

Lezione di Informatica Teorica

Traveling Salesperson Problem (TSP)

Appunti da Trascrizione Automatica

30 giugno 2025

Indice

1	Richiami: Macchine Oracolo e Gerarchia Polinomiale	2
2	Traveling Salesperson Problem (TSP)	2
2.1	Risoluzione di FTSP usando un Oracolo per TSP	2
2.1.1	Prima Soluzione (Naive)	3
2.1.2	Seconda Soluzione (Ricerca Binaria)	3
2.2	È FTSP in FP?	3
3	TSP è NP-Completo	4
3.1	Membership: $TSP \in NP$	4
3.2	Hardness: TSP è NP-arduo	4
3.2.1	Riduzione: $3SAT \leq_p DHC$	5
3.2.2	Riduzione: $DHC \leq_p HC$	7
3.2.3	Riduzione: $HC \leq_p TSP$	8
3.3	Conclusione della NP-Completezza di TSP	9

1 Richiami: Macchine Oracolo e Gerarchia Polinomiale

Riprendiamo il concetto di macchine oracolo e gerarchia polinomiale, già introdotto per problemi come il Vertex Cover.

Definizione 1 (Funzioni in FP^{NP}). FP^{NP} è l'insieme delle funzioni calcolabili da trasduttori deterministici in tempo polinomiale che hanno accesso a un oracolo in NP.

Osservazione 1. Un *oracolo* è una subroutine che risponde a interrogazioni sull'appartenenza di stringhe a un linguaggio (es. un linguaggio in NP) in un singolo passo di calcolo. Un oracolo può essere molto potente (es. oracolo NP, NEXP).

Le macchine oracolo permettono di definire gerarchie di complessità, come la Gerarchia Polinomiale (PH).

Definizione 2 (Σ_k^P). Σ_k^P è il k -esimo livello della Gerarchia Polinomiale. In particolare, Σ_2^P è la classe dei linguaggi decidibili da macchine in NP con accesso a un oracolo in NP.

Osservazione 2. I problemi Σ_k^P -completi sono intrinsecamente più difficili dei problemi NP-completi. Un problema Σ_2^P -completo, ad esempio, richiede algoritmi con due livelli di "backtracking" annidati, suggerendo un tempo di esecuzione doppiamente esponenziale su macchine deterministiche (se $P \neq NP$). Conoscere la classe di complessità di un problema aiuta a progettare algoritmi appropriati e a capire i limiti computazionali.

2 Traveling Salesperson Problem (TSP)

Vediamo il Traveling Salesperson Problem (TSP) sia nella sua versione computazionale (funzionale) che decisionale.

Definizione 3 (Grafo Pesato). Un *grafo pesato* è una tupla $G = (V, E, \lambda)$, dove V è l'insieme dei vertici, E è l'insieme degli archi (orientati o meno), e $\lambda : E \rightarrow \mathbb{N}^+$ è una funzione di pesatura che associa un peso (intero positivo) a ciascun arco.

Definizione 4 (Ciclo Hamiltoniano). Un *ciclo Hamiltoniano* in un grafo $G = (V, E)$ è un percorso che parte da un vertice, visita tutti gli altri vertici esattamente una volta, e termina nel vertice di partenza. Il peso di un percorso o ciclo è la somma dei pesi degli archi che lo compongono.

Definizione 5 (FTSP - Functional Traveling Salesperson Problem). Dato un grafo $G = (V, E, \lambda)$ non orientato e pesato, calcolare il peso del ciclo Hamiltoniano di peso minimo. Formalmente, $FTSP(G) = \min\{\lambda(\pi) \mid \pi \text{ è un ciclo Hamiltoniano di } G\}$.

Definizione 6 (TSP - Decision Traveling Salesperson Problem). Dato un grafo $G = (V, E, \lambda)$ non orientato e pesato, e un intero K , decidere se esiste un ciclo Hamiltoniano in G di peso al più K . Le istanze di TSP sono coppie (G, K) .

2.1 Risoluzione di FTSP usando un Oracolo per TSP

Supponiamo di avere un oracolo per TSP. Vogliamo risolvere FTSP usando chiamate a questo oracolo.

