Ricerca Operativa

a.a. 2018-2019

SAPIENZA
UNIVERSITÀ DI ROMA

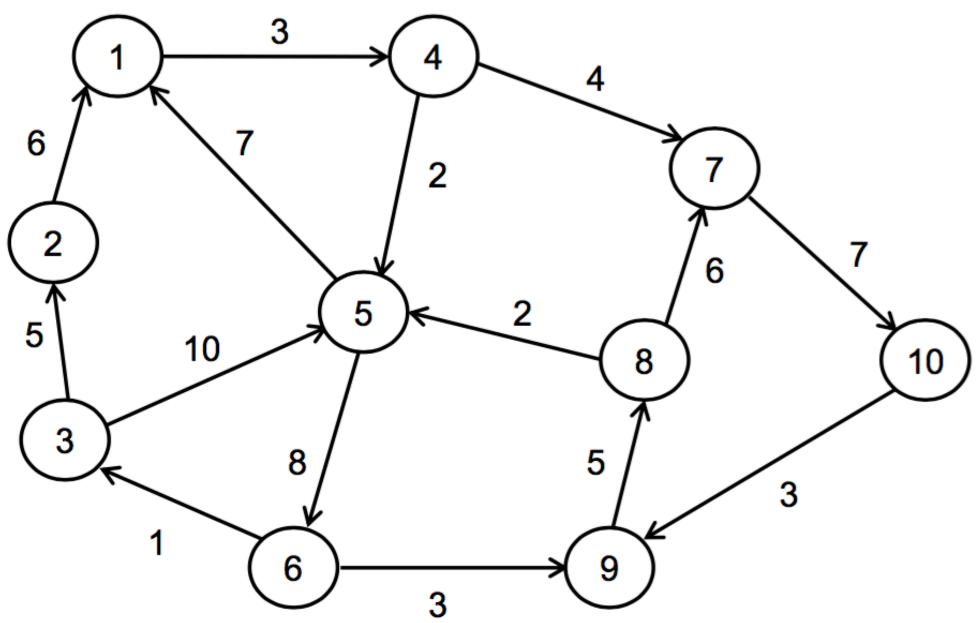
Dipartimento di Scienze Statistiche
Lavinia Amorosi
E-mail: lavinia.amorosi@uniroma1.it

Il Problema del Commesso Viaggiatore (Travel Salesman Problem)

Il problema del commesso viaggiatore (TSP)

Un problema di Ottimizzazione Combinatoria molto noto è quello del Commesso Viaggiatore o Travel Salesman Problem (TSP).

Il problema è definito su un grafo G = (N,A), con |N|=n e |A| = m, in cui ad ogni arco $(i,j) \in A$ è associato un costo o lunghezza $c_{ij} \geq 0$.

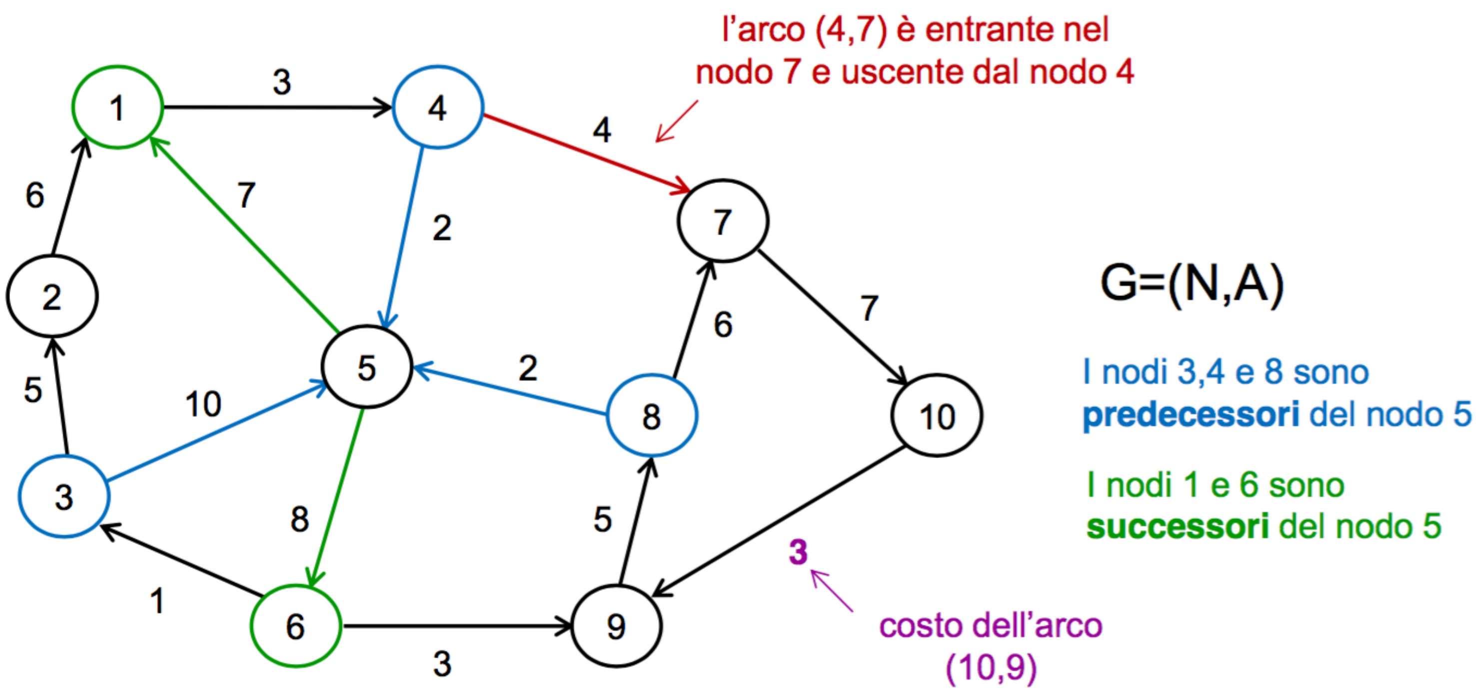


G=(N,A)

Il problema del commesso viaggiatore (TSP)

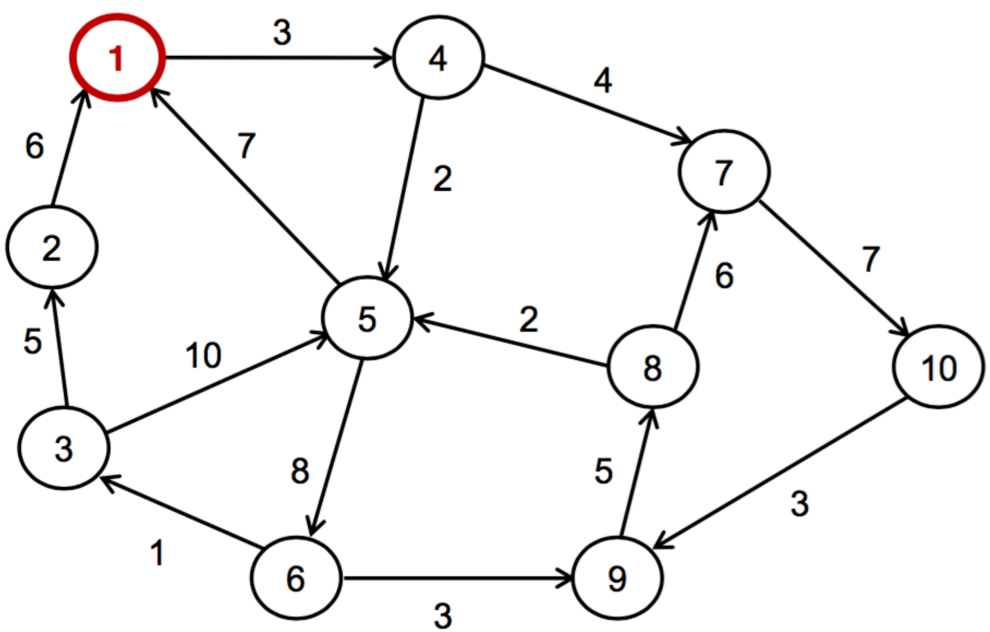
Un problema di Ottimizzazione Combinatoria molto noto è quello del Commesso Viaggiatore o Travel Salesman Problem (TSP).

Il problema è definito su un grafo G = (N,A), con |N|=n e |A| = m, in cui ad ogni arco $(i,j) \in A$ è associato un costo o lunghezza $c_{ij} \geq 0$.



Il problema del commesso viaggiatore (TSP)

Problema: Un commesso viaggiatore (Travel Salesman), partendo dalla città in cui vive, ogni giorno deve visitare n=10 città. Egli deve passare in ciascuna città esattamente una volta e alla fine del percorso deve ritornare nella città di partenza minimizzando il *costo totale degli spostamenti*.

