Relazione sul Problema del Weighted Feedback Vertex Set utilizzando un Algoritmo Ibrido Genetico (HGA)

Santi Lisi (1000044181)

Marzo 2025

Contents

1 Introduzione

Il problema del *Weighted Feedback Vertex Set* (WFVS) consiste nel trovare un sottoinsieme di vertici in un grafo, la cui rimozione renda il grafo aciclico, minimizzando la somma dei pesi dei vertici rimossi. Formalmente, dato un grafo $G = (V, E)$ e una funzione di peso $w : V \to \mathbb{R}^+$, l'obiettivo è:

$$\min_{S \subseteq V} \sum_{v \in S} w(v)$$

tale che il grafo $G' = (V \setminus S, E')$ sia aciclico.

1.1 Applicazioni ed Esempio

Il WFVS viene utilizzato in:

- Sistemi operativi, per la prevenzione dei deadlock.
- Sistemi distribuiti, per evitare stalli nelle comunicazioni.
- Verifica dei programmi e sicurezza informatica.

La Figura 1 mostra un esempio di grafo e la rimozione di vertici per ottenere un grafo aciclico.

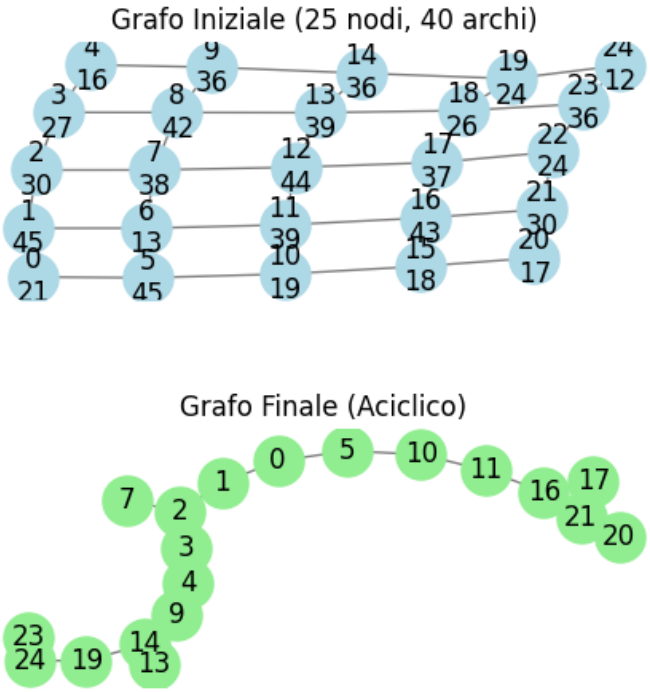

Figure 1: Esempio di grafo e rimozione dei vertici per ottenere un grafo aciclico. In cui una soluzione del FVS con 25 nodi e 40 archi è: 6,8,12,15,18,22

2 Algoritmo Utilizzato

Per risolvere il problema del *Weighted Feedback Vertex Set* (WFVS) è stato sviluppato un Algoritmo Ibrido Genetico (HGA) che integra operatori evolutivi, una funzione di riparazione e una strategia di ricerca locale. L'approccio scelto è particolarmente efficace per questo problema per i seguenti motivi:

- **Selezione a torneo:** Ad ogni torneo vengono scelti k individui casualmente e si seleziona il migliore, garantendo un equilibrio tra diversità e selezione del meglio.
- **Crossover a un punto:** I genitori vengono suddivisi in un punto casuale, scambiando le rispettive porzioni per generare nuovi individui, facilitando lo scambio di informazioni utili.
- **Mutazione:** La mutazione viene applicata su ogni gene (rappresentante di un vertice della soluzione) con una probabilità fissa (ad esempio, 0.065), garantendo l'esplorazione dell'intero spazio delle soluzioni.
- **Funzione di riparazione:** Se una soluzione genera un grafo che contiene cicli, la funzione di riparazione interviene rimuovendo il vertice con il peso minimo all'interno di un ciclo, assicurando così che il grafo risultante sia aciclico.
- **Ricerca locale:** Viene eseguito un hill climbing basato sul criterio di *best improvement*. In altre parole, viene esplorato un campione casuale dei geni e si accetta il miglior miglioramento possibile, affinando la soluzione in maniera efficace.

