

UNIVERSITÀ DEGLI STUDI DI TORINO  
DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN INFORMATICA



Laurea Magistrale

**Intelligenza Artificiale e Laboratorio**

Relazione dei Progetti di Laboratorio

Gabriele Picco - Stefano Locci

ANNO ACCADEMICO 2018/2019

# Indice

<b>1</b>	<b>Programmazione Logica</b>	<b>3</b>
1.1	Implementazione delle strategie di ricerca . . . . .	3
1.1.1	Strategie non informate . . . . .	3
1.1.1.1	Iterative Deepening . . . . .	3
1.1.2	Strategie con euristica . . . . .	3
1.1.2.1	A* . . . . .	4
1.1.2.2	IDA* . . . . .	4
1.2	Descrizione del dominio . . . . .	4
1.3	Funzioni Euristiche . . . . .	5
1.4	Confronto tra le strategie di ricerca . . . . .	6
1.5	Analisi delle prestazioni con costi non unitari . . . . .	7
1.6	Completezza e ottimalità delle strategie di ricerca . . . . .	8
1.6.1	Soluzione non appartenente allo spazio degli stati . . . . .	8
1.6.2	Più di una soluzione nello spazio degli stati . . . . .	8
<b>2</b>	<b>Answer Set Programming</b>	<b>9</b>
2.1	Problem encoding . . . . .	9
2.1.1	Gestione dell'orario prolungato con mensa della sezione A . . . . .	10
2.1.2	Integrity constraints . . . . .	11
2.2	Problem solution . . . . .	12
<b>3</b>	<b>Planning in CLIPS</b>	<b>16</b>
3.1	Descrizione del problema . . . . .	16
3.1.1	Dominio del problema . . . . .	16
3.2	Organizzazione dei moduli . . . . .	17
3.2.1	Modulo MAIN . . . . .	17
3.2.2	Modulo QUESTIONS . . . . .	17
3.2.3	Modulo ENRICH . . . . .	17
3.2.4	Modulo LOCATIONS . . . . .	18
3.2.5	Modulo GENERATE-PATH . . . . .	19
3.2.6	Modulo OPTIMIZE-PATH . . . . .	19
3.2.7	Modulo HOTEL . . . . .	20
3.2.8	Modulo TRIP . . . . .	21
3.2.9	Modulo TRIP-COST . . . . .	21
3.2.10	Modulo TRIP-SELECTION . . . . .	22
3.2.11	Modulo PRINT-RESULT . . . . .	23
3.2.12	Modulo FINAL-QUESTION . . . . .	23

3.3	Interazione tra i moduli . . . . .	24
3.4	Caso d'uso del sistema esperto . . . . .	25
<b>4</b>	<b>Incertezza</b>	<b>27</b>
4.1	MPE e MAP . . . . .	27
4.1.1	Most Probable Explanation (MPE) . . . . .	27
4.1.2	Maximum a Posteriori Hypotesis (MAP) . . . . .	27
4.1.3	Implementazione algoritmica . . . . .	28
4.1.3.1	Implementazione di MPE . . . . .	28
4.1.3.2	Implementazione MAP . . . . .	28
4.1.3.3	Correttezza delle implementazioni di MAP e MPE . . . . .	29
4.1.4	Analisi requisiti spaziali e temporali di MPE e MAP . . . . .	29
4.1.5	Analisi requisiti spaziali e temporali di MPE e MAP in relazione al numero di evidenze . . . . .	33
4.1.6	Analisi requisiti spaziali e temporali di MPE e MAP in relazione alla lontananza della rete da un polytree . . . . .	34
4.1.7	Analisi requisiti spaziali e temporali di MAP in relazione alle variabili di MAP selezionate . . . . .	37
4.2	Modelli probabilistici temporali: Kalman Filter . . . . .	39
4.2.1	Test . . . . .	39
4.2.2	Comportamento con modello di transizione non lineare . . . . .	46
4.2.3	Note . . . . .	48

# Capitolo 1

## Programmazione Logica

### 1.1 Implementazione delle strategie di ricerca

Di seguito vengono riportate le principali scelte implementative e le considerazioni relative agli algoritmi Iterative Deepening, A\*, IDA\* in linguaggio Prolog.

#### 1.1.1 Strategie non informate

Le strategie non informate cercano di esplorare in modo completo lo spazio degli stati nella ricerca della soluzione, mediante l'espansione combinatoria degli stati. Esse non utilizzano alcun tipo di conoscenza del problema al di là della definizione del problema stesso.

##### 1.1.1.1 Iterative Deepening

La strategia di ricerca Iterative Deepening sfrutta una ricerca in profondità limitata, incrementando il limite (inizialmente impostato a 1) iterativamente. L'implementazione in prolog è simile ad una ricerca in profondità limitata, con l'aggiunta di un predicato ausiliario per l'incremento della soglia in caso di fallimento. Viene inoltre gestito come caso particolare la situazione in cui la soluzione non appartenga allo spazio degli stati, come descritto nella sezione 6.1. Il predicato ausiliario, che si occupa dell'incremento iterativo della soglia, asserisce il fatto `maybe_solvable(false)` all'inizio del corpo della regola, prima del predicato ricorsivo per la DFS limitata. Il fatto `maybe_solvable(false)` viene rimosso e sostituito con `maybe_solvable(true)` dal primo ramo della DFS limitata che fallisce poiché limitato dalla soglia. E' infatti possibile che la soluzione si trovi a distanza maggiore della soglia, perciò ha senso proseguire la ricerca. In caso contrario, se cioè nessun fallimento è stato causato dal limite, l'intero spazio degli stati è stato esplorato. La ricerca perciò termina restituendo il fallimento.

#### 1.1.2 Strategie con euristica

Le strategie di ricerca con euristica, al contrario delle strategie non informate, sfruttano la conoscenza del dominio del problema, per guidare l'esplorazione dello spazio degli stati. L'algoritmo di ricerca informata è quindi dotato sia della funzione di ricerca che di una funzione di conoscenza (funzione euristica). La funzione di conoscenza determina l'ordine di ricerca dell'algoritmo al fine di ridurre il tempo necessario e/o il costo della ricerca.

### 1.1.2.1 A\*

L'algoritmo A\* basa l'esplorazione dello spazio degli stati su una funzione di stima corrispondente alla somma dell'euristica e il costo del cammino per arrivare allo stato S:  $F(S) = \text{Euristica}(S) + G(S)$ . La ricerca della soluzione utilizza la politica best-first, ovvero si espande sempre il nodo con  $F(S)$  minore. Tale strategia richiede che si memorizzi e si mantengano ordinati i nodi ed i relativi valori  $F(S)$  per tutti i nodi alternativi alla scelta corrente (frontiera). L'implementazione risulta perciò simile ad una BFS, ma tenendo in considerazione i valori  $F(S)$  per l'esplorazione. L'implementazione in prolog memorizza la frontiera in una lista di nodi, dove ogni nodo corrisponde ad una quadrupla ( $F(S)$ ,  $G$ , Stato, Azioni). La lista viene mantenuta ordinata dal predicato `build-in sort`, ad ogni espansione della frontiera. L'algoritmo inoltre deve considerare che non vengano espansi nodi i cui successori appartengono alla frontiera o siano già presenti nella soluzione parziale, per evitare situazioni cicliche.

### 1.1.2.2 IDA\*

L'algoritmo IDA\* è una variante di Iterative Deepening, che sfrutta l'idea di utilizzare una funzione stima per valutare il costo rimanente per raggiungere l'obiettivo, come in A\*. Poiché si tratta di un algoritmo di ricerca in profondità, l'utilizzo della memoria è inferiore rispetto ad A\*. Lo svantaggio dell'approccio è che lo stesso nodo può essere espanso più volte, poiché non viene memorizzata la frontiera come in A\*. L'implementazione in Prolog è molto simile a quella descritta precedentemente per Iterative Deepening, con l'unica differenza che la soglia che limita l'esplorazione è basata sulla funzione  $F(S)$  (la medesima di A\*) e la funzione ausiliaria che richiama la ricerca in profondità, dopo il fallimento, aggiorna la soglia basandosi sul minimo valore  $F(S)$  superiore alla soglia. Questo valore può essere facilmente ricavato asserendo un predicato `fvalue(F(S))` al momento del fallimento, nel caso in cui il predicato `fvalue(X)` abbia valore maggiore di  $F(S)$  o uguale alla soglia precedente. Basandosi inoltre sul fatto che la `FSoglia` è nota, prima di proseguire la ricerca impostando un nuovo valore di `FSoglia`, si può verificare che essi non corrispondano, evitando di ripetere una computazione uguale. Si può ricavare inoltre che tale condizione si verifica quando lo spazio degli stati è stato esplorato completamente, ma la soluzione non è stata trovata. In questo modo la non esistenza della soluzione risulterà in una esplorazione completa dello spazio degli stati ed un fallimento.

## 1.2 Descrizione del dominio

Il dominio di interesse simula la ricerca di un percorso, all'interno di un labirinto, da uno stato iniziale ad uno stato finale. Ogni casella  $C$  è rappresentata da un termine `pos(X,Y)` dove  $X$  e  $Y$  sono le coordinate spaziali. Gli ostacoli sono modellati da un termine `occupata(C)`. `finale(C)` e `iniziale(C)` rappresentano rispettivamente lo stato finale e iniziale. `num_colonne(CL)` e `num_righe(NR)` identificano la dimensione del labirinto. I predicati "applicabile", "trasforma" e "costo" determinano rispettivamente l'applicabilità delle azioni, le transizioni di stato e il costo delle azioni. Per effettuare la sperimentazione è stato implementato in Python un generatore algoritmico di labirinti, che costruisce un tabellone  $CL \times NR$  fornendo il codice prolog corrispondente e la relativa immagine.

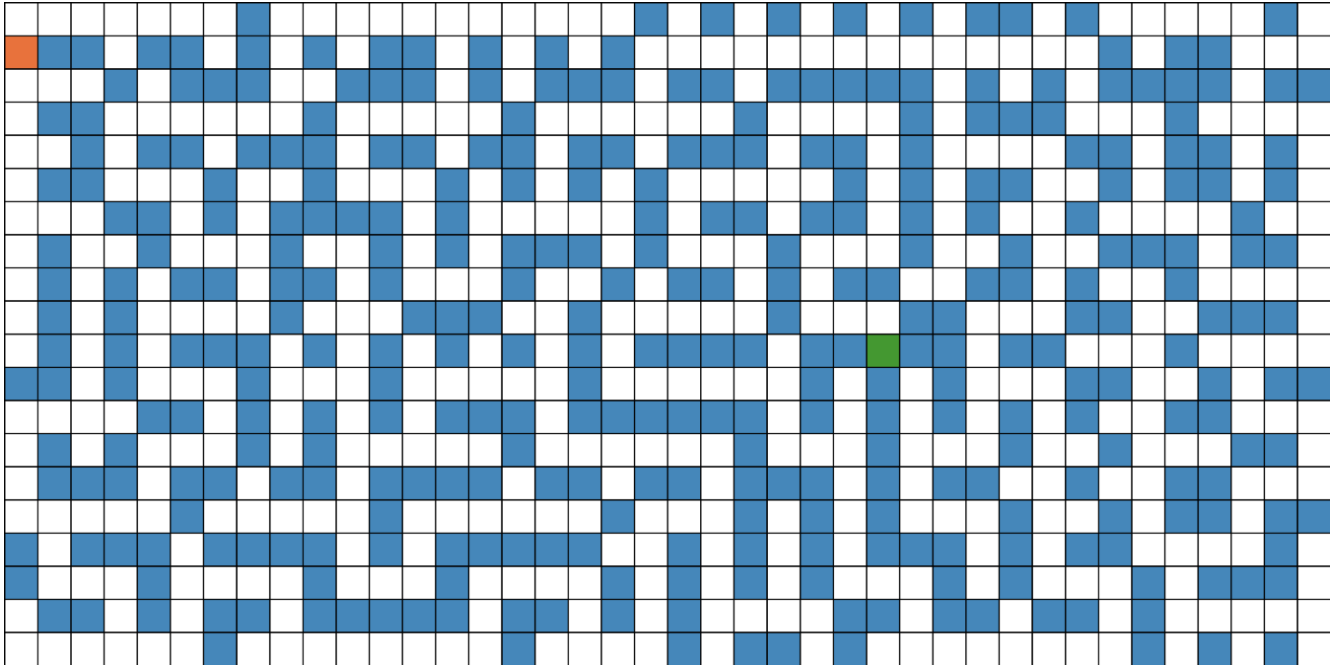


Figura 1.1: Esempio di labirinto 30 righe x 40 colonne generato automaticamente

### 1.3 Funzioni Euristiche

Una funzione euristica in una strategia di ricerca fornisce informazioni all'algoritmo (sfruttando la conoscenza del dominio) per ridurre il processo di ricerca della soluzione. Due proprietà importanti delle euristiche, che incidono sulla completezza e l'ottimalità degli algoritmi che le sfruttano sono ammissibilità e consistenza. L'ammissibilità asserisce che la funzione euristica non stimi mai per eccesso il costo per arrivare all'obiettivo:  $h(n) \leq h^*(n)$ . La consistenza di un euristica è una condizione più forte (consistente  $\Rightarrow$  ammissibile), che corrisponde alla monotonicità della funzione. Un'euristica è consistente se  $h(n) \leq c(n, a, n') + h(n')$ . Le euristiche possono essere costruite rilassando la definizione del problema, memorizzando in un database di pattern i costi precalcolati delle soluzioni, o apprendendo dall'esperienza. Di seguito vengono riportate 3 euristiche facilmente definibili nel dominio del labirinto, tutte consistenti.

*Distanza di Manhattan*: la distanza tra due punti è la somma del valore assoluto delle differenze delle loro coordinate.

$$D(C_1, C_2) = |x_1 - x_2| + |y_1 - y_2|$$

*Distanza Massima orizzontale o verticale (Max Distance)*:

$$D(C_1, C_2) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

*Distanza Euclidea*: misura del segmento avente per estremi i due punti approssimata all'intero più vicino.