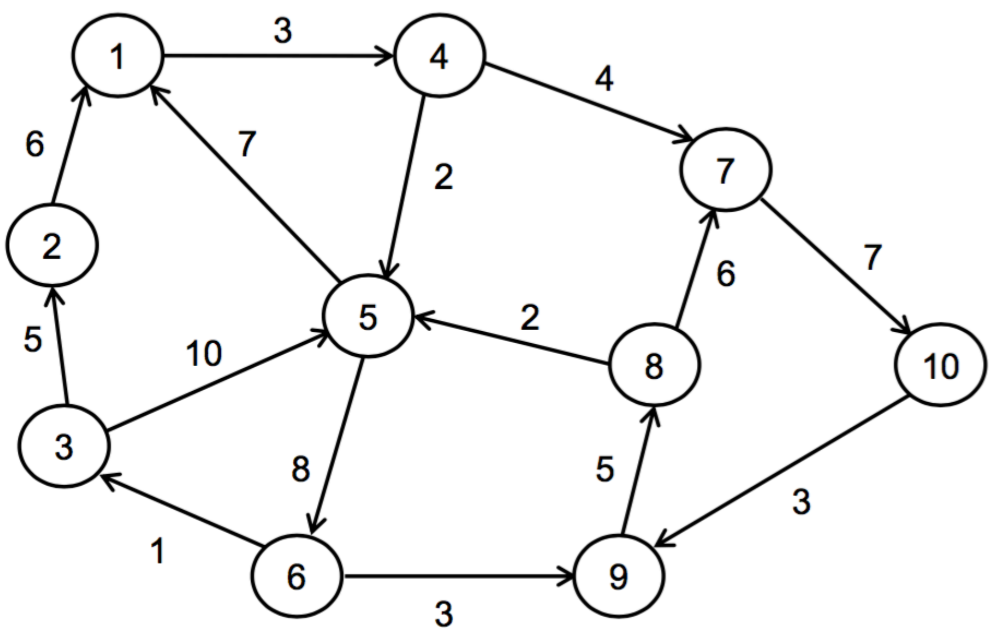


HP: G=(N,A) rappresenta le città (nodi in N) e i collegamenti tra città (archi in A).

Il problema del commesso viaggiatore (TSP)

TOUR

Fissato un nodo di G, ad esempio il nodo 1, un tour (o ciclo hamiltoniano) è un percorso (sequenza ordinata di nodi-archi) che inizia e termina nel nodo fissato (S.P.I.G. Il nodo 1) e che visita ogni nodo di G una e una sola volta.

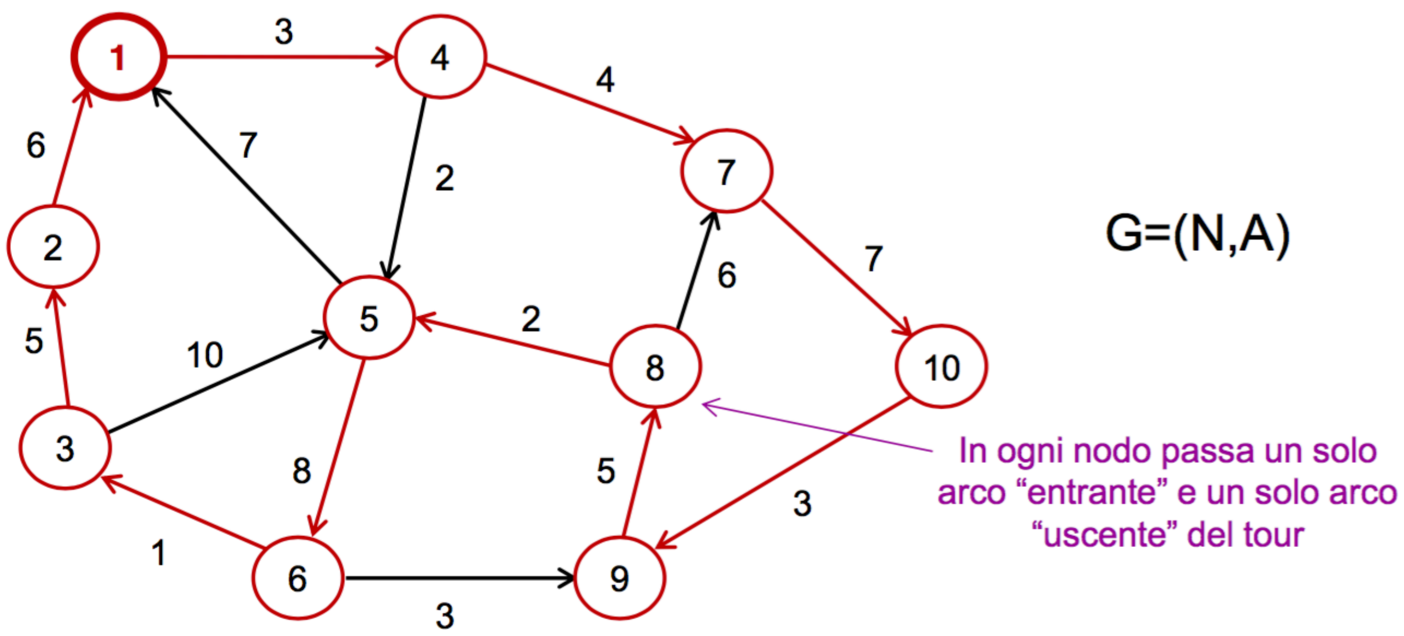


NOTA:
G=(N,A)
deve essere
'connesso'

Il problema del commesso viaggiatore (TSP)

TOUR

Fissato un nodo di G, ad esempio il nodo 1, un tour (o ciclo hamiltoniano) è un percorso (sequenza ordinata di nodi-archi) che inizia e termina nel nodo fissato (S.P.I.G. Il nodo 1) e che visita ogni nodo di G una e una sola volta.

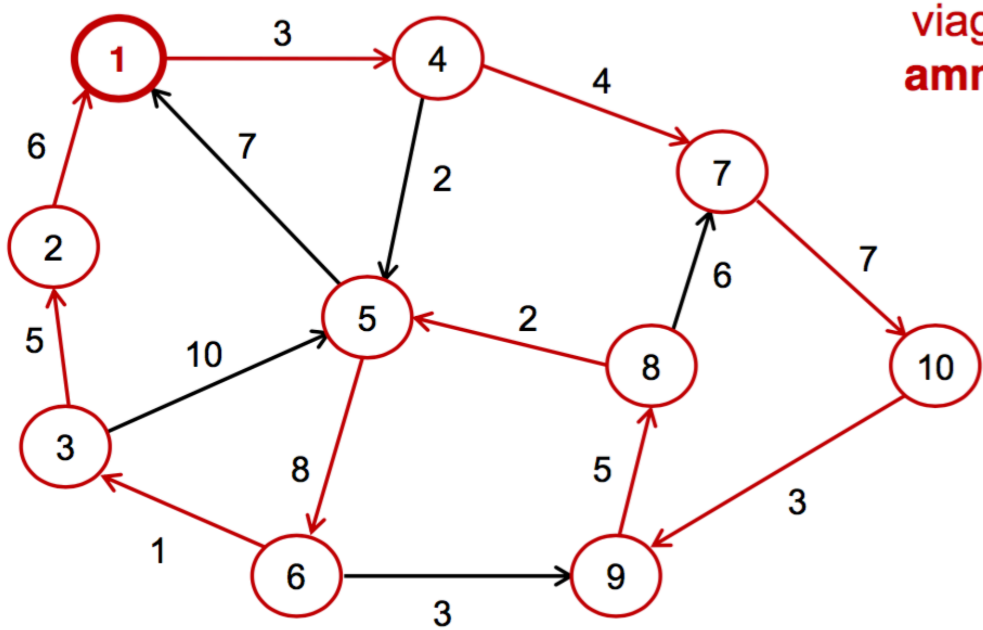


G=(N,A)

Il problema del commesso viaggiatore (TSP)

Problema: Un commesso viaggiatore (Travel Salesman), partendo dalla città in cui vive, ogni giorno deve visitare n=10 città. Egli deve passare in ciascuna città esattamente una volta e alla fine del percorso deve ritornare nella città di partenza minimizzando il *costo totale degli spostamenti*.

Nel problema del commesso viaggiatore le **soluzioni ammissibili** sono i **tour**.



