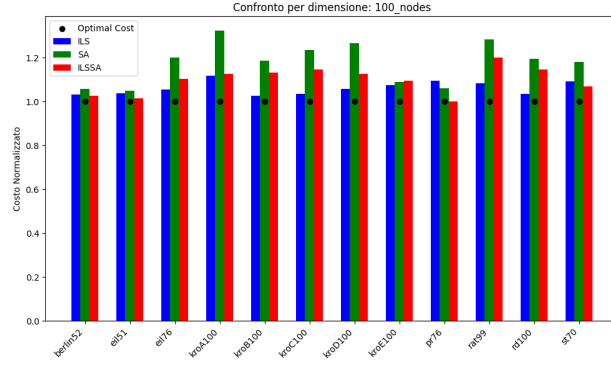
Il comportamento generale rimane simile a quello osservato per i grafi più piccoli. L'ILS continua a dominare le prestazioni, mentre l'ibrida e il SA non mostrano miglioramenti significativi.

- *Grafi con meno di 500 nodi:*

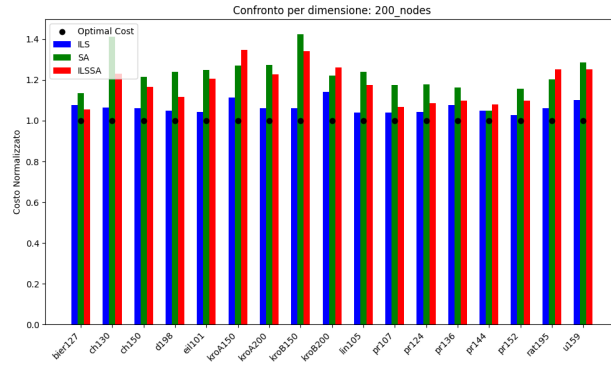
L'ILS mantiene un buon livello di efficacia, trovando soluzioni con un costo del 5-10% peggiore rispetto a quello ottimo. Tuttavia, il SA continua a fornire soluzioni con costi maggiorati del 20%. La metaeuristica ibrida, invece, inizia a peggiorare ulteriormente, generando soluzioni peggiori del 20% rispetto all'ottimo.

- *Grafi con meno di 1000 nodi:*

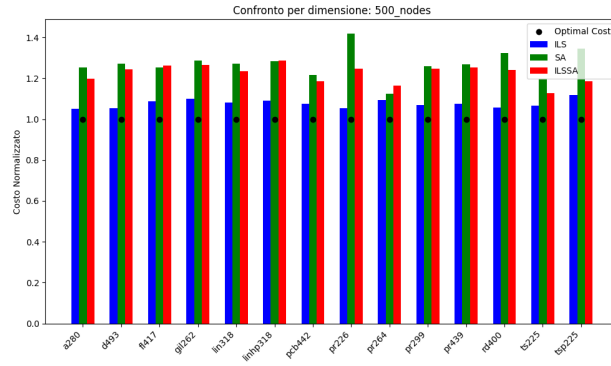
Con l'aumentare delle dimensioni, la performance di SA e della metaeuristica ibrida degrada ulteriormente, con soluzioni che superano il costo ottimale del 30%. L'ILS si dimostra ancora una volta la tecnica più robusta, con soluzioni mediamente peggiori solo del 10% rispetto a quelle ottimali.



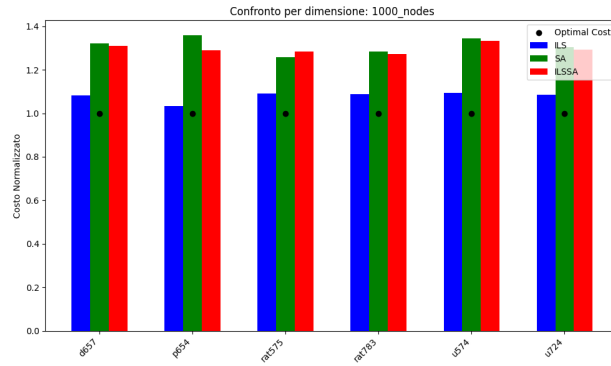
(a) 100 nodi



(b) 200 nodi



(c) 500 nodi



(d) 1000 nodi

Figure 1: Plots di EUC 2D con differenti numeri di nodi

5.2 EXPLICIT

L'analisi dei risultati ottenuti per le istanze con la matrice delle distanze esplicite evidenzia il seguente comportamento:

- *Grafi con meno di 100 nodi:*

La metaeuristica ibrida si distingue nettamente, trovando sempre la soluzione ottima. L'ILS si discosta in modo significativo dalla soluzione ottimale, mentre il Simulated Annealing (SA) si allontana solo leggermente.

