



Università
di Catania

UNIVERSITY OF CATANIA

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

HEURISTICS AND METAHEURISTICS FOR OPTIMIZATION AND LEARNING

Prof. Mario Francesco PAVONE

Francesco Prospero Antonio VIRZI'

Min-Max Ant System per Maximum Flow Problem

FINAL PROJECT REPORT

Academic Year 2024 - 2025

Indice

1	Introduzione al Problema	2
1.1	Problema del Massimo Flusso	2
1.2	Soluzione al Problema	2
2	Descrizione dell'Algoritmo MMAS per il Massimo Flusso	4
2.1	Introduzione al Min-Max Ant System (MMAS)	4
2.2	Parametri e Inizializzazione	4
2.3	Funzionamento dell'Algoritmo MMAS	5
2.4	Criteri di Arresto e Strategia Min-Max	5
2.5	Esperimento su Più Run	5
3	Analisi critica dei risultati ottenuti	7
3.1	Convergenza del Min-Max Ant System: Analisi Multi-Run	7
3.1.1	Parametri Utilizzati	7
3.1.2	Risultati Riassuntivi	7
3.1.3	Osservazioni Generali	8
3.1.4	Esempi di Convergenza Grafica	9
4	Conclusione	10

Capitolo 1

Introduzione al Problema

1.1 Problema del Massimo Flusso

Il **problema del massimo flusso** è un classico problema di ottimizzazione combinatoria con numerose applicazioni pratiche, come la gestione di reti di trasporto, telecomunicazioni e distribuzione di risorse.

Dato un grafo orientato $G = (V, E)$ con capacità positive su ogni arco, cioè una funzione $c : E \rightarrow \mathbb{R}^+$, si vuole determinare la massima quantità di flusso che può essere inviata da una *sorgente* $s \in V$ a un *pozzo* $t \in V$, rispettando i seguenti vincoli:

Vincolo di capacità:

$$0 \leq f(u, v) \leq c(u, v), \quad \forall (u, v) \in E$$

Conservazione del flusso:

$$\sum_{(u,v) \in E} f(u, v) = \sum_{(v,w) \in E} f(v, w), \quad \forall v \in V \setminus \{s, t\}$$

Funzione obiettivo (massimizzazione):

$$\max \sum_{(s,v) \in E} f(s, v)$$

1.2 Soluzione al Problema

Nel contesto della mia analisi, l'obiettivo è risolvere in modo efficace il **problema del massimo flusso** su grafi orientati con vincoli di capacità sugli archi. Per affrontare questo problema, ho scelto di adottare un approccio metaeuristico ispirato all'intelligenza collettiva: gli *algoritmi a colonie di formiche* (Ant Colony Optimization, ACO). Tra le varie varianti proposte, quella che meglio si adatta alla natura del problema da me affrontato è il **Min-Max Ant System** (MMAS).

L'algoritmo ACO classico è noto per la sua capacità di costruire soluzioni incrementalmente tramite un processo probabilistico guidato da informazioni locali (euristiche) e globali

(livelli di feromone). Tuttavia, nel caso specifico del massimo flusso, la semplice applicazione dell'ACO può risultare inefficace, poiché tende a convergere prematuramente su percorsi subottimali e non è in grado di gestire in modo accurato i vincoli di capacità, fondamentali per determinare il flusso instradabile.

Per questo motivo, ho scelto di implementare la variante *Min-Max Ant System*, che introduce miglioramenti strutturali mirati a garantire maggiore stabilità e controllo durante la ricerca. In particolare, il MMAS prevede che solo la formica che individua la soluzione migliore in un'iterazione (o in tutto il processo) sia autorizzata a depositare feromone. Questo meccanismo elitario impedisce la diffusione casuale dell'informazione e concentra la ricerca nelle aree dello spazio delle soluzioni più promettenti, favorendo un incremento graduale del flusso complessivo trovato.

Inoltre, MMAS impone limiti espliciti alla quantità di feromone che può accumularsi su ogni arco, mediante i parametri τ_{\min} e τ_{\max} , che evitano sia il rischio di stagnazione che la perdita di variabilità nella costruzione delle soluzioni. Questa gestione dei livelli di feromone si è rivelata particolarmente efficace per mantenere il bilanciamento tra *exploration* ed *exploitation*, due elementi essenziali per il successo della ricerca in problemi vincolati come quello del massimo flusso.