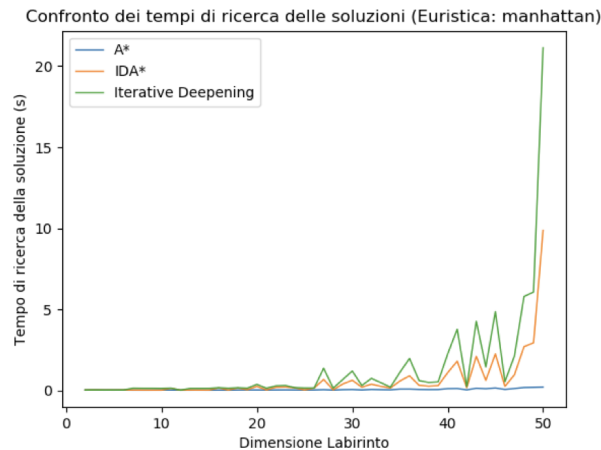
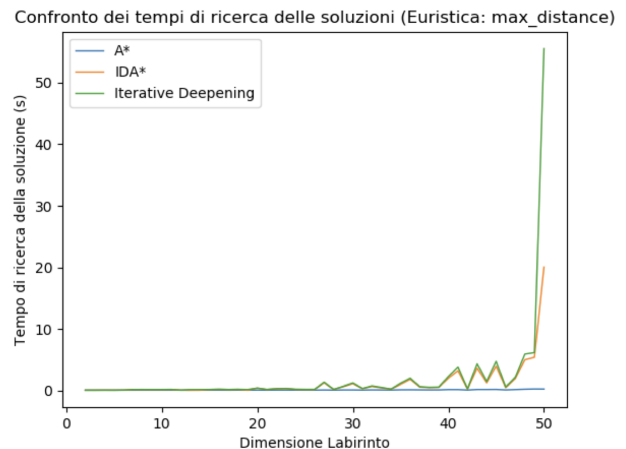
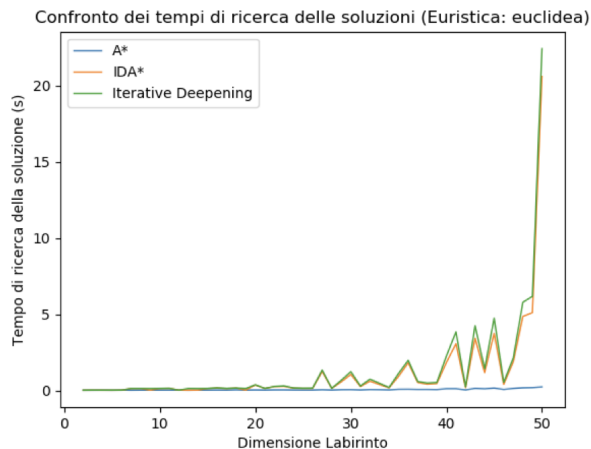
$$D(C_1, C_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Le euristiche definite verranno utilizzate per confermare le differenze di performance, poiché le prestazioni degli algoritmi di ricerca euristici dipendono dalla qualità delle funzioni euristiche.

## 1.4 Confronto tra le strategie di ricerca

Di seguito verranno riportati i risultati e le considerazioni derivanti dall'esecuzione dei tre algoritmi su una serie di labirinti generati automaticamente. In particolare verranno forniti i grafici relativi alla lunghezza delle soluzioni ed al tempo di ricerca.

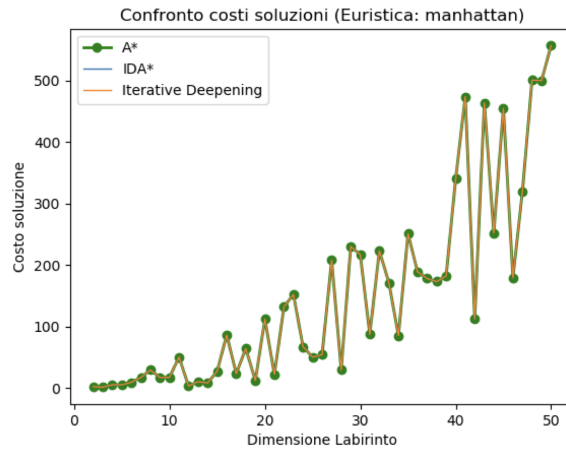
Considerando azioni con costo unitario e utilizzando le euristiche di Manhattan e Max Distance si ottengono i seguenti grafici:



Osservando i grafici si può osservare che A\*, come atteso, risulta sempre più performante rispetto a IDA\* e Iterative Deepening. IDA\* nel complesso è più veloce di Iterative Deepening. E' inoltre possibile osservare la qualità delle euristiche, ovvero quanto più si avvicinano all'euristica perfetta (costo reale per raggiungere l'obiettivo).

La distanza di Manhattan riduce considerevolmente i costi di ricerca. Dall'evidenza sperimentale si osserva che la distanza euclidea comporta tempi di ricerca leggermente superiori, mentre la max distanza (essendo sempre minore di entrambe le euristiche) è la meno efficace.

Il costo delle soluzioni trovate risulta coincidere per tutte le strategie di ricerca, poiché con azioni di costo unitario tutti gli algoritmi trovano la soluzione ottima.

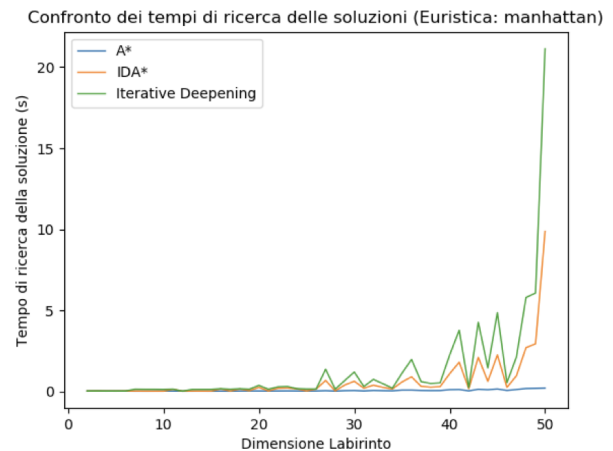
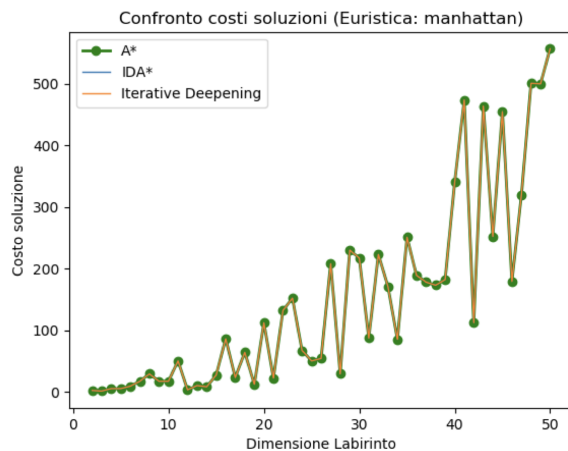


## 1.5 Analisi delle prestazioni con costi non unitari

Di seguito vengono analizzate le strategie di ricerca ipotizzando una situazione in cui le azioni hanno costo non unitario, ad esempio il costo delle azioni est e ovest potrebbe risultare maggiore rispetto alle azioni nord e sud a causa di un particolare servo sterzo. I fatti che rappresentano il costo delle azioni risulteranno quindi:

$$\text{costo}(\text{est}, 3) \cdot \text{costo}(\text{ovest}, 3) \cdot \text{costo}(\text{sud}, 1) \cdot \text{costo}(\text{nord}, 1).$$

Si riportano di seguito i grafici relativi alle prestazioni degli algoritmi e al costo delle soluzioni trovate:



Dai grafici si può osservare che le osservazioni precedenti rimangono valide e che tutte le strategie di ricerca trovano la soluzione ottima. La discussione sulla completezza e ottimalità degli algoritmi viene rimandata alla sezione successiva.



## 1.6 Completezza e ottimalità delle strategie di ricerca

Nell'implementazione utilizzata gli algoritmi  $A^*$  e  $IDA^*$  sono sia completi che ottimi, se l'euristica è consistente.

Iterative Deepening è completa, risulta inoltre ottima quando il costo del percorso è una funzione non-decrescente (monotona) della profondità del nodo. Per accorgersi delle differenze delle soluzioni trovate possiamo analizzare la seguente particolare situazione:

Le azioni nord e sud hanno costo negativo, in qualche modo ricaricano il robot (esempio pannelli solari), mentre le azioni est e ovest hanno costo 2.

$$\text{costo}(\text{est}, 2). \text{costo}(\text{ovest}, 2). \text{costo}(\text{sud}, -1). \text{costo}(\text{nord}, -1).$$

Formulando una funzione euristica ammissibile per il problema (solo per il caso particolare in figura 1.2):

$$H(S) = 2 * |x1 - x2| - |y1 - y2| - 2$$

gli algoritmi  $A^*$  e  $IDA^*$  individueranno la soluzione ottima *nord - est - est - sud* di costo 2, mentre l'algoritmo Iterative deepening troverà la soluzione *est - est* di costo 4. Bisogna notare che non essendo consistente l'euristica, non è garantito che  $A^*$  e  $IDA^*$  trovino la soluzione ottima, anche se questo succede nel particolare esempio presentato.

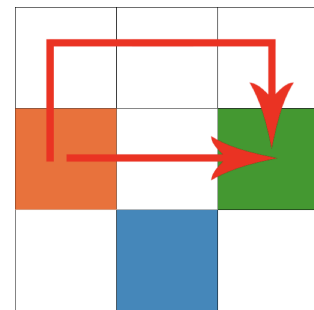


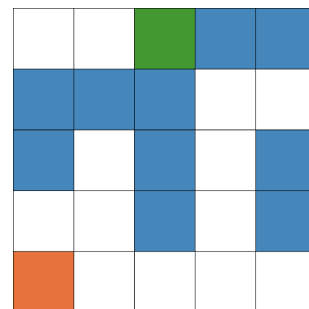
Figura 1.2

### 1.6.1 Soluzione non appartenente allo spazio degli stati

Come già illustrato nei paragrafi precedenti, tutte le implementazioni delle strategie di ricerca gestiscono il caso in cui la soluzione non appartenga allo spazio degli stati.

$A^*$  per definizione termina quando lo spazio degli stati è stato esplorato completamente.

$IDA^*$  e Iterative Deepening terminano quando le asserzioni individuano situazioni in cui la ricerca non può più esplorare nuovi percorsi per la ricerca della soluzione e ripeterebbe la computazione precedente. Iterative Deepening ha complessità spaziale  $O(bd)$ , dove  $b$  è il branching factor e  $d$  è la profondità della soluzione più vicina alla radice, nel caso peggiore quindi  $d$  corrisponde al path più lungo.  $IDA^*$  ravisiterà meno volte i nodi sui cammini a causa della funzione euristica, ma il caso pessimo risulta comunque computazionalmente proibitivo.



### 1.6.2 Più di una soluzione nello spazio degli stati

Nel caso in cui sia presente più di una soluzione nello spazio degli stati è possibile definire un predicato ausiliario che calcoli l'euristica per ogni stato finale, restituendo l'euristica minima per lo stato più promettente. In questo modo gli stati verranno valutati in relazione allo stato finale più promettente, indirizzando la ricerca della soluzione in questo senso.

# Capitolo 2

## Answer Set Programming

### 2.1 Problem encoding

Il problema richiedeva di comporre un orario scolastico di una scuola media rispettando determinati vincoli di ore per materia, numero di docenti e numero di ore per giorno, distinguendo le classi in due sezioni, Sezione A con tempo prolungato, ovvero con partecipazione ad attività extra-scolastiche e alla mensa; Sezione B con tempo normale composto dalle 6 ore al giorno per 5 giorni a settimana. Per questa fase di codifica del problema abbiamo seguito il paradigma Generate, Define and Test. In primo luogo si è deciso di esprimere i fatti relativi all'insegnamento di una o più materie da parte di un docente e all'assegnamento di una specifica aula per ogni materia nel seguente modo:

$$aula\_materia(matematica, aula\_matematica).$$
$$docente\_insegna(doc\_tecnologia, tecnologia).$$

Per quanto riguarda le ore di lezione per ogni materia sono stati utilizzati dei vincoli di cardinalità espressi mediante aggregati. In questo modo vengono generati degli insiemi di predicati "insegnamento\_ora(Classe, Giorno, Ora, Materia)" con cardinalità pari al numero di ore della materia da assegnare durante la settimana. Il corpo della regola esprime il fatto che il numero di ore di lezione della materia devono essere assegnate per ogni classe.

$$10\{insegnamento\_ora(Classe, G, O, lettere) : ora(O), giorno(G)\}10 \models classe(Classe).$$

Per rappresentare l'orario normale di lezione, formato da 6 ore al giorno per 5 giorni alla settimana, comune ad entrambe le sezioni, sono stati utilizzati degli aggregati di cardinalità i quali sono stati utilizzati in maniera congiunta per ottenere l'insieme di 30 ore settimanali per ogni classe.

$$5\{giorno\_lezione(Classe, Giorno) : giorno(Giorno)\}5 \models classe(Classe).$$

$$6\{ora\_lezione(Classe, Giorno, Ora) : ora(Ora)\}6 \models giorno\_lezione(Classe, Giorno).$$

A questo punto il predicato "ora\_lezione" è stato utilizzato per comporre l'orario scolastico tramite una regola la cui testa è composta da un aggregato che ha permesso di generare uno e un solo predicato "assegnamento\_ora(Classe, Giorno, Ora, Aula, Doc, Materia)" per ogni ora di lezione, di ogni giorno per ogni classe.

$$1\{assegnamento\_ora(C, G, O, A, D, M) : aula\_materia(M, A), docente\_insegna(D, M), \\ giorno\_lezione(C, G), insegnamento\_ora(C, G, O, M)\}1 \models \\ ora\_lezione(C, G, O).$$

In questo modo abbiamo ottenuto un orario completo per entrambe le sezioni A e B, in cui le variabili Materia e Aula vengono assegnate in modo tale da rispettare i fatti "aula\_materia" che assegnano ad ogni materia la propria aula; Doc e Materia fanno riferimento ai fatti "docente\_insegna" che assegnano ad ogni materia il proprio docente e Classe, Giorno, Ora e Materia devono essere assegnate rispettando i fatti espressi da "insegnamento\_ora" che stabilisce il numero di ore per ogni materia.