2.1.1 Prima Soluzione (Naive)

1. Inizializza $K = 1$. 2. Chiedi all'oracolo: "Esiste un ciclo Hamiltoniano di peso al più K in G ?" 3. Se l'oracolo risponde "Sì", allora K è il peso minimo. Termina. 4. Se l'oracolo risponde "No", incrementa K e ripeti dal passo 2. 5. Se G non ammette alcun ciclo Hamiltoniano, l'oracolo risponderà sempre "No". Per garantire la terminazione, il valore massimo di K da testare può essere la somma di tutti i pesi degli archi in G (chiamiamo questo valore S_{total}). Se l'oracolo risponde "No" anche per S_{total} , allora non esiste un ciclo Hamiltoniano.

Osservazione 3. Questa procedura non colloca FTSP in FP^{NP} . Il numero di chiamate all'oracolo è proporzionale a S_{total} . Poiché S_{total} può essere esponenziale nella dimensione dell'input (se i pesi sono grandi e rappresentati in binario), il trasduttore impiega un tempo esponenziale per effettuare le chiamate. La complessità è una funzione della taglia dell'input, non del suo valore numerico.

2.1.2 Seconda Soluzione (Ricerca Binaria)

Questa procedura è quella corretta per collocare FTSP in FP^{NP} .

1. Calcola S_{total} , la somma di tutti i pesi degli archi in G . 2. Chiedi all'oracolo: "Esiste un ciclo Hamiltoniano di peso al più S_{total} in G ?" * Se l'oracolo risponde "No", allora G non ammette alcun ciclo Hamiltoniano. Termina. * Se l'oracolo risponde "Sì", allora esiste almeno un ciclo Hamiltoniano. Procedi con una ricerca binaria nell'intervallo $[0, S_{total}]$ per trovare il peso minimo. 3. Esegui una ricerca binaria sull'intervallo $[L, R]$, inizialmente $[0, S_{total}]$: * Sia $M = \lfloor (L + R)/2 \rfloor$. * Chiedi all'oracolo: "Esiste un ciclo Hamiltoniano di peso al più M in G ?" * Se l'oracolo risponde "Sì", allora il peso minimo è $\leq M$, quindi imposta $R = M$. * Se l'oracolo risponde "No", allora il peso minimo è $> M$, quindi imposta $L = M + 1$. * Ripeti finché $L = R$. Il valore finale di L (o R) è il peso minimo.

Teorema 1. $FTSP \in FP^{NP}$.

Dimostrazione. Il numero di chiamate all'oracolo effettuate dalla procedura di ricerca binaria è logaritmico rispetto alla dimensione del dominio di ricerca $[0, S_{total}]$. La dimensione del dominio è S_{total} , che può essere esponenziale nella taglia dell'input (numero di bit per rappresentare S_{total}). Tuttavia, $\log(S_{total})$ è polinomiale nella taglia della rappresentazione binaria di S_{total} (e quindi polinomiale nella taglia dell'input). Ad esempio, se $S_{total} \approx 2^N$ dove N è la taglia dell'input, allora $\log(S_{total}) \approx N$, che è polinomiale. Il trasduttore deterministico esegue un numero polinomiale di chiamate all'oracolo e le operazioni ausiliarie (calcolo S_{total} , calcolo M , aggiornamento L, R) sono polinomiali. Pertanto, FTSP appartiene a FP^{NP} . \square

2.2 È FTSP in FP?

Ci chiediamo se $FTSP \in FP$, ovvero se può essere risolto da un algoritmo deterministico in tempo polinomiale senza l'ausilio di un oracolo.

Proposizione 1. Se $FTSP \in FP$, allora $TSP \in P$.

Dimostrazione. Supponiamo che esista un algoritmo deterministico polinomiale per FTSP. Per risolvere un'istanza (G, K) di TSP: 1. Usa l'algoritmo per FTSP per calcolare il peso minimo di un ciclo Hamiltoniano in G , sia esso W_{min} . 2. Confronta W_{min} con K . Se $W_{min} \leq K$, la risposta è "Sì"; altrimenti, la risposta è "No". Poiché sia il calcolo di W_{min} che il confronto sono polinomiali, anche TSP sarebbe risolvibile in tempo polinomiale. \square

Dato che $P \neq NP$ è un problema aperto e si ritiene improbabile che $P = NP$, la proposizione precedente implica che è altamente improbabile che FTSP sia in FP, a meno che NP non collassi su P. Questo sposta il nostro focus sulla complessità di TSP.

