



UNIVERSITÀ DI PISA

CORSO DI LAUREA IN MATEMATICA

**Algoritmi per il problema di
assegnamento ottimo nel caso di costo
concavo su un intervallo**

TESI DI LAUREA TRIENNALE
IN MATEMATICA

ANNO ACCADEMICO 2021/2022

RELATORE:

Dario Trevisan
UNIVERSITÀ DI PISA

CANDIDATO:

Giacomo Lagomarsini

Indice

1	Il problema dell'assegnamento nel caso concavo	4
1.1	Introduzione	4
1.2	Proprietà del caso concavo	6
2	Algoritmi per il caso concavo	9
2.1	L'assegnamento di Dyck	9
2.2	Un algoritmo esatto per l'assegnamento	12
2.2.1	Indicatori locali per catene	12
2.2.2	Dimostrazione del Teorema 1 (caso bilanciato)	18
2.2.3	Dimostrazione del caso non bilanciato	22
2.2.4	L'algoritmo e il caso generale	25
3	Analisi del caso di punti scelti casualmente	28
3.1	Costo medio dell'assegnamento di Dyck e dell'assegnamento ottimale	28
3.2	Complessità degli algoritmi	29
3.2.1	Complessità dell'algoritmo 2 su catene	30
3.2.2	Algoritmo esatto nel caso generale	30
4	Sperimentazioni numeriche	35
4.1	codici	35
4.1.1	Algoritmo per l'assegnamento di Dyck	35
4.1.2	Algoritmo per l'assegnamento su catene	36
4.1.3	Algoritmo esatto per l'assegnamento basato sulle catene	39
4.2	Risultati della sperimentazione	40
4.2.1	Esempi di assegnamenti	40
4.2.2	Andamento del costo dell'assegnamento di Dyck	40
4.2.3	Andamento del costo dell'assegnamento ottimale	43

4.2.4 Tempo di esecuzione dell'algoritmo basato su catene . . 43

Abstract

L'elaborato verte sull'analizzare le proprietà di vari algoritmi, esatti o approssimati, per il problema dell'assegnamento per due insiemi in \mathbb{R} nel caso di costo concavo. Dati due insiemi X e Y di N punti e una funzione di costo $c : X \times Y \rightarrow \mathbb{R}$ per connettere ogni punto del primo insieme ad ogni punto del secondo il problema dell'assegnamento consiste nel trovare la funzione di assegnamento $M : X \rightarrow Y$ che minimizzi il costo totale $\sum c(x, M(x))$. Si studia in particolare il caso di costo concavo in una dimensione, ovvero il caso in cui $X, Y \in \mathbb{R}$, e $c(x, y)$ è una funzione concava e crescente della distanza $|x - y|$.

Come evidenziato in [5] tali assunzioni sulla funzione di costo dovrebbero modellare meglio il problema del trasporto di beni su larga scala rispetto a un costo convesso.

Nel capitolo 1 vengono descritti il problema e le prime proprietà che derivano dalla concavità della funzione di costo c , su cui si basano i teoremi e gli algoritmi nei capitoli successivi.

Nella prima parte del capitolo 2 viene studiato un assegnamento approssimato, detto assegnamento di Dyck [2], che si può calcolare che ha le proprietà descritte nel capitolo 1, e viene descritto un algoritmo per trovare tale assegnamento. Nella seconda parte del capitolo 2 vengono introdotti gli indicatori locali [3], delle speciali funzione definite su catene, ovvero sottoinsiemi di $X \cup Y$ dove i punti di X e Y sono a due a due alternati. A partire da queste funzioni viene descritto un algoritmo che permette di trovare con relativa facilità l'assegnamento ottimale.

Nel capitolo 3 viene trattato il caso speciale in cui prendiamo i punti in maniera casuale. In particolare si fanno alcune considerazioni sul costo medio dell'assegnamento di Dyck, che nella maggior parte dei casi sembra non discostarsi molto dal costo medio ottimale. Inoltre si dimostra che mediamente l'algoritmo ottimale è molto più veloce che nel caso pessimo, attraverso alcune considerazioni statistiche sulla distribuzione dei punti.

Entrambi gli algoritmi sono stati implementati e nel capitolo 4 sono state fatte delle sperimentazioni nel caso di punti scelti in modo casuale in un intervallo, per verificare varie proprietà espone nei capitoli precedenti.

Capitolo 1

Il problema dell'assegnamento nel caso concavo

1.1 Introduzione

Un grafo bipartito completo è un grafo $G = (V, E)$, dove $V = X \cup Y$ è unione disgiunta di due insiemi finiti e $E = X \times Y$ è l'insieme degli archi. Ad ogni arco (x_i, y_j) associamo una funzione di costo $c_{ij} = c((x_i, y_j))$.

Il *problema dell'assegnamento ottimale* consiste nel trovare un assegnamento, ovvero un sottoinsieme $\mathcal{M} \subset E$ tale che

- per ogni $(x, y), (x', y') \in \mathcal{M}$, $x \neq x', y \neq y'$
- $|\mathcal{M}| = \min\{|X|, |Y|\}$,

che minimizzi

$$c(\mathcal{M}) = \sum_{(x,y) \in \mathcal{M}} c(x,y) \quad (1.1)$$

Alternativamente, possiamo considerare due insiemi finiti X e Y di cardinalità $|X| = M$, $|Y| = N$ e una funzione di *costo* $c : X \times Y \rightarrow \mathbb{R}$, e il *problema dell'assegnamento ottimale* diventa trovare una funzione di assegnamento iniettiva e definita su $\min\{M, N\}$ punti $m : X' \subseteq X \rightarrow Y$ che minimizzi il costo pagato per accoppiare i punti

$$c(m) = \sum_{x \in X'} c(x, m(x)). \quad (1.2)$$

Se $M = N$ parliamo di caso bilanciato, in cui ogni punto di x è accoppiato con un punto di y e viceversa; altrimenti siamo nel caso non bilanciato. A meno di scambiare X e Y possiamo assumere che sia $M < N$ e quindi f definita su tutto X .

Il problema può essere riformulato in termini di programmazione lineare intera: detti $X = \{x_1, \dots, x_M\}$, $Y = \{y_1, \dots, y_N\}$ vogliamo trovare

$$\min C(\gamma) = \sum_{i,j} c(x_i, y_j) \gamma_{ij} \quad (1.3)$$

con i vincoli

$$\sum_{i,j} \gamma_{ij} = \min\{M, N\}, \sum_{i=1}^M \gamma_{ij} \leq 1, \sum_{j=1}^N \gamma_{ij} \leq 1, \gamma_{ij} \in \{0, 1\} \quad (1.4)$$

Definizione 1. Nel caso in cui X e Y sono sottoinsiemi di un intervallo $I \subset \mathbb{R}$ e $c(x, y) = g(|x - y|)$ è una funzione (strettamente) concava e crescente della distanza euclidea, parliamo di *problema di assegnamento concavo*.

Possiamo, almeno nel caso bilanciato, vedere un problema di assegnamento come un caso particolare di problema di trasporto ottimale, che nel caso discreto corrisponde al trasportare una certa quantità di beni da dei punti $x_i \in X$ che possono offrire una quantità s_i di risorse a dei punti $y_j \in Y$ che hanno una richiesta d_j , pagando il minore costo possibile. Il problema di assegnamento corrisponde al caso in cui le domande e le risorse abbiano peso unitario ovvero $s_i = d_j = 1$ per ogni i e j . Il problema unidimensionale in un intervallo è stato completamente risolto nel caso in cui c sia una funzione *convessa* della distanza. In tal caso l'assegnamento ottimale non dipende dalla funzione di costo c , ma è ottenuto prendendo un assegnamento monotono, come dimostriamo nella Proposizione n.

Il caso concavo ha goduto di molta meno attenzione, tuttavia se pensiamo all'assegnamento come un particolare problema di trasporto, un costo concavo modella bene viaggi su larga scala come evidenziato in [5]: l'idea è che per viaggi che percorrono più distanze il prezzo pagato per percorrere un'unità di lunghezza fissata diminuisca, e un singolo viaggio lungo costi meno di tanti viaggi più corti che sommati percorrono la stessa distanza, perciò è favorita una divisione tra fornitori a breve e a lunga distanza. A differenza

del caso convesso, poi si vede una forte dipendenza dalla forma della funzione di costo c : ad esempio nel caso bilanciato, se $X = \{x_1 < \dots < x_N\}$, $Y = \{y_1 < \dots < y_N\}$, $m(x_i) = y_i$ è ottimale per ogni funzione di costo c convessa. Questo risultato non si estende al caso concavo: ad esempio pren-

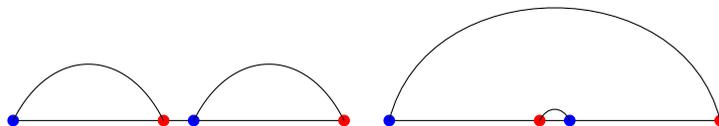


Figura 1.1: A sinistra l'assegnamento ottimale per il costo $c = |x - y|^{0.5}$, a destra per il costo $c(x, y) = |x - y|^{0.9}$

dendo $X = \{x_1, x_2\}$ e $Y = \{y_1, y_2\}$ con $x_1 = 0, x_2 = 1.2$ e $y_1 = 1, y_2 = 2.2$ si vede che se $c(x, y) = |x - y|^{0.5}$ l'assegnamento ottimale è $\{(x_2, y_1), (x_1, y_2)\}$; se invece $c(x, y) = |x - y|^{0.9}$ è meno costoso prendere $\{(x_1, y_1), (x_2, y_2)\}$ (Figura 1.1). Per una funzione di costo convessa verrà invece sempre scelta la soluzione di sinistra. Dunque la discussione nel caso concavo risulta più complessa rispetto al caso convesso, e l'assegnamento ottimale è dipendente della funzione di costo: dare una descrizione analitica è per la maggior parte dei casi impossibile, ed è necessario ricorrere ad algoritmi per trovare la configurazione di costo minimo.

1.2 Proprietà del caso concavo

La concavità della funzione di costo c e il fatto che il problema sia unodimensionale possono essere sfruttati per dimostrare alcune proprietà necessarie affinché un assegnamento sia ottimale:

Definizione 2. Diciamo che un assegnamento m è *crossing* se esistono $x_1, x_2 \in X$ e $y_1, y_2 \in Y$ tali che $f(x_1) = y_1, f(x_2) = y_2$ e valga una delle seguenti:

1. $x_1 < x_2 < y_1 < y_2$,
2. $y_1 < x_2 < x_1 < y_2$,
3. $x_1 < y_2 < y_1 < x_2$,

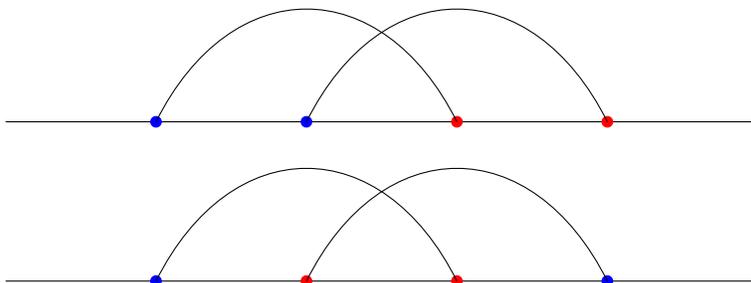


Figura 1.2: Due assegnamenti crossing.

4. $y_1 < y_2 < x_1 < x_2$.

L'idea è che se un assegnamento è crossing, due suoi archi si intersecano, se disegnati come semicerchi sopra il segmento I , come mostrato nella figura 1.2

Proposizione 1. (non-crossing rule) *Se m è un assegnamento ottimale, allora non è crossing.*

Dimostrazione. Per assurdo. Sia m ottimale; supponiamo che in m ci sia una coppia di punti come in 2. nella definizione di assegnamento crossing. Sia m' uguale a m sui punti diversi da x_1 e x_2 e con $m'(x_1) = y_2, m'(x_2) = y_1$. Si ha $|x_1 - y_2| < |x_1 - y_1|$ e $|x_2 - y_1| < |x_2 - y_2|$, perciò $c(m') < c(m)$ poiché g è crescente. Supponiamo invece invece che m accoppi due paia di punti come in 1. Dato che $y_1 - x_2 < y_2 - x_2 < y_2 - x_1$, possiamo scrivere come combinazione convessa, per qualche $0 < t < 1$,

$$(y_2 - x_2) = (1 - t)(y_1 - x_2) + t(y_2 - x_1).$$

Notiamo inoltre che

$$(y_1 - x_1) = t(y_1 - x_2) + (1 - t)(y_2 - x_1).$$

Ma allora per la concavità di g si ha

$$\begin{aligned} g(y_2 - x_2) &\geq (1 - t)g(y_1 - x_2) && +tg(y_2 - x_1) \\ g(y_1 - x_1) &\geq tg(y_1 - x_2) && +(1 - t)g(y_2 - x_1) \end{aligned}$$

CAPITOLO 1. IL PROBLEMA DELL'ASSEGNAZIONE NEL CASO CONCAVO8

Sommando le due disequazioni si ottiene che $c(x_1, y_2) + c(x_2, y_1) \leq c(x_2, y_2) + c(x_1, y_1)$, in contraddizione con l'ottimalità di m . I casi 3. e 4. sono analoghi rispettivamente ai casi 2. e 1. \square

Una conseguenza della non-crossing rule è il seguente fatto:

Proposizione 2. *Se due punti x e y sono assegnati l'uno all'altro da un assegnamento ottimale, ad $I = (\min(x, y), \max(x, y))$ appartengono un numero uguale di punti in X e in Y .*

Dimostrazione. Supponiamo inizialmente che tutti i punti in I siano assegnati; allora per la non-crossing rule ogni punto z in I è assegnato ad un punto z' sempre in I .