Questa combinazione consente di bilanciare l'esplorazione globale della popolazione con un affinamento locale accurato, garantendo soluzioni valide e ottimizzate per il problema WFVS.

2.1 Parametri del HGA

I parametri dell'algoritmo HGA implementato sono i seguenti:

- **Dimensione della popolazione**: il numero di soluzioni iniziali generate casualmente. Un'ampia popolazione favorisce una migliore esplorazione dello spazio delle soluzioni.
- **Numero di generazioni**: il numero di iterazioni in cui la popolazione viene evoluta. Un numero elevato di generazioni permette un affinamento progressivo, anche se aumenta il tempo di esecuzione.
- **Dimensione del torneo**: il numero di individui selezionati casualmente per determinare il migliore durante la selezione a torneo. Questo parametro bilancia la pressione selettiva con il mantenimento della diversità.
- **Tasso di mutazione**: la probabilità che ogni gene (rappresentante di un vertice) venga invertito. Un valore moderato aiuta a evitare la stagnazione mantenendo la possibilità di esplorare nuove soluzioni.

- **Tasso di crossover**: la probabilità che due individui vengano incrociati per generare figli. Un alto tasso di crossover favorisce lo scambio di informazioni tra le soluzioni, contribuendo a miglioramenti significativi.
- **Dimensione del sottoinsieme per la local search**: il numero di geni scelti casualmente da esplorare durante la ricerca locale (hill climbing). Questo parametro controlla il compromesso tra efficienza computazionale e capacità di affinamento della soluzione.

Grazie a questi parametri, l'algoritmo riesce a esplorare efficacemente lo spazio delle soluzioni e a trovare soluzioni progressivamente più ottimali.

2.2 Pseudocodice dell'Algoritmo Ibrido Genetico

Algorithm 1 Hybrid Genetic Algorithm (Essenziale)

Require: Grafo G, pesi w, costo massimo $maxCost$, dimensione popolazione pop_size, numero di generazioni G_{max}, parametro torneo k, tasso di crossover p_c, tasso di mutazione p_m, parametro local search ls_sample

Inizializza popolazione P con soluzioni casuali
for $gen = 1$ **to** G_{max} **do**
  Valuta la fitness di ogni soluzione in P
  $new\_population \leftarrow [\,]$
  while $|new\_population| < pop\_size$ **do**
    $p_1 \leftarrow$ **TournamentSelection**(P, k)
    $p_2 \leftarrow$ **TournamentSelection**(P, k)
    if random $< p_c$ **then**
      $(c_1, c_2) \leftarrow$ **Crossover**(p_1, p_2)
    else
      $c_1 \leftarrow p_1$, $c_2 \leftarrow p_2$
    end if
    $c_1 \leftarrow$ **Mutate**(c_1, p_m), $c_2 \leftarrow$ **Mutate**(c_2, p_m)
    $c_1 \leftarrow$ **LocalSearch**(c_1, ls_sample), $c_2 \leftarrow$ **LocalSearch**(c_2, ls_sample)
    Aggiungi c_1, c_2 a $new\_population$
  end while
  $P \leftarrow new\_population[0 : pop\_size]$
  if $gen \mod 10 = 0$ **then**
    Stampa lo stato attuale (best fitness)
  end if
end for
return La migliore soluzione trovata in P

2.3 Pseudocodice degli Operatori Evolutivi

2.3.1 Tournament Selection

Algorithm 2 Tournament Selection

Require: Popolazione P, fitness f, parametro k
1: Seleziona casualmente k individui da P
2:
3: **return** l'individuo con il valore di fitness minimo

2.3.2 One Point Crossover

Algorithm 3 One Point Crossover

Require: Genitori P_1 e P_2
1: **if** lunghezza di P_1 o $P_2 < 2$ **then**
2: **return** P_1, P_2
3: **end if**
4: Scegli un punto di taglio p casuale, con $1 \leq p \leq (\text{lunghezza} - 1)$
5: **Child1** $\leftarrow$ primi p geni di P_1 concatenati con i restanti geni di P_2
6: **Child2** $\leftarrow$ primi p geni di P_2 concatenati con i restanti geni di P_1
7: **return** **Child1**, **Child2**