3 TSP è NP-Completo

Dimostriamo ora che TSP è NP-completo. Per fare ciò, dobbiamo mostrare due cose: 1. $TSP \in NP$ (membership). 2. TSP è NP-arduo (hardness).

3.1 Membership: $TSP \in NP$

Teorema 2. $TSP \in NP$.

Dimostrazione. Per mostrare che TSP è in NP, dobbiamo esibire un certificato di dimensione polinomiale che possa essere verificato in tempo polinomiale da una macchina deterministica. Sia (G, K) un'istanza di TSP.

- **Guess (Certificato):** La macchina non deterministica "indovina" una sequenza di vertici $v_1, v_2, \dots, v_{|V|}, v_1$ che rappresenta un potenziale ciclo Hamiltoniano. Questa sequenza ha lunghezza $|V| + 1$, quindi è polinomiale nella dimensione dell'input.
- **Check (Verifica):** Una macchina deterministica può verificare in tempo polinomiale che la sequenza:
 1. Formi un ciclo: $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_{|V|} \rightarrow v_1$. Questo significa controllare che per ogni $i \in \{1, \dots, |V| - 1\}$, esista un arco (v_i, v_{i+1}) e un arco $(v_{|V|}, v_1)$ in E .
 2. Visiti tutti i vertici: Controllare che ogni vertice in V appaia esattamente una volta nella sequenza $v_1, \dots, v_{|V|}$.
 3. Abbia peso al più K : Calcolare la somma dei pesi degli archi lungo il ciclo $\sum_{i=1}^{|V|-1} \lambda(v_i, v_{i+1}) + \lambda(v_{|V|}, v_1)$ e verificare che sia $\leq K$.

Tutti questi passaggi possono essere eseguiti in tempo polinomiale rispetto a $|V|$ e $|E|$. Pertanto, TSP è in NP.

□

3.2 Hardness: TSP è NP-arduo

Per dimostrare che TSP è NP-arduo, mostreremo una catena di riduzioni polinomiali da un problema noto NP-completo (3SAT) a TSP:

$$3SAT \leq_P DHC \leq_P HC \leq_P TSP$$

dove:

- **DHC (Directed Hamiltonian Cycle):** Dato un grafo orientato G , decidere se G ammette un ciclo Hamiltoniano.
- **HC (Undirected Hamiltonian Cycle):** Dato un grafo non orientato G , decidere se G ammette un ciclo Hamiltoniano.

3.2.1 Riduzione: $3SAT \leq_P DHC$

Questa è la riduzione più complessa. Dobbiamo trasformare una formula booleana in 3CNF in un grafo orientato G tale che G ammetta un ciclo Hamiltoniano se e solo se la formula è soddisfacibile.

Sia $\Phi = C_1 \wedge C_2 \wedge \dots \wedge C_m$ una formula 3CNF con m clausole e n variabili $X = \{x_1, x_2, \dots, x_n\}$. Assumiamo che nessuna clausola contenga un letterale e la sua negazione (es. $x \vee \neg x$).

Costruzione del Grafo G_Φ : Il grafo G_Φ sarà costruito in modo da codificare sia gli assegnamenti di verità delle variabili che la soddisfazione delle clausole.

- **Sezioni Variabili:** Per ogni variabile $x_i \in X$, creiamo una "catena" di nodi composta da m coppie di nodi, una per ogni clausola. Ogni coppia è $(x'_{i,j}, x''_{i,j})$ per $j = 1, \dots, m$. Aggiungiamo nodi intermedi ausiliari (piccoli pallini) tra le coppie e nodi diamante all'inizio e alla fine di ciascuna catena.