Infatti se così non fosse le coppie (x, y) e (z, z') renderebbero l'assegnamento crossing, e quindi non ottimale, perciò tutti i punti in I sono assegnati tra di loro, e quindi i punti di X e quelli di Y in I sono in numero uguale.

Se inoltre esistesse un punto $y' \in Y$ in I non assegnato, l'assegnamento m' uguale a m tranne in x e con $m'(x) = y'$ sarebbe meno costoso di m : infatti $C(m') = C(m) - c(x, y) + c(x, y')$, e $c(x, y') = g(|y' - x|) < g(|y - x|) = c(x, y)$, poiché g è crescente e $|y' - x| < |y - x|$. \square

Definizione 3. Un punto $z \in X \cup Y$ è detto *esposto* per un assegnamento m se per ogni x, y tali che $m(x) = y$, $z \notin (\min(x, y), \max(x, y))$.

Corollario 1. *Nel caso non bilanciato, in cui $|X| < |Y|$, ogni punto $y \in Y$ non assegnato da un assegnamento ottimale è esposto.*

Dimostrazione. Come già visto nella dimostrazione della Proposizione 2, se ci fosse un punto non esposto e non assegnato, sarebbe meno costoso assegnarlo. \square

Capitolo 2

Algoritmi per il caso concavo

Come già visto, l'assegnamento si può riscrivere come un problema di programmazione lineare, e quindi può essere risolto attraverso tecniche come l'algoritmo del simplesso oppure riconducendo il problema a un problema di flusso di costo minimo. Un algoritmo di natura combinatoria usato per trovare un assegnamento ottimale per una qualsiasi funzione di costo è l'algoritmo ungherese. Nel nostro caso si può tuttavia migliorare di molto in termini di efficienza sfruttando le proprietà geometriche del problema, come la concavità della funzione di costo e il fatto che sia unidimensionale.

In questo capitolo descriveremo un particolare assegnamento, detto assegnamento di Dyck, che ha in modo ovvio le proprietà descritte dalle Proposizioni 1 e 2. Tale assegnamento è particolarmente facile da calcolare, e inoltre pone le basi per la discussione sull'algoritmo di assegnamento sulle catene descritto nelle sezioni successive.

2.1 L'assegnamento di Dyck

Dati due insiemi disgiunti di punti $X, Y \subset I \subset \mathbb{R}$, chiamiamo $S = \{s_1, s_2, \dots, s_{M+N}\} = X \cup Y$, con $s_i < s_j \Leftrightarrow i < j$. Ad ogni punto $s_i \in S$ associamo un segno $\sigma_i \in \{-1, 1\}$, con

$$\sigma_i = \begin{cases} 1 & \text{se } s_i \in X \\ -1 & \text{se } s_i \in Y \end{cases}$$

Assumiamo che il problema sia bilanciato, ovvero che $|X| = |Y| = N$. Il *cammino di Dyck* associato a X e Y è il cammino sul reticolo \mathbb{Z}^2 così

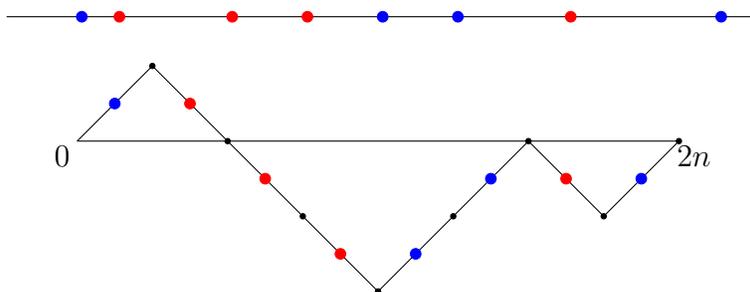


Figura 2.1: Un'istanza bilanciata del problema di assegnamento e il relativo cammino di Dyck. Al centro di ogni passo è disegnato il colore del punto che produce tale passo. X è perciò rappresentato da punti blu, Y da punti rossi

definito: partendo da $(0, 0)$, per ogni punto s_i facciamo un passo della forma $step_i = (1, \sigma_i)$, ovvero facciamo un passo all'insù se $s_i \in X$, facciamo un passo all'ingiù se $s_i \in Y$. La definizione è spiegata graficamente nella Figura 2.1.

Per ogni punto $x_i \in X$, $i = 1, \dots, N$ possiamo definire l'altezza di $h(x_i)$ come l'ordinata del punto medio dell' i -esimo passo ascendente del cammino di Dyck (meno $\frac{1}{2}$ in modo da avere coordinate intere), e allo stesso modo per ogni $y_j \in Y$, $j = 1 \dots N$, $h(y_j)$ come l'ordinata del punti medio del j -esimo passo discendente. Chiamiamo uno *strato* ogni insieme

$$\mathcal{C}_h = \{x_i \mid h(x_i) = h\} \cup \{y_j \mid h(y_j) = h\}$$

di punti tutti alla stessa altezza. Osserviamo che, poiché il problema è bilanciato, l'ultimo punto del cammino avrà altezza pari a 0. Dalla Proposizione 2 segue il seguente lemma:

Lemma 1. *Se due punti x_i e y_j sono accoppiati da un assegnamento ottimale, allora appartengono allo stesso strato*

Dimostrazione. Due punti sono alla stessa altezza se e solo se fra di loro si trovano un numero uguale di punti in X e in Y . \square

Fermo restando che vogliamo la proprietà che i punti accoppiati siano alla stessa altezza, definiamo l'assegnamento di Dyck nel modo più semplice possibile:

Definizione 4. *L'assegnamento di Dyck, che verrà indicato con m_{Dyck} , è ottenuto appaiando ogni punto non assegnato con il punto immediatamente successivo alla stessa altezza (vedasi Figura 2.2).*

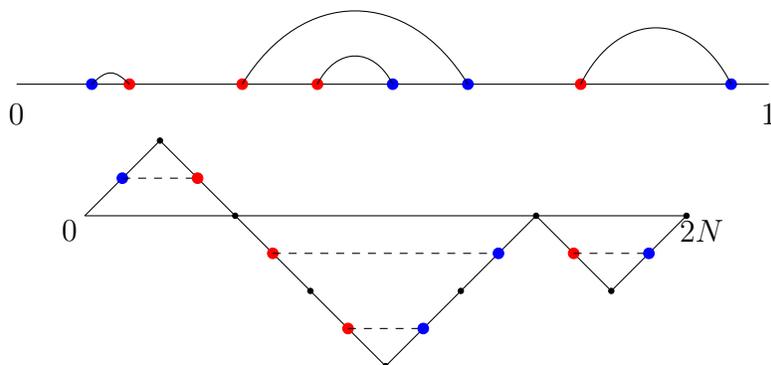


Figura 2.2: Cammino e assegnamento di Dyck relativi all'esempio nella Figura 2.1

Si noti che il sottoinsieme di punti alla stessa altezza è composto di punti di X e Y alternati, perciò l'assegnamento è ben definito. Inoltre l'assegnamento è non crossing: se due punti sono alla stessa altezza per ogni altra altezza diversa, tra i due punti ci sarà un numero uguale di punti alla stessa altezza appartenenti a X e Y , che verranno perciò accoppiati. Si può osservare anche che se due punti sono accoppiati, tra di essi non esistono punti ad altezza 0. I punti cioè appartengono alla stessa escursione irriducibile, ovvero un sottocammino in cui solo le estremità hanno altezza 0. Più semplicemente possiamo osservare il seguente fatto:

Osservazione 1. Ad ogni passo, se nel cammino di Dyck ci allontaniamo da $\{x = 0\}$, il punto associato a tale passo è assegnato nell'assegnamento di Dyck ad un punto successivo, altrimenti il punto sarà da assegnare ad un punto precedente.

Quest'ultima osservazione ci porta ad un algoritmo molto semplice per calcolare l'assegnamento di Dyck (Algoritmo 1).

Osservazione 2. L'assegnamento di Dyck dipende solo da $(\sigma_1, \dots, \sigma_{2N})$, cioè solo dalle posizioni relative dei punti, e non dalle loro posizioni effettive, né dalla funzione di costo c .

In [4] viene dimostrato che per una funzione di costo della forma $c(x, y) = |x - y|^t$, per $t \rightarrow 1^-$ il costo dell'assegnamento di Dyck tende al costo ottimale per una qualsiasi istanza, e per $t = 1$ l'assegnamento è in effetti ottimale.

Tuttavia l'assegnamento di Dyck non è ottimo nella maggior parte dei casi: per esempio nell'esempio illustrato nella Figura 1.1 tale assegnamento

Algoritmo 1 Algoritmo per l'assegnamento di Dyck

```

Escursione = []
for  $k = 1, \dots, 2N$  do
  if  $h(k) = 0$  then
    segno(Escursione) =  $\sigma(k)$ 
    Escursione(0) =  $k$ 
     $k = k + 1$ 
  else
    if segno(Escursione) =  $\sigma(k)$  then
      appendiamo  $k$  a Escursione
    else
      assegniamo  $k$  all'ultimo elemento di Escursione
      cancelliamo l'ultimo elemento di Escursione
    end if
  end if
end for

```

sarà sempre quello a sinistra, ottimale per $c(x, y) = |x - y|^{0.9}$ ma non per $c(x, y) = |x - y|^{0.5}$. Motivati dal fatto che in un accoppiamento ottimo ogni paio di punti accoppiati è formato da punti con la stessa altezza nel cammino di Dyck, nella prossima sezione vogliamo affinare il metodo con cui accoppiamo i punti appartenenti allo stesso strato.

2.2 Un algoritmo esatto per l'assegnamento

In questa sezione descriviamo un algoritmo esatto per l'assegnamento ottimale nel caso concavo, seguendo la descrizione in [3].

2.2.1 Indicatori locali per catene

Si è già osservato che per ogni strato i punti appartenenti allo strato sono alternati tra appartenenti a X e appartenenti a Y . Formalizziamo questa proprietà con una definizione:

Definizione 5. Diciamo che due insiemi P e Q formano una *catena bilanciata* se $P = \{p_1, \dots, p_n\}$, $Q = \{q_1, \dots, q_n\}$, con $p_1 < q_1 < p_2 < q_2 \dots < p_n < q_n$. Se

invece $P = \{p_1, \dots, p_{n+1}\}$ e Q è come sopra, con $p_1 < q_1 < \dots < p_n < q_n < p_{n+1}$, diciamo che formano una catena non bilanciata.

Osservazione 3. Se il problema è bilanciato dato un qualunque strato, la catena di punti appartenenti allo strato è bilanciata.

Osservazione 4. Dato uno strato \mathcal{C}_h composto da $2k_h$ punti, l'assegnamento di Dyck accoppia p_i con q_i per ogni $i = 1, \dots, k_h$

Introduciamo ora delle funzioni che ci aiuteranno a capire la forma dell'assegnamento ottimale. Osserviamo che la non-crossing rule ci garantisce che in una catena almeno una coppia di punti consecutivi è accoppiata. Il risultato principale della sezione ci dirà quali sono tali punti.

Definizione 6. Date P e Q due catene, definiamo gli indicatori locali di ordine k relativi all'indice i come

$$I_k^p(i) = c(p_i, q_{i+k}) + \sum_{j=0}^{k-1} c(p_{i+j+1}, q_{i+j}) - \sum_{j=0}^k c(p_{i+j}, q_{i+j}) \quad (2.1)$$

e

$$I_k^q(i) = c(p_{i+k+1}, q_i) + \sum_{j=1}^k c(p_{i+j}, q_{i+j}) - \sum_{j=0}^k c(p_{i+j+1}, q_{i+j}). \quad (2.2)$$

I due apici p e q indicano se il primo punto considerato r_i , appartiene a P o a Q , cioè se $r_i = p_i$ o $r_i = q_i$.

Per una catena bilanciata, nel primo caso $1 \leq k \leq n-1$ e $1 \leq i \leq n-k$, e nel secondo caso $1 \leq k \leq n-2$ e $1 \leq i \leq n-k-1$. Per una catena non bilanciata invece abbiamo in entrambi i casi $1 \leq k \leq n-k$ e $1 \leq i \leq n-k$. Nella Figura 2.3 è rappresentato graficamente un indicatore per $k=2$.

I punti interni all'indicatore sono i punti $\{q_i, p_{i+1}, q_{i+1}, \dots, p_{i+k}\}$ per $I_k^p(i)$, e $\{p_{i+1}, q_{i+1}, \dots, p_{i+k}, q_{i+k}\}$ per $I_k^q(i)$, ovvero i punti "all'interno" del primo arco nella definizione degli indicatori (vedasi ancora la figura 2.3 a sinistra). Possiamo ora enunciare il teorema principale della sezione.

Teorema 1. *Data una catena bilanciata (risp. non bilanciata), e una funzione di costo c concava, strettamente crescente, siano $k_0 \leq n-1$ e $i_0 \leq n-k_0$ tali che*

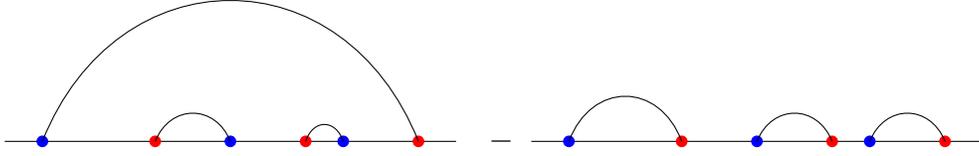


Figura 2.3: Un indicatore di ordine $k = 2$, è uguale alla somma dei costi delle coppie nella figura a sinistra, meno la somma dei costi delle coppie nella figura a destra.