### 2.1.1 Gestione dell'orario prolungato con mensa della sezione A

Le ore di lezione extra-scolastiche per il corso A sono state gestite in modo separato rispetto all'orario normale comune per entrambe le sezioni. Si è deciso di generare, in modo analogo all'orario normale, una distribuzione di 2 ore extra al giorno per 5 giorni a settimana aggiungendo una pausa mensa tra le ore normali di lezione e quelle extra.

$$1\{assegnamento\_ora(C, G, O, A, D, M) : giorno\_lezione\_extra(C, G), aula(A), \\ docente\_insegna\_extra(D, M), insegnamento\_ora(C, G, O, M)\}1 \models \\ ora\_lezione\_extra(C, G, O).$$

$$5\{giorno\_lezione\_extra(C, G) : giorno(G)\}5 \models classe\_tempo\_prolungato(C).$$

$$2\{ora\_lezione\_extra(C, G, O) : ora\_extra(O)\}2 \models giorno\_lezione\_extra(C, G).$$

Dato che le ore extra di lezione si svolgono dopo le ore normali di lezione, si è assunto che esse possano essere svolte in un'aula qualunque anche se specifica per un'altra materia.

## 2.1.2 Integrity constraints

L'orario è stato modellato tenendo in considerazione i seguenti vincoli:

- Una classe non può essere in due aule diverse contemporaneamente
- Un'aula non può essere occupata da due classi diverse nello stesso giorno alla stessa ora
- Un docente non può essere in aule diverse o due classi diverse lo stesso giorno alla stessa ora
- Una classe non può avere due docenti diversi per la stessa materia
- Una classe non può avere più di 2 ore per giorno della stessa materia nello stesso giorno
- Per comodità del professore, se una stessa classe ha due ore di una materia in un giorno, le ore devono essere consecutive

Per i primi quattro vincoli si è specificato semplicemente quali configurazioni di "assegnamento\_ora" non possono essere vere contemporaneamente. In particolare, il vincolo "una classe non può avere due docenti diversi per la stessa materia" è stato pensato per dare uniformità agli insegnamenti che hanno più di un professore per la stessa materia, evitando quindi i casi in cui per una stessa materia in una stessa classe si debbano alternare professori diversi. Questo vincolo aumenta il tempo di calcolo delle soluzioni passando da un tempo inferiore ad un secondo a circa 3 minuti.

```
% una classe non può essere in due aule diverse contemporaneamente
:- assegnamento_ora(Classe, Giorno, Ora, Aula1, _, _), assegnamento_ora(Classe, Giorno, Ora, Aula2, _, _), Aula1 <> Aula2.

% % Un'aula non può essere occupata da due classi diverse nello stesso giorno alla stessa ora
:- assegnamento_ora(Classe1, Giorno, Ora, Aula, _, _), assegnamento_ora(Classe2, Giorno, Ora, Aula, _, _), Classe1 <> Classe2.

% Un docente non può essere in aule diverse o due classi diverse lo stesso giorno alla stessa ora
:- assegnamento_ora(Classe1, Giorno, Ora, _, Doc, _), assegnamento_ora(Classe2, Giorno, Ora, _, Doc, _), Classe1 <> Classe2.

% Una classe non può avere due docenti diversi per la stessa materia
:- assegnamento_ora(Classe, _, _, Doc1, Materia), assegnamento_ora(Classe, _, _, Doc2, Materia), Doc1 <> Doc2.
```

Per il vincolo "Una classe non può avere più di 2 ore per giorno della stessa materia nello stesso giorno" è stato utilizzato l'aggregato #count per contare il numero di ore di una determinata materia in un giorno. La somma delle ore viene limitata da integrity constraint a 2 al giorno.

```
% Una classe non può avere più di 2 ore per giorno della stessa materia
conteggio_ore_materia(Classe, Giorno, Materia, N) :-
    N = #count { (Classe, Giorno, Ora, Materia) : assegnamento_ora(Classe, Giorno, Ora, _, _, Materia) }
    , classe(Classe), giorno(Giorno), materia(Materia).

:-conteggio_ore_materia(Classe, Giorno, Materia, N), N > 2.
```

Il vincolo delle ore consecutive è stato realizzato considerando definendo una relazione temporale tra le ore, mediante i predicati "ora\_successiva" e "ora\_precedente". In questo modo è stato possibile definire un integrity constraint che impedisse il verificarsi di due ore di una stessa materia in uno stesso giorno non consecutive. Questo vincolo, sebbene renda la soluzione più realistica ed applicabile, aumenta considerevolmente i tempi di esecuzione (passando da circa 3 minuti a circa 8). Si è scelto di utilizzare un integrity constraint, invece di filtrare le soluzioni direttamente in fase di generazione dell'orario, perché in questo modo si ha a disposizione un vincolo generico per

contare il numero di ore in un giorno per una materia e quindi per organizzarne l'ordine durante la giornata in modo più immediato.

```
% Per comodità del professore, se una stessa classe ha due ore
% di una materia in un giorno, le ore devono essere consecutive
:-conteggio_ore_materia(Classe, Giorno, Materia, N), N = 2,
    assegnamento_ora(Classe, Giorno, Ora1, _, _, Materia), assegnamento_ora(Classe, Giorno, Ora2, _, _, Materia),
    not ora_precedente(Ora1,Ora2), not ora_successiva(Ora1,Ora2), Ora1 <> Ora2.
```

## 2.2 Problem solution

Per semplificare la rappresentazione dei risultati si è deciso di sviluppare un semplice script in Python che ha permesso di ottenere delle rappresentazioni leggibili, confrontabili e verificabili dei risultati.

Prima A		lunedì	martedì	mercoledì	giovedì	venerdì
prima_ora	Aula	aula lettere2	aula lettere2	aula lettere2	aula musica	lab arte
	Docente	doc lettere_2	doc lettere_2	doc lettere_2	doc musica	doc arte
	Materia	lettere	lettere	lettere	musica	arte
seconda_ora	Aula	aula lettere2	aula lettere1	aula lettere1	aula religione	lab scienze
	Docente	doc lettere_2	doc lettere_2	doc lettere_2	doc religione	doc matematica_2
	Materia	lettere	lettere	lettere	religione	scienze
terza_ora	Aula	aula inglese	lab arte	aula spagnolo	aula lettere1	aula lettere2
	Docente	doc inglese	doc arte	doc spagnolo	doc lettere_2	doc lettere_2
	Materia	inglese	arte	spagnolo	lettere	lettere
quarta_ora	Aula	aula tecnologia	aula matematica	aula tecnologia	aula lettere2	aula lettere1
	Docente	doc tecnologia	doc scienze_2	doc tecnologia	doc lettere_2	doc lettere_2
	Materia	tecnologia	matematica	tecnologia	lettere	lettere
quinta_ora	Aula	aula matematica	aula inglese	palestra	aula inglese	palestra
	Docente	doc scienze_2	doc inglese	doc ed fisica	doc inglese	doc ed fisica
	Materia	matematica	inglese	educazione fisica	inglese	educazione fisica
sesta_ora	Aula	lab scienze	aula spagnolo	aula matematica	aula matematica	aula musica
	Docente	doc matematica_2	doc spagnolo	doc scienze_2	doc scienze_2	doc musica
	Materia	scienze	spagnolo	matematica	matematica	musica
settima_ora	-	mensa	mensa	mensa	mensa	mensa
ottava_ora	Aula	aula lettere1	aula lettere2	palestra	lab scienze	aula lettere2
	Docente	doc matematica_2	doc matematica_2	doc matematica_2	doc ed civica	doc scienze_1
	Materia	recupero matematica	recupero matematica	recupero matematica	educazione civica	chimica
nona_ora	Aula	aula lettere2	aula lettere1	lab arte	lab arte	aula lettere2
	Docente	doc matematica_2	doc lab disegno	doc scienze_1	doc ed civica	doc lab disegno
	Materia	recupero matematica	lab disegno	chimica	educazione civica	lab disegno



Seconda A		lunedì	martedì	mercoledì	giovedì	venerdì
prima_ora	Aula	aula_spagnolo	aula_lettere1	aula_lettere1	aula_matematica	aula_spagnolo
	Docente	doc_spagnolo	doc_lettere_1	doc_lettere_1	doc_matematica_2	doc_spagnolo
	Materia	spagnolo	lettere	lettere	matematica	spagnolo
seconda_ora	Aula	lab_scienze	aula_lettere2	aula_lettere2	aula_tecnologia	aula_religione
	Docente	doc_matematica_2	doc_lettere_1	doc_lettere_1	doc_tecnologia	doc_religione
	Materia	scienze	lettere	lettere	tecnologia	religione
terza_ora	Aula	aula_matematica	palestra	aula_matematica	aula_lettere2	aula_lettere1
	Docente	doc_matematica_2	doc_ed_fisica	doc_matematica_2	doc_lettere_1	doc_lettere_1
	Materia	matematica	educazione_fisica	matematica	lettere	lettere
quarta_ora	Aula	aula_musica	aula_inglese	palestra	aula_lettere1	aula_lettere2
	Docente	doc_musica	doc_inglese	doc_ed_fisica	doc_lettere_1	doc_lettere_1
	Materia	musica	inglese	educazione_fisica	lettere	lettere
quinta_ora	Aula	aula_lettere2	lab_arte	aula_inglese	lab_arte	aula_matematica
	Docente	doc_lettere_1	doc_arte	doc_inglese	doc_arte	doc_matematica_2
	Materia	lettere	arte	inglese	arte	matematica
sesta_ora	Aula	aula_lettere1	aula_tecnologia	aula_inglese	aula_musica	lab_scienze
	Docente	doc_lettere_1	doc_tecnologia	doc_inglese	doc_musica	doc_matematica_2
	Materia	lettere	tecnologia	inglese	musica	scienze
settima_ora	- - -	mensa	mensa	mensa	mensa	mensa
ottava_ora	Aula	aula_lettere2	aula_musica	lab_scienze	aula_lettere1	lab_arte
	Docente	doc_matematica_1	doc_matematica_1	doc_matematica_1	doc_lab_disegno	doc_matematica_1
	Materia	recupero_matematica	recupero_matematica	recupero_matematica	lab_disegno	recupero_matematica
nona_ora	Aula	lab_scienze	palestra	aula_lettere2	aula_lettere2	aula_lettere1
	Docente	doc_ed_civica	doc_ed_civica	doc_lab_disegno	doc_scienze_1	doc_scienze_1
	Materia	educazione_civica	educazione_civica	lab_disegno	chimica	chimica

Terza A		lunedì	martedì	mercoledì	giovedì	venerdì
prima_ora	Aula	aula_lettere1	palestra	aula_matematica	aula_spagnolo	aula_matematica
	Docente	doc_lettere_1	doc_ed_fisica	doc_scienze_1	doc_spagnolo	doc_scienze_1
	Materia	lettere	educazione_fisica	matematica	spagnolo	matematica
seconda_ora	Aula	aula_lettere1	lab_scienze	lab_arte	aula_matematica	aula_inglese
	Docente	doc_lettere_1	doc_scienze_2	doc_arte	doc_scienze_1	doc_inglese
	Materia	lettere	scienze	arte	matematica	inglese
terza_ora	Aula	aula_musica	aula_matematica	aula_tecnologia	palestra	aula_musica
	Docente	doc_musica	doc_scienze_1	doc_tecnologia	doc_ed_fisica	doc_musica
	Materia	musica	matematica	tecnologia	educazione_fisica	musica
quarta_ora	Aula	aula_religione	aula_spagnolo	lab_arte	lab_scienze	aula_tecnologia
	Docente	doc_religione	doc_spagnolo	doc_arte	doc_scienze_2	doc_tecnologia
	Materia	religione	spagnolo	arte	scienze	tecnologia
quinta_ora	Aula	aula_inglese	aula_lettere1	aula_lettere2	aula_lettere1	aula_lettere2
	Docente	doc_inglese	doc_lettere_1	doc_lettere_1	doc_lettere_1	doc_lettere_1
	Materia	inglese	lettere	lettere	lettere	lettere
sesta_ora	Aula	aula_inglese	aula_lettere1	aula_lettere2	aula_lettere1	aula_lettere1
	Docente	doc_inglese	doc_lettere_1	doc_lettere_1	doc_lettere_1	doc_lettere_1
	Materia	inglese	lettere	lettere	lettere	lettere
settima_ora	- - -	mensa	mensa	mensa	mensa	mensa
ottava_ora	Aula	aula_tecnologia	palestra	aula_inglese	aula_spagnolo	aula_musica
	Docente	doc_chimica	doc_ed_civica	doc_ed_civica	doc_arte	doc_matematica_2
	Materia	chimica	educazione_civica	educazione_civica	lab_disegno	recupero_matematica
nona_ora	Aula	aula_lettere1	aula_tecnologia	lab_scienze	aula_musica	lab_scienze
	Docente	doc_arte	doc_chimica	doc_matematica_2	doc_matematica_2	doc_matematica_2
	Materia	lab_disegno	chimica	recupero_matematica	recupero_matematica	recupero_matematica

Prima B		lunedì	martedì	mercoledì	giovedì	venerdì
prima_ora	Aula Docente Materia	aula_religione	lab_scienze	aula_tecnologia	aula_lettere2	aula_musica
		doc_religione	doc_matematica_1	doc_tecnologia	doc_lettere_2	doc_musica
		religione	scienze	tecnologia	lettere	musica
seconda_ora	Aula Docente Materia	aula_matematica	aula_tecnologia	palestra	aula_lettere1	aula_matematica
		doc_scienze_1	doc_tecnologia	doc_ed_fisica	doc_lettere_2	doc_scienze_1
		matematica	tecnologia	educazione_fisica	lettere	matematica
terza_ora	Aula Docente Materia	aula_spagnolo	aula_musica	aula_lettere1	aula_inglese	aula_spagnolo
		doc_spagnolo	doc_musica	doc_lettere_2	doc_inglese	doc_spagnolo
		spagnolo	musica	lettere	inglese	spagnolo
quarta_ora	Aula Docente Materia	lab_scienze	palestra	aula_lettere1	aula_inglese	aula_inglese
		doc_matematica_1	doc_ed_fisica	doc_lettere_2	doc_inglese	doc_inglese
		scienze	educazione_fisica	lettere	inglese	inglese
quinta_ora	Aula Docente Materia	aula_lettere1	aula_lettere2	aula_matematica	aula_matematica	aula_lettere1
		doc_lettere_2	doc_lettere_2	doc_scienze_1	doc_scienze_1	doc_lettere_2
		lettere	lettere	matematica	matematica	lettere
sesta_ora	Aula Docente Materia	aula_lettere2	aula_lettere2	lab_arte	lab_arte	aula_lettere2
		doc_lettere_2	doc_lettere_2	doc_arte	doc_arte	doc_lettere_2
		lettere	lettere	arte	arte	lettere