2.3.3 Mutazione

Algorithm 4 Mutate

Require: Individuo I, tasso di mutazione r
1: **for** ogni gene in I **do**
2: **if** numero casuale $< r$ **then**
3: Inverti il gene (se 0 diventa 1, se 1 diventa 0)
4: **end if**
5: **end for**
6: **return** l'individuo mutato

2.3.4 Ricerca Locale (Hill Climbing con First Improvement)

Algorithm 5 Local Search (Hill Climbing First Improvement)

Require: Soluzione s, grafo G, pesi w, costo massimo $maxCost$, lista di archi E, parametro $sample\_size$

$s \leftarrow$ **Ripara**(s, G, w, E)
$best \leftarrow s$, $best\_fit \leftarrow$ Fitness($best$, G, w, $maxCost$, E)
$improved \leftarrow true$
while $improved$ **do**
  $improved \leftarrow false$
  Definisci l'insieme degli indici $I \leftarrow \{1, \ldots, |s|\}$
  if $sample\_size$ **è definito then**
    $I \leftarrow$ Campione casuale di indici di dimensione $\min(sample\_size, |s|)$
  end if
  for ogni indice $i \in I$ **do**
    $neighbor \leftarrow best$
    Modifica $neighbor[i] \leftarrow 1 - neighbor[i]$
    $new\_fit \leftarrow$ Fitness($neighbor$, G, w, $maxCost$, E)
    if $new\_fit < best\_fit$ **then**
      $best \leftarrow neighbor$, $best\_fit \leftarrow new\_fit$
      $improved \leftarrow true$
      break Prima soluzione migliorante trovata
    end if
  end for
end while
return $best$

2.4 Motivazioni dell'Approccio

L'approccio HGA è particolarmente adatto al WFVS per le seguenti ragioni:

- La selezione a torneo garantisce una buona diversità mantenendo al contempo la pressione evolutiva verso soluzioni migliori.
- Il crossover a punto unico favorisce lo scambio di informazioni tra soluzioni, combinando tratti efficaci.
- La mutazione, applicata a ogni gene, permette una rapida esplorazione dello spazio delle soluzioni, evitando il rischio di stagnazione.
- La funzione di riparazione, insieme alla ricerca locale basata su hill climbing con first improvement, assicura che le soluzioni siano valide (grafo aciclico) e continuamente migliorate.

Questo approccio ibrido combina l'esplorazione globale (attraverso operatori evolutivi) con l'ottimizzazione locale, rendendolo particolarmente efficace per un problema complesso come il WFVS.

3 Risultati

L'algoritmo è stato testato su tutte le istanze fornite, sia di tipo Grid che di tipo Rand. Per ciascuna istanza sono state eseguite 10 iterazioni. Durante l'esecuzione, alcuni parametri evolutivi, quali il tasso di mutazione (0,065), il tasso di crossover (0,8) e la dimensione del sottoinsieme per la local search (0,5), sono rimasti invariati, poiché diversi esperimenti hanno evidenziato che tali valori garantiscono risultati e prestazioni ottimali. Gli altri parametri, come la dimensione della popolazione, il numero di generazioni e la dimensione del torneo, sono stati variabili e ottimizzati in base alla grandezza e alla densità del grafo. I risultati includono curve di convergenza, grafici dei costi finali per iterazione e una tabella riassuntiva.

3.1 Visualizzazione dei Risultati

In allegato a questa relazione sono presenti i plot dei risultati ottenuti per tutte le sei tipologie di problemi forniti. Di seguito vengono presentate due immagini esemplificative, organizzate come segue:

Figura 2 che mostra:

- Il grafo iniziale e il grafo finale (dove vengono rimossi i vertici indicati dalla soluzione migliore);
- Il grafico di convergenza, ottenuto dalla concatenazione delle curve di costo di tutte le iterazioni;
- Il grafico dei costi finali per iterazione.