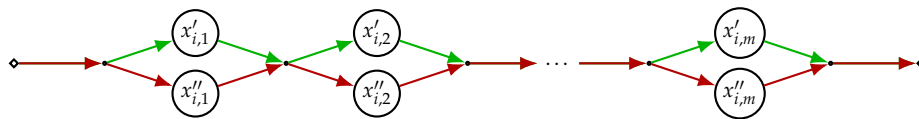


Figura 1: Struttura della Catena per la Variabile x_i

- **Struttura Generale del Grafo:** Connettiamo le n sezioni di variabili in serie. Ci sarà un nodo di inizio globale S e un nodo di fine globale T .

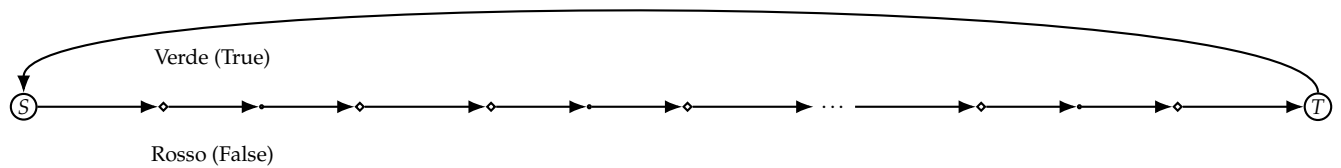


Figura 2: Struttura Generale del Grafo G_Φ

In ogni sezione di variabile (es. tra `start_x1` e `end_x1`), un ciclo Hamiltoniano dovrà scegliere di percorrere o tutti gli archi "verdi" (associati a $x_i = \text{TRUE}$) o tutti gli archi "rossi" (associati a $x_i = \text{FALSE}$) per passare tutti i nodi interni di quella sezione. Questo crea 2^n percorsi distinti da S a T , uno per ogni possibile assegnamento di verità.

- **Nodi Clausola:** Per ogni clausola C_j , aggiungiamo un nodo c_j . Questi nodi devono essere visitati da qualsiasi ciclo Hamiltoniano. I nodi clausola sono agganciati alle catene delle variabili.
 - Se il letterale x_i appare positivamente in C_j : Aggiungi archi orientati da $x'_{i,j}$ a c_j e da c_j a $x''_{i,j}$. Questo crea un "detour" dalla catena "verde" (True) di x_i .
 - Se il letterale $\neg x_i$ appare negativamente in C_j : Aggiungi archi orientati da $x''_{i,j}$ a c_j e da c_j a $x'_{i,j}$. Questo crea un "detour" dalla catena "rossa" (False) di x_i .

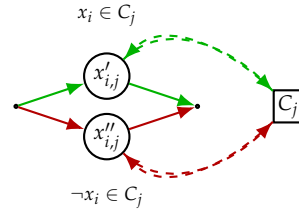


Figura 3: Agganci tra Nodi Variabile e Nodi Clausola

Dimostrazione:

Teorema 3. Φ è soddisfacibile se e solo se G_Φ ammette un ciclo Hamiltoniano.

Dimostrazione. (\Rightarrow) **Se Φ è soddisfacibile:** Sia σ un assegnamento di verità che soddisfa Φ . Costruiamo un ciclo Hamiltoniano π in G_Φ come segue: 1. Inizia da S . 2. Per ogni variabile x_i , se $\sigma(x_i) = \text{TRUE}$, attraversa la catena della variabile x_i (ovvero, da start_xi a end_xi) seguendo gli archi "verdi" (da $x'_{i,j}$ a $x'_{i,j+1}$ tramite nodi ausiliari). Se $\sigma(x_i) = \text{FALSE}$, segui gli archi "rossi" (da $x''_{i,j}$ a $x''_{i,j+1}$). 3. Quando si arriva a un nodo $x'_{i,j}$ (se stiamo sul percorso verde) o $x''_{i,j}$ (se stiamo sul percorso rosso), controlla se la clausola C_j è soddisfatta da questo assegnamento di x_i . Poiché σ è un assegnamento che soddisfa Φ , ogni clausola C_j è soddisfatta da almeno un letterale. * Se $x_i \in C_j$ e $\sigma(x_i) = \text{TRUE}$, allora mentre si passa per $x'_{i,j}$, possiamo deviare verso c_j (arco da $x'_{i,j}$ a c_j) e poi tornare a $x''_{i,j}$ (arco da c_j a $x''_{i,j}$), e poi proseguire il percorso principale. * Se $\neg x_i \in C_j$ e $\sigma(x_i) = \text{FALSE}$, allora mentre si passa per $x''_{i,j}$, possiamo deviare verso c_j (arco da $x''_{i,j}$ a c_j) e poi tornare a $x'_{i,j}$ (arco da c_j a $x'_{i,j}$), e poi proseguire il percorso principale. 4. Questo meccanismo assicura che tutti i nodi clausola c_j vengano visitati (dato che ogni C_j è soddisfatta da almeno un letterale x_i o $\neg x_i$ il cui percorso si sta già attraversando). I nodi intermedi (pallini) e i nodi $x'_{i,j}, x''_{i,j}$ vengono tutti visitati. 5. Infine, da T si torna a S chiudendo il ciclo. Questo percorso π visita tutti i nodi esattamente una volta (inclusi S, T , tutti i nodi delle catene x_i , e tutti i nodi clausola c_j) e forma un ciclo. Quindi G_Φ ammette un ciclo Hamiltoniano.