Seconda B		lunedì	martedì	mercoledì	giovedì	venerdì
prima_ora	Aula Docente Materia	palestra	lab_arte	aula_inglese	aula_lettere1	aula_lettere1
		doc_ed_fisica	doc_arte	doc_inglese	doc_lettere_1	doc_lettere_1
		educazione_fisica	arte	inglese	lettere	lettere
seconda_ora	Aula Docente Materia	aula_inglese	aula_spagnolo	aula_matematica	aula_lettere2	aula_lettere2
		doc_inglese	doc_spagnolo	doc_scienze_1	doc_lettere_1	doc_lettere_1
		inglese	spagnolo	matematica	lettere	lettere
terza_ora	Aula Docente Materia	aula_lettere1	aula_lettere1	aula_lettere2	aula_spagnolo	aula_tecnologia
		doc_lettere_1	doc_lettere_1	doc_lettere_1	doc_spagnolo	doc_tecnologia
		lettere	lettere	lettere	spagnolo	tecnologia
quarta_ora	Aula Docente Materia	aula_lettere2	aula_lettere1	aula_lettere2	aula_matematica	aula_religione
		doc_lettere_1	doc_lettere_1	doc_lettere_1	doc_scienze_1	doc_religione
		lettere	lettere	lettere	matematica	religione
quinta_ora	Aula Docente Materia	palestra	aula_tecnologia	aula_musica	lab_scienze	lab_arte
		doc_ed_fisica	doc_tecnologia	doc_musica	doc_matematica_1	doc_arte
		educazione_fisica	tecnologia	musica	scienze	arte
sesta_ora	Aula Docente Materia	aula_matematica	lab_scienze	aula_musica	aula_inglese	aula_matematica
		doc_scienze_1	doc_matematica_1	doc_musica	doc_inglese	doc_scienze_1
		matematica	scienze	musica	inglese	matematica



Terza B		lunedì	martedì	mercoledì	giovedì	venerdì
prima_or	Aula	aula_tecnologia	aula_musica	aula_spagnolo	aula_religione	aula_lettere2
	Docente	doc_tecnologia	doc_musica	doc_spagnolo	doc_religione	doc_lettere_2
	Materia	tecnologia	musica	spagnolo	religione	lettere
seconda_or	Aula	aula_tecnologia	aula_inglese	aula_inglese	palestra	aula_lettere1
	Docente	doc_tecnologia	doc_inglese	doc_inglese	doc_ed_fisica	doc_lettere_2
	Materia	tecnologia	inglese	inglese	educazione_fisica	lettere
terza_or	Aula	aula_lettere2	aula_lettere2	lab_arte	aula_matematica	aula_matematica
	Docente	doc_lettere_2	doc_lettere_2	doc_arte	doc_matematica_1	doc_matematica_1
	Materia	lettere	lettere	arte	matematica	matematica
quarta_or	Aula	aula_lettere1	aula_lettere2	aula_matematica	aula_musica	lab_arte
	Docente	doc_lettere_2	doc_lettere_2	doc_matematica_1	doc_musica	doc_arte
	Materia	lettere	lettere	matematica	musica	arte
quinta_or	Aula	lab_scienze	aula_matematica	aula_lettere1	aula_lettere2	lab_scienze
	Docente	doc_matematica_1	doc_matematica_1	doc_lettere_2	doc_lettere_2	doc_matematica_1
	Materia	scienze	matematica	lettere	lettere	scienze
sesta_or	Aula	aula_spagnolo	palestra	aula_lettere1	aula_lettere2	aula_inglese
	Docente	doc_spagnolo	doc_ed_fisica	doc_lettere_2	doc_lettere_2	doc_inglese
	Materia	spagnolo	educazione_fisica	lettere	lettere	inglese



# Capitolo 3

## Planning in CLIPS

### 3.1 Descrizione del problema

Il problema richiedeva la costruzione di un sistema esperto in grado di suggerire dei pacchetti vacanza ad un utente che fornisce informazioni parziali sulla sua idea di viaggio. Il sistema quindi viene utilizzato da un'agenzia viaggi e si deve occupare di elaborare le preferenze inserite dall'utente e di fornire un insieme di almeno 2 e al massimo 5 soluzioni con un determinato fattore di certezza. Una soluzione è composta da una lista di località da visitare, un prezzo totale ed un fattore di certezza che indica quanto essa sia adatta all'utente sulla base delle preferenze che ha fornito. Il sistema, inoltre, non deve fornire soluzioni che favoriscono sempre il medesimo hotel, in modo tale da garantire un'equa distribuzione dei clienti.

#### 3.1.1 Dominio del problema

Nel comporre il pacchetto vacanza sono state considerate differenti località con diversi tipi di turismo:

- Turismo Balneare
- Turismo Montano
- Turismo Lacustre
- Turismo Naturalistico
- Turismo Termale
- Turismo Culturale
- Turismo Religioso
- Turismo Sportivo
- Turismo Enogastronomico

Ogni località ha un punteggio per ogni tipo di turismo che va da 0 a 5. Ci possono essere più hotel per ogni località con un numero di stelle che va da 1 a 4, dove il prezzo per una notte in hotel da 1 stella costa 50 euro e 25 euro per ogni stella in più.

## 3.2 Organizzazione dei moduli

Il sistema esperto si compone dei seguenti moduli: MAIN, QUESTIONS, ENRICH, LOCATIONS, GENERATE-PATH, OPTMIZE-PATH, HOTEL, TRIP, TRIP-COST, TRIP-SELECTION, PRINT-RESULTS, FINAL-QUESTIONS

### 3.2.1 Modulo MAIN

Il modulo MAIN è il modulo di partenza che contiene la regola *start* la quale definisce la sequenza di attivazione dei moduli. In questo modulo sono presenti le regole per la combinazione dei certainty factors (entrambi positivi, entrambi negativi o di segno opposto) relativi ai fatti *attribute* asseriti durante l'esecuzione. Alle regole di combinazione viene impostato l'*auto-focus* a True in modo tale da garantire che i fatti vengano unificati prima di essere utilizzati da altre regole.

Inoltre, è presente la funzione **ask-a-question** che si occupa di stampare la domanda a video, di controllare la validità della risposta e di restituire la risposta validata.

### 3.2.2 Modulo QUESTIONS

In questo modulo contiene il fatto non ordinato per creare le domande e le regole per validare le risposte. Nello specifico la regola **ask-a-question** si occupa di chiamare la funzione *ask-question* per ogni fatto non ordinato *question* che è stato asserito. Di conseguenza viene prodotto un fatto temporaneo *attribute-intention* con valore uguale alla risposta alla domanda. Successivamente, verranno attivate le regole di verifica delle risposte che si occuperanno o di rimuovere il fatto temporaneo qualora l'utente non abbia risposto oppure creare un nuovo fatto *attribute* definitivo contenente la risposta alla domanda, un nome identificativo relativo alla domanda alla quale l'utente ha risposto e un certainty factor pari a 100.

In questo modulo sono presenti anche le regole relative alla verifica dei *precursors* delle domande, infatti alcune delle domande non verranno proposte all'utente se prima non si è risposto ad altre, ad esempio l'utente non potrà rispondere alla domanda "Le persone sono disponibili a dormire in camere doppie?" se prima non avrà indicato il numero di persone che desiderano partecipare al viaggio.

### 3.2.3 Modulo ENRICH

Una volta espresse le preferenze relative alla tipologia di turismo, il sistema inferirà alcune informazioni correlate per arricchire ed espandere la scelta dell'utente, sfruttando l'*expertise* del sistema esperto. Se ad esempio l'utente seleziona un tipo di turismo "religioso", il sistema potrebbe inferire come correlato un turismo di tipo "culturale". Le informazioni inferite sono pesate da uno specifico certainty factor ed influenzeranno in modo minore la ricerca rispetto alle preferenze espresse esplicitamente dall'utente. In questo modulo, quindi, sono presenti delle regole di expertise che arricchiscono ed espandono il contesto e le informazioni per la selezione dei viaggi, utilizzando la conoscenza specifica del dominio applicativo codificata nel sistema esperto.

### 3.2.4 Modulo LOCATIONS

Questo modulo si occupa di definire i fatti ordinati relativi alle località, ovvero asserisce quali sono le città disponibili per l'organizzazione del viaggio. Una determinata località è composta dai seguenti slot:

- name: slot contenente il nome della località
- region: slot contenente la regione di cui la località fa parte
- latitude: slot contenente la latitudine della località
- longitude: slot contenente la longitudine della località
- tourism-type: multislot che contiene i tipi di turismo con la relativa valutazione, ad esempio "balneare 5.0 enogastronomico 4.7"

Gli slot *latitude* e *longitude* sono stati definiti per il calcolo della distanza tra le località attraverso la formula dell'emisenoverso, una formula trigonometrica delle sfere utilizzata comunemente per la navigazione:

$$\begin{aligned}a &= \sin^2(\Delta d / R) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta \lambda / 2) \\c &= 2 \cdot \operatorname{atan} 2(\sqrt{a}, \sqrt{1-a}) \\d &= R \cdot c\end{aligned}$$

dove  $\varphi$  è la latitudine,  $\lambda$  è la longitudine e  $R$  è il raggio della Terra (pari a 6731 km).

Le preferenze dell'utente relativamente alla regione e al tipo di turismo che desidera vengono intercettate dalle regole **update-tourism-type-fit** e **update-region-fit**. Queste due regole si occupano di effettuare un match per ogni località e per ogni preferenza espressa dall'utente relativa al tipo di turismo e alla regione e asseriranno un nuovo fatto che si presenta nel seguente modo per il tipo di turismo:

(specification (name location-fit) (subject ?l) (value (\* ?rank (/ ?c 100))))

mentre per la regione:

(specification (name location-fit) (subject ?l) (value (\* 15.0 (/ ?c 100))))

dove ?c rappresenta il certainty factor del fatto che coincide con la preferenza dell'utente. Il valore di ?c sarà pari a 50.0 se l'utente non ha espresso preferenza in merito ad una determinata domanda, sarà invece 100.0 se l'utente ha risposto alla domanda. Questa distinzione è stata pensata per pesare diversamente l'idoneità di una località andando a considerare se l'utente ha espresso esplicitamente la preferenza o meno.

Per quanto riguarda il calcolo dei valori di *location-fit*, nel primo caso si prende in considerazione la valutazione (rank) del tipo di turismo per una località, il quale viene moltiplicato per 0,5 (ovvero il certainty factor pari a 50 diviso per 100) nel caso in cui l'utente non abbia espresso preferenze a riguardo, mentre viene moltiplicato per 1 (certainty factor pari a 100 diviso 100) qualora l'utente abbia espresso esplicitamente la preferenza. Nel caso della regione, invece si è scelto di moltiplicare

per 15 lo stesso valore di modulazione del primo caso. Il valore "15" è stato scelto considerando che i fatti *specification* verranno combinati andando a sommare i loro valori al fine di ottenere un unico valore di *location-fit* per ogni località. Nello specifico si è considerato il caso particolare in cui un utente ha espresso la preferenza per 3 tipi di turismo in una determinata regione, ora, immaginando che ci siano due località di cui: la prima ha i tre tipi di turismo corrispondenti alle preferenze dell'utente e una valutazione pari a 5 per ognuno di essi, ma in una regione diversa da quella esplicitata dall'utente, mentre la seconda ha valutazioni inferiori a 5 per i tipi di turismo, ma nella regione scelta dall'utente, si otterrebbe che la località al di fuori della regione scelta avrebbe una valutazione totale pari a 15, mentre la località che si trova nella regione espressa dall'utente avrebbe una valutazione minore di 15. In questo modo si è voluto dare una maggiore importanza al match con la regione modulando il peso dei match con i tipi di turismo.

### 3.2.5 Modulo GENERATE-PATH

Il modulo GENERATE-PATH si occupa di generare i possibili percorsi intercettando le località che sono in accordo con le preferenze dell'utente. La scelta delle località da aggiungere al percorso avviene quindi considerando: la distanza massima esplicitata dall'utente (100 km di default) con un margine di tolleranza del 20%; le regioni in cui l'utente desidera effettuare il viaggio (nessuna preferenza di default) e il numero di località che l'utente vuole visitare (3 di default).

Ogni volta che viene aggiunta una città al percorso, viene calcolata la distanza totale e aggiornata sul piano di viaggio.

### 3.2.6 Modulo OPTIMIZE-PATH

In questo modulo sono state implementate due regole di ottimizzazione dei percorsi generati: la prima si occupa di eliminare i percorsi incompleti, ovvero quelli per i quali il sistema non è riuscito ad assegnare il numero di località richiesto dall'utente, mentre la seconda elimina i percorsi sub-ottimi, ovvero quelli che contengono le stesse località di un altro percorso, ma che prevedono di percorrere una distanza totale maggiore. Nella figura 3.1 è rappresentato un percorso per cui partendo da Alassio ci si sposta a Sanremo e successivamente a Laigueglia percorrendo un totale di circa 106km, mentre considerando un percorso migliore ci si potrebbe spostare da Alassio a Laigueglia e successivamente a Sanremo percorrendo circa 55km ovvero quasi la metà!

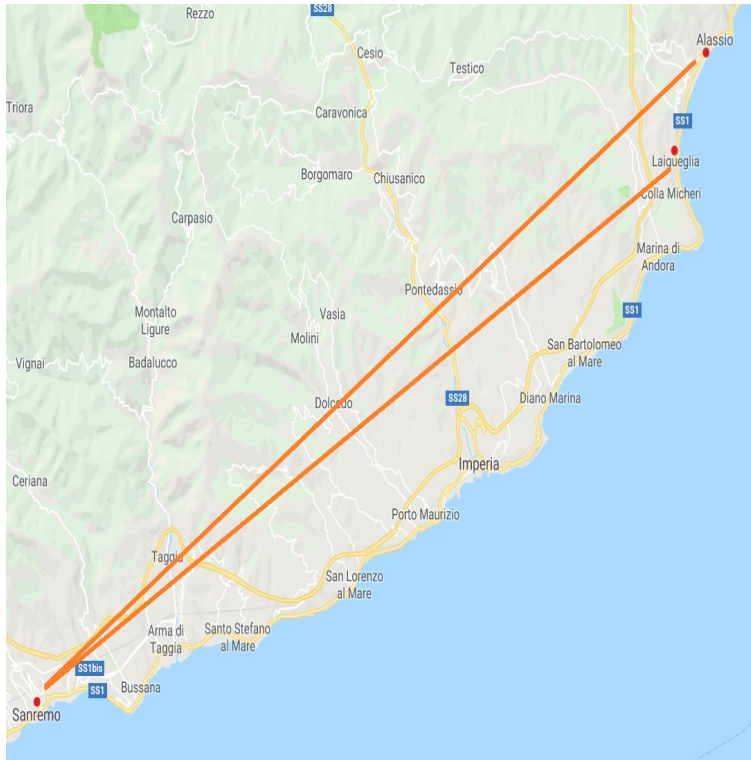


Figura 3.1: Esempio di percorso sub-ottimo

In situazioni analoghe a questa presentata in figura il sistema esperto andrà ad eliminare i percorsi che, tra una località e un'altra di uno stesso percorso, hanno una distanza maggiore rispetto ad altri.