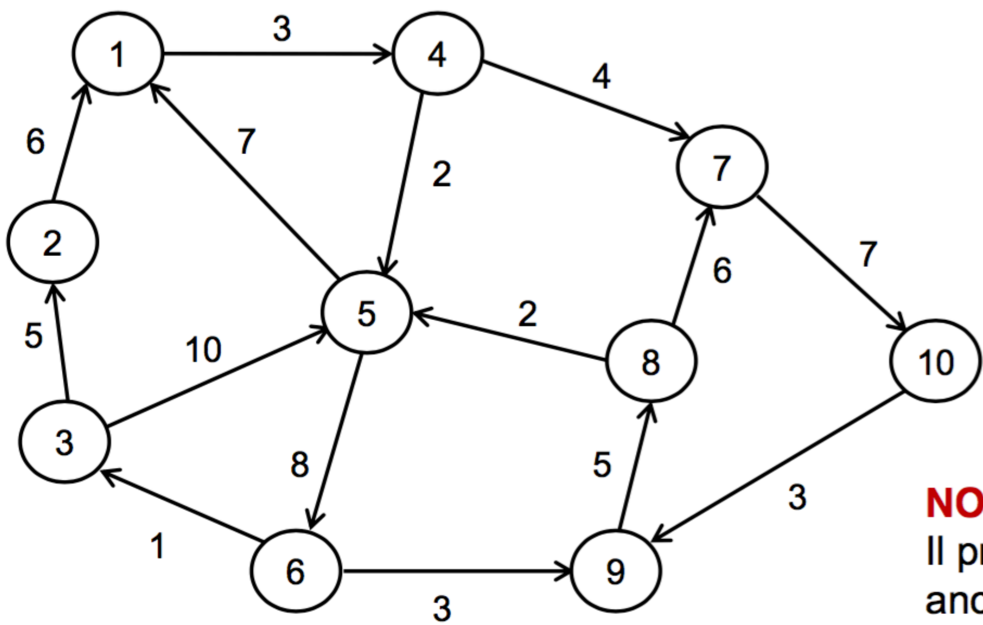
HP: G=(N,A) rappresenta le città (nodi in N) e i collegamenti tra città (archi in A).

Il problema del commesso viaggiatore (TSP)

Problema del TSP

Dato un grafo G=(N,A), con |N|=n e |A|=m, e con **costi non negativi associati agli archi**, individuare il tour di costo totale minimo.

Il **costo del tour** è dato dalla somma dei costi degli archi che sono stati inseriti nel ciclo.



G=(N,A)

NOTA

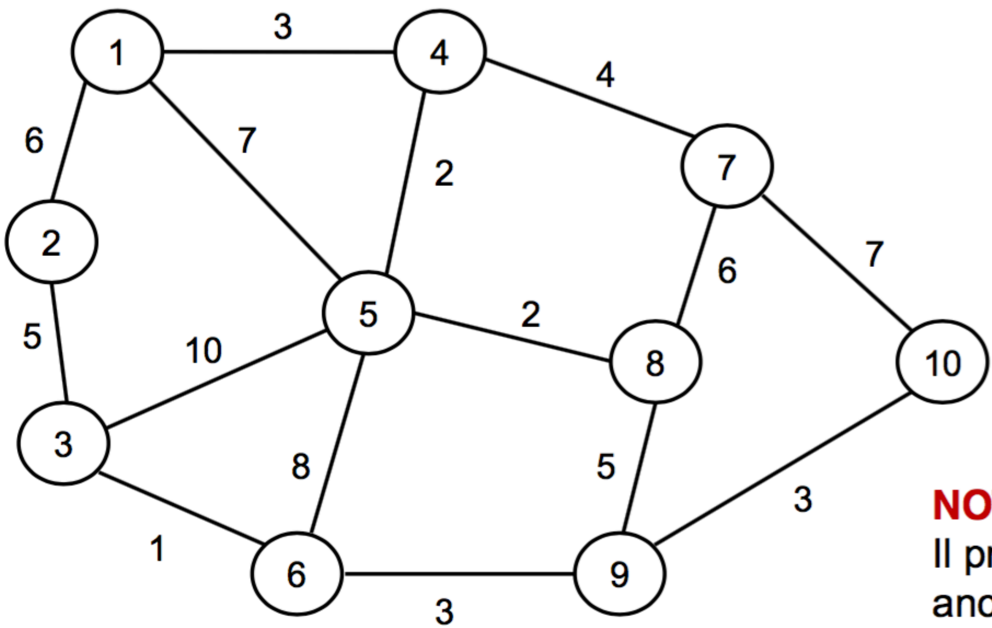
Il problema può essere formulato anche con riferimento a un grafo non orientato (TSP simmetrico).

Il problema del commesso viaggiatore (TSP)

Problema del TSP

Dato un grafo G=(N,A), con |N|=n e |A|=m, e con **costi non negativi associati agli archi**, individuare il tour di costo totale minimo.

Il **costo del tour** è dato dalla somma dei costi degli archi che sono stati inseriti nel ciclo.



G=(N,A)

NOTA

Il problema può essere formulato anche con riferimento a un grafo non orientato (TSP simmetrico).

Il problema del commesso viaggiatore (TSP)

Il TSP descrive il problema combinatorio della **ricerca su G di un tour di costo minimo** ed incorpora per questo l'essenza di tutti quei problemi di ottimizzazione discreta noti con il nome di **problemi di routing**.

Alcuni esempi sono:

- **Veichle Routing Problem** (consegne merci, pick-up passeggeri)
  Un insieme di veicoli, ciascuno con la sua capacità di carico, deve visitare un insieme di clienti. Il problema è assegnare clienti a veicoli e poi determinare **per ciascun veicolo il tour dei suoi clienti di costo minimo**.
- **Scheduling di lavori su una macchina** (ad es., operazioni di set up in un macchinario tra la lavorazione di un pezzo e del successivo)
  Un macchinario che vernicia sportelli di auto, **tra la lavorazione di un pezzo e quella del successivo**, deve eseguire alcune operazioni di ripristino o set up (come, ad es., pulire le bocchette per la verniciature, ricaricare i colori, riposizionare gli spruzzatori, ecc) e ciascuna di queste operazioni ha un costo (es.: tempo).
  Stabilite tramite un grafo G=(N,A) le **relazioni di precedenza** tra le operazioni di set up da eseguire, il problema consiste nell'individuare la sequenza di operazioni (città) che il macchinario (commesso viaggiatore) deve eseguire per effettuare il set up completo al minimo costo totale.

Il problema del commesso viaggiatore (TSP)

Il TSP descrive il problema combinatorio della **ricerca su G di un tour di costo minimo** ed incorpora per questo l'essenza di tutti quei problemi di ottimizzazione discreta noti con il nome di **problemi di routing**.

Per il problema del TSP esistono **molte formulazioni** (di Programmazione Lineare a Intera o Mista) diverse, anche a seconda che si tratti della versione su grafo orientato (TSP asimmetrico) o non orientato (TSP simmetrico).

Purtroppo il TSP è un problema **computazionalmente difficile** e pertanto le formulazioni possono essere utilizzate per risolverlo attraverso tecniche di tipo B&B, ma senza garanzie sui tempi di calcolo.

In alternativa al B&B (o a tecniche simili) sono stati studiati algoritmi di soluzione ad hoc per il TSP che ovviamente **non sono di natura esatta**:

Algoritmo di **approssimazione** di Christofides (1975).

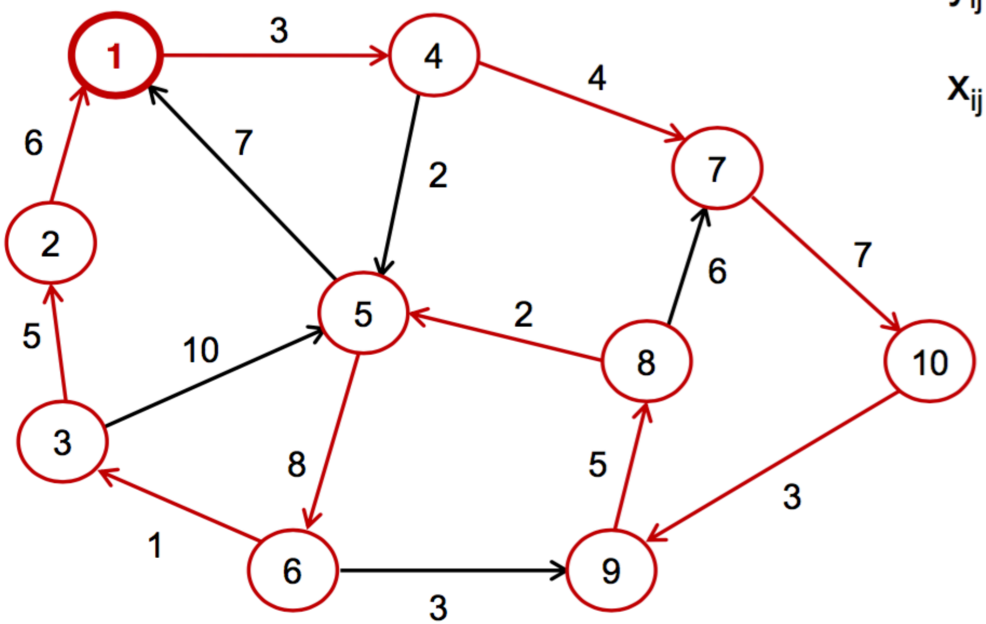
Algoritmo **euristico di Ricerca Locale** di Lin e Kernighan (1973).