(\Leftarrow) **Se G_Φ ammette un ciclo Hamiltoniano:** Sia π un ciclo Hamiltoniano in G_Φ . Dobbiamo dimostrare che da π si può derivare un assegnamento di verità σ che soddisfa Φ .

Lemma 1. In un ciclo Hamiltoniano π di G_Φ :

- Il percorso attraverso ciascuna sezione di variabile x_i (da start_xi a end_xi) deve seguire esclusivamente gli archi "verdi" (da $x'_{i,j}$ a $x'_{i,j+1}$) o esclusivamente gli archi "rossi" (da $x''_{i,j}$ a $x''_{i,j+1}$). Non è possibile alternare tra percorsi verdi e rossi all'interno della stessa sezione, altrimenti si lascerebbero nodi non visitati o si violerebbe la proprietà di ciclo.
- Se π devia in un nodo clausola c_j da $x'_{i,j}$, deve tornare a $x''_{i,j}$. Simmetricamente, se devia da $x''_{i,j}$, deve tornare a $x'_{i,j}$. Non sono possibili "salti" a nodi di altre sezioni di variabili o altre clausole.

Dimostrazione del Lemma (Sketch). La prima proprietà è assicurata dalla struttura delle catene: se si alterna, si lascerebbe un "lato" della catena non visitato (es. tutti i nodi $x''_{i,k}$ se si sceglie il percorso verde e poi si cambia). Per la seconda proprietà, consideriamo il nodo c_j . Ha solo due archi in ingresso e due in uscita, provenienti dalle sezioni variabili. Se si entra in c_j da $x'_{i,j}$, l'unica via per visitare $x''_{i,j}$ (che deve essere visitato se non lo è già) è tornare a $x''_{i,j}$ direttamente da c_j . Un salto verso

un altro $x_{k,l}$ impedirebbe di visitare correttamente i nodi rimanenti della catena di x_i o violerebbe il ciclo Hamiltoniano. \square

Grazie al Lemma, un ciclo Hamiltoniano π ha una struttura ben definita: 1. Per ogni variabile x_i , π o attraversa tutti i nodi $x'_{i,j}$ (usando gli archi "verdi") o tutti i nodi $x''_{i,j}$ (usando gli archi "rossi"). Questo ci permette di definire un assegnamento σ : * Se π attraversa la sezione x_i sul percorso verde, poniamo $\sigma(x_i) = \text{TRUE}$. * Se π attraversa la sezione x_i sul percorso rosso, poniamo $\sigma(x_i) = \text{FALSE}$. 2. Poiché π è un ciclo Hamiltoniano, deve visitare tutti i nodi del grafo, inclusi tutti i nodi clausola c_j . 3. Per ogni nodo c_j visitato, π deve aver eseguito un detour da un nodo $x'_{i,j}$ o $x''_{i,j}$. * Se il detour è da $x'_{i,j}$ (percorso verde), significa che $\sigma(x_i) = \text{TRUE}$ e $x_i \in C_j$. Questo rende la clausola C_j vera. * Se il detour è da $x''_{i,j}$ (percorso rosso), significa che $\sigma(x_i) = \text{FALSE}$ e $\neg x_i \in C_j$. Questo rende la clausola C_j vera. Poiché tutti i nodi c_j sono visitati, tutte le clausole C_j devono essere soddisfatte dall'assegnamento σ . Quindi Φ è soddisfacibile. \square