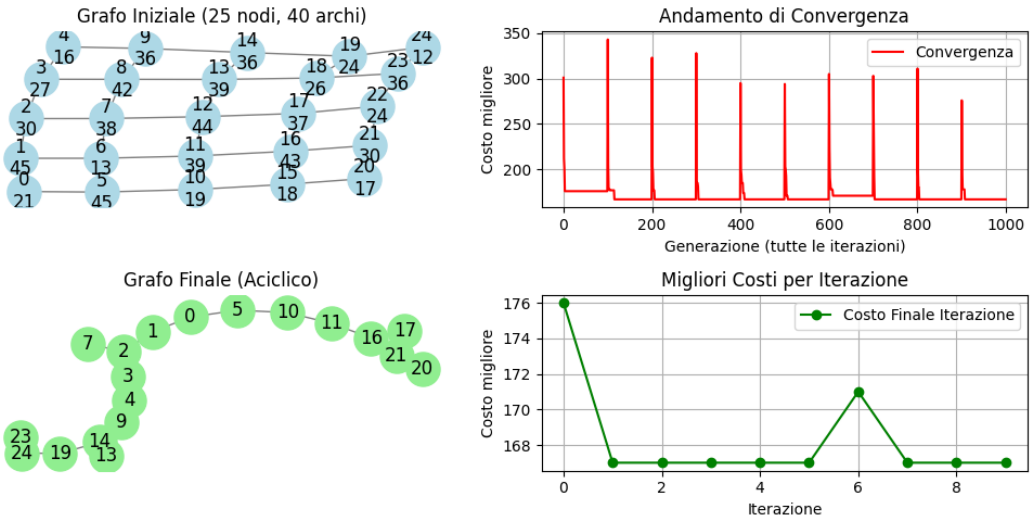

Figure 2: Visualizzazione dei grafici: grafo iniziale, grafo finale, convergenza e costi finali per iterazione.

Figura 3 mostra una tabella riassuntiva dei risultati per cinque istanze della stessa categoria (es. Grid), riportando costo minimo, costo medio, deviazione standard, tempo medio e valutazioni.

Risultati istanza 'Grid_25_40'

seed	min_costo	media_costi	dev_std	evaluations	tempo
107	198	198.80	0.98	41939.2	1.32 s
115	211	211.20	0.60	41863.6	1.22 s
83	226	227.80	2.27	42738.4	1.36 s
91	197	197.20	0.60	42155.9	1.80 s
99	167	168.30	2.83	42153.4	1.42 s

Risultati dell' algoritmo HGA eseguito sull'istanza 'Grid_25_40'.
Media costo tra i seed: 199.8
L'istanza migliore ha costo 167 con std 2.830194339616981
Parametri usati:
pop_size = 50
generations = 100
k (tournament) = 4
mutation_rate = 0.065
crossover_rate = 0.8
local_search_sample_size = 5

Figure 3: Tabella riassuntiva dei risultati.

Figura 4 illustra il plot dello *andamento di convergenza* zoommato per una singola iterazione di una delle istanze. Questo grafico permette di osservare in dettaglio il comportamento dell'algoritmo durante una singola esecuzione, evidenziando i miglioramenti locali ottenuti.

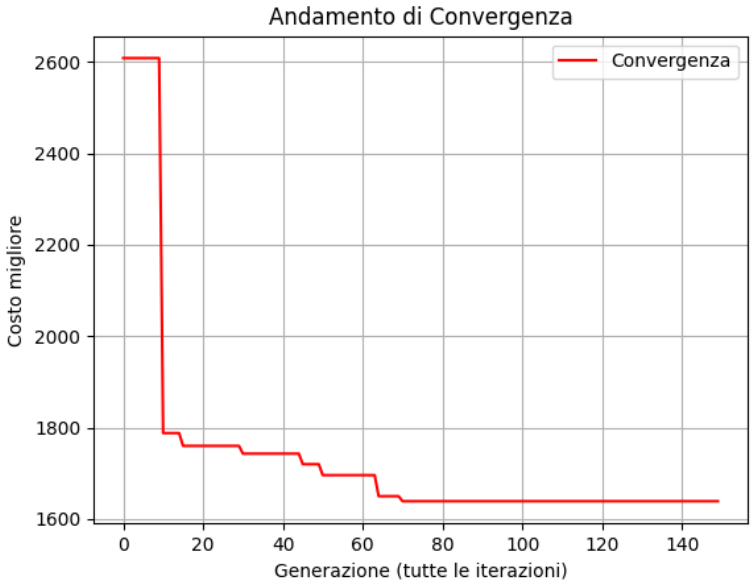

Figure 4: Andamento di convergenza zoommato per una singola iterazione.