### 3.2.7 Modulo HOTEL

Una volta generati i possibili percorsi si passa all'assegnamento degli hotel per il pernottamento nelle diverse località. In primo luogo il sistema cerca di assegnare un hotel che rispetti il numero di stelle esplicitato dall'utente (altrimenti vengono presi in considerazione tutti gli hotel per default), considerando anche se i partecipanti al viaggio sono disponibili a dormire in stanze doppie (impostato a falso per default). Nel caso in cui per una stessa località ci fossero più hotel che corrispondono alle preferenze dell'utente, viene confrontata la disponibilità di camere degli hotel e viene assegnato quello che ha più posti disponibili, in questo modo non si tenderà a favorire sempre lo stesso hotel. Ad esempio prendendo in considerazione Finale Ligure dove sono presenti due hotel a 4 stelle, con le opportune preferenze per far figurare Finale ligure nei risultati si ottiene:

```

(hotel (name "lido resort")
      (city "Finale Ligure")
      (region liguria)
      (rooms 30)
      (stars 4)
)
(hotel (name "Hotel Noris")
      (city "Finale Ligure")
      (region liguria)
      (rooms 40)
      (stars 4)
)

```

```

*****
* Trip suggestion (with certainty 58)
* - Journey length: 5 days
* - Number of locations: 3 city
* - Number of people: 3
* - Total journey: 26 km
* - Hotels cost: 1500 euro (with double rooms)
* - Journey:
*   City: Laigueglia
*   Hotel: Hotel Pineta (4 stars)
*   Permanence: 1 day
*
*   City: Alassio
*   Hotel: Hotel Riva (4 stars)
*   Permanence: 1 day
*
*   City: Finale Ligure
*   Hotel: Hotel Noris (4 stars)
*   Permanence: 3 days
*
*****

```

Figura 3.2: A sinistra i fatti relativi agli hotel di Finale Ligure con stesso numero di stelle ma con disponibilità diversa. A destra soluzione in cui viene selezionato Hotel Noris a Finale Ligure, ovvero quello con più disponibilità a parità di stelle.

L'expertise del sistema è stata progettata per cercare sempre di far risparmiare l'utente, infatti nel caso in cui non ci fossero hotel disponibili con il numero di stelle espresso dall'utente, viene assegnato l'hotel con più disponibilità di posti tra gli hotel più economici. Il prezzo degli hotel viene calcolato contando che il prezzo per una notte in hotel a 1 stella costa 50 euro e 25 euro in più per ogni stella aggiuntiva, inoltre viene considerata anche la preferenza relativa alla disponibilità di dormire in camere doppie, se i partecipanti non sono disposti verrà assegnata una stanza per ogni partecipante.

### 3.2.8 Modulo TRIP

Una volta assegnati gli hotel, in questo modulo viene organizzata la distribuzione dei giorni nelle varie località. Il sistema, attraverso le regole **optimize-trip-left-to-right** e **optimize-trip-right-to-left**, andrà a distribuire i giorni totali della durata del viaggio in base alle seguenti assunzioni: considerando ogni località del piano di viaggio, se una località ha una *location-fit* migliore o uguale ad un'altra e il prezzo dell'hotel è inferiore, allora viene spostato un giorno di permanenza nella località migliore. In questo modo il sistema distribuirà i giorni nelle località seguendo le preferenze dell'utente e mantenendo la politica volta a far risparmiare l'utente dal punto di vista del costo.

### 3.2.9 Modulo TRIP-COST

Il costo effettivo dell'intero piano di viaggio viene calcolato dopo aver distribuito correttamente i giorni totali di vacanza nei diversi hotel delle località. In questo modulo sono presenti due regole, la prima **calculate-path-price** per il calcolo effettivo del costo del viaggio e la seconda **modify-trip-path-price** per completare il fatto relativo al viaggio aggiungendo l'informazione sul prezzo.

### 3.2.10 Modulo TRIP-SELECTION

Una volta generati i piani di viaggio completi il sistema individuerà quali sono i piani di viaggio migliori e quali tra questi proporrà all'utente con un determinato fattore di certezza. In questo modulo vengono computati i certainty factor dei diversi piani di viaggio generati. Per un singolo piano di viaggio vengono generati 5 fatti *attribute* con lo slot name *path-confidence* relativi a:

- *Fit* delle località presenti sul piano di viaggio. Il certainty factor è stato calcolato nel modo seguente:

$$\frac{\text{fit del piano di viaggio}}{(((5 * \text{numero di preferenze espresse e inferite}) + 15) * (\text{numero di località} * 1.5))) * 100}$$

- *Prezzo* delle notti nei diversi hotel delle località considerando che l'utente ha espresso un budget massimo. Il certainty factor relativo viene calcolato utilizzando una funzione esponenziale se il prezzo calcolato per il viaggio è maggiore del budget dell'utente, altrimenti viene utilizzata una funzione lineare:

$$\text{Certainty factor} = \begin{cases} \max \left\{ \frac{(\frac{\text{prezzo}}{\text{budget}})^2}{4} * -1; -1 \right\} * 100, & \text{se prezzo} > \text{budget} \\ 1 - \frac{\text{prezzo}}{\text{budget}} * 100, & \text{se prezzo} \leq \text{budget} \end{cases}$$

In questo modo si ottengono i seguenti casi:

- Se il prezzo è maggiore del budget, più il prezzo del viaggio sarà alto rispetto al budget, più il certainty factor sarà basso fino ad un minimo di -100 nel caso in cui il prezzo sia il doppio rispetto al budget
- Se il prezzo è inferiore o uguale al budget il certainty factor avrà un minimo pari a 0 quando il prezzo è uguale al budget, mentre seguirà andamento lineare nel caso in cui sia minore.
- *Prezzo* delle notti nei diversi hotel delle località considerando che l'utente non ha espresso un budget massimo: in questo caso viene calcolato un budget di default considerando il numero totale di giorni del viaggio moltiplicato per il costo di un hotel a 3 stelle. In questo caso se il prezzo totale del viaggio risulta essere inferiore alla stima, l'incremento del CF sarà lineare, se uguale alla stima sarà uguale a 0, altrimenti sarà minore di 0 se il prezzo supera la stima.
- *Distanza* totale percorsa durante l'intero viaggio. Il certainty factor viene calcolato seguendo la seguente funzione:

$$1 - \frac{\text{distanza totale viaggio}}{\text{massima distanza percorribile}} * \frac{\text{cf-km}}{100}$$

In questo modo più la distanza totale calcolata per il piano di viaggio sarà maggiore minore sarà il certainty factor. *cf-km* rappresenta il certainty factor relativo al fatto se l'utente ha espresso o meno un numero massimo di km percorribili, così facendo il fattore di certezza verrà modulato in base alle preferenze espresse dall'utente.

- *Stelle* degli hotel che corrispondono alle preferenze dell'utente: In questo caso il certainty factor è calcolato sul numero di hotel assegnati per il viaggio che hanno stelle diverse da quanto esplicitato dall'utente.

$$\frac{\# \text{ hotel con stelle diverse}}{\# \text{ località del piano di viaggio}} * -60$$

Si è deciso limitare il cf a -60 (quando gli hotel sono tutti assegnati senza rispettare le preferenze dell'utente) in modo tale da non dare un peso eccessivo per calcolo totale del certainty factor dell'intero viaggio.

### 3.2.11 Modulo PRINT-RESULT

I piani di viaggio vengono stampati in questo modulo. Le soluzioni che vengono mostrate all'utente contengono tutte le informazioni riguardo il viaggio:

- Fattore di certezza della proposta di viaggio
- Numero di giorni totali
- Numero di località che si visiteranno durante il viaggio
- Numero di partecipanti
- Totale dei chilometri che si dovranno percorrere
- Costo delle notti in hotel
- Pernottamento in stanza con un'altra persona o singolarmente

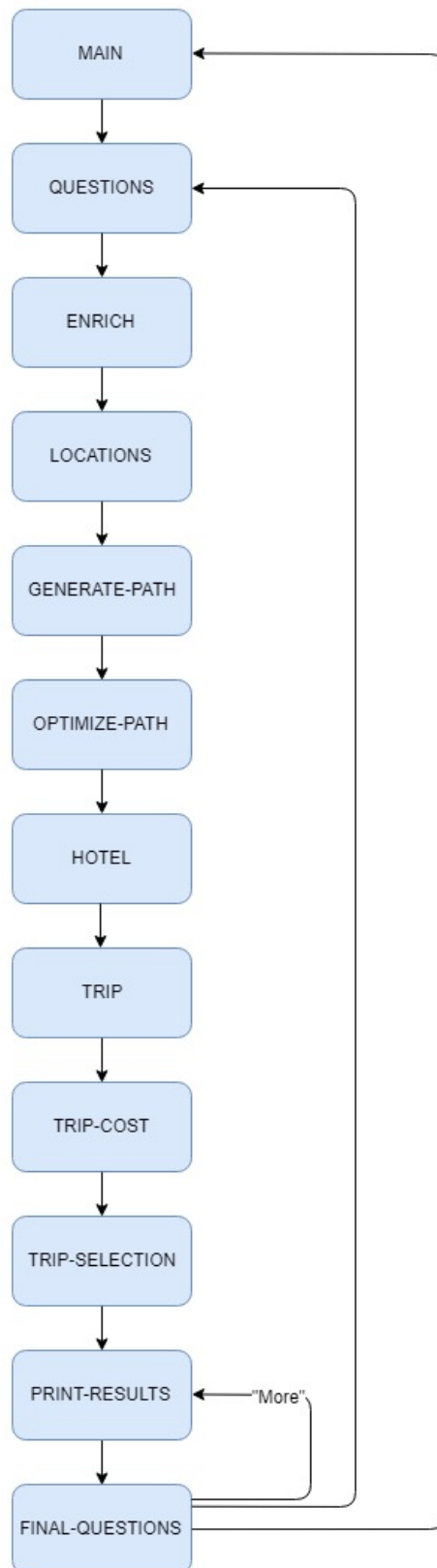
Le soluzioni proposte all'utente sono 3, ma nel prossimo modulo viene data la possibilità di richiedere ulteriori soluzioni fino all'esaurimento.

### 3.2.12 Modulo FINAL-QUESTION

Una volta conclusa l'interazione con il sistema e mostrate all'utente 3 soluzioni, viene data la possibilità di richiedere ulteriori soluzioni con le stesse preferenze (comando "**more**", di raffinare la soluzione rispondendo a nuove domande oppure di ricominciare l'interazione da capo. Nel caso in cui si scelga di raffinare la soluzione, se l'utente non cambia dei parametri che influiscono sul percorso che è stato calcolato (ad esempio: rispondendo alla domanda relativa al numero di persone che parteciperanno al viaggio si influenzerà solo il prezzo e non le località da visitare) il percorso non viene ricalcolato da zero, ma viene aggiornato con le nuove preferenze. Nel caso in cui invece le nuove preferenze influenzino il calcolo del percorso del viaggio, le soluzioni precedenti verranno scartate e ne verranno compute di nuove.



### 3.3 Interazione tra i moduli



### 3.4 Caso d'uso del sistema esperto

Si ipotizza il caso in cui un utente interagisce con il sistema senza avere un'idea precisa del viaggio che desidera fare. In primo luogo decide di vedere quali sono i possibili pacchetti viaggio senza esprimere alcuna preferenza, ovvero non rispondendo a nessuna delle domande poste dal sistema, in questo caso il sistema proporrà le seguenti tre soluzioni:

```
*****
* Trip suggestion (with certainty 50)
* - Journey length: 3 days
* - Number of locations: 3 city
* - Number of people: 2
* - Total journey: 8 km
* - Hotels cost: 350 euro (with double rooms)
* - Journey:
*   City: Garda
*   Hotel: Hotel Parco del Garda (3 stars)
*   Permanence: 1 day
*
*   City: Bardolino
*   Hotel: Hotel Lungolago (2 stars)
*   Permanence: 1 day
*
*   City: Lazise
*   Hotel: Hotel Miralago (3 stars)
*   Permanence: 1 day
*
*****

*****
* Trip suggestion (with certainty 44)
* - Journey length: 3 days
* - Number of locations: 3 city
* - Number of people: 2
* - Total journey: 41 km
* - Hotels cost: 300 euro (with double rooms)
* - Journey:
*   City: Napoli
*   Hotel: Hotel Albina (1 stars)
*   Permanence: 1 day
*
*   City: Positano
*   Hotel: Hotel Delle Stelle (3 stars)
*   Permanence: 1 day
*
*   City: Amalfi
*   Hotel: Hotel Marebello (2 stars)
*   Permanence: 1 day
*
*****

*****
* Trip suggestion (with certainty 41)
* - Journey length: 3 days
* - Number of locations: 3 city
* - Number of people: 2
* - Total journey: 26 km
* - Hotels cost: 350 euro (with double rooms)
* - Journey:
*   City: Finale Ligure
*   Hotel: Hotel Mare Inn (2 stars)
*   Permanence: 1 day
*
*   City: Alassio
*   Hotel: Hotel Savoia (3 stars)
*   Permanence: 1 day
*
*   City: Laigueglia
*   Hotel: Hotel Delfino (3 stars)
*   Permanence: 1 day
*
*****
```

In questo caso l'utente non ha espresso alcuna preferenza, quindi il sistema ha generato delle soluzioni basandosi esclusivamente su alcuni valori di default i quali prevedono:

- 3 località da visitare
- 3 giorni di durata del viaggio

- 2 persone partecipanti al viaggio
- Hotel con un qualunque numero di stelle

Una volta che l'utente ha visto quali sono le proposte potrà decidere se esplorarne altre attraverso il comando "more" oppure introdurre qualche preferenza con il comando "affine". Supponendo ora che l'utente si sia fatto un'idea del viaggio che vuole fare immaginiamo che voglia iniziare ad inserire qualche preferenza:

- 3 partecipanti
- preferenza turismo balneare ed enogastronomico
- hotel solo 3 e 4 stelle
- 7 giorni di vacanza
- in Campania
- budget 2500 euro

Date queste informazioni, il sistema proporrà le seguenti soluzioni:

```

*****
* Trip suggestion (with certainty 78)
* - Journey length: 7 days
* - Number of locations: 3 city
* - Number of people: 3
* - Total journey: 41 km
* - Hotels cost: 1750 euro (with double rooms)
* - Journey:
*   City: Amalfi
*   Hotel: Hotel Venere (3 stars)
*   Permanence: 3 days
*
*   City: Positano
*   Hotel: Hotel Delle Stelle (3 stars)
*   Permanence: 2 days
*
*   City: Napoli
*   Hotel: Hotel Tiempo (3 stars)
*   Permanence: 2 days
*
*****

```

```

*****
* Trip suggestion (with certainty 75)
* - Journey length: 7 days
* - Number of locations: 3 city
* - Number of people: 3
* - Total journey: 32 km
* - Hotels cost: 1900 euro (with double rooms)
* - Journey:
*   City: Capri
*   Hotel: Hotel Isola (4 stars)
*   Permanence: 3 days
*
*   City: Positano
*   Hotel: Hotel Delle Stelle (3 stars)
*   Permanence: 2 days
*
*   City: Amalfi
*   Hotel: Hotel Venere (3 stars)
*   Permanence: 2 days
*
*****

```

Anche in questo caso l'utente potrà far mostrare ulteriori soluzioni con il comando "more", inserire ulteriori preferenze con il comando "affine", terminare l'esecuzione con il comando "exit" oppure ricominciare da capo con il comando "reset".