1. per ogni $1 \leq k < k_0$ e per ogni $i_0 \leq i \leq i_0 + k_0 - k$, $I_k^p(i) \geq 0$,
2. per ogni $1 \leq k < k_0 - 1$ e per ogni $i_0 \leq i \leq i_0 + k_0 - k - 1$, $I_k^q(i) \geq 0$, ($i_0 \leq i \leq i_0 - k_0 - k$ nel caso non bilanciato)
3. $I_{k_0}^p(i_0) < 0$.

Allora per ogni assegnamento ottimale m si ha

$$m(p_{i_0+1}) = q_{i_0}, \dots, m(p_{i_0+k_0+1}) = q_{i_0+k_0}.$$

Se invece continuano a valere 1. e 2. e vale

- 3'. $I_{k_0}^q(i_0) < 0$, (con le stesse ipotesi su k_0 e i_0 nel caso non bilanciato, e $1_0 \leq k_0 \leq n - 2$, $1 \leq i_0 \leq i_0 + k_0 - k$ nel caso bilanciato)

allora per ogni assegnamento ottimale m vale

$$m(p_{i_0+1}) = q_{i_0+1}, \dots, m(p_{i_0+k_0}) = q_{i_0+k_0}.$$

Il teorema ci dice cioè che se troviamo un indicatore negativo di ordine k_0 $I_{k_0}^*(i_0)$ e tutti gli indicatori di ordine $1 \leq k < k_0$ che coinvolgono un sottoinsieme dei punti interessati da $I_{k_0}^*(i_0)$ sono positivi, allora i punti interni a $I_{k_0}^*(i_0)$ sono assegnati tra di loro senza che ci siano accoppiamenti annidati (come nell'assegnamento di Dyck).

Per procedere con la dimostrazione, dimostriamo prima alcuni lemmi preliminari

Lemma 2. Sia $c(p, q) = g(|p - q|)$ con g strettamente concava. Dati $1 \leq k \leq n - 1$ e $1 \leq i \leq n - k$, definiamo per ogni $x, y \in \mathbb{R}^+$

$$\varphi_{i,k}^p(x, y) = g(x+y+q_{i+k}-p_i) + \sum_{j=0}^{k-1} c(p_{i+j+1}, q_{i+j}) - g(x) - g(y) - \sum_{j=1}^{k-1} c(p_{i+j}, q_{i+j}). \quad (2.3)$$

Inoltre, per $1 \leq k \leq n - 2$ e $1 \leq i \leq n - k - 1$ nel caso bilanciato, e per $1 \leq k \leq n - k$ e $1 \leq i \leq n - k$ nel caso non bilanciato, definiamo

$$\varphi_{i,k}^q(x, y) = g(x+y+p_{i+k+1}-q_i) + \sum_{j=1}^k c(p_{i+j}, q_{i+j}) - g(x) - g(y) - \sum_{j=1}^{k-1} c(p_{i+j+1}, q_{i+j}). \quad (2.4)$$

Entrambe le funzioni sono strettamente decrescenti rispetto a ciascuna delle due variabili.

Dimostrazione. Dimostriamo che $\varphi_{i,k}^p(x, y)$ è decrescente rispetto a x , gli altri casi sono analoghi. Poiché il secondo, il quarto e il quinto termine della definizione sono costanti rispetto a x , basta dimostrare che $g(x + y - q_{i+k} - p_i) - g(x)$ sia decrescente rispetto a x . Si nota che $y - q_{i+k} - p_i = \epsilon$ è positivo e costante in x , perciò ci siamo ridotti a dimostrare che $g(x + \epsilon) - g(x)$, $\epsilon > 0$ è decrescente.

Ma g è strettamente concava, quindi per ogni ϵ , il rapporto incrementale $\frac{g(x + \epsilon) - g(x)}{\epsilon}$ è strettamente decrescente in x per ϵ fissato, da cui la tesi. \square

Per lavorare sul caso non bilanciato, ci servono anche i seguenti lemmi. Il primo è chiamato

Lemma 3. (regola dei tre) Supponiamo g strettamente crescente. Dati $p < q < p' < q'$, se vale

$$c(p, q') + c(p', q) < c(p, q) + c(p', q'),$$

allora $|p' - q| < \min\{|p - q|, |p' - q'|\}$

Dimostrazione. Poiché g è crescente e $|p - q'| > \max\{|p - q|, |p' - q'|\}$, si ha $c(p, q') > \max(c(p, q), c(p', q'))$, e quindi perché valga la disequazione dev'essere $c(p, q') < \min(c(p, q), c(p', q'))$. La tesi segue ancora dal fatto che g è crescente. \square

Nel nostro caso servirà una generalizzazione nel caso delle catene:

Lemma 4. *Se valgono le ipotesi 1. e 3. del Teorema 1, allora per ogni $i \in \{i_0 + 1, \dots, i_0 + k_0 - 1\}$ vale*

$$|q_i - p_{i+1}| < \min(|p_{i_0} - q_i|, |p_{i+1} - q_{i_0+k_0}|) \quad (2.5)$$

Se invece valgono 2. e 3., si trova che per ogni $i \in \{i_0 + 1, \dots, i_0 + k_0\}$

$$|p_i - q_i| < \min(|q_{i_0} - p_i|, |q_i - p_{i_0+k_0+1}|) \quad (2.6)$$

Dimostrazione. Dimostriamo la prima disuguaglianza, l'altra è analoga. Dall'ipotesi 3. del 1 si ha, portando tutto a destra della disequazione tranne il primo termine e sommando $c(p_{i+1}, q_i)$ con i come nelle ipotesi ad entrambi i lati, che

$$c(p_{i+1}, q_i) + c(p_{i_0}, q_{i_0+k_0}) < \sum_{j=i_0}^{i_0+k_0} c(p_j, q_j) - \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j) + c(p_{i+1}, q_i).$$

Da 1. abbiamo che $I_{i-i_0}^p(i_0) \geq 0$ e $I_{i_0+k_0-i-1}(i+1) \geq 0$, perciò valgono

$$\sum_{j=i_0}^i c(p_j, q_j) \leq c(p_{i_0}, q_i) + \sum_{j=i_0}^{i-1} c(p_{j+1}, q_j)$$

e

$$\sum_{j=i+1}^{i_0+k_0} c(p_j, q_j) \leq c(p_i, q_{i_0+k_0}) + \sum_{j=i+1}^{i_0+k_0-1} c(p_{j+1}, q_j)$$

Sommando a destra e a sinistra le ultime due disuguaglianze si ottiene

$$\sum_{j=i_0}^{i_0+k_0} c(p_j, q_j) \leq c(p_{i_0}, q_i) + c(p_i, q_{i_0+k_0}) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j) - c(p_{i+1}, q_i).$$

Perciò dalla prima disuguaglianza arriviamo a

$$c(p_{i+1}, q_i) + c(p_{i_0}, q_{i_0+k_0}) < c(p_{i_0}, q_i) + c(p_i, q_{i_0+k_0}),$$

per cui grazie al Lemma 3, $|q_i - p_{i+1}| < \min(|p_{i_0} - q_i|, |p_i - q_{i_0+k_0}|)$. \square

L'ultimo lemma è un risultato sulle somme parziali.

Lemma 5. (somme parziali) *Se valgono le ipotesi 1. e 3. del Teorema 1, per ogni $i \in \{i_0 + 1, \dots, i_0 + k_0\}$ e per ogni $i' \in \{i_0, \dots, i_0 + k_0 - 1\}$ valgono le seguenti disequazioni:*

$$\sum_{j=i_0}^{i-1} c(p_j, q_j) > \sum_{j=i_0}^{i-1} c(p_{j+1}, q_j) \quad (2.7)$$

e

$$\sum_{j=i'+1}^{i_0+k_0} c(p_j, q_j) > \sum_{j=i'}^{i_0+k_0-1} c(p_{j+1}, q_j). \quad (2.8)$$

Se valgono 2. e 3'. invece si ha che valgono

$$\sum_{j=i_0}^{i-1} c(p_{j+1}, q_j) > \sum_{j=i_0+1}^i c(p_j, q_j) \quad (2.9)$$

e

$$\sum_{j=i'+1}^{i_0+k_0} c(p_{j+1}, q_j) > \sum_{j=i'+1}^{i_0+k_0} c(p_j, q_j). \quad (2.10)$$

Dimostrazione. Dimostriamo la disequazione (2.7): $I_{k_0}^p(i_0) < 0$, perciò per ogni i tale che $i_0 < i \leq i_0 + k_0$ vale

$$\begin{aligned} \sum_{j=i_0}^{i-1} c(p_j, q_j) &= \sum_{j=i_0}^{i_0+k_0} c(p_j, q_j) - \sum_{j=i}^{i_0+k_0} c(p_j, q_j) \\ &> c(p_{i_0}, q_{i_0+k_0}) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j) - \sum_{j=i}^{i_0+k_0} c(p_j, q_j) \end{aligned}$$

Inoltre, poiché $I_{i_0+k_0-i}^p(i) \geq 0$, si ha che

$$c(p_i, q_{i_0+k_0}) + \sum_{j=i}^{i_0+k_0-1} c(p_{j+1}, q_j) \geq \sum_{j=i}^{i_0+k_0} c(p_j, q_j),$$

e quindi

$$\begin{aligned}
 \sum_{j=i_0}^{i-1} c(p_j, q_j) &> c(p_{i_0}, q_{i_0+k_0}) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j) - c(p_i, q_{i_0+k_0}) - \sum_{j=i}^{i_0+k_0-1} c(p_{j+1}, q_j) \\
 &= c(p_{i_0}, q_{i_0+k_0}) + \sum_{j=i_0}^{i-1} c(p_{j+1}, q_j) - c(p_i, q_{i_0+k_0}) \\
 &> \sum_{j=i_0}^{i-1} c(p_{j+1}, q_j),
 \end{aligned}$$

dove l'ultima disuguaglianza segue dal fatto che $|p_{i_0} - q_{i_0+k_0}| > |p_i - q_{i_0+k_0}|$ e quindi $g(|p_{i_0} - q_{i_0+k_0}|) - g(|p_i - q_{i_0+k_0}|) = c(p_{i_0}, q_{i_0+k_0}) - c(p_i, q_{i_0+k_0}) > 0$ poiché g è strettamente crescente. Le disuguaglianze (2.8) - (2.10) si dimostrano in modo analogo. \square

Con questi lemmi possiamo procedere con la dimostrazione del Teorema 1. Dimosteremo prima il caso di catene bilanciate, poi estenderemo il risultato a catene non bilanciate.

2.2.2 Dimostrazione del Teorema 1 (caso bilanciato)

Ci concentriamo come al solito sul caso in cui valga $I_{k_0}^p(i_0) < 0$, l'altro caso è identico, come si può dedurre facilmente dalla simmetria del problema. Vedremo che l'insieme $\mathcal{S}_{i_0}^{k_0}$ dei punti interni all'indicatore $I_{k_0}^p(i_0) < 0$ è invariante per ogni assegnamento ottimale $m : P \rightarrow Q$, cioè $m(\mathcal{S}_{i_0}^{k_0}) = \mathcal{S}_{i_0}^{k_0}$.

Impossibilità di assegnamenti annidati: Ciò ci porta alla conclusione che i punti interni all'indicatore sono accoppiati come nel Teorema 1: infatti grazie alle ipotesi 1. e 2. possiamo escludere coppie di assegnamenti annidati tra punti di $\mathcal{S}_{i_0}^{k_0}$, ovvero assegnamenti della forma $\pi(p) = q$, $\pi(p') = q'$ con $p' < p < q < q'$. Per la non-crossing rule esiste una coppia di assegnamenti annidato se e solo se sono accoppiati due punti della forma p_i, q_{i+k} . Si può dimostrare che un tale accoppiamento è impossibile per induzione su k :

$k = 2$ Supponiamo per assurdo che $m(p_i) = q_{i+2}$. Allora per la non-crossing rule, $m(p_{i+1}) = q_i$ sono accoppiati, ma se così fosse, per l'ipotesi 1. $I_2^p(i) \geq 0$, e quindi l'assegnamento che accoppia p_i a q_{i+1} e p_{i+1} a q_{i+2} sarebbe migliore.

$k > 2$ Se $m(p_i) = q_{i+k}$, Allora per la non-crossing rule i punti $\{p_{i+1}, \dots, p_{i+k}\}$ sono assegnati ai punti $\{q_i, \dots, q_{i+k-1}\}$. Per ipotesi induttiva $m(p_{i+j+1}) = q_{i+j}$ per $j = 0, \dots, k-1$. Ma per l'ipotesi 1., $I_k^p(i) \geq 0$, ovvero $c(p_i, q_{i+k}) + \sum_{j=0}^{k-1} c(p_{i+j+1}, q_{i+j}) \geq \sum_{j=0}^k c(p_{i+j}, q_{i+j})$, perciò l'assegnamento non sarebbe ottimale.

Analogamente, usando l'ipotesi 2 si dimostra che p_{i+k} non può essere assegnato a q_i per nessun $k \geq 1$, perciò abbiamo dimostrato che se $\mathcal{S}_{i_0}^{k_0}$ è preservato da un assegnamento ottimale, allora i suoi punti sono assegnati come nel Teorema 1.