- *Grafi di dimensioni maggiori:*

In queste istanze più grandi, l'ILS emerge come la tecnica migliore, ottenendo risultati più vicini alla soluzione ottimale. Sia la metaeuristica ibrida che il SA mostrano un calo di prestazioni, con soluzioni che si distaccano in media del 20% rispetto all'ottimo.

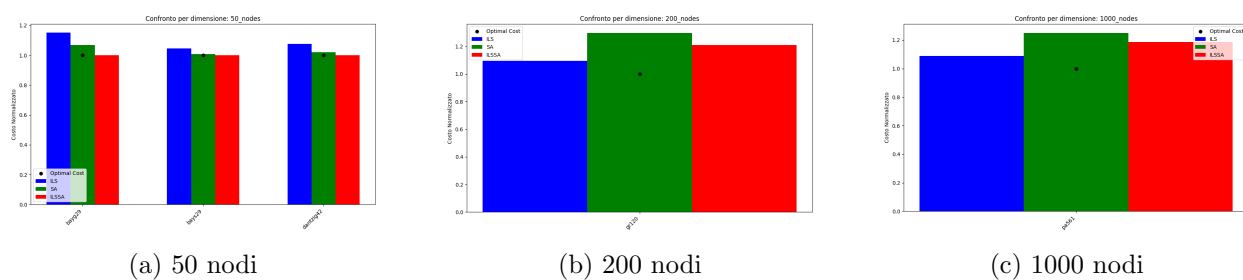


Figure 2: Plots di EXPLICIT con differenti numeri di nodi

5.3 GEO

L'analisi dei risultati per le istanze geografiche mostra il seguente comportamento:

- *Grafi piccoli (meno di 50 o 100 nodi):*
La metaeuristica ibrida si comporta in modo eccellente, trovando sempre la soluzione ottima. Anche il Simulated Annealing (SA) ottiene risultati altrettanto validi, mentre l'Iterated Local Search (ILS) si discosta leggermente dall'ottimo.
- *Grafi di medie dimensioni (meno di 200 nodi):*
La metaeuristica ibrida continua a ottenere ottimi risultati, mantenendo prestazioni costanti.
- *Grafi più grandi (meno di 500 e 1000 nodi):*
Con l'aumento delle dimensioni, sia la metaeuristica ibrida che il SA subiscono un peggioramento delle prestazioni, trovando soluzioni significativamente meno ottimali. L'ILS, al contrario, mostra una maggiore stabilità e rimane costante nelle sue prestazioni.