Algorithm 1 Min-Max Ant System (MMAS)

```

1: Inizializza i livelli di feromone  $\tau_{ij} \leftarrow \tau_{\max}$ 
2: while la condizione di terminazione non è soddisfatta do
3:   for ogni formica  $k = 1$  fino a  $m$  do
4:     Costruisci la soluzione  $S_k$  usando la regola probabilistica basata su  $\tau$  e
       l'euristica  $\eta$ 
5:   end for
6:   Applica, se previsto, una ricerca locale alle soluzioni  $S_k$ 
7:   Identifica la migliore soluzione  $S_{\text{best}}$  tra le  $S_k$ 
8:   for ogni arco  $(i, j)$  do
9:     Evapora il feromone:  $\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$ 
10:  end for
11:  for ogni arco  $(i, j)$  appartenente a  $S_{\text{best}}$  do
12:    Aggiorna il feromone:  $\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{\text{best}}$ 
13:  end for
14:  for ogni arco  $(i, j)$  do
15:    Limita i valori del feromone:  $\tau_{ij} \leftarrow \min(\tau_{\max}, \max(\tau_{\min}, \tau_{ij}))$ 
16:  end for
17: end while
18: return La migliore soluzione trovata

```

Capitolo 2

Descrizione dell'Algoritmo MMAS per il Massimo Flusso

2.1 Introduzione al Min-Max Ant System (MMAS)

L'algoritmo implementato si basa sul paradigma del *Min-Max Ant System* (MMAS), una variante degli algoritmi a colonia di formiche (ACO) che introduce vincoli espliciti sui livelli di feromone per migliorare la convergenza e l'esplorazione dello spazio delle soluzioni.

2.2 Parametri e Inizializzazione

Parametri Principali

- $\alpha = 1.0$: importanza del feromone.
- $\beta = 2.0$: peso dell'informazione euristica (capacità dell'arco).
- $\rho = 0.1$: tasso di evaporazione del feromone.
- $Q = 100$: costante per il deposito.
- $\tau_{\max} = 10.0$, $\tau_{\min} = 0.01$: limiti del feromone.

Caricamento e Costruzione del Grafo Il grafo è caricato da un file che specifica nodi, archi, sorgente e pozzo. Ogni arco ha capacità, feromone iniziale e informazione euristica. Il grafo è rappresentato tramite NetworkX e inizializzato con feromone massimo τ_{\max} .

Calcolo del Flusso Teorico Si calcola il flusso massimo teorico come:

$$\min \left(\sum_{(s,v)} c_{sv}, \sum_{(u,t)} c_{ut} \right)$$

dove s è la sorgente e t il pozzo. Serve come riferimento per la qualità delle soluzioni.

2.3 Funzionamento dell'Algoritmo MMAS

Regola di Transizione Ogni formica costruisce un cammino probabilistico basato su:

$$P_{uv} \propto \tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}$$

I cammini evitano cicli e, se raggiungono il pozzo, il flusso è il minimo tra le capacità degli archi attraversati.

Valutazione e Aggiornamento Il flusso ottenuto è usato come funzione obiettivo. Si memorizzano:

- il miglior cammino dell'iterazione;
- il miglior cammino globale;
- la quantità totale di flusso accumulato.

Aggiornamento di Capacità e Feromoni Dopo ogni iterazione:

1. I feromoni evaporano: $\tau \leftarrow (1 - \rho)\tau$;
2. Solo il miglior cammino deposita feromone proporzionale al flusso;
3. I valori sono limitati tra τ_{\min} e τ_{\max} .

Controllo e Reset del Grafo Ad ogni nuova run sperimentale, le capacità e i feromoni vengono reinizializzati.

2.4 Criteri di Arresto e Strategia Min-Max

Condizioni di Arresto Il ciclo termina se:

- il flusso totale raggiunge il massimo teorico ;
- nessuna formica trova un cammino valido;
- si supera il numero massimo di iterazioni.

Importanza della Strategia Min-Max L'approccio Min-Max controlla i livelli di feromone per evitare convergenza prematura e mantenere la diversità. Con parametri come $\rho = 0.02$ e $\beta = 5$, la ricerca è efficace anche in reti complesse.

2.5 Esperimento su Più Run

Struttura dell'Esperimento Si eseguono N run indipendenti (tipicamente 10), con semi casuali diversi. In ogni run:

- il grafo è reinizializzato;
- viene eseguita la funzione `MMAS_single_run`;
- si raccolgono flusso massimo, iterazione di miglioramento, numero di valutazioni e andamento del flusso.