TSP asimmetrico

Il problema del commesso viaggiatore (TSP)

TOUR

Fissato un nodo di G, ad esempio il nodo 1, un tour (o ciclo hamiltoniano) è un percorso (sequenza ordinata di nodi-archi) che inizia e termina nel nodo fissato (S.P.I.G. Il nodo 1) e che visita ogni nodo di G una e una sola volta.



Le variabili y_{ij} sono variabili indicatrici di quali archi appartengono al tour:

y_{ij} =1 se l'arco (i,j) è nel tour
y_{ij} =0 altrimenti

x_{ij} sono variabili di "flusso"

G=(N,A)

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i=1 \\ -1 & \text{se } i\neq 1 \end{cases} \qquad \forall i \in N$$

$$x_{ij} \le (n-1)\, y_{ij} \qquad \forall (i,j)\in A$$

$$x_{ij} \ge 0 \qquad \forall (i,j)\in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j)\in A$$

Le variabili y_{ij} sono variabili indicatrici di quali archi appartengono al tour:

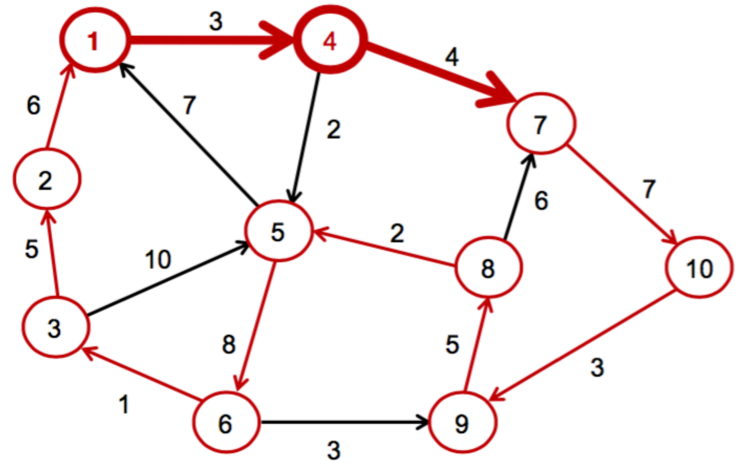
y_{ij} =1 se l'arco (i,j) è nel tour
y_{ij} =0 altrimenti

x_{ij} sono variabili di “flusso”

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$



Questi vincoli garantiscono che per ogni nodo visitato ci sia **esattamente** un arco del tour entrante e **esattamente** uno uscente (vincoli di assegnamento).

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i = 1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N$$

$$x_{ij} \le (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

$$x_{ij} \ge 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

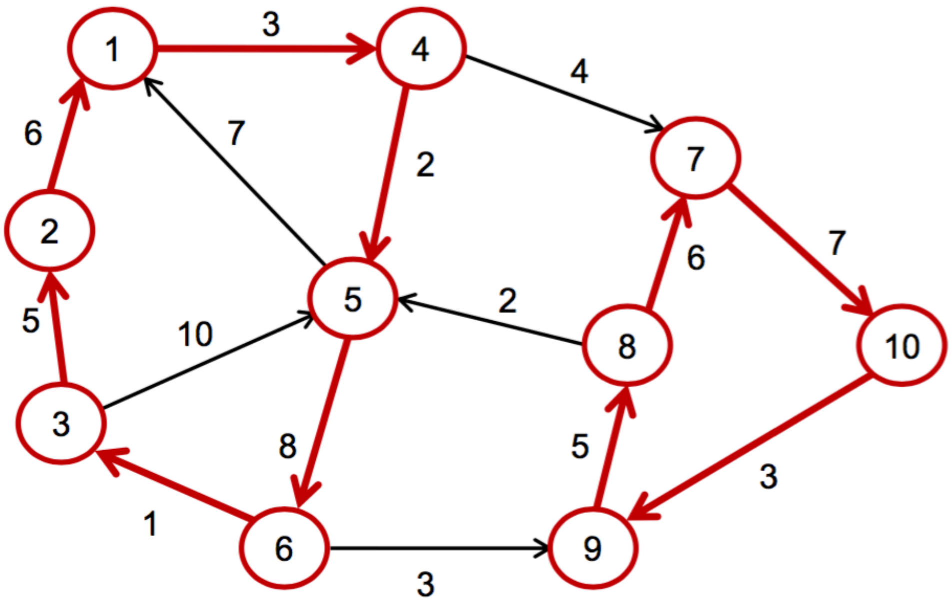
NOTA: Questi vincoli non sono sufficienti per caratterizzare un tour.

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$



La selezione degli archi rossi soddisfa i vincoli?
Corrisponde a un tour?

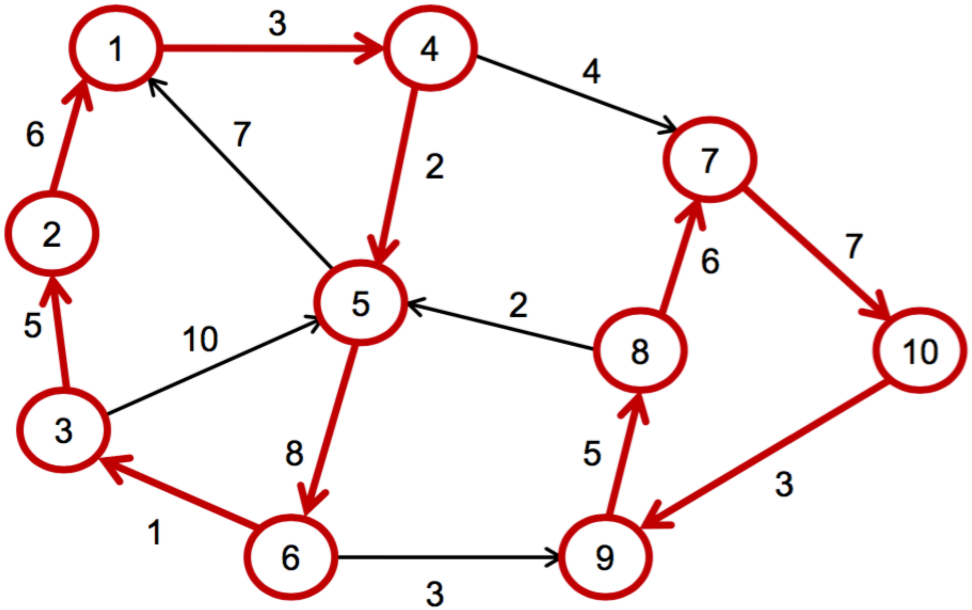
NOTA: Questi vincoli non sono sufficienti per caratterizzare un tour.

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$



Questa scelta di archi **non corrisponde** a **un** ciclo (tour) ma a **due** cicli disgiunti (subtour).

La selezione degli archi rossi soddisfa i vincoli?
Corrisponde a un tour?

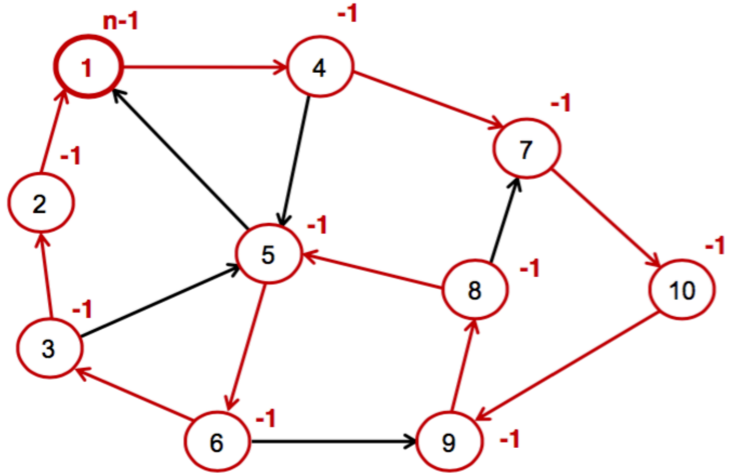
- ✓ per ogni nodo i un solo arco uscente e un solo arco entrante;
- ✓ ogni nodo è attraversato esattamente una volta.

NOTA: Questi vincoli non sono sufficienti per caratterizzare un tour.