Come si può notare i percorsi vengono calcolati da zero andando a scartare tutte le soluzioni che non corrispondono alle preferenze dell'utente. Inoltre, come specificato nella Sezione 3.2.8, i giorni vengono distribuiti nelle località che ottengono un punteggio di *fit* maggiore, ovvero risultano migliori rispetto alle corrispondenze con l'utente relativamente a stelle degli hotel e punteggio dei vari tipi di turismo della località.

# Capitolo 4

## Incertezza

### 4.1 MPE e MAP

MAP (Maximum a Priori) e MPE (Most Probable Explanation) rappresentano due task di ragionamento per le reti Bayesiane. Uno degli scopi principali per cui si utilizzano le reti Bayesiane riguarda l'*Explanation* ovvero cercare di capire quali sono le **cause** di una determinata probabilità. Il termine "explanation" può essere inteso in diversi modi e MAP ed MPE ne rappresentano due possibili definizioni.

#### 4.1.1 Most Probable Explanation (MPE)

In questo caso ci si chiede qual è l'istanza  $x$  più probabile di tutte le variabili  $X$  data l'evidenza  $e$ , Ovvero:

$$\text{MPE}(e) = \operatorname{argmax}_x P(x, e)$$

Quindi la definizione di "explanation" che si vuole dare con MPE significa che alla richiesta di spiegazione di una data evidenza  $e$  viene risposto che la spiegazione è data dall'istanziamento delle altre variabili ( $X$ ) secondo l'assegnamento  $x$  poiché  $x$  è quella che ha la probabilità più grande di tutti di essere congiunta con l'evidenza  $e$ .

Questo tipo di spiegazione non è sempre quello più desiderabile, infatti, ad esempio, nel caso in cui si hanno molte variabili e poche evidenze MPE effettua l'assegnamento di tutte le variabili, mentre potrebbe bastare un sottoinsieme di esse (MAP)

#### 4.1.2 Maximum a Posteriori Hypotesis (MAP)

MAP è una generalizzazione di MPE. Anche in questo caso si ha un'evidenza  $e$ , ma non si chiede di dare un assegnamento a tutte le variabili della rete che massimizzano la probabilità congiunta con  $e$ , ma si chiede di restituire l'assegnamento più probabile di un *sottoinsieme* di variabili ritenute rilevanti, ovvero quelle variabili che vengono ritenute interessanti dal punto di vista dell'impatto esplicativo verso l'evidenza  $e$ . Il sottoinsieme di variabili rilevanti ( $M \subseteq X$ ) viene passato direttamente alla MAP che produrrà l'assegnamento. Quindi MAP calcola:

$$\text{MAP}(e) = \operatorname{argmax}_m P(m, e)$$

Ovvero calcola l'assegnamento  $m$  delle variabili  $M$  che massimizza la probabilità congiunta di  $m$  con  $e$ .

### 4.1.3 Implementazione algoritmica

L'implementazione algoritmica di MAP ed MPE è stata realizzata in Python, espandendo e modificando la libreria `aimacode/aima-Python`.

Sono state necessarie le seguenti estensioni (prerequisiti delle funzioni di MAP e MPE o classi/metodi utili per i test):

- Implementazione di una classe *HybridBayesNet* in grado di gestire variabili con dominio discreto (la classe *BayesNet* presente nel pacchetto supporta unicamente variabili con dominio booleano).
- Implementazione di una classe *HybridBayesNode* per rappresentare i nodi delle rete bayesiana, con i relativi metodi per la memorizzazione e l'accesso ad i valori delle *conditional probability table (CPT)*.
- Implementazione di una libreria di parsing in grado di analizzare ed instanziare una rete *HybridBayesNet* formalizzata in un file in formato *.bif* o da una stringa in memoria.

Di seguito si presentano, da un punto di vista concettuale, le implementazioni di MPE e MAP realizzate.

#### 4.1.3.1 Implementazione di MPE

L'implementazione di MPE risulta molto simile all'algoritmo di *Variable Elimination* (presente in `aima-core`), ma invece di effettuare l'operazione di "*summing out*" nella fase di eliminazione di una variabile da un fattore, effettua l'operazione di "*maxing out*", individuando le configurazioni con probabilità massima e memorizzandone gli assegnamenti. Il fattore risultante, oltre alla CPT calcolata, memorizza il valore della variabile eliminata che massimizza la probabilità per ogni entry della CPT.

Inoltre si tiene traccia di tutti i fattori generati dalle operazioni di "*maxing out*", generando una catena di puntatori. Dal momento che tutte le variabili non di evidenza vengono eliminate dal *max out*, è possibile ricostruire l'assegnamento più probabile partendo dal valore individuato dell'ultima variabile eliminata e ricostruendo l'assegnamento globale percorrendo la catena di fattori al contrario. La probabilità congiunta  $P(mpe, e)$  viene individuata mediante il prodotto punto per punto dei fattori all'ultimo *step*.

#### 4.1.3.2 Implementazione MAP

L'implementazione di MAP estende MPE con una fase iniziale di eliminazione delle variabili non MAP, mediante l'operazione di "*summing out*", seguendo l'ordine topologico inverso delle variabili della rete. La seconda fase di MAP consiste in un MPE sui fattori risultati dalla fase precedente, realizzata applicando l'operazione di *maxing out* alle variabili MAP (sempre utilizzando l'ordine topologico inverso). La stessa catena di fattori descritta nella sezione precedente permette di ricostruire l'assegnamento più probabile partendo dal valore dell'ultima variabile eliminata che ne massimizza la probabilità.

#### 4.1.3.3 Correttezza delle implementazioni di MAP e MPE

La correttezza degli algoritmi è stata verificata sperimentalmente, implementando una classe di *unit testing* con il compito di eseguire gli algoritmi di MAP e MPE su diverse tipologie di reti presenti alla seguente pagina <http://www.bnlearn.com/bnrepository> e diverse configurazioni di chiamate a MAP e MPE. I risultati ottenuti, all'interno dei test, vengono confrontati con i valori e gli assegnamenti precalcolati con l'applicativo software *Samiam*

#### 4.1.4 Analisi requisiti spaziali e temporali di MPE e MAP

Una prima analisi dei requisiti spaziali e temporali di MPE e MAP è stata effettuata variando le dimensioni di una rete generata dinamicamente. In particolare è stato implementato un metodo in grado di restituire una rete a "catena" di dimensione arbitraria  $n$ , ovvero una rete bayesiana (polialbero) composta da  $n + 1$  nodi, in cui ogni nodo ha un unico padre e figlio, ad eccezione del nodo iniziale  $S_0$  che non ha padre (rappresentazione in figura 4.1).

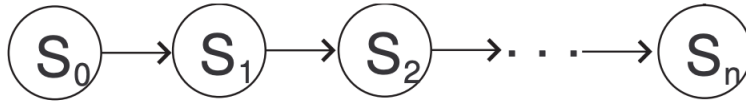


Figura 4.1: Chain network generata dinamicamente, di dimensione  $n$

I requisiti di tempo e spazio degli algoritmi MPE e MAP, basati sull'eliminazione di variabili sono dominati dal fattore più grande costruito, che a sua volta è determinato dall'ordine di eliminazione delle variabili e della struttura della rete.

La rete bayesiana a catena rappresenta un particolare caso in cui, ad ogni passo di eliminazione di una variabile (mediante un'operazione di *sum out* o *maxout*) il fattore risultante ha dimensione uguale al precedente.

Il comportamento di MPE atteso, in termini temporali, al crescere della dimensione della rete bayesiana a catena, è un incremento lineare. L'ipotesi è verificata dal grafico riportato in figura 4.2.

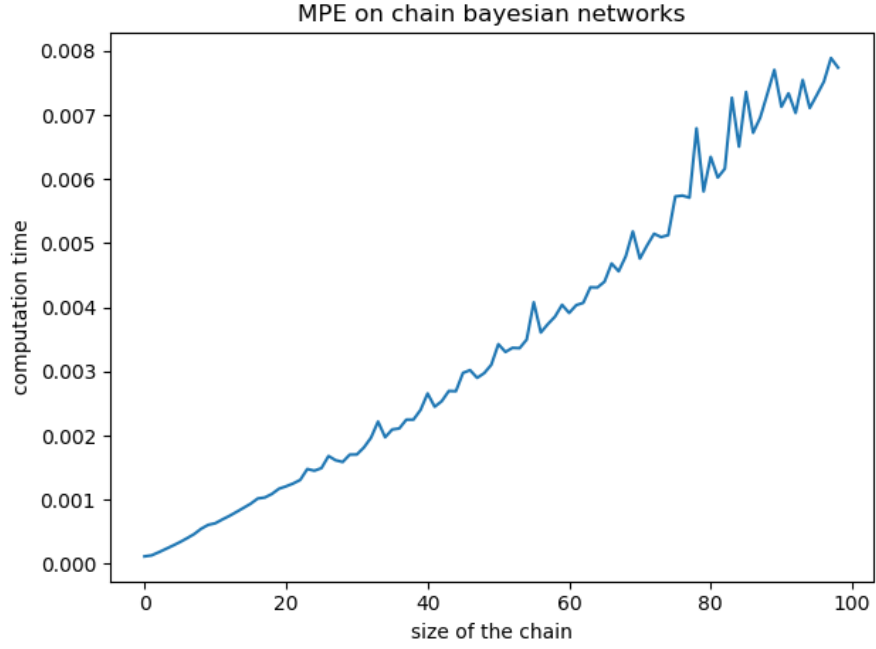


Figura 4.2: Tempi di esecuzione di MPE al variare della dimensione  $n$  di una chain network generata dinamicamente (con evidenza ( $S_N = True$ ))

Analizzando i requisiti di MAP mediante lo stesso esperimento si ottengono risultati simili (figura 4.3). Anche in questo esperimento è verificato l'incremento temporale lineare.

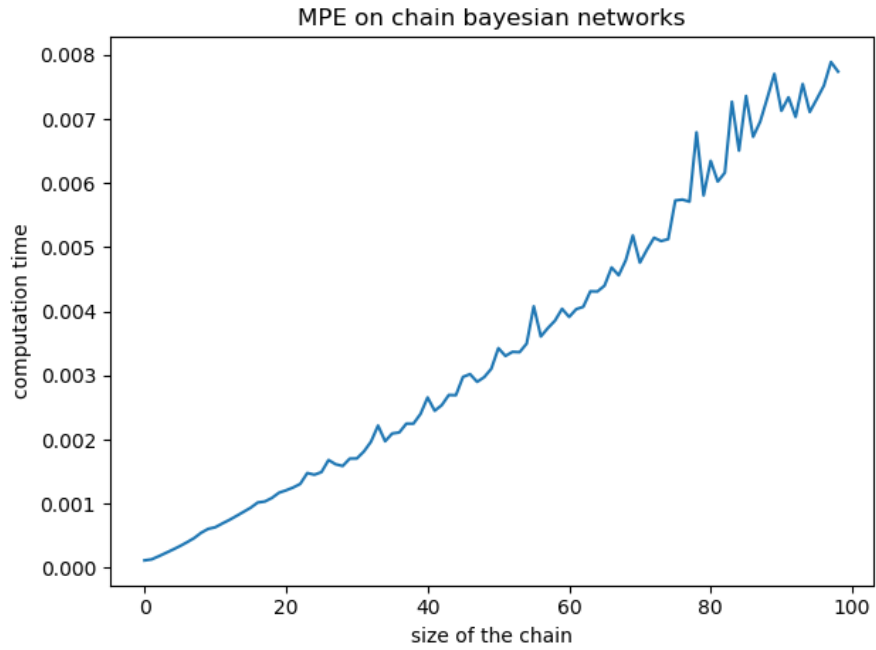


Figura 4.3: Tempi di esecuzione di MAP al variare della dimensione  $n$  di una chain network generata dinamicamente (con evidenza ( $S_N = True$ ) e con variabili di MAP  $S_1 \dots S_{\frac{n}{2}}$ )

La sostanziale differenza tra MAP ed MPE è il fatto che MAP ammette meno ordini di eliminazioni delle variabili rispetto ad MPE. Questa restrizione sul numero di ordini possibili è dovuta al fatto che MAP deve processare prima le variabili non MAP mediante l'operazione di *sum out* e successivamente applica l'operazione di *maxing out*.