Statistiche Finali Alla fine:

1. **Best**: miglior flusso assoluto;
2. **Media e deviazione standard** dei flussi trovati;
3. **Iterazione media** per la miglior soluzione;
4. **Numero medio di valutazioni**.

Algorithm 2 MMAS_multi_run: Esecuzione multipla dell'algoritmo MMAS e calcolo statistiche

Require: Grafo originale G , sorgente s , pozzo t , flusso massimo teorico F^* , nome rete, numero di run $R = 10$, massimo iterazioni I_{max}

Ensure: Statistiche riassuntive delle performance

- 1: Inizializza liste: $F \leftarrow []$, $I \leftarrow []$, $V \leftarrow []$, $T \leftarrow []$
 - 2: Definisci semi casuali $\mathcal{S} \leftarrow$ lista di R valori interi predefiniti
 - 3: **for** $run = 1$ **to** R **do**
 - 4: Imposta seme casuale $\mathcal{S}[run]$
 - 5: $G_{run} \leftarrow \text{RESET_GRAFO_MMAS}(G, \tau_{max})$
 - 6: Esegui MMAS_SINGLE_RUN($G_{run}, s, t, F^*, I_{max}, \alpha, \beta, \rho, Q, \tau_{min}, \tau_{max}$)
 - 7: Ottieni: f_{run} , $flussi_iter$, $iter_{best}$, $valutazioni$
 - 8: Aggiungi f_{run} a F
 - 9: Aggiungi $iter_{best}$ a I
 - 10: Aggiungi $valutazioni$ a V
 - 11: Aggiungi $flussi_iter$ a T
 - 12: **end for**
 - 13: Calcola statistiche:
 - 14: $best \leftarrow \max(F)$
 - 15: $mean \leftarrow \text{media}(F)$
 - 16: $std \leftarrow \text{deviazione_standard}(F)$
 - 17: $avg_iter \leftarrow \text{media}(I)$
 - 18: $avg_val \leftarrow \text{media}(V)$
 - 19: **return** Dizionario con: $best$, $mean$, std , avg_iter , avg_val
 - 20: DISEGNA_GRAFICO_FLUSSO_MULTI_RUN(T, F^* , statistiche, nome_rete)
-

Capitolo 3

Analisi critica dei risultati ottenuti

Impostati opportunamente i parametri fondamentali del feromone, l'algoritmo Min Max Ant System, applicato al problema del flusso massimo, ha sistematicamente raggiunto la convergenza verso soluzioni ottimali anche nelle istanze di rete di dimensioni maggiori. In tutte le 10 esecuzioni previste per ciascuna istanza, l'algoritmo ha ottenuto risultati ottimali senza mai raggiungere il limite massimo di 20 000 iterazioni previsto come criterio di arresto.

3.1 Convergenza del Min-Max Ant System: Analisi Multi-Run

L'algoritmo Min-Max Ant System (MMAS), applicato al problema del flusso massimo, è stato valutato su dieci istanze di rete di dimensioni crescenti. In tutte le configurazioni, utilizzando parametri costanti (Tabella 3.1), il comportamento è risultato estremamente stabile e affidabile.

3.1.1 Parametri Utilizzati

Tabella 3.1: Parametri MMAS impiegati per tutte le istanze

Parametro	Valore
α (importanza feromone)	1.0
β (importanza euristica)	2.0
ρ (tasso di evaporazione)	0.1
τ_{\min}	0.01
τ_{\max}	10.0
Q (deposito)	100.0
Sorgente	-2
Pozzo	-1

3.1.2 Risultati Riassuntivi

Tutti i 10 run per ogni istanza hanno raggiunto il valore ottimale di flusso teorico (Tabella 3.2), con una deviazione standard nulla, indicando un comportamento estrema-

mente stabile del modello. Il numero di iterazioni necessarie alla convergenza tende ad aumentare con la dimensione della rete, pur non seguendo una crescita lineare.