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$



$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\boxed{\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i = 1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N}$$

S(i) insieme dei successori del nodo i

P(i) insieme dei predecessori del nodo i

$$x_{ij} \le (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

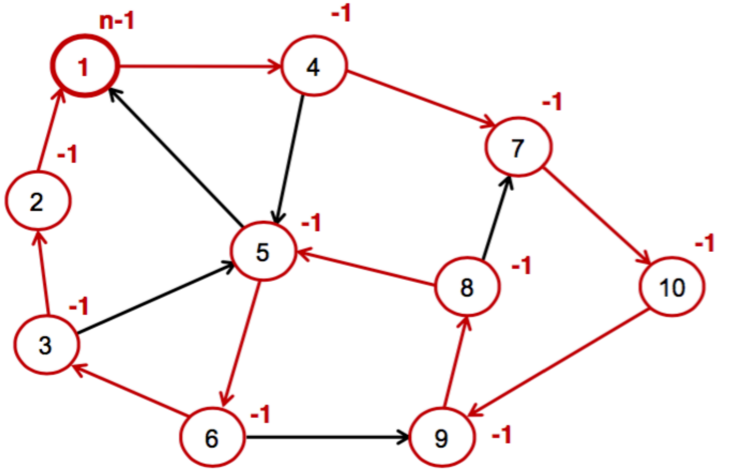
$$x_{ij} \ge 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$



$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

Questi vincoli garantiscono che ogni nodo sia effettivamente visitato attraverso un **unico** tour (vincoli di conservazione del flusso).

$$\boxed{\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i = 1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N}$$

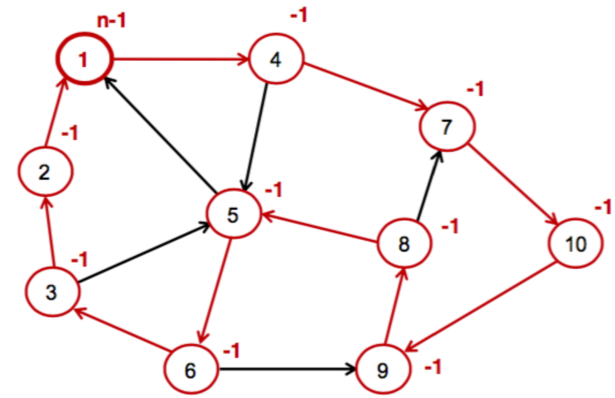
$$x_{ij} \leq (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico



$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

Il nodo 1 è considerato come una "sorgente" da cui si genera un flusso pari a n-1

$$\underbrace{\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi}} = \begin{cases} n-1 & \text{se } i=1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N$$

Ogni altro nodo è considerato come un "pozzo" in cui viene assorbita una unità di flusso

Afflusso netto del nodo i =
flusso uscente da i al netto del flusso entrante in i

$$x_{ij} \le (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

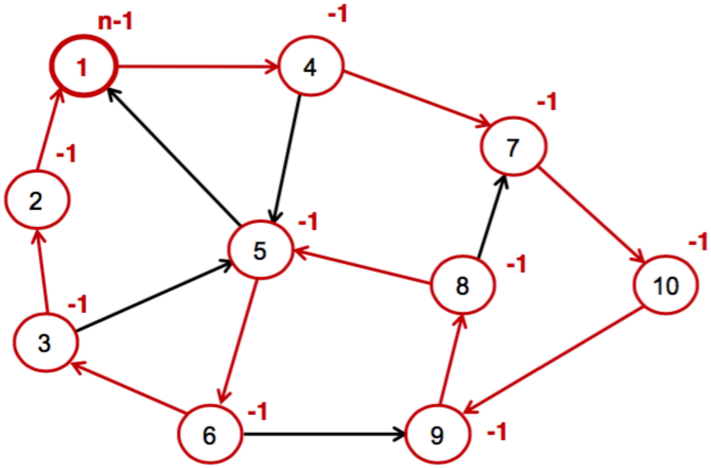
$$x_{ij} \ge 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$



$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

Questi vincoli garantiscono che ogni nodo sia effettivamente visitato attraverso una **unica struttura di connessione** (vincoli di conservazione del flusso).

$$\boxed{\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i=1 \\ -1 & \text{se } i\neq 1 \end{cases} \qquad \forall i \in N}$$

$$x_{ij} \leq (n-1)\, y_{ij} \qquad \forall (i,j)\in A$$

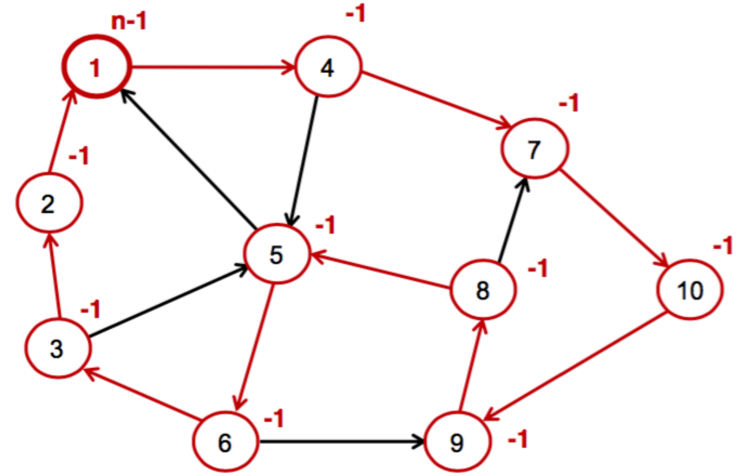
$$x_{ij} \geq 0 \qquad \forall (i,j)\in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j)\in A$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$



$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i = 1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N$$

Questi vincoli (logici) mettono in relazione archi "bagnati dal flusso" con archi inseriti nel tour (vincoli logici di condizionamento). →

$$\boxed{x_{ij} \leq (n-1)\, y_{ij} \qquad \forall (i,j) \in A}$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

La funzione obiettivo misura il costo totale del tuor

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i=1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N$$

$$x_{ij} \le (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

$$x_{ij} \ge 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

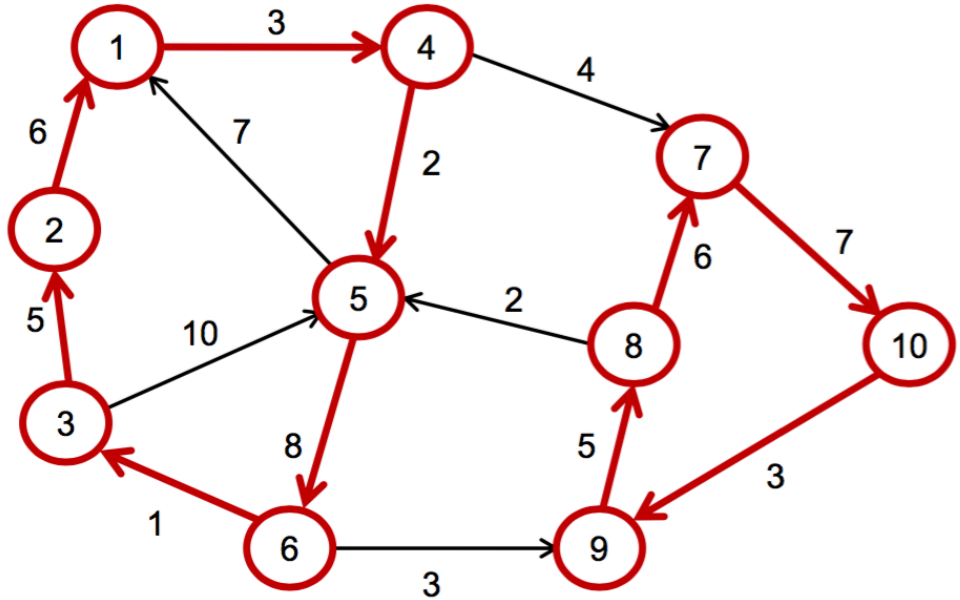
Le variabili y_{ij} sono variabili indicatrici di quali archi appartengono al tour:

y_{ij} =1 se l'arco (i,j) è nel tour
y_{ij} =0 altrimenti

x_{ij} sono variabili di "flusso"

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico



$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\boxed{\begin{array}{ll} \sum_{(i,j)\in A} y_{ij} = 1 & \forall i \in N \\ \sum_{(i,j)\in A} y_{ij} = 1 & \forall j \in N \end{array}}$$

$$\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i = 1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N$$

$$x_{ij} \leq (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

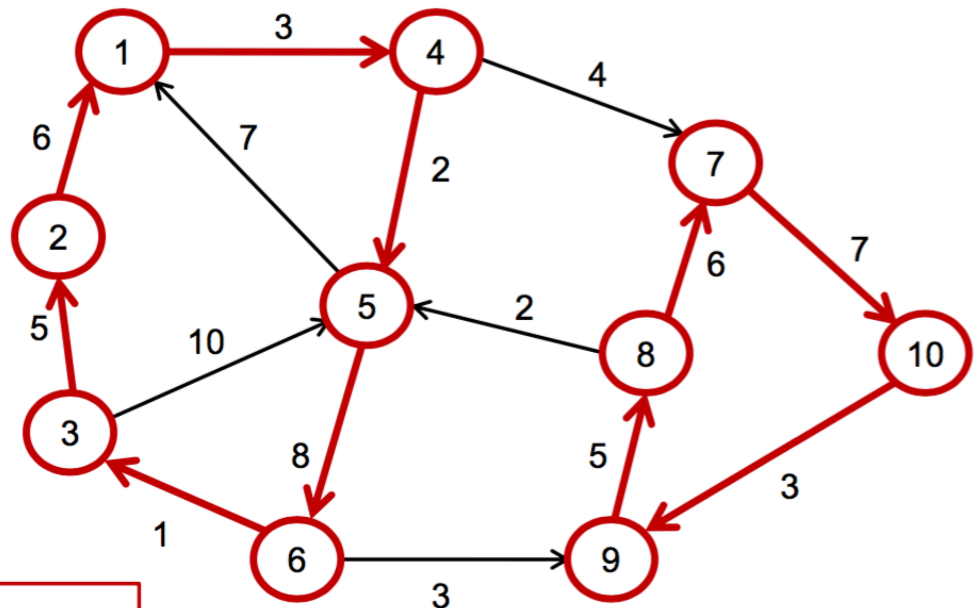
$$x_{ij} \geq 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

1) I vincoli di assegnamento implicano che gli archi del sottoinsieme **Ay** = { (i,j)∈A: y_{ij}=1 } formano una **unione di cicli nodi-disgiunti** tali che ogni nodo di G appartiene ad uno solo di questi cicli.