Questo conclude la riduzione $3\text{SAT} \leq_P \text{DHC}$. La trasformazione è polinomiale perché il numero di nodi e archi nel grafo G_Φ è polinomiale rispetto a n e m (circa $O(nm)$ nodi e $O(nm)$ archi).

3.2.2 Riduzione: $\text{DHC} \leq_P \text{HC}$

Dobbiamo trasformare un grafo orientato $G = (V, E)$ in un grafo non orientato $H = (V', E')$ tale che G ammette un ciclo Hamiltoniano se e solo se H ammette un ciclo Hamiltoniano.

Costruzione del Grafo H : Per ogni vertice $v \in V$ in G , creiamo tre vertici in H : v_{in}, v_{mid}, v_{out} .

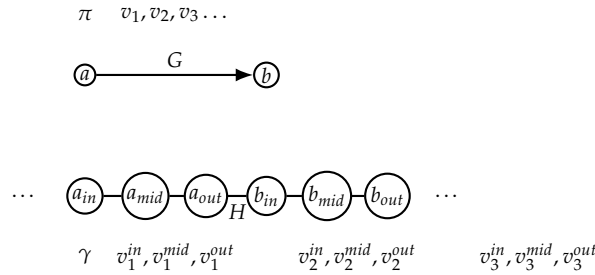


Figura 4: Struttura della Riduzione $\text{DHC} \leq_P \text{HC}$

- **Nodi in H :** $V' = \{v_{in}, v_{mid}, v_{out} \mid v \in V\}$. Quindi $|V'| = 3|V|$.
- **Archetti in H :**
 1. Per ogni $v \in V$, aggiungi archi non orientati (v_{in}, v_{mid}) e (v_{mid}, v_{out}) . Questi archi "forzano" l'attraversamento in sequenza.
 2. Per ogni arco orientato $(u, v) \in E$ in G , aggiungi un arco non orientato (u_{out}, v_{in}) in H .

Dimostrazione:

Teorema 4. G ammette un ciclo Hamiltoniano se e solo se H ammette un ciclo Hamiltoniano.

Dimostrazione. (\Rightarrow) **Se G ammette un ciclo Hamiltoniano:** Sia $\pi = v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k \rightarrow v_1$ un ciclo Hamiltoniano in G (dove $k = |V|$). Costruiamo un ciclo in H come segue: $\gamma = v_{1,in} \rightarrow v_{1,mid} \rightarrow v_{1,out} \rightarrow v_{2,in} \rightarrow v_{2,mid} \rightarrow v_{2,out} \rightarrow \dots \rightarrow v_{k,in} \rightarrow v_{k,mid} \rightarrow v_{k,out} \rightarrow v_{1,in}$. Questo percorso visita tutti i $3|V|$ nodi di H esattamente una volta:

- Le transizioni $v_{i,in} \rightarrow v_{i,mid} \rightarrow v_{i,out}$ sono garantite dalla costruzione degli archi all'interno di ogni tripletto.
- Le transizioni $v_{i,out} \rightarrow v_{i+1,in}$ (e $v_{k,out} \rightarrow v_{1,in}$ per chiudere il ciclo) sono garantite dall'esistenza degli archi (v_i, v_{i+1}) in G , che corrispondono agli archi $(v_{i,out}, v_{i+1,in})$ in H .

Pertanto, γ è un ciclo Hamiltoniano in H .