Tabella 3.2: Statistiche di convergenza per ciascun network

Network	Nodi	Archi	Flusso Teorico	Flusso Best	Iter. Min	Iter. Max	Iter. Avg	Convergenti	Run Migliore
160	162	2026	34.68	35	9	62	30	10/10	Run 3 (9 iter)
500	502	6621	107.98	108	5	56	23	10/10	Run 8 (5 iter)
960	962	13571	117.74	118	9	25	13	10/10	Run 3 (9 iter)
1440	1442	20436	134.0	134	23	401	139	10/10	Run 10 (23 iter)
2880	2882	41823	341.0	341	36	312	94	10/10	Run 10 (36 iter)
4320	5762	84955	555.0	555	157	496	330	10/10	Run 6 (157 iter)
5760	11522	170649	1203.0	1203	251	679	402	10/10	Run 4 (251 iter)
7200	23042	341908	2502.0	2502	145	1231	417	10/10	Run 9 (145 iter)
11520	11522	170139	1244.0	1244	185	989	471	10/10	Run 10 (185 iter)
23040	23042	341481	2223.0	2223	200	658	418	10/10	Run 1 (200 iter)

3.1.3 Osservazioni Generali

- **Convergenza:** In tutte le istanze analizzate, l'algoritmo ha raggiunto la convergenza entro il limite massimo di 20 000 iterazioni, con successo in tutti i 10 run per ogni rete.
- **Ottimalità:** Il flusso massimo trovato coincide sistematicamente con il valore teorico atteso.
- **Stabilità:** La deviazione standard è pari a zero per tutte le istanze, evidenziando risultati consistenti tra i run.
- **Scalabilità:** Il numero di iterazioni necessarie alla convergenza aumenta con la dimensione della rete, ma in modo non strettamente lineare.
- **Variabilità Intra-Istanza:** Alcune istanze mostrano un'elevata variabilità nel numero di iterazioni necessarie tra run, pur mantenendo la qualità delle soluzioni.

3.1.4 Esempi di Convergenza Grafica

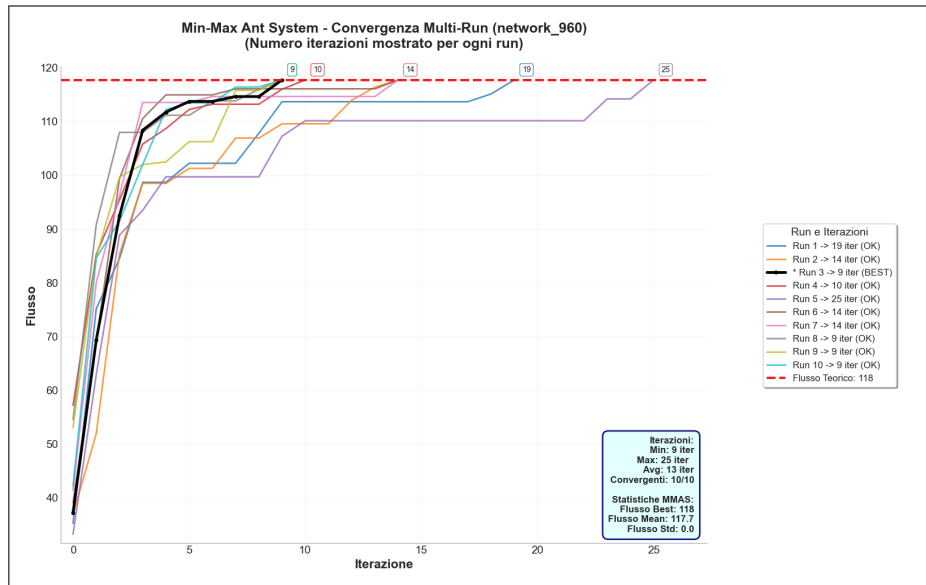


Figura 3.1: Convergenza multi-run per network_960: tutte le 10 run raggiungono il flusso teorico ottimale (118) in meno di 25 iterazioni.