3.2 Tabella Riassuntiva dei Risultati Medi e dei parametri usati

Di seguito una tabella che riassume, per 6 tipi di istanze, la media dei costi, il numero medio di valutazioni e il tempo medio:

Tipo	PS	G	K
Grid 25_40	50	100	4
Grid 49_84	50	100	4
Grid 81_144	100	150	5
Rand 100_841	100	150	5
Rand 100_3069	250	500	6
Rand 200_3184	300	400	8

Table 1: Parametri usati in ogni istanza: PS = dimensione della popolazione; G = numero di generazioni; K = dimensione del toreno. Altri parametri usati sono: il tasso di mutazione (0.065); tasso di crossover (0.8); la dimensione del sottoinsieme per la local search (5).

Tipo	Costo Medio	Valutazioni Medie	Tempo Medio (s)
Grid 25_40	199.8	42169.6	1.42 s
Grid 49_84	253.2	46065.9	3.19 s
Grid 81_144	1177.2	149182.66	16.54 s
Rand 100_841	1759.2	141198.76	22.35 s
Rand 100_3069	1134.4	948675.28	141.16 s
Rand 200_3184	5263.0	1150266.68	465.12 s

Table 2: Risultati medi trovati per i diversi tipi di istanze. Costo Medio: Media dei costi minimi trovati nelle 5 istanze di un ogni tipo; Valutazioni Medie: numero di chiamate medio della funzione di valutazione(fitness) che viene eseguito durante un'iterazione dell'algoritmo HGA su una istanze delle 5; Tempo Medio: Tempo medio (espresso in secondi) che rappresenta la durata di un'iterazione dell'algoritmo HGA su una instanza delle 5.

4 Conclusioni e Prospettive Future

L'algoritmo ibrido genetico implementato ha dimostrato efficacia nel risolvere il problema WFVS, ottenendo soluzioni di alta qualità in tempi ragionevoli. In particolare:

- La combinazione di operatori evolutivi e la ricerca locale ha accelerato la convergenza .
- La funzione di riparazione garantisce che le soluzioni siano sempre valide (grafo aciclico).
- I risultati sperimentali mostrano stabilità e performance soddisfacenti.

Nonostante i risultati ottenuti con l'algoritmo HGA, sono possibili ulteriori ottimizzazioni che potrebbero portare a performance ancora migliori. Tra le possibili direzioni di ricerca si suggerisce:

- **Ricerca Locale:**
  - Sperimentare con altri metodi di ricerca locale, quali il la Tabu Search, per migliorare l'abilità dell'algoritmo di uscire dai minimi locali o provare con un hill climbing con best improvement che non si ferma quando vede un primo miglioramento ma che continua la ricerca.
- **Parametrizzazione:**
  - Aumentare o adattare dinamicamente la dimensione della popolazione e il numero di generazioni in funzione della complessità e della densità del grafo.
- **Selezione dei Genitori:**
  - Sperimentare altre modalità di selezione, come la selezione a roulette o strategie elitiste, per mantenere una maggiore diversità e garantire un migliore mix delle soluzioni.
- **Crossover:**
  - Valutare alternative al crossover a un punto, come il crossover uniforme o multi-punto, per favorire uno scambio più efficace delle informazioni tra i genitori.

Queste possibili modifiche potrebbero non solo migliorare la qualità delle soluzioni ottenute, ma anche ridurre i tempi di esecuzione e aumentare la robustezza dell'algoritmo in presenza di istanze particolarmente complesse.

5 Bibliografia

I riferimenti utilizzati per questo progetto sono stati principalmente le slide del corso, unitamente a fonti online autorevoli.

References

[1] Slide del corso.

[2] "Genetic Algorithms", disponibile su: https://www.sciencedirect.com/topics/computer-science/genetic-algoritahm.

[3] Wikipedia, "Hill climbing", disponibile all'indirizzo: https://en.wikipedia.org/wiki/Hill_climbing (Accesso: marzo 2025).