Questa restrizione sull'ordine di eliminazione delle variabili di MAP porta tendenzialmente a fattori di eliminazione più grandi (in termini di CPT) rispetto ad MPE.

Si riporta in figura 4.4 l'evidenza sperimentale presentata nell'articolo *Complexity Results and Approximation Strategies for MAP Explanations*<sup>1</sup>.

Il grafico illustra il variare della width, definita come  $\log_2((\max(\text{size}(\text{Fattore}))) - 1)$  al crescere del numero delle variabili di MAP su 100 reti bayesiane generate casualmente. MPE, che corrisponde a MAP con tutte e 100 le variabili come *MAP variables*, presenta tipicamente *width* (di conseguenza massima dimensione dei fattori) minori rispetto a molti casi di map. Questo spiega le possibili differenze in termini di requisiti temporali e spaziali tra MAP e MPE.

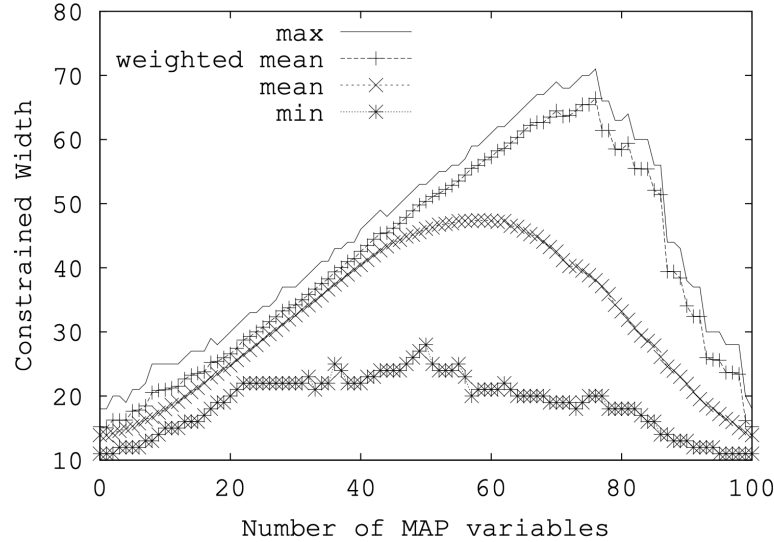


Figura 4.4: variazione della width, definita come  $\log_2((\max(\text{size}(\text{Fattore}))) - 1)$  al crescere del numero delle variabili di MAP su 100 reti bayesiane generate casualmente

Per evidenziare meglio l'incidenza delle variabili MAP in relazione con la struttura della rete è stato implementato un metodo in grado di generare dinamicamente una rete bayesiana con una particolare topologia, descritta nell'articolo precedentemente citato<sup>2</sup>.

La rete bayesiana dinamica generata è un polytree, di cui ne è riportata una rappresentazione in figura 4.5.

<sup>1</sup>Complexity Results and Approximation Strategies for MAP Explanations: <https://arxiv.org/pdf/1107.0024.pdf>, Figura 2

<sup>2</sup>Complexity Results and Approximation Strategies for MAP Explanations: <https://arxiv.org/pdf/1107.0024.pdf>, Figura 4



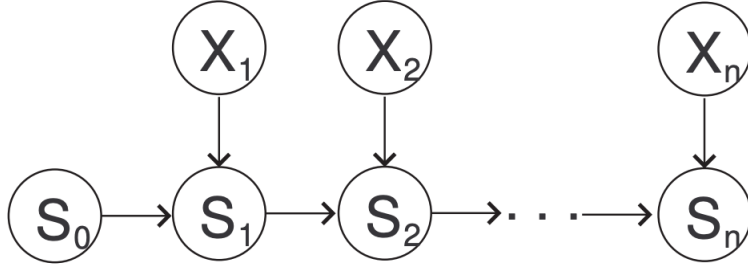


Figura 4.5: Polytree generato dinamicamente, di dimensione  $n$

Nonostante il fatto che la rete bayesiana sia un polialbero, impostando come variabili di MAP  $X_1, \dots, X_n$  e data l'evidenza  $S_n = True$  MAP richiede risorse esponenziali, poichè risulta necessario eliminare prima tutte le variabili  $S_0, \dots, S_{n-1}$  mediante l'operazione di *sum out* e successivamente le variabili  $X_1, \dots, X_n$  con l'operazione di *max out*. I tempi di computazione sono riportati in figura 4.6, che mostrano la crescita esponenziale dei tempi all'aumentare della dimensione della rete. Sempre secondo l'articolo citato, selezionando come variabili di MAP  $X_1, \dots, X_{\frac{n}{2}, S_0, \dots, S_{\frac{n}{2}}}$  invece di  $X_1, \dots, X_n$  è possibile individuare un ordine di eliminazione delle variabili che richiede requisiti temporali e spaziali lineari.

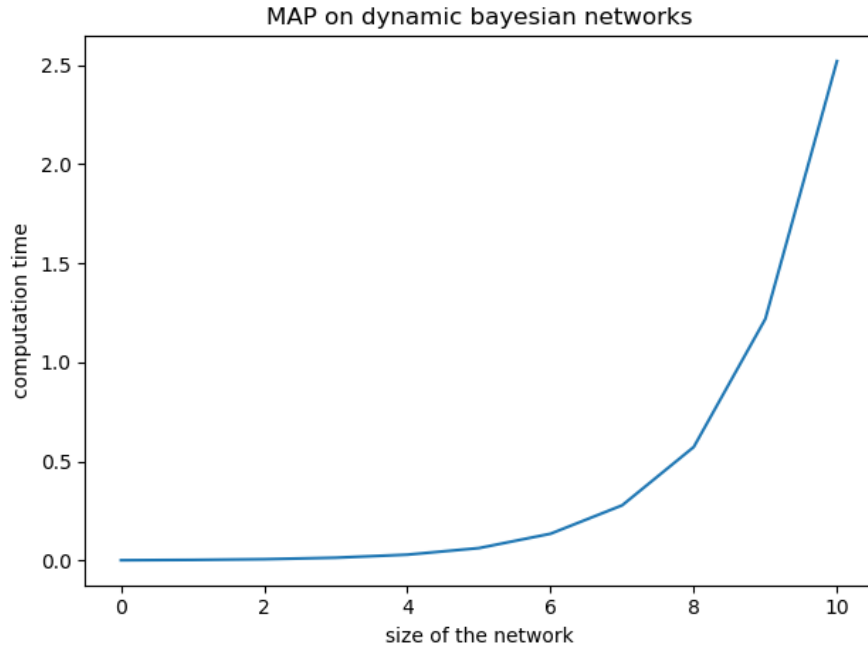


Figura 4.6: Tempi di esecuzione di MAP al variare della dimensione  $n$  di una rete con tipologia illustrata in figura 4.5, impostando come variabili di MAP  $X_1, \dots, X_n$  e data l'evidenza  $S_n = True$

#### 4.1.5 Analisi requisiti spaziali e temporali di MPE e MAP in relazione al numero di evidenze

Entrambi gli algoritmi non effettuano l'eliminazione su variabili a cui è assegnato un valore dall'evidenza. Il comportamento atteso in termini spaziali e temporali perciò è una diminuzione del tempo/spazio necessario per la computazione all'aumentare dell'evidenza.

Gli esperimenti effettuati sono stati eseguiti calcolando i tempi di computazione di MAP ed MPE all'aumentare del numero di evidenze.

Si riportano i risultati ottenuti per MAP e MPE sulla rete Alarm, scaricata dal seguente indirizzo <http://www.bnlearn.com/bnrepository/discrete-medium.html#alarm>, con la seguente configurazione:

- Number of nodes: 37
- Number of arcs: 46
- Number of parameters: 509
- Average Markov blanket size: 3.51
- Average degree: 2.49
- Maximum in-degree: 4

I risultati sono riportati in figura 4.7 e 4.8.

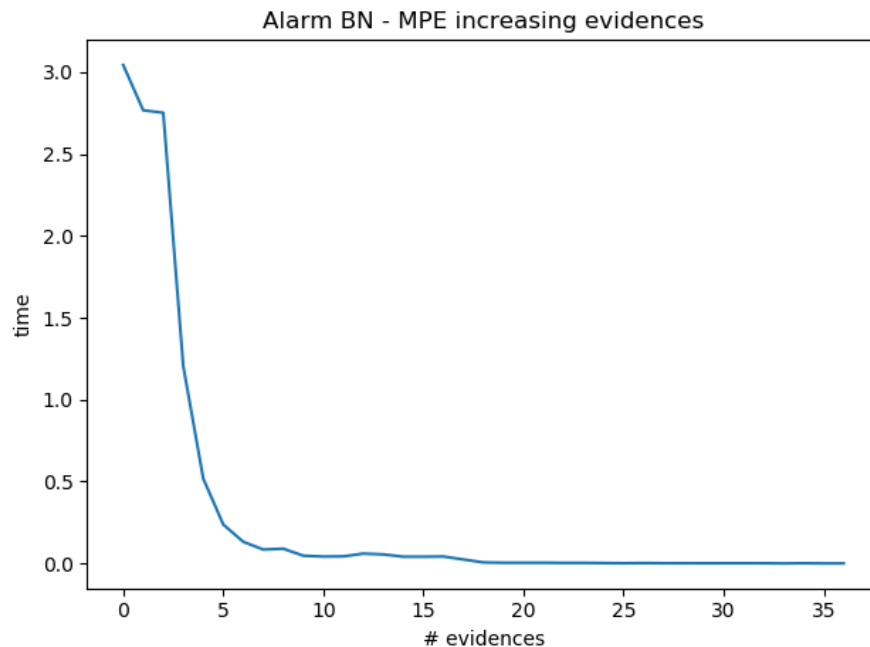


Figura 4.7: Tempi di esecuzione di MPE in relazione al numero di evidenze

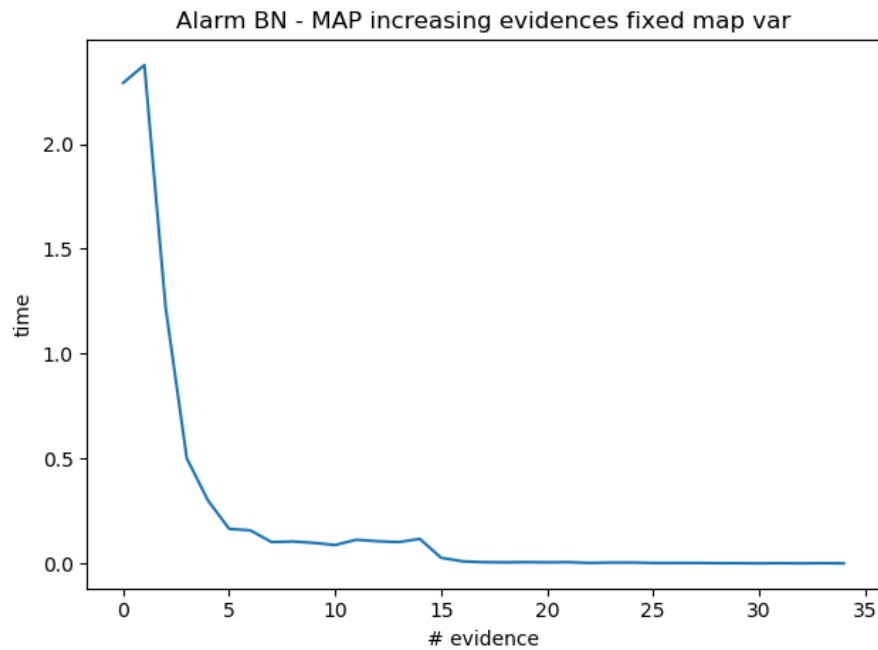


Figura 4.8: Tempi di esecuzione di MAP in relazione al numero di evidenze, impostando una variabile scelta a caso come MAP

#### 4.1.6 Analisi requisiti spaziali e temporali di MPE e MAP in relazione alla lontananza della rete da un polytree

Un polytree è una struttura a grafo in cui tra due nodi qualsiasi c'è al più un cammino non orientato.

L'ipotesi che si vuole verificare è che la complessità spaziale/temporale di MPE e MAP aumentano all'aumentare della lontananza di una rete da un polytree.

Per la sperimentazione si è deciso di comparare 3 diverse reti, con diverse configurazioni: la prima è la rete Alarm precedentemente descritta, le altre sono:

- Child:
  - Number of nodes: 20
  - Number of arcs: 25
  - Number of parameters: 230
  - Average Markov blanket size: 3.00
  - Average degree: 1.25
  - Maximum in-degree: 2
- Insurance:
  - Number of nodes: 27
  - Number of arcs: 52
  - Number of parameters: 984

- Average Markov blanket size: 5.19
- Average degree: 3.85
- Maximum in-degree: 3

Per valutare la lontananza delle reti da un polytree si utilizza un euristica basata sull'*Average Markov blanket size* (AMbs), in particolare si assume che una rete sia tanto più distante da un polytree tanto più il valore di AMbs è alto.

L'AMbs specifica il numero di figli, genitori e genitori dei figli medio per ogni nodo, più questo valore è alto più è probabile che il numero di percorsi diretti che collegano due nodi sia elevato. Sfruttando questa considerazione e trascurando gli altri parametri delle rete, i tempi di computazione per MAP e MPE attesi dovrebbero essere minori per la rete Child, intermedi per la rete Alarm e maggiori per la rete Insurance.