(\Leftarrow) **Se H ammette un ciclo Hamiltoniano:** Sia γ un ciclo Hamiltoniano in H . Tutti i nodi v_{mid} hanno solo due vicini: v_{in} e v_{out} . Pertanto, qualsiasi ciclo Hamiltoniano che passi per v_{mid} deve per forza percorrere gli archi (v_{in}, v_{mid}) e (v_{mid}, v_{out}) in sequenza (o viceversa). Questo implica che i nodi di ogni tripletto $(v_{in}, v_{mid}, v_{out})$ devono essere visitati consecutivamente in γ in una delle due direzioni. Quindi γ deve essere della forma: $\dots \rightarrow u_{out} \rightarrow v_{in} \rightarrow v_{mid} \rightarrow v_{out} \rightarrow w_{in} \rightarrow \dots$. Da questa sequenza, possiamo costruire un ciclo Hamiltoniano in G : $u \rightarrow v \rightarrow w \rightarrow \dots \rightarrow u$. Ogni volta che γ si sposta da u_{out} a v_{in} , ciò implica l'esistenza dell'arco (u_{out}, v_{in}) in H , che a sua volta significa che esisteva l'arco orientato (u, v) in G . Poiché γ visita tutti i nodi di H , ogni vertice di G (rappresentato dal suo tripletto di nodi in H) deve essere visitato. Poiché ogni tripletto è visitato esattamente una volta, ogni vertice in G è visitato esattamente una volta. Questo forma un ciclo Hamiltoniano in G . \square

Questa riduzione è polinomiale poiché $|V'| = 3|V|$ e il numero di archi in H è $2|V| + |E|$, il che è polinomiale in $|V|$ e $|E|$.

3.2.3 Riduzione: $HC \leq_P TSP$

Dobbiamo trasformare un grafo non orientato $G = (V, E)$ in un'istanza (H, K) di TSP.

Costruzione dell'Istanza (H, K) per TSP: 1. Il grafo H è uguale a G : $H = G$. Tutti i vertici e gli archi di G sono i vertici e gli archi di H . 2. Assegna un peso a tutti gli archi di H : Per ogni arco $e \in E$, imposta il peso $\lambda(e) = 1$. 3. Imposta il valore limite K : $K = |V|$ (il numero di vertici in G).

Dimostrazione:

Teorema 5. G ammette un ciclo Hamiltoniano se e solo se H ammette un ciclo Hamiltoniano di peso al più K .

Dimostrazione. (\Rightarrow) **Se G ammette un ciclo Hamiltoniano:** Sia π un ciclo Hamiltoniano in G . Per definizione, π visita tutti i $|V|$ vertici esattamente una volta e consiste di esattamente $|V|$ archi. Poiché tutti gli archi in H hanno peso 1, il peso del ciclo π in H è $1 \times |V| = |V|$. Dato che $K = |V|$, il ciclo π in H ha peso esattamente K , che è al più K . Pertanto, H ammette un ciclo Hamiltoniano di peso al più K .

(\Leftarrow) **Se H ammette un ciclo Hamiltoniano di peso al più K :** Sia γ un ciclo Hamiltoniano in H con peso $\lambda(\gamma) \leq K$. Per definizione di ciclo Hamiltoniano, γ visita tutti i $|V|$ vertici di H (e quindi di G) esattamente una volta e consiste di esattamente $|V|$ archi. Il peso di γ è la somma dei pesi di questi $|V|$ archi. Dato che tutti i pesi degli archi in H sono 1 (e sono positivi), l'unico modo per avere un peso $\leq K = |V|$ per un ciclo di $|V|$ archi è che ogni arco nel ciclo abbia peso esattamente 1. Questo significa che γ è un ciclo Hamiltoniano in G (che non ha pesi o ha pesi unitari). \square

Questa riduzione è chiaramente polinomiale poiché la trasformazione consiste solo nell'aggiungere pesi unitari e definire K , operazioni che richiedono tempo polinomiale.

3.3 Conclusione della NP-Completezza di TSP

Avendo dimostrato che:

- $TSP \in NP$.
- $3SAT \leq_P DHC \leq_P HC \leq_P TSP$.

E sapendo che 3SAT è NP-completo, per la transitività delle riduzioni polinomiali, concludiamo che TSP è NP-completo.

Teorema 6. *TSP è NP-completo.*

Ciò rafforza l'idea che FTSP è improbabile che sia in FP, a meno di una sorprendente risoluzione di P vs NP.