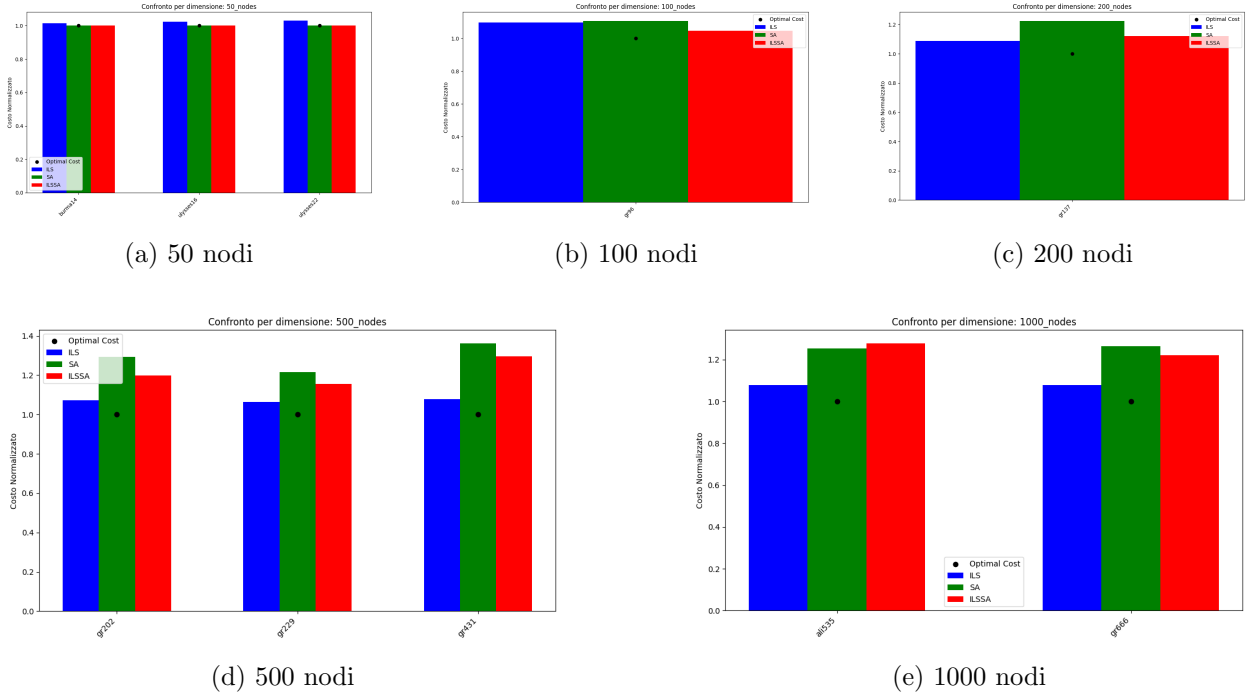


Figure 3: Plots di GEO con differenti numeri di nodi

6 Conclusioni

Dall'analisi delle tre tipologie di istanze (euclidee, con matrici esplicite e geografiche), emergono diverse considerazioni generali sulle prestazioni delle metaeuristiche analizzate:

- **Metaeuristica Ibrida**

Mostra prestazioni eccellenti su grafi di piccole dimensioni, specialmente per le istanze geografiche e quelle con matrici esplicite, dove riesce spesso a raggiungere la soluzione ottima. Tuttavia, con l'aumentare della dimensione del grafo, il suo comportamento peggiora significativamente, soprattutto nelle istanze euclidee, arrivando a soluzioni anche del 30% peggiori rispetto all'ottimo. Questo suggerisce che la metaeuristica ibrida non scala bene su grafi più grandi.

- **Iterated Local Search (ILS)**

È la metaeuristica più consistente e affidabile, specialmente nelle istanze euclidee e con matrici esplicite, dove riesce a mantenere uno scarto rispetto all'ottimo entro il 10% anche per grafi di grandi dimensioni. Nonostante qualche difficoltà su grafi molto piccoli, ILS si dimostra robusto e versatile, confermandosi come una soluzione ottimale per problemi di dimensioni medio-grandi.

- **Simulated Annealing (SA)**

Presenta prestazioni accettabili solo su istanze piccole, come quelle geografiche o euclidee con meno di 100 nodi, ma peggiora rapidamente con l'aumentare delle dimensioni del grafo. Il distacco rispetto all'ottimo, che inizia con un 20% sui grafi piccoli, si amplia progressivamente fino a raggiungere il 30% o più su grafi molto grandi. Questo indica che l'SA è meno adatto per affrontare istanze di grandi dimensioni.

Quindi tra le metaeuristiche analizzate, ILS si distingue come la più stabile e performante, specialmente su grafi più complessi o di grandi dimensioni. La metaeuristica ibrida, pur mostrando eccellenti prestazioni iniziali, si rivela meno scalabile, mentre il Simulated Annealing, nonostante la semplicità, soffre di limiti evidenti in termini di qualità delle soluzioni per istanze grandi. La scelta della metaeuristica più adatta dipende quindi dal tipo di istanza e dalla dimensione del grafo, ma in generale, ILS rappresenta un compromesso ottimale tra precisione e scalabilità.

Un aspetto critico da evidenziare è che, sebbene l'Iterated Local Search (ILS) si sia dimostrata la metaeuristica più solida e stabile in termini di qualità delle soluzioni, i suoi tempi di esecuzione sono significativamente più elevati rispetto a quelli del Simulated Annealing (SA) e della metaeuristica ibrida (ILSSA). Questo è principalmente dovuto al costo computazionale legato alla generazione dei vicinati nella fase di local search. Per grafi con più di 1000 nodi, ILS diventa praticamente inutilizzabile a causa dei tempi di calcolo proibitivi, e già per grafi di grandi dimensioni i tempi di esecuzione si sono rivelati molto elevati, limitandone la scalabilità in scenari reali.