Vediamo ora che $\mathcal{S}_{i_0}^{k_0}$ è effettivamente invariante per ogni assegnamento ottimale. Per semplificare la notazione chiamiamo π la permutazione di $\{1, \dots, N\}$ associata a m , ovvero

$$m(p_i) = q_j \Leftrightarrow \pi(i) = j.$$

Se per assurdo $\mathcal{S}_{i_0}^{k_0}$ non fosse invariante, abbiamo 3 possibilità:

- a) esistono i_1 tale che $1 \leq i_1 \leq i_0$ e $i_0 \leq \pi(i_1) \leq i_0 + k_0 - 1$
 e i'_1 tale che $\pi(i_1) + 1 \leq i'_1 \leq i_0 + k_0$ e $i_0 + k_0 \leq \pi(i'_1) \leq N$;
- b) esiste i_2 con $i_0 + 1 \leq i_2 \leq i_0 + k_0$ e $1 \leq \pi(i_2) \leq i_0 - 1$
- c) esiste i_2 con $i_0 + k_0 < i_2 \leq N$ e $i_0 \leq \pi(i_2) < i_0 + k_0$

Caso a) Possiamo assumere senza perdere generalità che $\pi(i_1)$ sia l'indice più grande che soddisfi $1 \leq i_1 \leq i_0$ e $i_0 \leq \pi(i_1) \leq i_0 + k_0 - 1$ e che i'_1 sia l'indice più piccolo che soddisfi $\pi(i_1) + 1 \leq i'_1 \leq i_0 + k_0$ e $i_0 + k_0 \leq \pi(i'_1) \leq N$. Assumiamo inoltre che non valgano b) e c).

Grazie a queste assunzioni, per la non-crossing rule troviamo che l'insieme $\{p_i | \pi(i_1) + 1 \leq i \leq i'_1 - 1\} \cup \{q_i | \pi(i_1) + 1 \leq i \leq i'_1 - 1\}$ è invariante per m . Poiché tale insieme è un sottoinsieme di $\mathcal{S}_{i_0}^{k_0}$, valgono le ipotesi 1. e 2., e quindi con la stessa dimostrazione si vede che tali punti devono essere per forza accoppiati consecutivamente, perciò $\pi(i) = i$ per ogni $i = \pi(i_1) + 1, \dots, i'_1 - 1$. Dall'ottimalità di m si ottiene che

$$c(p_{i_1}, q_{\pi(i_1)}) + c(p_{i'_1}, q_{\pi(i'_1)}) + \sum_{j=\pi(i_1)+1}^{i'_1-1} c(p_j, q_j) \leq c(p_{i_1}, q_{\pi(i'_1)}) + \sum_{j=\pi(i_1)}^{i'_1-1} c(p_{j+1}, q_j).$$

Per il Lemma 2 si ottiene inoltre

$$c(p_{i_0}, q_{\pi(i_1)}) + c(p_{i'_1}, q_{i_0+k_0}) + \sum_{j=\pi(i_1)+1}^{i'-1} c(p_j, q_j) \leq c(p_{i_0}, q_{i_0+k_0}) + \sum_{j=\pi(i_1)}^{i'-1} c(p_{j+1}, q_j).$$

Sommando i termini $c(p_{j+1}, q_j)$ in modo da completare la sommatoria a destra, si trova che

$$\begin{aligned} & c(p_{i_0}, q_{\pi(i_1)}) + c(p_{i'_1}, q_{i_0+k_0}) + \sum_{j=\pi(i_1)+1}^{i'-1} c(p_j, q_j) \\ & + \sum_{j=i_0}^{\pi(i_1)-1} c(p_{j+1}, q_j) + \sum_{j=\pi(i'_1)}^{i_0+k_0-1} c(p_{j+1}, q_j) \leq c(p_{i_0}, q_{i_0+k_0}) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j). \end{aligned} \quad (2.11)$$

Per l'ipotesi (1), $I_{\pi(i_1)-i_0}^p(i_0) \geq 0$, e $I_{i_0+k_0-i'_1}^p(i_1) \geq 0$, perciò

$$\begin{aligned} \sum_{j=i_0}^{\pi(i_1)} c(p_j, q_j) & \leq c(p_{i_0}, q_{\pi(i_1)}) + \sum_{j=i_0}^{\pi(i_1)-1} c(p_{j+1}, q_j) \\ \sum_{j=i_1}^{i_0+k_0} c(p_j, q_j) & \leq c(p_{i_1}, q_{i_0+k_0}) + \sum_{j=i_1}^{i_0+k_0-1} c(p_{j+1}, q_j). \end{aligned}$$

Grazie alle ultime due disequazioni e a 2.11, troviamo che

$$\sum_{j=i_0}^{i_0+k_0} \leq c(p_{i_0}, q_{i_0+k_0}) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j),$$

ovvero che $I_{k_0}^p(i_0)$ è negativo, contraddicendo l'ipotesi (3).

Caso b) Il caso b) e il caso c) sono analoghi, perciò trattiamo il solo caso b). Possiamo supporre che i_2 sia il più piccolo indice che soddisfa le ipotesi del caso b). Vediamo che esiste un unico i'_2 tale per cui $i_0 \leq \pi(i'_2) \leq i_2 - 1$ e $i'_2 \leq i_0$. Infatti

- ne esiste almeno uno: $|Q \cap (p_{i_0}, p_{i_2})| = |P \cap (p_{i_0}, p_{i_2})| + 1$, perciò almeno un punto q_i con $i_0 \leq i < i_2$ è accoppiato ad un punto p_j con $j \notin (i_0, i_2)$. Per la non-crossing rule dev'essere $j \leq i_0$;

- ne esiste al più uno: se esistessero $i'_2 \neq i''_2$ come sopra, per la non-crossing rule ci sarebbe almeno un punto p_i con $i_0 + 1 \leq i \leq i_2 - 1$ accoppiato ad un punto q_j con $j \leq i_0 - 1$. Ma allora $i < i_2$ soddisfa b), contraddicendo che i_2 sia l'indice minore che soddisfa tali ipotesi.

Sempre per la non-crossing rule allora, gli insiemi (possibilmente vuoti) $\{p_i | i_0 + 1 \leq i \leq \pi(i'_2)\} \cup \{q_i | i_0 \leq i \leq \pi(i'_2) - 1\}$ e $\{p_i | \pi(i_2) + 1 \leq i \leq i'_2 - 1\} \cup \{q_i | \pi(i'_2) + 1 \leq i \leq i'_2 - 1\}$ sono preservati.

Per la dimostrazione già fatta, ogni punto di questi due insiemi essere per forza accoppiati con il suo vicino, ovvero $\pi(i) = i - 1$ per $i_0 + 1 \leq i \leq \pi(i'_2)$ e $\pi(i) = i$ per $\pi(i'_2) + 1 \leq i \leq i'_2 - 1$. Poichè abbiamo assunto l'accoppiamento ottimale, si ha

$$\begin{aligned} c(p_{i_2}, q_{\pi(i_2)}) + c(p_{i'_2}, q_{\pi(i'_2)}) + \sum_{j=i_0+1}^{\pi(i'_2)} c(p_j, q_{j+1}) + \sum_{j=\pi(i'_2)+1}^{i_2-1} c(p_j, q_j) \\ \leq c(p_{i'_2}, q_{\pi(i_2)}) + \sum_{j=i_0+1}^{i_2} c(p_j, q_{j-1}) \end{aligned}$$

Applicando ancora il Lemma 2 si ottiene anche che

$$\begin{aligned} c(p_{i_2}, q_{\pi(i_2)}) + c(p_{i_0}, q_{\pi(i'_2)}) + \sum_{j=i_0+1}^{\pi(i'_2)} c(p_j, q_{j+1}) + \sum_{j=\pi(i'_2)+1}^{i_2-1} c(p_j, q_j) \\ \leq c(p_{i_0}, q_{\pi(i_2)}) + \sum_{j=i_0+1}^{i_2} c(p_j, q_{j-1}). \quad (2.12) \end{aligned}$$

Inoltre si ha che $|p_{i_0} - q_{\pi(i_2)}| \leq |p_{i_2} - q_{\pi(i_2)}|$ e quindi $c(p_{i_0}, q_{\pi(i_2)}) \leq c(p_{i_2}, q_{\pi(i_2)})$, dunque (2.12) implica che

$$c(p_{i_0}, q_{\pi(i'_2)}) + \sum_{j=i_0+1}^{\pi(i'_2)} c(p_j, q_{j+1}) + \sum_{j=\pi(i'_2)+1}^{i_2-1} c(p_j, q_j) \leq \sum_{j=i_0+1}^{i_2} c(p_j, q_{j-1}).$$

Completando la sommatoria a destra otteniamo

$$\begin{aligned}
 c(p_{i_0}, q_{\pi(i'_2)}) + \sum_{j=i_0+1}^{\pi(i'_2)} c(p_j, q_{j+1}) + \sum_{j=\pi(i'_2)+1}^{i_2-1} c(p_j, q_j) + \sum_{j=i_2+1}^{i_0+k_0} c(p_j, q_{j-1}) \\
 \leq \sum_{j=i_0+1}^{i_0+k_0} c(p_j, q_{j-1}). \quad (2.13)
 \end{aligned}$$

Per l'ipotesi (1), $I_{\pi(i'_2)-i_0}^p(i_0) \geq 0$, cioè

$$\sum_{j=i_0}^{\pi(i'_2)} c(p_j, q_j) \leq c(p_{i_0}, q_{\pi(i'_2)}) + \sum_{j=i_0}^{\pi(i'_2)-1} c(p_{j+1}, q_j).$$

Unendo l'ultima disuguaglianza e (2.12), si ottiene

$$\sum_{\substack{j=0 \\ j \neq i_2}}^{i_0+k_0} c(p_j, q_j) \leq \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j)$$

. Infine notando che $c(p_{i_2}, q_{i_2}) \leq c(p_{i_0}, q_{i_0+k_0})$ si ottiene

$$\sum_{j=i_0}^{i_0+k_0} c(p_j, q_j) \leq c(p_{i_0}, q_{\pi(i_0+k_0)}) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j),$$

che ancora una volta contradduce l'ipotesi (3).

Abbiamo dunque dimostrato che $\mathcal{S}_{i_0}^{k_0}$ è preservata da ogni assegnamento ottimale. Come abbiamo già visto all'inizio della sezione, ciò implica che i punti in $\mathcal{S}_{i_0}^{k_0}$ sono accoppiati come nel Teorema 1. \square

2.2.3 Dimostrazione del caso non bilanciato

Vediamo ora che il Teorema 1 rimane valido nel caso di catene non bilanciate. Ci concentreremo sul caso $I_{k_0}^p(i_0) < 0$. Dividiamo la dimostrazione in due parti: dimostriamo prima che i punti interni vengono tutti assegnati a qualche punto. Poi vediamo che p_{i_0} viene anch'esso assegnato. Una volta verificato che valgano queste due condizioni possiamo applicare lo stesso ragionamento fatto sul caso bilanciato anche al caso non bilanciato, e concludere la dimostrazione del teorema

I punti interni vengono tutti assegnati: Supponiamo, per assurdo, che esista $l \in \{i_0 + 1, \dots, i_0 + k_0\}$ tale che p_l sia il punto che rimane non assegnato in un assegnamento ottimale m . Ancora una volta per le ipotesi (1) e (2) tutti i punti in $\mathcal{S}_{i_0}^{k_0}$ se sono assegnati, lo sono con uno dei loro vicini oppure con punti fuori da $\mathcal{S}_{i_0}^{k_0}$. Per il Lemma 5,

$$\sum_{j=i_0}^{l-1} c(p_j, q_j) > \sum_{j=i_0}^{l-1} c(p_{j+1}, q_j), \quad (2.14)$$

perciò non può essere $m(p_j) = j$ per ogni $j = i_0, \dots, l$, altrimenti potremmo ridurre il costo con un assegnamento m' tale che $m'(p_{j+1}) = q_j$. Esistono allora $i_1 \notin \{i_0, \dots, l-1\}$ e $i_2 \in \{i_0, \dots, l-1\}$ tale per cui $\pi(i_1) = i_2$. Inoltre per il Corollario 1, p_l è esposto, perciò dev'essere $i_1 < i_0$. Scegliendo i_2 come il più grande indice che soddisfi tale proprietà, siccome non ci possono essere assegnamenti annidati, $\pi(j) = j$ per ogni j nell'intervallo (possibilmente vuoto) $i_2 + 1 \leq j \leq l-1$. Poiché g è crescente si ha

$$\begin{aligned} c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{l-1} c(p_j, q_j) &> c(p_{i_0}, q_{i_2}) + \sum_{j=i_2+1}^{l-1} c(p_j, q_j) \\ &= c(p_{i_0}, q_{i_2}) + \sum_{j=i_0}^{l-1} c(p_j, q_j) - \sum_{j=i_0}^{i_2} c(p_j, q_j) \end{aligned}$$

Usando di nuovo la disuguaglianza (2.14), si deduce dall'ultima uguaglianza che

$$\begin{aligned} c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{l-1} c(p_j, q_j) &> c(p_{i_0}, q_{i_2}) + \sum_{j=i_0}^{l-1} c(p_{j+1}, q_j) + \sum_{j=i_0}^{i_2} c(p_j, q_j) = \\ &= c(p_{i_0}, q_{i_2}) + \sum_{j=i_0}^{i_2-1} c(p_{j+1}, q_j) + \sum_{j=i_2}^{l-1} c(p_{j+1}, q_j) - \sum_{j=i_0}^{i_2} c(p_j, q_j) \quad (2.15) \end{aligned}$$

Osserviamo ora che $I_{i_0+i_2-1}^p(i_0) \geq 0$ perciò

$$c(p_{i_0}, q_{i_2}) + \sum_{j=i_0}^{i_2-1} c(p_{j+1}, q_j) - \sum_{j=i_0}^{i_2} c(p_j, q_j) \geq 0,$$

da cui si ottiene, unendo con la disuguaglianza precedente, che

$$c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{l-1} c(p_j, q_j) > \sum_{j=i_2}^{l-1} c(p_{j+1}, q_j),$$

ovvero è meno costoso assegnare i p_j al loro vicino destro e escludere p_{i_1} che assegnare p_i con q_i , q_{i_2} con p_{i_1} ed escludere p_l . In ogni caso p_l non può rimanere non assegnato.

p_{i_0} viene assegnato Assumiamo che p_{i_0} sia il punto in P non assegnato. Se $\pi(j) = j - 1$ per ogni $j = i_0 + 1, \dots, i_0 + k_0$, la tesi del Teorema 1 è soddisfatta e non c'è nulla da dimostrare. Altrimenti sia $i_2 \in \{i_0, \dots, i_0 + k_0 - 1\}$ il più grande indice tale che esiste $i_1 > i_0 + k_0$ con $\pi(i_1) = i_2$. Osserviamo che i_2 esiste perché i_0 è esposto e quindi non può essere $\pi(i'_1) = i_2$ con $i'_1 < i_0$.