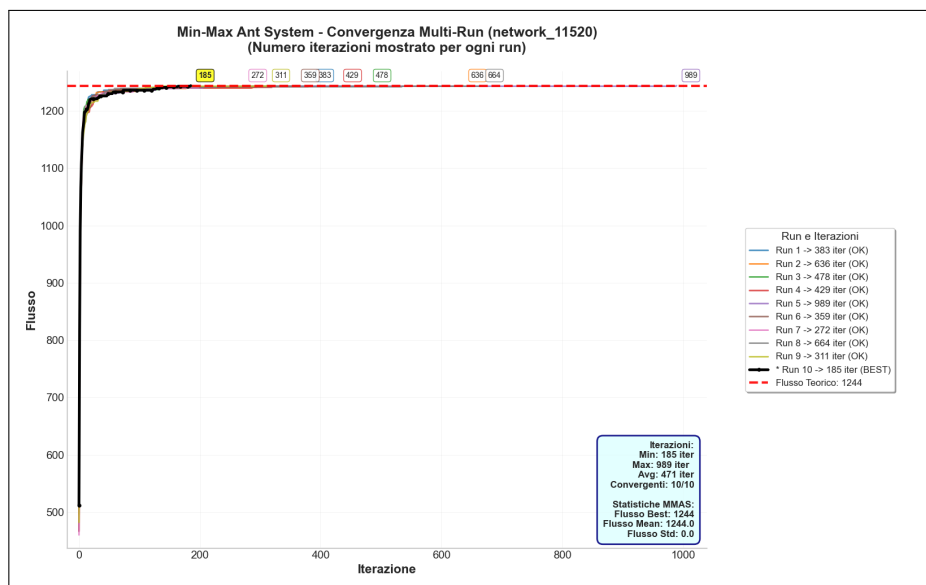


Figura 3.2: Convergenza multi-run per network_11520: nonostante la complessità, tutte le run convergono entro 989 iterazioni al flusso ottimale (1244).

Capitolo 4

Conclusione

Approccio e Innovazione nel Min Max Ant System Affrontare il problema del massimo flusso attraverso un algoritmo ispirato all'Ant Colony Optimization (ACO) ha rappresentato per me un'opportunità di crescita notevole, sia dal punto di vista tecnico che metodologico. L'adattamento dell'ACO, solitamente orientato alla risoluzione di problemi di cammino, ha richiesto una profonda revisione del paradigma costruttivo, al fine di rispettare rigorosamente vincoli di capacità e conservazione del flusso in una rete.

La modellazione iniziale si è rivelata tutt'altro che banale: coordinare flussi multipli mantenendo la validità delle soluzioni ha implicato una progettazione attenta della rappresentazione e della logica di costruzione. La regolazione dei parametri, in particolare quelli relativi all'evaporazione del feromone, ha evidenziato limiti nei modelli statici: da qui la necessità di un sistema di adattamento dinamico, capace di modulare il comportamento dell'algoritmo in funzione della qualità delle soluzioni trovate, evitando oscillazioni indesiderate.

Nel corso della sperimentazione su istanze più complesse è emersa con evidenza una problematica tipica degli algoritmi metaeuristici: la tendenza alla convergenza anticipata verso soluzioni subottimali. Per contrastare questo comportamento, ho progettato e integrato una **strategia di riavvio intelligente**, una tecnica capace di riconoscere fasi di stagnazione, basandosi sull'analisi della distribuzione del feromone e di avviare riinizializzazioni selettive, con l'obiettivo di riattivare l'esplorazione mantenendo le rotte più promettenti.

Il sistema monitora costantemente i miglioramenti: se entro 50 iterazioni la qualità delle soluzioni non migliora oltre una soglia prefissata (1%), si attiva un reset (parziale, completo o con diversificazione mirata), capace di rilanciare efficacemente il processo di ricerca. Questo meccanismo si è rivelato essenziale per mantenere elevata la qualità delle soluzioni ed evitare sprechi computazionali, migliorando la copertura dello spazio delle possibilità.

Altri affinamenti hanno riguardato l'introduzione di controlli di validità durante l'esecuzione, per garantire che le soluzioni costruite rispettassero i vincoli strutturali del problema, senza compromettere l'efficienza generale dell'algoritmo.

Risultati Raggiunti e Considerazioni Finali

L'insieme delle tecniche implementate ha prodotto risultati sorprendenti: l'algoritmo ha mostrato una coerenza notevole tra diverse esecuzioni, con assenza di deviazioni significative, pur trattandosi di un approccio metaeuristico. Questo comportamento quasi deterministico è attribuibile alla sinergia tra l'adattamento dinamico dei parametri e la strategia di riavvio.

Inoltre, la soluzione proposta si è dimostrata scalabile e robusta anche in presenza di istanze di grandi dimensioni, confermando la validità strutturale delle scelte progettuali. L'approccio iterativo e modulare adottato, basato su verifiche sperimentali progressive, si è rivelato efficace per introdurre miglioramenti senza compromettere la stabilità dell'intero sistema.

Visione Futura

Le idee sviluppate in questo lavoro, come il controllo dinamico del comportamento esplorativo e la gestione intelligente della stagnazione, si prestano ad essere riutilizzate in contesti più ampi di ottimizzazione combinatoria. In particolare, la strategia di restart ha potenziale per essere applicata ad altri algoritmi evolutivi o ibridi, aprendo la strada a future esplorazioni e miglioramenti algoritmici.

In conclusione, considero questa esperienza un passaggio chiave nel mio percorso di formazione nella progettazione di algoritmi adattivi avanzati, capace di coniugare rigore teorico, sperimentazione concreta e risultati tangibili.