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico



$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\boxed{\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i=1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N}$$

$$x_{ij} \le (n-1)\, y_{ij} \qquad \forall (i,j) \in A$$

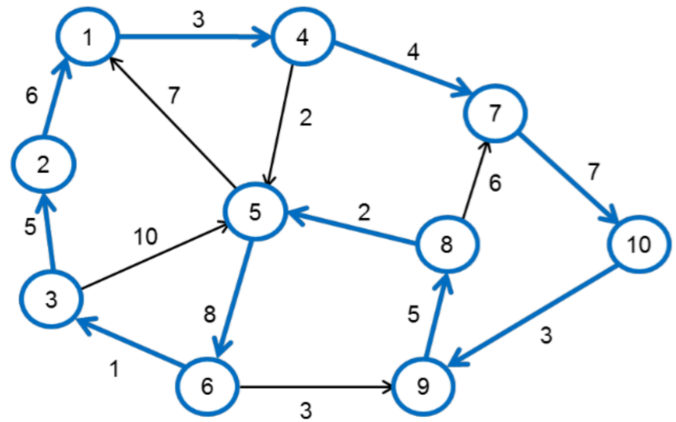
$$x_{ij} \ge 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

2) I **vincoli di flusso** garantiscono la **connessione degli archi bagnati**, cioè degli archi del sottoinsieme **Ax** = { (i,j)∈A: x_{ij}>0 }.

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP asimmetrico



$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N$$

$$\sum_{k\in S(i)} x_{ik} - \sum_{h\in P(i)} x_{hi} = \begin{cases} n-1 & \text{se } i=1 \\ -1 & \text{se } i \neq 1 \end{cases} \qquad \forall i \in N$$

$$\boxed{x_{ij} \leq (n-1)\, y_{ij} \qquad \forall (i,j) \in A}$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

3) I **vincoli logici** mettono in relazione i due sottoinsiemi Ax e Ay:

Ax $\subseteq$ Ay

e, siccome ogni y_{ij}=1 determina un costo aggiuntivo, nella selezione ottima degli archi del tour si considererà Ay = Ax.

Siccome Ax è connesso, **anche Ay sarà connesso** e quindi gli archi in Ay formeranno un **unico ciclo**.

TSP simmetrico

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP **simmetrico**

Consideriamo G=(N,E) non orientato con |N|=n, |E|=m.

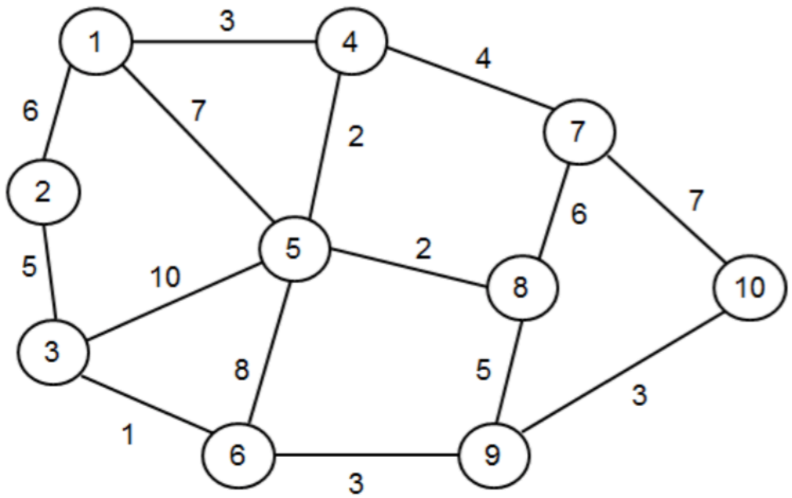
$$\min \sum_{(i,j)\in E} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A(i)} y_{ij} \le 2 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A(S)} y_{ij} \le |S|-1 \qquad \forall S \subset \{1,2,\ldots,n\}$$

$$\sum_{(i,j)\in E} y_{ij} = n$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in E$$



dove:

A(i) è l'insieme degli spigoli che hanno un estremo nel vertice i

A(S) è l'insieme degli spigoli formati da nodi nell'insieme S

Le variabili y_{ij} sono variabili indicatrici di quali spigoli apartengono al tour:

y_{ij} =1 se (i,j) è nel tour

y_{ij} =0 altrimenti

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP **simmetrico**

Ci sono tre tipi di vincoli.

$$\min \sum_{(i,j)\in E} c_{ij} y_{ij}$$

MASSIMO GRADO DI UN NODO:
ogni nodo deve avere **grado al più pari a due**.

$$\boxed{\sum_{(i,j)\in A(i)} y_{ij} \leq 2 \qquad \forall i \in N}$$

$$\sum_{(i,j)\in A(S)} y_{ij} \leq |S|-1 \qquad \forall S \subset \{1,2,\ldots,n\}$$

$$\sum_{(i,j)\in E} y_{ij} = n$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in E$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP **simmetrico**

Ci sono tre tipi di vincoli.

$$\min \sum_{(i,j)\in E} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A(i)} y_{ij} \le 2 \qquad \forall i \in N$$

NO SUBTOUR:
non ci devono essere cicli formati da 2,3,…,n-1 nodi

$$\boxed{\sum_{(i,j)\in A(S)} y_{ij} \le |S|-1 \qquad \forall S \subset \{1,2,\ldots,n\}}$$

$$\sum_{(i,j)\in E} y_{ij} = n$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in E$$

Il problema del commesso viaggiatore (TSP)

Formulazione del problema del TSP **simmetrico**

Ci sono tre tipi di vincoli.

$$\min \sum_{(i,j)\in E} c_{ij} y_{ij}$$

$$\sum_{(i,j)\in A(i)} y_{ij} \leq 2 \qquad \forall i \in N$$

$$\sum_{(i,j)\in A(S)} y_{ij} \leq |S| - 1 \qquad \forall S \subset \{1,2,\ldots,n\}$$

CARDINALITÀ DEL CICLO:
In totale dobbiamo selezionare n spigoli

$$\boxed{\sum_{(i,j)\in E} y_{ij} = n}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in E$$

Euristiche di Ricerca Locale per il TSP

Un algoritmo di Ricerca Locale per il TSP

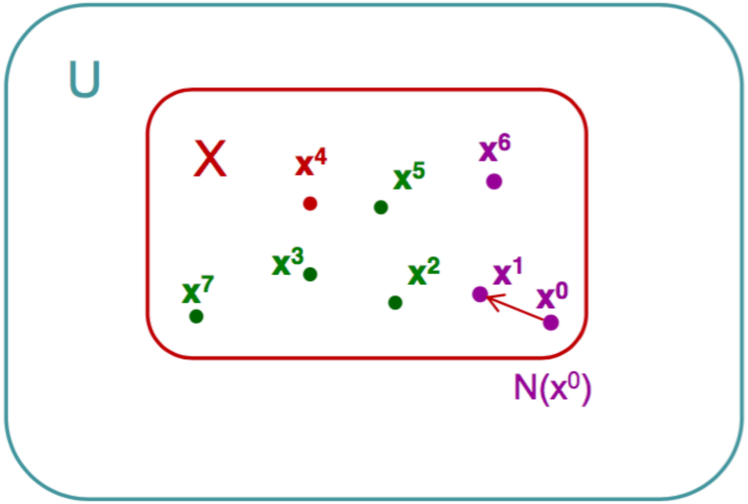
Algoritmo di Lin e Kernighan, 1973

Per semplicità consideriamo G=(N,E) non orientato.

S.P.I.G., possiamo supporre che G sia completo, cioè che esista in G uno spigolo (i,j) per ogni coppia di nodi i e j.

NOTA: Se così non fosse, S.P.I.G., potremmo "completare" il grafo G aggiungendo archi fittizi con costo altissimo (MULTA).