In realtà si ha che $i_2 < i_0 - k_0 - 1$: supponiamo al contrario che sia $i_2 = i_0 + k_0 - 1$, allora per le ipotesi del teorema e per la variante della regola dei tre (Lemma 4) si ha che $|p_{i_0+k_0} - q_{i_0-k_0-1}| < |p_{i_0+k_0} - q_{i_0-k_0}|$. D'altra parte l'assegnamento è ottimale, perciò la regola dei tre (Lemma 3) implica che $|p_{i_0+k_0} - q_{i_0+k_0-1}| > |p_{i_0+k_0} - q_{\pi(i_0+k_0)}|$. Per la non-crossing rule $\pi(i_0 + k_0) < i_0 + k_0$, da cui $|p_{i_0+k_0} - q_{i_0-k_0-1}| > |p_{i_0+k_0} - q_{i_0-k_0}|$.

Perciò possono accadere due casi:

1. O $\pi(j) = j$ per ogni $j \in \{i_2 + 1, \dots, i_0 + k_0\}$;
2. Oppure esiste un unico i_3 con $i_3 \in \{i_2 + 1, \dots, i_0 + k_0\}$ tale che $\pi(i_3) = i_4 > i_0 + k_0$. Osserviamo che tale i_3 è unico, altrimenti esisterebbe un altro i'_2 con $i'_2 \in \{i_2 + 1, \dots, i_0 + k_0\}$ tale che $\pi^{-1}(i'_2) > i_0 + k_0$, ma abbiamo supposto che i_2 sia il più grande indice con tale proprietà.

Nel primo caso per l'equazione (2.8), si ha che

$$\begin{aligned} c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{i_0+k_0} c(p_j, q_j) &> c(p_{i_1}, q_{i_2}) + \sum_{j=i_2}^{i_0+k_0-1} c(p_{j+1}, q_j) \\ &> c(p_{i_1}, q_{i_0+k_0}) + \sum_{j=i_2}^{i_0+k_0-1} c(p_{j+1}, q_j). \end{aligned}$$

Ciò contraddice l'ottimalità di m .

Nel secondo caso, poiché $I_{k_0}^p(i_0) < 0$,

$$\begin{aligned} c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{i_3-1} c(p_j, q_j) + c(p_{i_3}, q_{i_4}) &> c(p_{i_1}, q_{i_2}) + c(p_{i_3}, q_{i_4}) \\ &+ c(p_{i_0}, q_{i_0+k_0}) - \sum_{j=i_0}^{i_2} c(p_j, q_j) + \sum_{j=i_0}^{i_0+k_0-1} c(p_{j+1}, q_j) - \sum_{j=i_3}^{i_0+k_0} c(p_j, q_j), \end{aligned} \quad (2.16)$$

notiamo infatti che il termine a destra della disequazione è uguale al termine a sinistra sommato a $I_{k_0}^p(i_0) < 0$.

Inoltre si ha $I_{i_2-i_0}^p(i_0) \geq 0$ e $I_{i_0+k_0-i_3}^p(i_3) \geq 0$, perciò sottraendo i due indicatori al termine a destra

$$\begin{aligned} c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{i_3-1} c(p_j, q_j) + c(p_{i_3}, q_{i_4}) &> c(p_{i_1}, q_{i_2}) \\ &+ c(p_{i_3}, q_{i_4}) - c(p_{i_3}, q_{i_0+k_0}) + c(p_{i_0}, q_{i_0+k_0}) - c(p_{i_0}, q_{i_2}) + \sum_{j=i_2}^{i_3-1} c(p_{j+1}, q_j). \end{aligned} \quad (2.17)$$

Le due differenze che appaiono a destra della disuguaglianza sono entrambe positive, inoltre $c(p_{i_1}, q_{i_2}) \geq c(p_{i_1}, q_{i_4})$, perciò abbiamo ottenuto che

$$\begin{aligned} c(p_{i_1}, q_{i_2}) + \sum_{j=i_2+1}^{i_3-1} c(p_j, q_j) + c(p_{i_3}, q_{i_4}) &> c(p_{i_1}, q_{i_4}) \\ &+ \sum_{j=i_2}^{i_3-1} c(p_{j+1}, q_j). \end{aligned}$$

Il caso $I_{k_0}^q(i_0)$ vale per simmetria. \square

2.2.4 L'algoritmo e il caso generale

Dal Teorema appena dimostrato possiamo ricavare un semplice algoritmo che ci restituisce un assegnamento ottimale nel caso di catene bilanciate. L'algoritmo per catene non bilanciate è identico, cambiando opportunamente gli indici di i e k .

Algoritmo 2 Assegnamento su catene

Sia $k = 1$, C la catena, $N = \frac{|C|}{2}$
while $C \neq \emptyset$ e $k < N$ **do**
 calcola gli indicatori di ordine k per ogni $i = 1, \dots, N - k$
 if \exists indicatori negativi **then**
 - Assegna i punti corrispondenti come nel teorema
 - Rimuovi i punti appena assegnati dalla catena
 - Sia C' la nuova catena
 - Ricomincia con $k = 1$, $C = C'$, $N = \frac{|C'|}{2}$
 else
 $k = k + 1$
 end if
end while
if $k = N$ **then** ci sono punti non ancora assegnati
 assegna ogni punto p con il punto q successivo
end if

Osservazione 5. Il teorema ci dà condizioni per accoppiare punti vicini, perciò appena trovati dei punti che possiamo accoppiare, li togliamo dalla catena considerata e ricominciamo da capo con una nuova catena più piccola

Osservazione 6. Ogni volta che troviamo un indicatore negativo siamo sicuri che le ipotesi del Teorema 1 sono soddisfatte: infatti poiché calcoliamo gli indicatori in ordine crescente, se stiamo calcolando un indicatore di ordine k su una data catena, abbiamo già calcolato tutti gli indicatori di ordine $k' < k$, e siamo sicuri che siano tutti positivi, altrimenti avremmo iniziato di nuovo l'iterazione da $k = 1$ con una catena più piccola.

Osservazione 7. Il costo per calcolare un indicatore di indice $k > 2$ è costante se conosciamo gli indicatori in ordine $k - 1$: infatti

$$I_k^p(i) = I_{k-1}^p(i) + c(p_i, q_{i+k}) + c(p_{i+k}, q_{i+k-1}) - c(p_i, q_{i+k-1}) - c(p_{i+k}, q_{i+k})$$

e

$$I_k^q(i) = I_{k-1}^q(i) + c(p_{i+k+1}, q_i) + c(p_{i+k}, q_{i+k}) - c(p_{i+k}, q_i) - c(p_{i+k+1}, q_{i+k})$$

perciò è conveniente calcolare dinamicamente gli indicatori. Questa osservazione può portare una riduzione fino ad un ordine di grandezza nella complessità dell'algoritmo.

Tornando al caso generale di X e $Y \in I$: abbiamo già visto nel Lemma 1 che un assegnamento ottimale assegna tra loro solo punti appartenenti allo stesso strato. Notiamo che tali strati formano sempre catene bilanciate se il problema è bilanciato, mentre se il problema è non bilanciato alcune catene saranno non bilanciate. Tali catene conterranno un punto appartenente a Y in più (ricordiamo che abbiamo assunto $|X| < |Y|$). Una volta formate le catene possiamo accoppiare i punti secondo l'algoritmo appena mostrato.

Capitolo 3

Analisi del caso di punti scelti casualmente

Analizziamo ora le proprietà dei due algoritmi nel caso di punti scelti casualmente in un intervallo $[0, L]$. In particolare possiamo scegliere nel caso bilanciato i $2N$ punti come selezionati con misura uniforme nell'intervallo $[0, 2N]$.

3.1 Costo medio dell'assegnamento di Dyck e dell'assegnamento ottimale

In [2] si dimostra che, scegliendo N punti in X e N punti in Y uniformemente in un intervallo $[0, 2N]$ oppure con intertempi dati da variabili aleatorie esponenziali di parametro 1, vale il seguente teorema per $N \rightarrow \infty$

Teorema 2. *Se $c(x, y) = |x - y|^t$, allora*

$$\mathbb{E}(c(m_{Dyck})) \simeq \begin{cases} N & \text{se } 0 < t < \frac{1}{2} \\ N \ln N & \text{se } t = \frac{1}{2} \\ N^{\frac{1}{2}+t} & \text{se } \frac{1}{2} < t \leq 1 \end{cases} \quad (3.1)$$

dove si intende che $a(N) \simeq b(N)$ se $\frac{a(N)}{b(N)} \rightarrow l \neq 0$.

Sempre in [2] si propone la seguente congettura:

Congettura 1. Chiamando m_{ott} l'assegnamento ottimale si ha che

$$\frac{\mathbb{E}(m_{ott})}{\mathbb{E}(m_{Dyck})} \rightarrow k_t,$$

con $0 < k_t < 1$ una costante dipendente da t

In [1] (Corollario 3.2) si dimostra la seguente stima

Proposizione 3. Mediamente il costo dell'assegnamento ottimale è

$$\mathbb{E}(m_{ott}) \leq \begin{cases} N & 0 < t < \frac{1}{2} \\ N\sqrt{\ln N} & t = \frac{1}{2} \\ N^{\frac{1}{2}+t} & \frac{1}{2} < t \leq 1 \end{cases} \quad (3.2)$$

Perciò la congettura va almeno corretta come segue

Congettura 2.

$$\frac{\mathbb{E}(m_{ott})}{\mathbb{E}(m_{Dyck})} \rightarrow k_t,$$

Con k_t che è una costante dipendente da t e $0 < k_t < 1$ se $p \neq \frac{1}{2}$, mentre $k_{\frac{1}{2}} = 0$.

3.2 Complessità degli algoritmi

Osserviamo subito che l'algoritmo per l'assegnamento di Dyck ha evidentemente complessità lineare nel numero di punti per qualsiasi istanza, infatti tutto ciò che facciamo è scorrere per una volta il vettore $\sigma = (\sigma_1, \dots, \sigma_{2N})$, assegnando mano a mano i punti. Questo algoritmo può essere perciò molto utile se vogliamo una stima del costo ottimale di un assegnamento per una certa istanza, anche alla luce di quanto visto nella sezione 3.1. La complessità dell'algoritmo esatto, invece è dipendente dall'istanza in input e dalla forma della funzione di costo: per esempio si può notare che per una funzione di costo $c(x, y) = |x - y|^p$, l'algoritmo sembra essere più efficiente per p vicino a 0, mentre è più lento per $p \rightarrow 1$. Per analizzare il comportamento dell'algoritmo esatto consideriamo prima il caso in cui sia applicato ad una catena di lunghezza $2n$. Passeremo poi ad un caso di assegnamento non bilanciato generale.

3.2.1 Complessità dell'algoritmo 2 su catene

Iniziamo ad analizzare l'algoritmo per l'assegnamento su catene. Assumiamo che la catena sia bilanciata, il caso non bilanciato è analogo. Possiamo stimare la complessità dell'algoritmo attraverso il numero di operazioni fatte per calcolare gli indicatori. In particolare prendiamo due casi opposti:

1. in un caso supponiamo di trovare sempre un indicatore di ordine 1;
2. nell'altro caso supponiamo di non trovare nessun indicatore negativo sulla catena di partenza, e di dove perciò calcolare $I_k^p(i)$ per ogni $1 \leq k < n$, $1 \leq i \leq n - k - 1$

Il primo caso è il più fortunato, poiché sono necessarie un numero lineare di valutazioni della funzione c e un numero lineare di addizioni per portare a termine l'algoritmo: infatti ogni punto della catena è rimosso attraverso indicatori di ordine 1, perciò per rimuovere ciascuna coppia di punti è necessario un numero costante di operazioni.

Il secondo caso è il caso pessimo: dobbiamo calcolare per ogni $k = 1, \dots, n - 1$ $n - k$ indicatori della forma $I_k^p(i)$ e $n - k - 1$ della forma $I_k^q(i)$, perciò il numero di operazioni totali diventa

$$\sum_{k=1}^{N-1} ((n - k) + (n - k - 1)) f(k)$$

dove $f(k)$ è il numero di operazioni necessarie per calcolare un indicatore di ordine k . Calcolare un indicatore di ordine k costerebbe un numero di operazioni lineare in k , ma per l'Osservazione 7, possiamo assumere $f(k)$ costante, calcolando gli indicatori di ordine k a partire da quelli di ordine $k - 1$. Perciò il costo totale diventa

$$\sum_{k=1}^{N-1} ((N - k) + (N - k - 1)) O(1) = O(n^2).$$

3.2.2 Algoritmo esatto nel caso generale

Vogliamo ora calcolare la complessità dell'algoritmo nel caso generale. Prendiamo un'istanza in generale non bilanciata, assegnando ad ognuno dei $2N$ punti casuali in $[0, 2N]$ una probabilità di $\frac{1}{2}$ di appartenere a X o a Y . In questo modo il cammino di Dyck può essere visto come una passeggiata aleatoria

che ha una probabilità di $\frac{1}{2}$ di fare un passo all'insù o all'ingiù. chiamiamo infatti $(Z_i)_{i=1}^{2N}$ le variabili tali che

$$Z_i = \begin{cases} 1 & \text{se } z_i \in X \\ -1 & \text{se } z_i \in Y. \end{cases}$$

Co questa notazione l'altezza raggiunta dal cammino dopo il passo n-esimo è $S_n = \sum_{i=1}^n Z_i$.

In questo modo si nota sperimentalmente che il numero di catene nel cammino è proporzionale a \sqrt{n} . Questa osservazione si può giustificare teoricamente usando la nota disuguaglianza massimale di Kolmogorov.

Proposizione 4. (Disuguaglianza massimale di Kolmogorov) *Siano $(Z_i)_{i=1}^{\infty}$ delle variabili aleatorie con media 0 e con momento secondo finito. Chiamiamo $S_n = \sum_{i=1}^n Z_i$. Vale allora*

$$\mathbb{P} \left(\max_{1 \leq k \leq n} |S_k| > \lambda \right) \leq \frac{\mathbb{E}[S_n^2]}{\lambda^2}$$

Nel nostro caso $\mathbb{E}[S_N^2] = \sum_{i=1}^N \mathbb{E}[Z_i^2] = N$, perciò si ha che per ogni $\epsilon > 0$ la probabilità che l'altezza catena esca da $\left[-\sqrt{N} \frac{1}{\sqrt{\epsilon}}, \sqrt{N} \frac{1}{\sqrt{\epsilon}}\right]$ è

$$\mathbb{P} \left(\max_{1 \leq k \leq N} |S_k| > \sqrt{N} \frac{1}{\sqrt{\epsilon}} \right) \leq \epsilon$$

Perciò per esempio con una probabilità del 90% l'altezza della catena non usciranno da $[-\sqrt{10N}, \sqrt{10N}]$ Inoltre possiamo vedere che le catene più lunghe si concentrano attorno all'altezza 0. In particolare nella Figura 4.6 vediamo la lunghezza media delle catene per ogni altezza. Si nota che le catene più lunghe hanno un numero di elementi dell'ordine di $\sqrt{2N}$. Chiamiamo infatti $\mathcal{N}_k = \mathcal{N}_k^{(2N)}$ il numero di elementi della catena ad altezza k con $2N \leq k \leq 2N$. Il nostro obiettivo è stimare \mathcal{N}_k per ogni k . Iniziamo con \mathcal{N}_0 .

Lemma 6. *Con la notazione appena definita, abbiamo che il numero medio di punti della catena ad altezza 0 è*

$$\mathbb{E}[\mathcal{N}_0] = (2N + 1) \binom{2N}{N} \left(\frac{1}{2}\right)^{2N} - 1.$$

CAPITOLO 3. ANALISI DEL CASO DI PUNTI SCELTI CASUALMENTE 32

Dimostrazione. Notiamo preliminarmente che la probabilità che per ogni N la passeggiata aleatoria si trovi ad altezza 0 dopo $2N$ passi è

$$P_{2N} = \binom{2N}{N} \left(\frac{1}{2}\right)^{2N}.$$

La dimostrazione è per induzione su N : Per $N = 0$, $\mathbb{E}[\mathcal{N}_0^{(0)}] = 0$.

Supponiamo ora che

$$\mathbb{E}[\mathcal{N}_0^{(2N-2)}] = (2N-1) \binom{2N-2}{N-1} \left(\frac{1}{2}\right)^{2N-2} - 1.$$

Allora

$$\begin{aligned} \mathbb{E}[\mathcal{N}_0^{(2N)}] &= \mathbb{E}[\mathcal{N}_0^{(2N-2)}] + P_{2N} \\ &= (2N-1) \binom{2(N-1)}{N-1} \left(\frac{1}{2}\right)^{2(N-1)} - 1 + \binom{2N}{N} \left(\frac{1}{2}\right)^{2N} \\ &= \binom{2N}{N} \left(\frac{1}{2}\right)^{2N} \left(1 + 2^2(2N-1) \frac{N^2}{(2N)(2N-1)}\right) - 1 \\ &= (2N+1) \binom{2N}{N} \left(\frac{1}{2}\right)^{2N} - 1. \end{aligned}$$

□

Abbiamo inoltre la seguente stima asintotica del coefficiente binomiale, che deriva dalla formula di Stirling:

$$\binom{2N}{N} \simeq \sqrt{\frac{1}{\pi N}} 2^{2N}.$$

Usando tale stima otteniamo che

$$\mathbb{E}[\mathcal{N}_0] \approx 2N \sqrt{\frac{1}{\pi N}} 2^{2N} \left(\frac{1}{2}\right)^{2N} = 2\sqrt{\frac{N}{\pi}}$$

Per la disuguaglianza di Markov, per ogni $\epsilon > 0$ esiste quindi una costante $\kappa(\epsilon)$ tale che

$$\mathbb{P}(\mathcal{N}_0 > \kappa(\epsilon)\sqrt{N}) \leq \epsilon. \quad (3.3)$$

Detta T_k la variabile aleatoria che esprime il tempo in cui $S_{T_k} = k$ per la prima volta, per la proprietà di Markov forte delle passeggiate aleatorie si ha che $\mathcal{N}_k^{(2N)}$ ha la stessa legge di $\mathcal{N}_0^{(2N-T_k)}$. Tale variabile aleatoria è chiaramente in media minore uguale a $\mathcal{N}_0^{(2N)}$. Perciò $\mathbb{E}[\mathcal{N}_k] \leq \mathbb{E}[\mathcal{N}_0]$.

Poiché il numero di operazioni eseguite dall'algoritmo su una catena di lunghezza ℓ è $O(\ell^2)$ al caso pessimo, si ha che per ogni $h \in \mathbb{Z}$ tale che lo strato \mathcal{C}_h è non vuoto, si ha che l'algoritmo esegue $O(\sum_h \mathcal{N}_h^2)$ operazioni.

Con probabilità maggiore o uguale di $1 - \epsilon$ si ha dunque che in media il costo totale dell'algoritmo è

$$\mathbb{E} \left[O \left(\sum_{h \leq \kappa(\epsilon)\sqrt{N}} \mathcal{N}_h^2 \right) \right] = O \left(\sum_{h \leq \kappa(\epsilon)\sqrt{N}} \mathbb{E}[\mathcal{N}_h^2] \right) \leq O \left(\kappa(\epsilon)\sqrt{N} \mathbb{E}[\mathcal{N}_0^2] \right). \quad (3.4)$$

Possiamo stimare anche $\mathbb{E}[\mathcal{N}_0^2]$ notando che

$$\begin{aligned} \mathbb{E}[\mathcal{N}_0^2] &= \mathbb{E} \left[\left(\sum_{k=1}^N I_{\{S_{2k}=0\}} \right)^2 \right] \\ &= \sum_{k=1}^N \mathbb{P}(S_{2k} = 0) + 2 \sum_{h < k} \mathbb{P}(S_{2h} = 0, S_{2k} = 0) \\ &\simeq \sum_{k=1}^N \binom{2k}{k} \frac{1}{2^{2k}} + 2 \sum_{h < k} \mathbb{P}(S_{2h} = 0, S_{2k} = 0). \end{aligned}$$

Per la formula di Stirling si ha che il primo termine è asintoticamente equivalente a $M \sum_{k=1}^N \sqrt{N} \approx M' \sqrt{N}$ per delle opportune costanti M e M' .

Notiamo che per la proprietà di Markov, per ogni $h, k \in \mathbb{Z}$ con $h < k$

$$\begin{aligned} \mathbb{P}(S_{2h} = 0, S_{2k} = 0) &= \mathbb{P}(S_{2h} = 0) \mathbb{P}(S_{2k} = 0 | S_{2h} = 0) \\ &= \mathbb{P}(S_{2h} = 0) \mathbb{P}(S_{2(k-h)} = 0) \\ &\simeq \frac{1}{\sqrt{h}} \frac{1}{\sqrt{k-h}}, \end{aligned}$$

dunque si ha che il secondo termine è

$$\begin{aligned}
 2 \sum_{h < k} \mathbb{P}(S_{2h} = 0, S_{2k} = 0) &\simeq 2 \sum_{h < k} \frac{1}{\sqrt{h}} \frac{1}{\sqrt{k-h}} \\
 &\leq 2 \int_0^N \frac{1}{\sqrt{y}} \int_y^N \frac{1}{\sqrt{x-y}} dx dy \\
 &= 4 \int_0^N \frac{\sqrt{N-y}}{\sqrt{y}} dy \\
 &\leq 4 \int_0^N \frac{\sqrt{N}}{\sqrt{y}} dy \leq 8N
 \end{aligned}$$

Unendo la disuguaglianza (3.3) con l'ultima stima, abbiamo ottenuto che con alta probabilità l'algoritmo eseguirà $O(N^{\frac{3}{2}})$ operazioni.

Osservazione 8. Nel caso fortunato in cui troviamo subito indicatori negativi (caso 1 della sezione 3.2.1) l'algoritmo eseguirà un numero di operazioni minore o uguale a $O(\sqrt{N}\mathbb{E}[\mathcal{N}_0]) = O(N)$. In pratica (Figura 4.4) questo caso si verifica per esempio quando $c(x, y) = |x - y|^t$ con t molto vicino a 0. In [3] si propone la seguente congettura per costi della forma $|x - y|^t$:

Congettura 3. *Se $I_k(i) < 0$ per una funzione di costo $c(x, y) = |x - y|^t$ con $0 < t \leq 1$, allora $I_k(i) < 0$ per ogni altra funzione di costo $c'(x, y) = |x - y|^{t'}$ con $0 < t' < t$.*

Nello stesso articolo si dà una dimostrazione per $k = 1$.

Questo algoritmo perciò rappresenta perciò un significativo miglioramento per questo caso particolare rispetto a un algoritmo generico per l'assegnamento come l'Algoritmo Ungherese che ha complessità in tempo $O(N^3)$.

Capitolo 4

Sperimentazioni numeriche

4.1 codici

Tutti gli algoritmi sono stati implementati in Matlab.

4.1.1 Algoritmo per l'assegnamento di Dyck

```
1 function [cost, matching] = dyckmatching(blue, red, p)
2 %--[matching, cost] = dyckmatch (blue , red, p)
3 %This function finds the Dyck matching of two arrays of the
4   same size k.
5 %blue and red are the two arrays, p is a real number
6   between 0 and 1
7 %matching is an array of size 2*k; matching(i)=j if in the
8   ordered array,
9 %i-th point is red and jth point is blue or viceversa,
10  and i is matched with j.
11 k=size(blue,2);
12 matching=zeros(1,2*k);
13 cost=0;
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 %costruisco l' array s di dim k+k che ordina i punti in
16 [0,1]
17 %costruisco l' array sign di dim k+k che mi dice se il
18 punto i e' rosso o blu
19 s=[blue, red];
20 sign=[ones(1,k), -ones(1,k)];
21 s = s + 1i*sign;
22 s=sort(s);
```

```

17 sign=imag(s);
18 s=real(s);
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 curr=k+1;
21 tree=zeros(1,2*k+1);
22 for j=1:2*k
23     sgn=sign(j);
24     if curr==k+1 %inizio di un'escursione positiva o
                %negativa;
25         % in questo caso j e' sempre ad altezza 0
26         tree(k+1)=j;
27         curr=curr+sgn;
28     else
29         % 4 casi a 2 a 2 simmetrici:
30         % escursione positiva e j blu <->escursione negativa e j
                %rosso;
31         % escursione negativa e j blu<-> escursione positiva e j
                %rosso;
32         % nel primi casi virtualmente si aggiunge un nuovo nodo
                %all'albero
33         if sign(j)*(curr-k-1)>0
34             tree(curr)=j;
35             curr=curr+sgn;
36
37         else
38
39             curr=curr+sgn;
40             cost=cost+ abs(s(j)-s(tree(curr)))^p;
41             matching(j)=tree(curr);
42             matching(tree(curr))=j;
43         end
44     end
45 end

```

4.1.2 Algoritmo per l'assegnamento su catene

```

1 function[cst, matching] = chainmatching(p, q, t)
2     N= length(p);
3     cst=0; % total cost
4     c = @(x,y) cost(x,y,t);
5     matching=zeros(1, N);
6     k=1;
7     up= p; % unmatched points in set P
8     uq= q; % unmatched points in set Q