Caso discreto



ALGORITMO DI RL:

- intorno
- soluzione iniziale
- mossa
- arresto

Un algoritmo di Ricerca Locale per il TSP

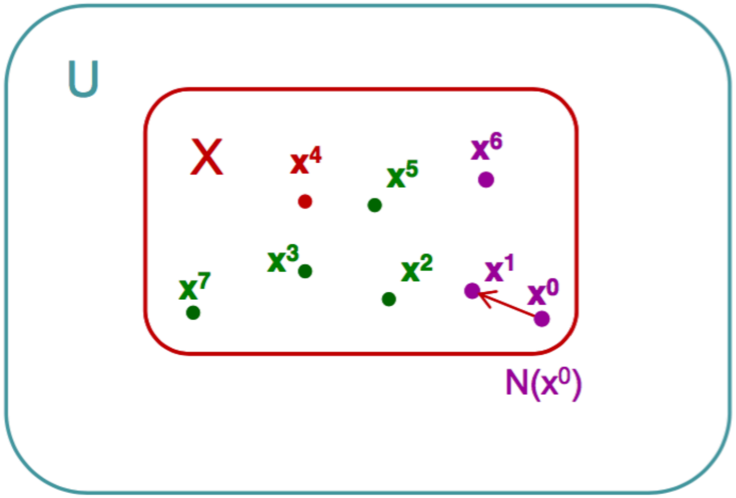
Algoritmo di Lin e Kernighan, 1973

Una soluzione ammissibile del TSP corrisponde a un tour in G.

L'insieme **X** del problema di ottimizzazione è dato da **tutti i possibili tour in G**.

Sia x^t il vettore che identifica il tour ammissibile corrente, dobbiamo definire l'intorno di x^t, $N(x^t)$

Caso discreto



In altre parole, dobbiamo caratterizzare una soluzione "vicina" a x^t, $x \in N(x^t)$, e definire la regola per passare da x^t a una nuova $x^{t+1} \in N(x^t)$ (**mossa**).

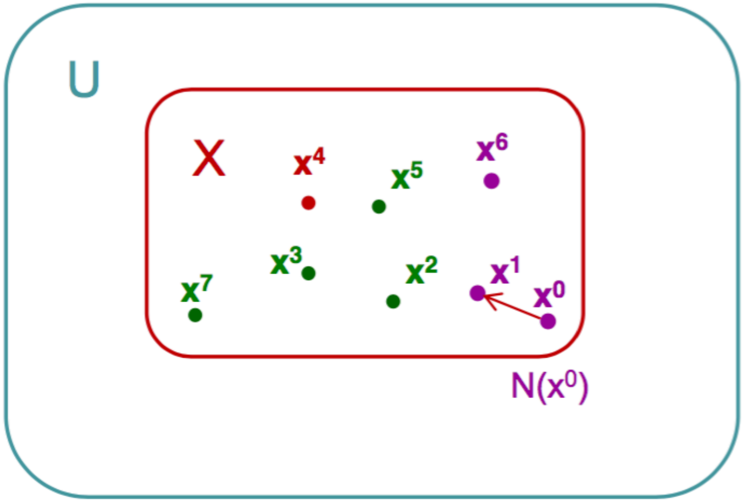
LK (1973) introducono l'operazione di **k-scambio** che corrisponde alla mossa per passare da un tour ammissibile ad un altro tour ammissibile.

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.

Caso discreto

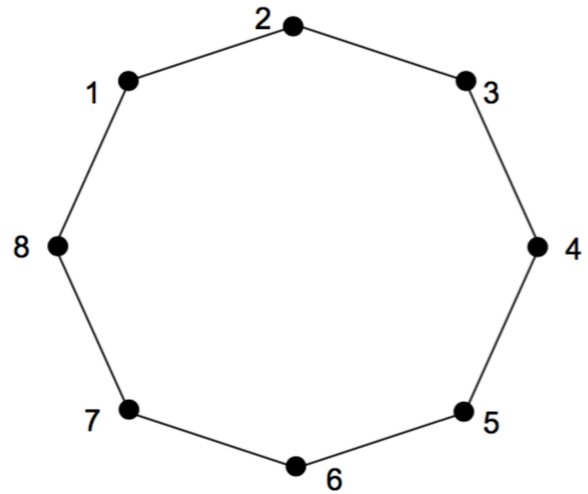


➡ Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



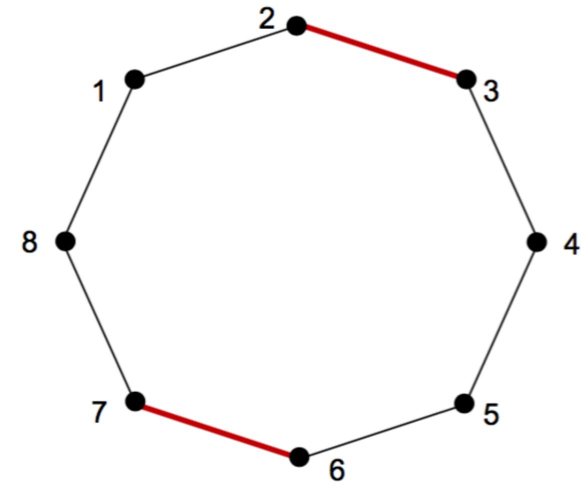
<**1**,2,3,4,5,6,7,8,**1**>

Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



<**1**,2,3,4,5,6,7,8,**1**>

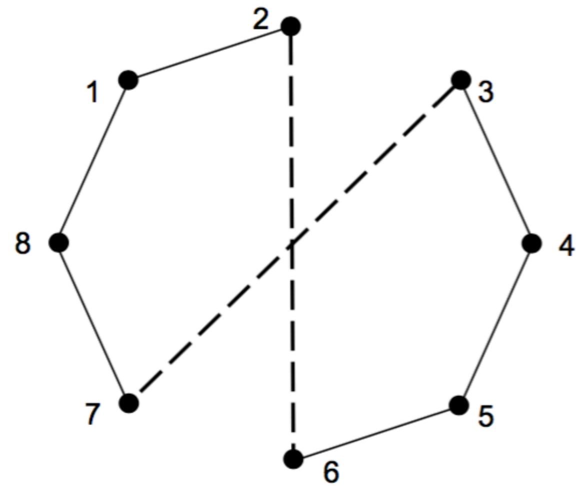
Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

Consideriamo il caso più semplice (e più comunemente adottato) di k=2:
2-opt → 2-scambio

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



<**1**,2,3,4,5,6,7,8,**1**>

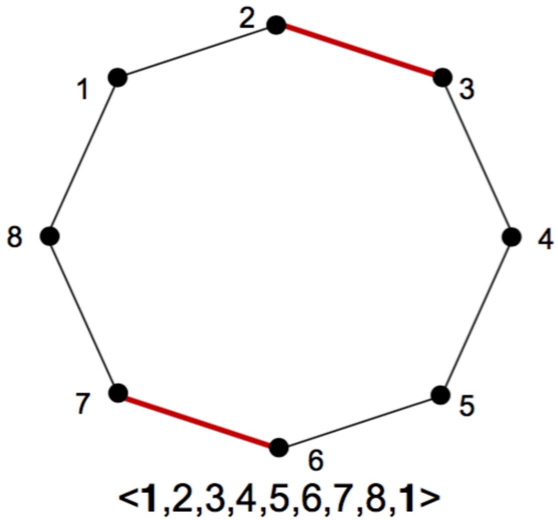
Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

Consideriamo il caso più semplice (e più comunemente adottato) di k=2:
2-opt → 2-scambio

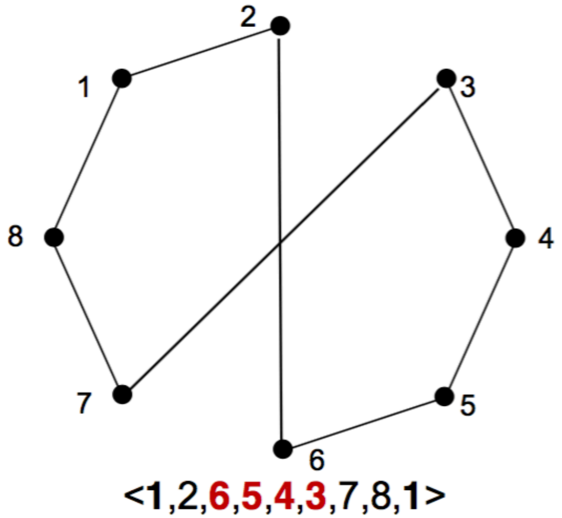
Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

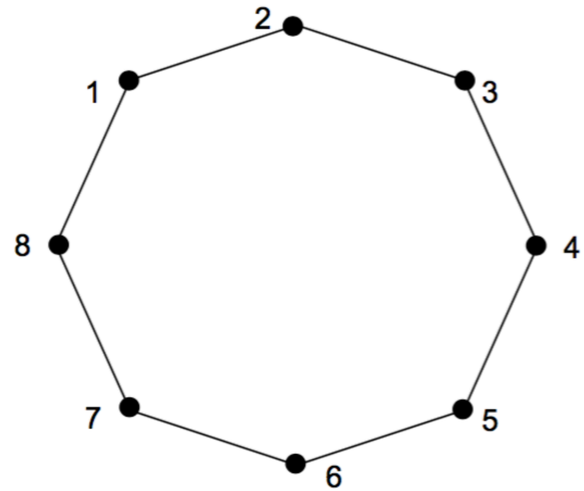


Consideriamo il caso più semplice (e più comunemente adottato) di k=2:
2-opt → 2-scambio

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



<**1**,2,3,4,5,6,7,8,**1**>

Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

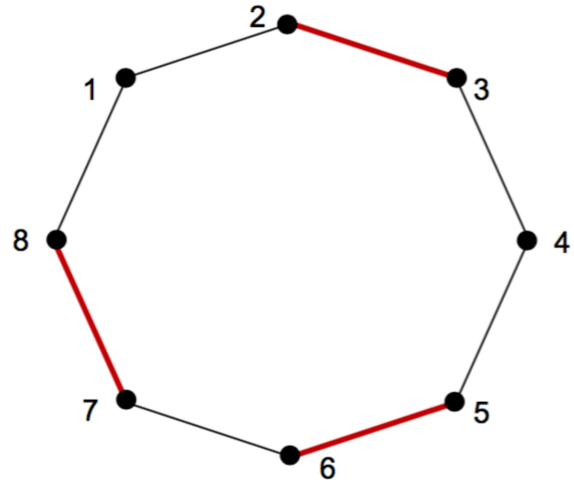
Consideriamo il caso k=3:

3-opt → 3-scambio

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



<**1**,2,3,4,5,6,7,8,**1**>

Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

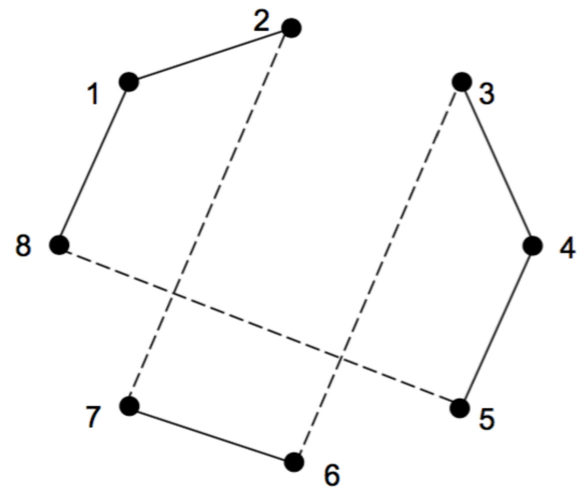
Consideriamo il caso k=3:

3-opt → **3-scambio**

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



<**1**,2,3,4,5,6,7,8,**1**>

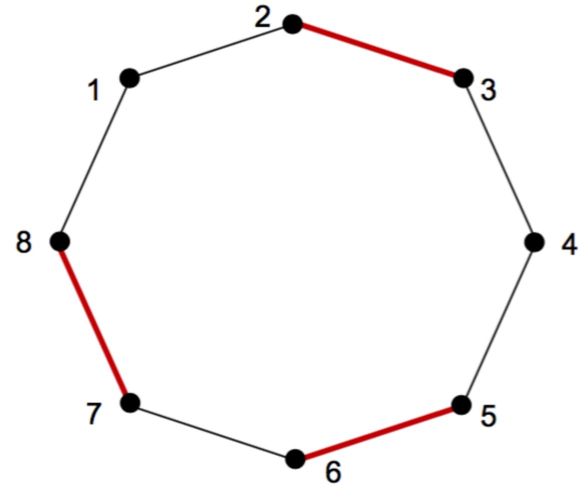
Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).

Consideriamo il caso k=3:
3-opt → 3-scambio

Un algoritmo di Ricerca Locale per il TSP

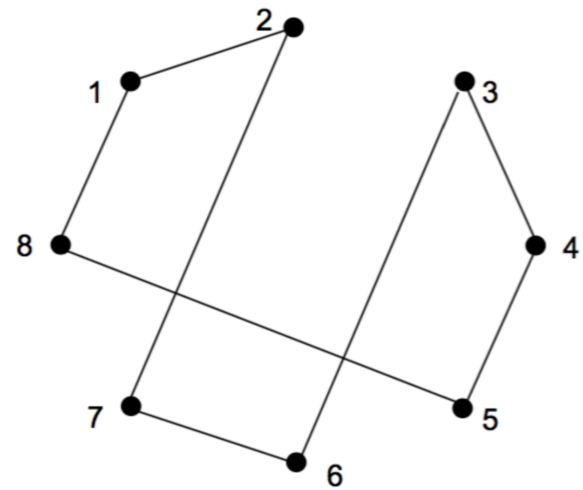
Algoritmo di Lin e Kernighan, 1973

Sia $x^t \in X$ un tour di G, una operazione (mossa) di **k-scambio** consiste nel rimuovere da x^t **k spigoli non consecutivi** e sostituirli con altri k spigoli in modo tale da **ottenere un nuovo tour** x^{t+1} diverso da x^t.



<**1**,2,3,4,5,6,7,8,**1**>

Il **k-scambio** è noto con il nome di **k-opt** (k=2,3).



<**1**,2,**7**,**6**,3,4,5,**8**,**1**>

Consideriamo il caso k=3:

3-opt → 3-scambio

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

Per semplicità consideriamo G=(N,E) non orientato.

S.P.I.G., possiamo supporre che G sia completo, cioè che esista in G uno spigolo (i,j) per ogni coppia di nodi i e j.

ALGORITMO 2-OPT

(inizializzazione)
Individuare un tour di partenza qualsiasi attraverso una visita del grafo G (**NOTA:** sotto l'ipotesi che G sia completo, basta fissare una qualsiasi sequenza ordinata dei nodi in N – operazione semplice).

(iterazione t: 2-scambio)
Sia x^t il tour corrente:
Rimuovere da x^t **due spigoli non consecutivi**
Inserire i due spigoli che generano un nuovo tour x^{t+1} diverso da x^t

se il costo di x^{t+1} è inferiore al costo di x^t:

$$x^t \leftarrow x^{t+1}$$

Un algoritmo di Ricerca Locale per il TSP

Algoritmo di Lin e Kernighan, 1973

ALGORITMO 2-OPT

(inizializzazione)
Individuare un tour di partenza qualsiasi attraverso una visita del grafo G (**NOTA:** sotto l'ipotesi che G sia completo, basta fissare una qualsiasi sequenza ordinata dei nodi in N – operazione semplice).

(iterazione t: 2-scambio)
Sia x^t il tour corrente:
Rimuovere da x^t **due spigoli non consecutivi**
Inserire i due spigoli che generano un nuovo tour x^{t+1} diverso da x^t

se il costo di x^{t+1} è inferiore al costo di x^t:
$x^t \leftarrow x^{t+1}$

INPUT: {min f(x), x in X} ; x^0 in X

OUTPUT: x* in X (ottimo locale)

Passo 0 (inizializzazione)
x:= x^0 in X

Passo1 (arresto)
Se N(x)∩X è vuoto
STOP: x* := x
Altrimenti andare al Passo 2

Passo 2 (avanzamento)
Selezionare una x' in N(x)∩X
N(x) := N(x) – {x'}
Se f(x') < f(x)
x := x'
aggiornare N(x)
Tornare al Passo 1

X è l'insieme di tutti i possibili tour di G

N(x) è l'insieme di tutti i possibili tour di G (diversi da x) che possono essere ottenuti con una operazione 2-opt.

f(x) è il costo del tour x.

Algoritmo generale di Discesa

INPUT: $G=(N,E)$ (completo); costi $c_{ij} \geq 0 \ \forall \ (i,j) \in N$;
x^0=permutazione degli indici 1,2...,n

OUTPUT: x^* tour di G di costo totale minimo

Specificando opportunamente INPUT, OUTPUT e N(x) nell'algoritmo generale di RL si ottiene un algoritmo di RL per il TSP.

Passo 0 (inizializzazione)
$x := x^0$ in X

Passo1 (arresto)
Se $N(x) \cap X$ è vuoto
STOP: $x^* := x$
Altrimenti andare al Passo 2

Passo 2 (avanzamento)
Selezionare una x' in $N(x) \cap X$
$N(x) := N(x) - \{x'\}$
Se $f(x') < f(x)$
$x := x'$
aggiornare N(x)
Tornare al Passo 1

Algoritmo generale di Discesa

OSSERVAZIONI

L'algoritmo di Lin e Kernighan (1973) è un algoritmo di ricerca locale euristico e dunque non garantisce una soluzione di ottimo globale, ma solo di ottimo locale.

All'arresto dell'algoritmo 2-opt non è detto che la soluzione sia di ottimo globale perché è possibile che esistano tour più convenienti di quello corrente ma che **non** possono essere ottenuti da y^t con mosse di tipo 2-opt.

Il **vantaggio** di questo algoritmo (come sempre nella RL) è che è concettualmente semplice e capace di valutare velocemente molte soluzioni ammissibili.

→ Per **problemi TSP di dimensioni molto grandi** è più indicato usare un algoritmo di questo tipo piuttosto che un algoritmo di tipo B&B.