```

```

9     lp=1:N;
10    lq=lp;
11    while not(isempty(up))&&not(isempty(uq)) && ( k<N )
12
13        %-----
14        % 1.) Compute k-order indicator for each i
15        %-----
16        if k==1
17            Ip=[];
18            Iq=[];
19
20            for i= 1: N-2
21                Ip(i) = indicator(up, uq, i, k, t, 'p');
22
23                Iq(i) =  indicator(up, uq, i, k, t, 'q');
24            end
25            Ip(N-1) = indicator(up, uq, N-k, k, t, 'p');
26
27            else %k>1
28                for i=1:N-k-1
29
30                    Ip(i) = Ip(i)+ c(up(i),uq(i+k)) + c(up(i+k),
31                    uq(i+k-1))...
32                    - c(up(i),uq(i+k-1)) - c(up(i+k)
33                    ,uq(i+k));
34
35                    Iq(i) = Iq(i)+ c(up(i+k+1),uq(i)) + c(up(i+k)
36                    ,uq(i+k))...
37                    - c(up(i+k),uq(i)) - c(up(i+k+1)
38                    ,uq(i+k));
39
40                end
41                Ip(N-k)=Ip(N-k)+c(up(N-k),uq(N))+c(up(N),uq(N-1))
42                ...
43                -c(up(N-k),uq(N-1))-c(up(N),uq(N));
44                Ip(end)=[];
45                Iq(end)=[];
46            end
47
48        %-----
49        % 2.) Find negative indicator
50        %-----
51        Negative_p = find(Ip<0);
52        Negative_q = find(Iq<0);

```

```

49 %-----
50 % If a negative indicator is found, match
51 % the points as in the theorem
52 % i.e. points "internal" to the indicator
53 %-----
54
55
56 del_index_p=[];
57 del_index_q=[];
58 if not(isempty(Negative_p))
59     for i=1:k
60         matching(lp(Negative_p+i)) = lq(Negative_p+i
-1) ;
61         del_index_p=[del_index_p Negative_p+i];
62         del_index_q=[del_index_q Negative_p+i-1];
63
64     end
65 end
66 if not(isempty(Negative_q))
67     for i=1:k
68         matching(lp(Negative_q+i))=lq(Negative_q+i);
69         del_index_p=[del_index_p Negative_q+i];
70         del_index_q=[del_index_q Negative_q+i];
71
72     end
73 end
74
75 lp(del_index_p)=[];
76
77 lq(del_index_q)=[];
78 up(del_index_p)=[];
79 uq(del_index_q)=[];
80
81 %-----
82 % if negative indicator were found in the previous
steps,
83 % Set N = length of the new chain and start over
with k=1.
84 % else
85 % Compute order k+1 indicators on the same chain.
86 %-----
87 if(not(isempty(Negative_p))||not(isempty(Negative_q))
)
88
89     k=1;

```

```

90         N=length(up);
91         else
92             k=k+1;
93         end
94     end
95     if(k==N)
96         for i=1:N
97             matching(lp(i))=lq(i);
98         end
99     end
100
101
102     %-----
103     % compute total cost
104     %-----
105     for i=1:length(p)
106         cst=cst+ abs(p(i)- q(matching(i)))^t;
107     end

```

4.1.3 Algoritmo esatto per l'assegnamento basato sulle catene

```

1 function [cst, matching]= chain_algorithm(blue, red, t)
2 % sorting blue and red points in a single array, keeping
3 % track of their
4 % type.
5 N=length(blue);
6 blue=reshape(blue, [N,1]);
7 red=reshape(red, [N,1]);
8 s = [blue;red];
9 signs = [ones(N,1); -ones(N,1)];
10 s = [s signs];
11 s= sortrows(s);
12 signs= s(:,2);
13 s= s(:,1);
14 % computing heighth of each point; same heighth equals same
15 % chain.
16 h=zeros(1, 2*N);
17 for i= 2:2*N
18     h(i)=h(i-1)+(1/2)*signs(i-1)+ (1/2)*signs(i);
19 end
20 chains = cell(2*N + 1,1); % used to store chains, entry j
21 %will contain

```

```

20                                     % chain at height j-N-1
21 chains_ind=cell(2*N +1,1);
22 match=cell(2*N+1,1);
23 for i = 1:2*N
24     chains{h(i) + N+1} (end+1) = s(i);
25     chains_ind{h(i) + N+1} (end+1)= i;
26 end
27
28 %compute matching cost for each chain
29 chains = chains(~cellfun('isempty', chains));
30 cst=0;
31 for i =1:length(chains)
32     p = chains{i}(1:2:end);
33     q = chains{i}(2:2:end);
34     [cs, match{i}]=chainmatching(p,q, t);
35     cst = cst + cs;
36 end

```

4.2 Risultati della sperimentazione

Esponiamo ora i risultati delle varie sperimentazioni eseguite con gli algoritmi esposti nel documento.

4.2.1 Esempi di assegnamenti

Nella figura 4.1 sono riportati degli esempi di assegnamenti ottenuti con l'algoritmo basato sulle catene. In particolare per la stessa istanza bilanciata con 25 punti blu e 25 punti rossi, abbiamo preso 3 diversi assegnamenti ottimali dipendenti dalla funzione di costo $c(x, y) = |x - y|^t$ con $t = 10^{-3}, \frac{1}{2}, 1 - 10^{-3}$, e l'assegnamento di Dyck (che ricordiamo non dipendere dalla funzione di costo). Si può notare come per $t = 10^{-3}$ ci siano archi molto lunghi, mentre per $t = 1 - 10^{-3}$ l'assegnamento ottimo equivale a quello di Dyck.

4.2.2 Andamento del costo dell'assegnamento di Dyck

Nella figura 4.2 sono riportati i costi medi dell'assegnamento di Dyck per una funzione di costo della forma $|x - y|^t$, con $t \in \{0.3, 0.5, 0.7\}$. Per ogni $N \in \{10.000, 20.000, \dots, 100.000\}$ e per ogni t abbiamo preso 500 istanze di punti generati casualmente in $[0, 2N]$ con probabilità uniforme e calcolato il costo medio per ogni N . I costi sono stati tracciati in funzione di $f_t(N)$ dove

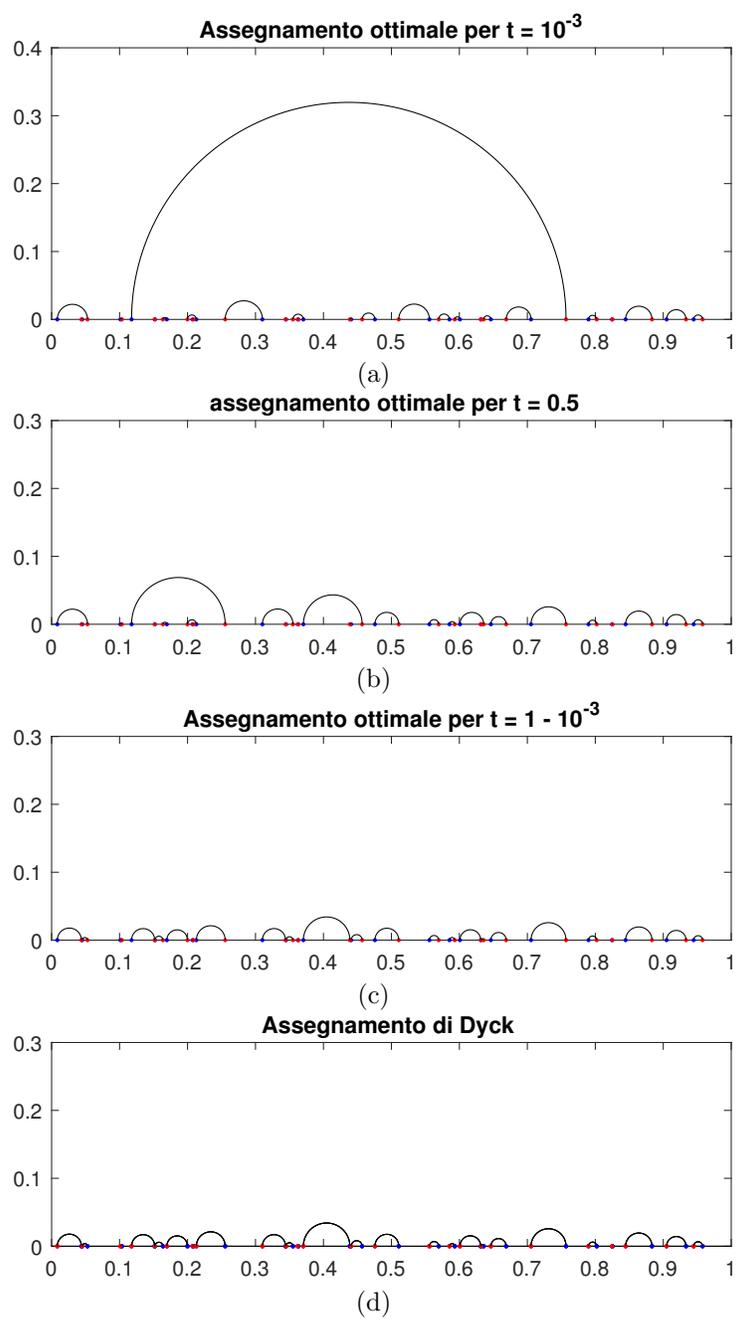
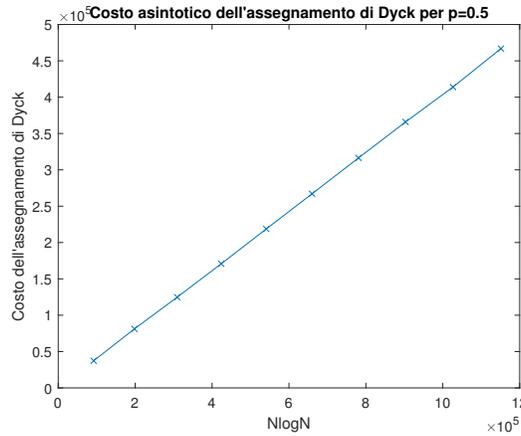
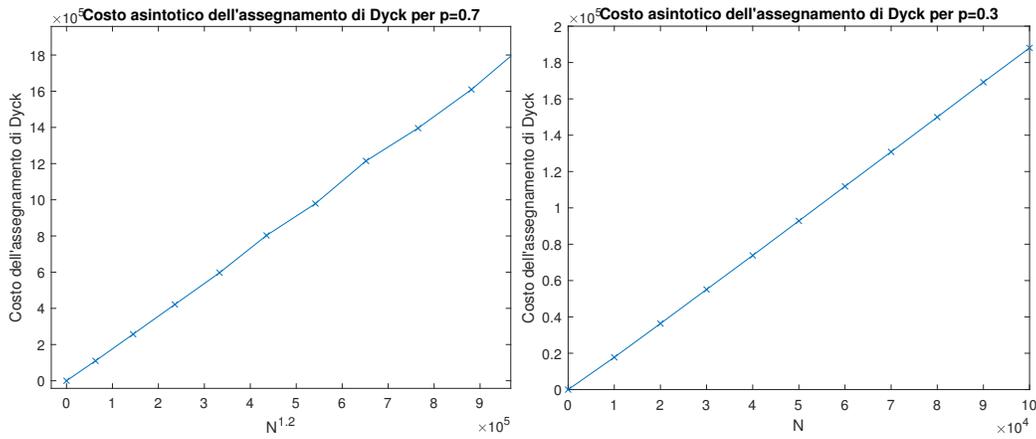


Figura 4.1

f_t è la funzione a cui i costi tendono asintoticamente a meno di una costante, come visto nel Teorema 2. In tutti i casi si vede come il grafico del costo dell'assegnamento in funzione di f_t sia molto simile a quello di una retta, confermando i risultati del Teorema 2.



(a) Costo assegnamento per $p = 0.5$



(b) Costo assegnamento per $p = 0.7$

(c) Costo assegnamento per $p = 0.3$

Figura 4.2: Costo dell'assegnamento di Dyck per vari valori di p , tracciati rispetto a $N \ln(N)$ per $p = \frac{1}{2}$, a N per $p < \frac{1}{2}$, ed a $N^{\frac{1}{2}+p}$ per $p > \frac{1}{2}$.

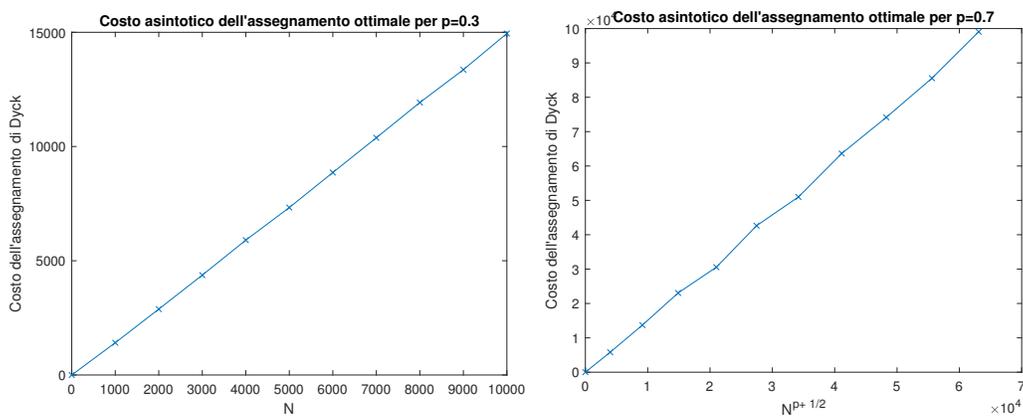
4.2.3 Andamento del costo dell'assegnamento ottimale

Nella figura 4.2 sono riportati i costi medi dell'assegnamento ottimale per una funzione di costo della forma $|x - y|^t$, con $t \in \{0.3, 0.5, 0.7\}$. In questo caso abbiamo preso $N \in \{1000, 2000, \dots, 10000\}$. Abbiamo preso 100 istanze per ogni valore di (t, N) . Abbiamo tracciato il grafico dei costi rispetto alle funzioni descritte nell'equazione (3.2); inoltre per $t = \frac{1}{2}$, abbiamo anche tracciato il grafico in funzione di $N \ln N$, come da Teorema 2 e Congettura 1. Si nota come $N \sqrt{\ln(N)}$ sia più vicino al costo dell'assegnamento, tuttavia il grafico sembra lineare anche in funzione di $N \ln N$, poiché la differenza tra le due funzioni è minima per questi ordini di grandezza.

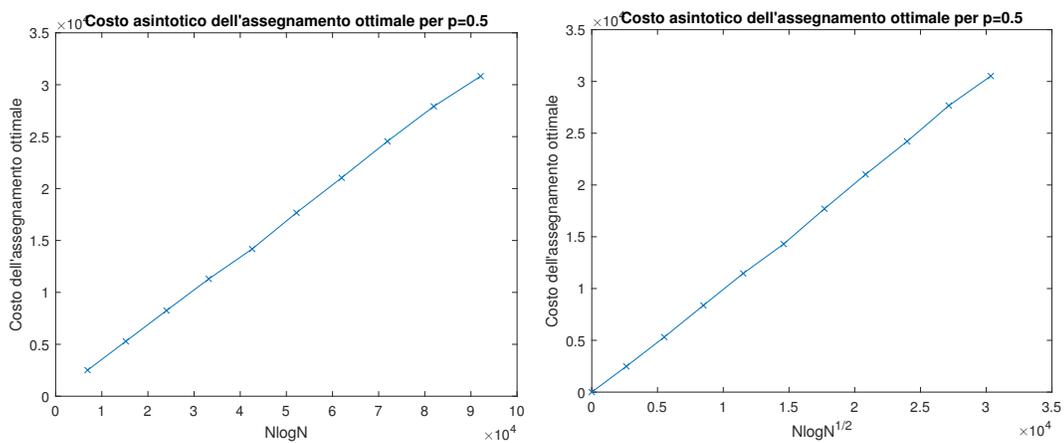
4.2.4 Tempo di esecuzione dell'algoritmo basato su catene

Nella figura 4.6 è riportata per ogni altezza h la lunghezza media della catena che forma lo strato \mathcal{C}_h . Tale media è stata fatta su 500 istanze bilanciate formata da $2N$ punti per $N = 100, 1000$. Come si può notare, per ogni catena, il numero di punti appartenenti alla catena è proporzionale a \sqrt{N} , e le catene con un numero di punti diverso da 0 sono anch'esse circa \sqrt{N} .

Nella figura 4.4 i riporta poi il tempo di esecuzione dell'algoritmo al variare della funzione di costo $|x - y|^t$ per vari valori di t . È evidente come per $t \ll 1$ il tempo di esecuzione sia pressoché lineare, mentre per t molto vicino a 1 si può notare come l'algoritmo impieghi tempo più che lineare, ma comunque non quadratico. Tali risultati rispettano in pieno le previsioni fatte nell'Osservazione 8 e nella Congettura 3: per $t \ll 1$ per ogni catena si trovano subito indicatori negativi per k piccolo, perciò per ogni catena l'algoritmo impiega un numero di operazioni lineare. Per $t \rightarrow 1^-$ invece gli indicatori negativi di ordine piccolo saranno molti meno. Si riporta a tale proposito nella Figura 4.5 un grafico che mostra al variare di $t \in (0, 1]$ l'ordine massimo k raggiunto dall'Algoritmo 2 prima di trovare un indicatore negativo, in una catena bilanciata di 100 punti scelti casualmente (in questo caso la catena non proviene da un insieme di livello di un'istanza generale).



(a) Costo assegnamento ottimale per $p = 0.3$ (b) Costo assegnamento ottimale per $p = 0.7$



(c) Costo assegnamento ottimale per $p = 0.5$, tracciato in funzione di $N \ln(N)$ (Congettura 1) (d) Costo assegnamento ottimale per $p = 0.5$, tracciato in funzione di $N \sqrt{\ln(N)}$ (Come da 3.2)

Figura 4.3: Costi dell'assegnamento ottimale per vari valori di p .

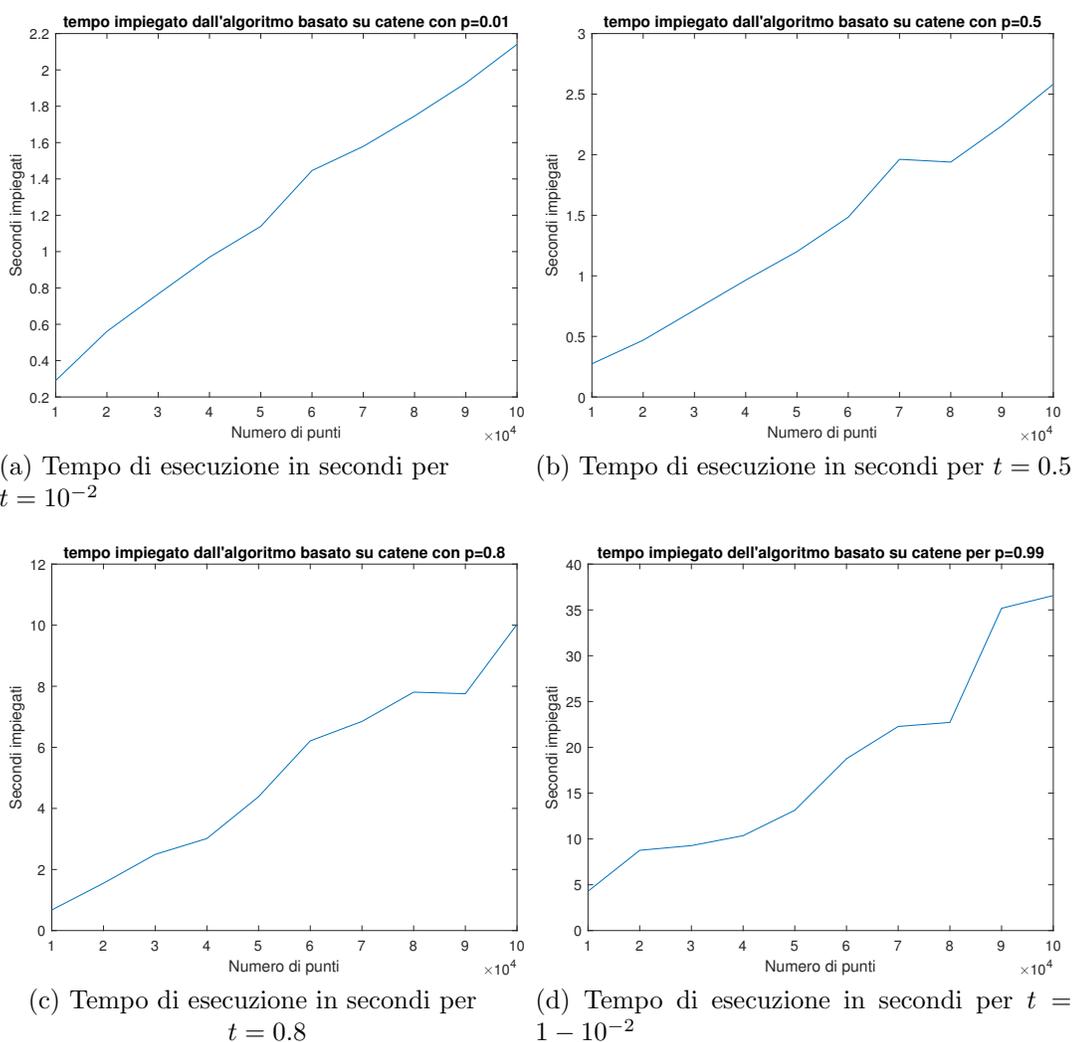


Figura 4.4: Tempo impiegato dall' algoritmo ottimale per vari valori di t .

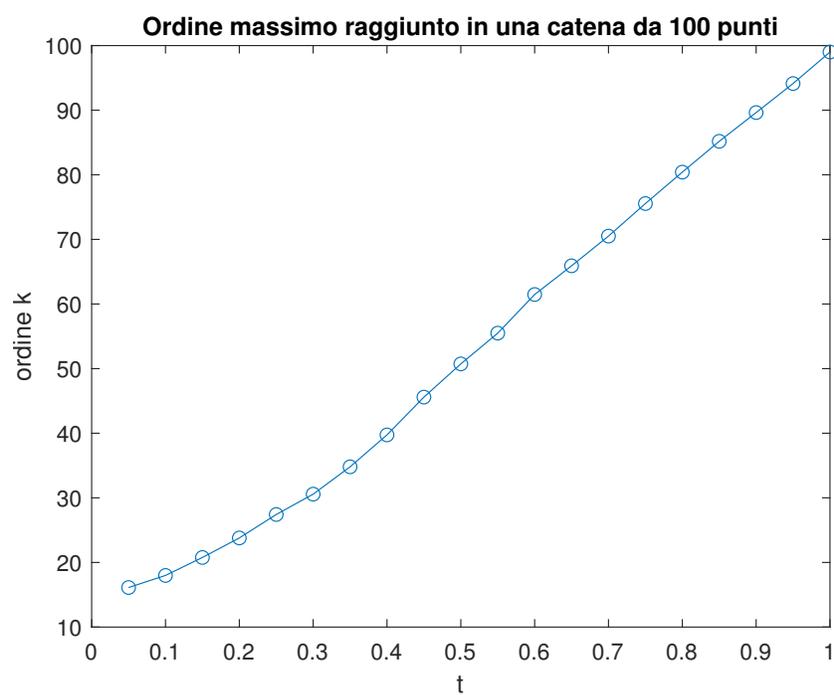
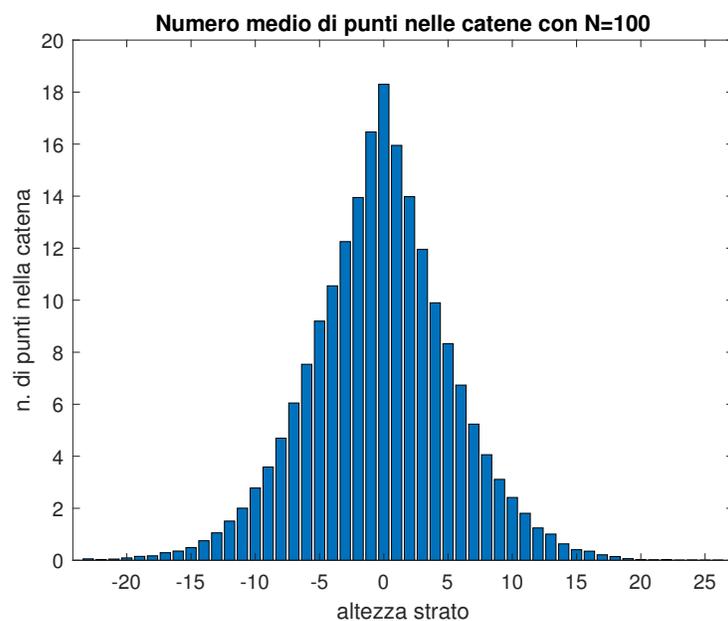
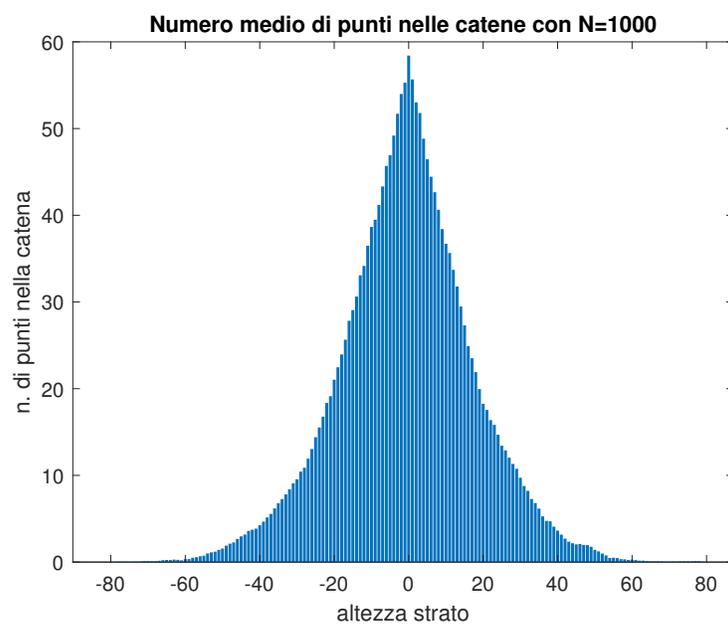


Figura 4.5: Ordine massimo raggiunto dall'algoritmo partendo da una catena di 100 punti prima di trovare un indicatore negativo



(a) Lunghezza media delle catene relative ad ogni strato data un'istanza casuale di $2N$ punti con $N = 100$



(b) Lunghezza media delle catene relative ad ogni strato data un'istanza casuale di $2N$ punti con $N = 1000$

Figura 4.6: Lunghezza media delle catene

Bibliografia

- [1] Sergey G Bobkov and Michel Ledoux. Transport inequalities on euclidean spaces for non-euclidean metrics. *Journal of Fourier Analysis and Applications*, 26(4):1–27, 2020.
- [2] S. Caracciolo, M. P. D’Achille, V. Erba, and A. Sportiello. The dyck bound in the concave 1-dimensional random assignment model. *Journal of Physics A: Mathematical and Theoretical*, 53(6):064001, 2020.
- [3] Julie Delon, Julien Salomon, and Andrei Sobolevski. Local matching indicators for transport problems with concave costs. *SIAM Journal on Discrete Mathematics*, 26(2):801–827, 2012.
- [4] Nicolas Juillet. On a solution to the monge transport problem on the real line arising from the strictly concave case. *SIAM Journal on Mathematical Analysis*, 52(5):4783–4805, 2020.
- [5] Robert J McCann. Exact solutions to the transportation problem on the line. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 455(1984):1341–1380, 1999.