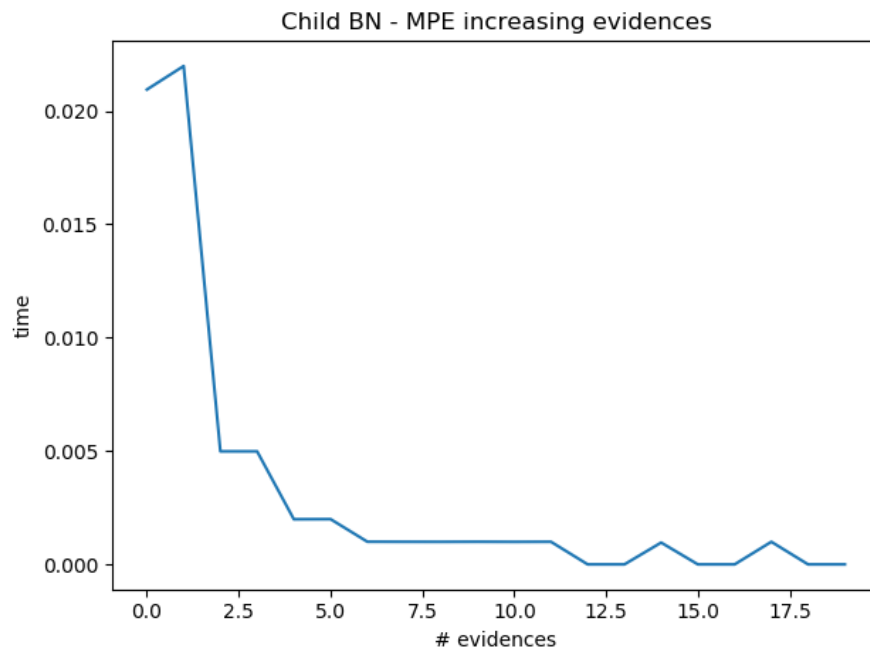


Figura 4.9: Tempi di esecuzione di MPE sulla rete Child in relazione al numero di evidenze

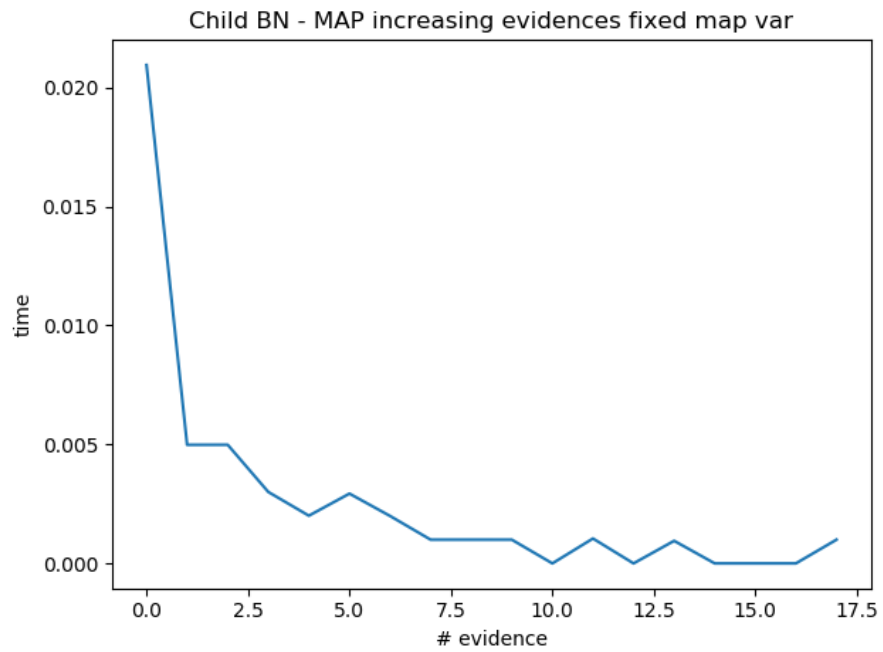


Figura 4.10: Tempi di esecuzione di MAP sulla rete Child in relazione al numero di evidenze, impostando una varaibile scelta a caso come MAP

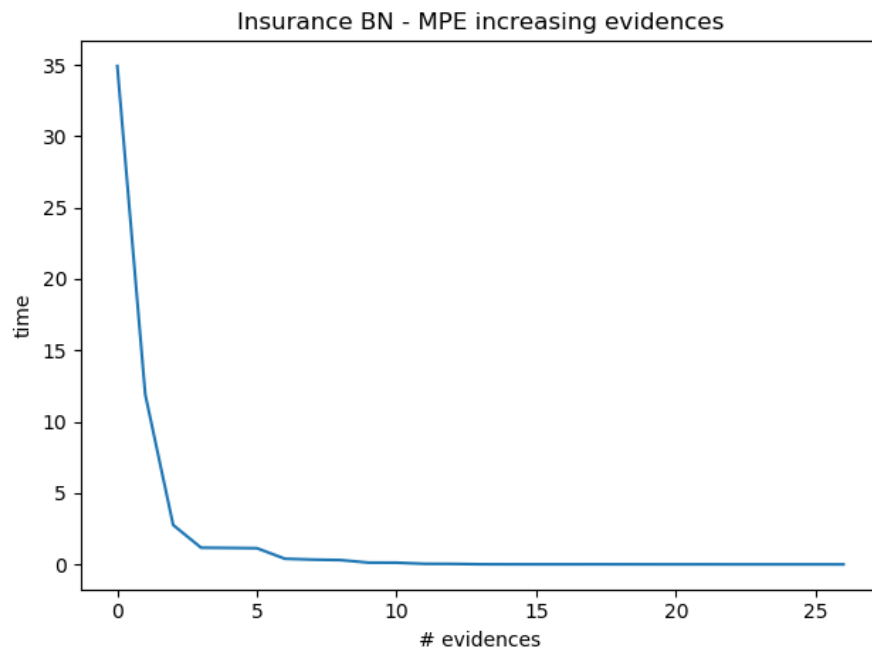


Figura 4.11: Tempi di esecuzione di MPE sulla rete Insurance in relazione al numero di evidenze

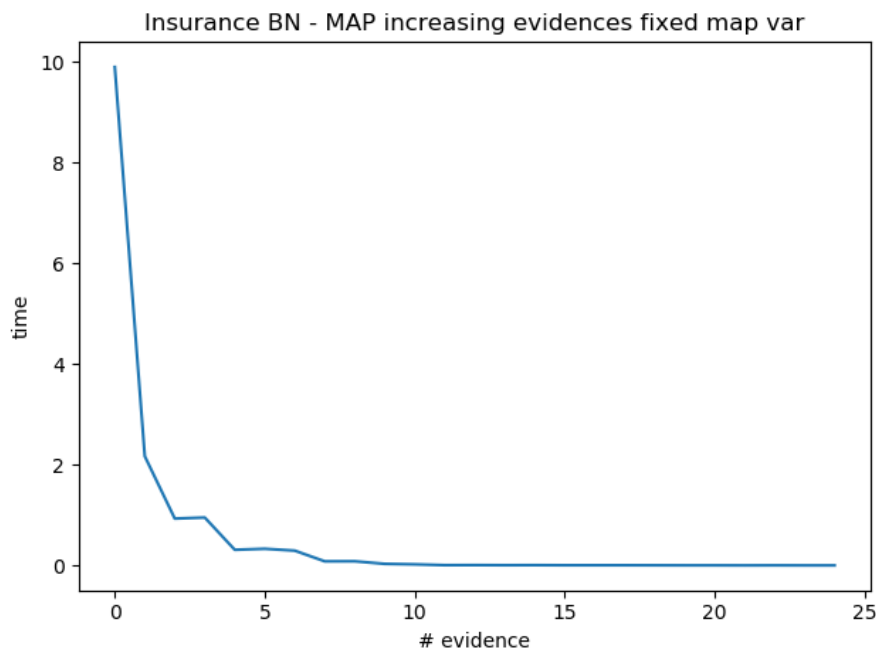


Figura 4.12: Tempi di esecuzione di MAP sulla rete Insurance in relazione al numero di evidenze, impostando una varaibile scelta a caso come MAP

Confrontando i grafici dei tempi di computazione in relazione al variare del numero di evidenze si riscontra come atteso che MPE e MAP sono più performanti sulla rete Child. Inoltre la rete Insurance, pur avendo un numero di nodi minore di Alarm, ottiene tempi di computazione più alti. L'evidenza è giustificata dal fatto che l'AVms di Insurance è più alto (5.19), indicando che la rete è maggiormente connessa, il che risulta in fattori più grandi al momento dell'eliminazione delle variabile e di conseguenza maggiori requisiti spaziali e temporali.

#### 4.1.7 Analisi requisiti spaziali e temporali di MAP in relazione alle variabili di MAP selezionate

Come già discusso nella sezione 4.1.4 la selezione delle variabili di MAP ha un rilevanza importante sull'ordine di eliminazione delle variabili e di conseguenza sui requisiti spaziali e temporali.

In figura 4.13 e 4.14 sono riportati i grafici dei tempi di computazione relativi al variare del numero di variabili di MAP.

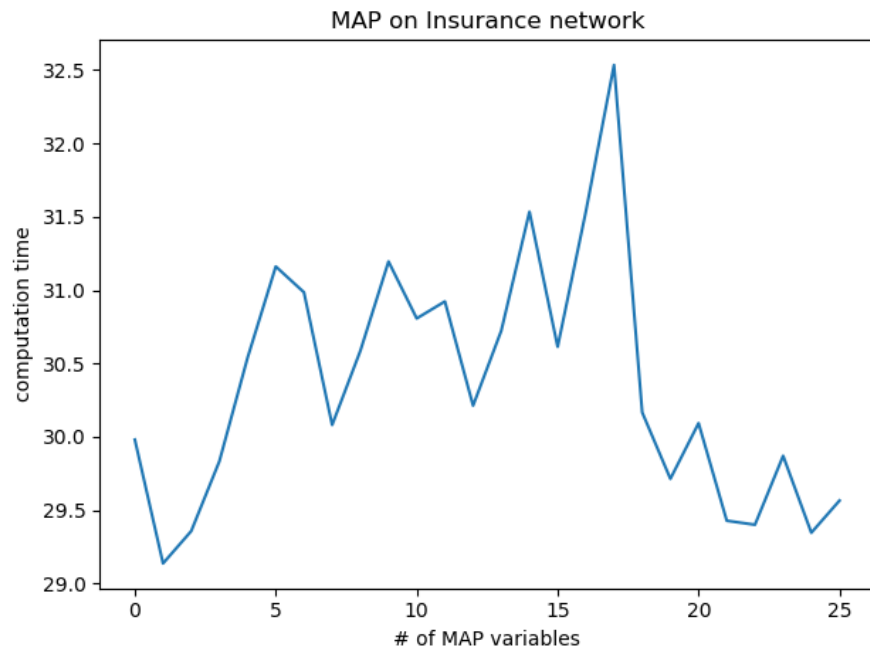


Figura 4.13: Tempi di esecuzione di MAP in relazione al numero di variabili di MAP sulla rete Insurance

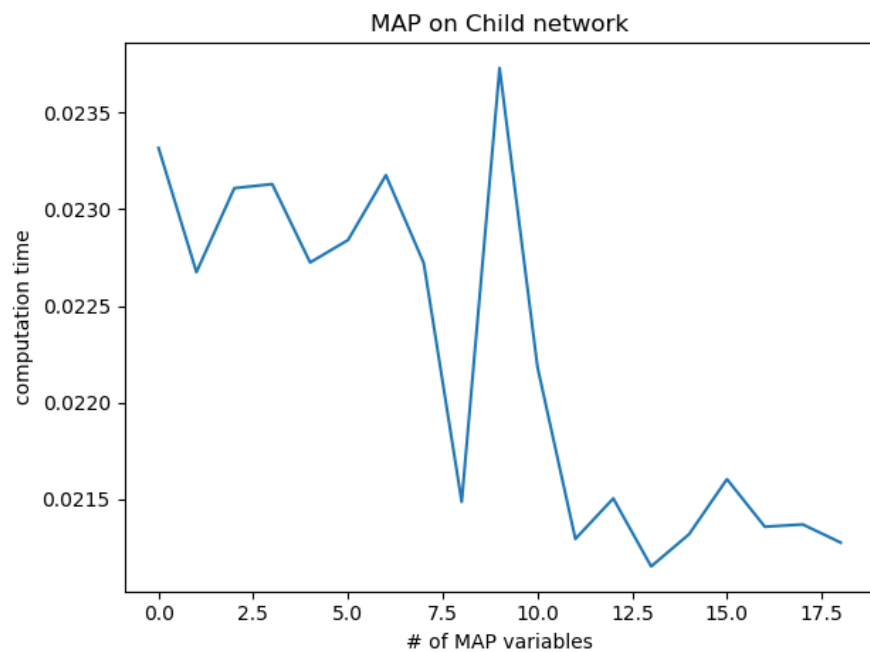


Figura 4.14: Tempi di esecuzione di MAP in relazione al numero di variabili di MAP sulla rete Child

Le considerazioni precedentemente discusse per la figura 4.4 permettono di giustificare la conformazione dei grafici, da cui si evince l'inefficienza di MAP rispetto ad MPE (MAP con tutte le variabili come variabili di MAP) in diverse configurazioni delle inferenze.

## 4.2 Modelli probabilistici temporali: Kalman Filter

Gli esperimenti con il Kalman Filter sono stati svolti in linguaggio Python utilizzando la libreria `pykalman` disponibile al link: `pykalman:https://pykalman.github.io/`. E' stato necessario effettuare una piccola modifica al codice della libreria per poter stampare a video il Kalman Gain poiché il valore veniva calcolato nella funzione `filter`, ma non veniva restituito, si vedano le note (sezione 4.2.3) in fondo per effettuare la medesima modifica.

### 4.2.1 Test

La matrice di transizione prevede due variabili, una relativa alla posizione e una alla velocità. Si ha quindi una matrice `[[posizione_x, posizione_y], [velocità_x, velocità_y]]` inizializzata nel modo seguente:

$$transition\_matrix = [[0.5, 0.1], [0.5, 0.9]]$$

Inoltre, sono stati inizializzati a 1 i parametri `transition_offset` e `observation_offset` del Kalman Filter, corrispondenti al coefficiente  $b$  nella formula di aggiornamento delle medie delle gaussiane delle osservazioni e delle transizioni. La formula di aggiornamento transizioni è:

$$P(X_{t+1}|X_t) = N(aX_t + b, \sigma_x)$$

mentre per le osservazioni si ha:

$$P(Z_t|X_t) = N(cX_t + b, \sigma_z)$$

Lo stato iniziale del processo, invece è stato inizializzato nel punto  $[0, 0]$  il quale verrà modificato a  $[1, 1]$  con aggiunta di rumore gaussiano in sede di esperimenti con punto di partenza  $P0$  sfalsato. Infine, le matrici di covarianza sono state inizializzate con delle matrici diagonali soggette a rumore gaussiano.

Gli esperimenti prevedevano l'esecuzione del filtro con diverse combinazioni di parametri relativi a:

- Rumore delle osservazioni
- Rumore del processo
- Stato iniziale di partenza (perfetto o variato con rumore)

I risultati sono stati i seguenti:

- Partenza perfetta:  $P0 = X0$ 
  - Rumore osservazione e transizione alti



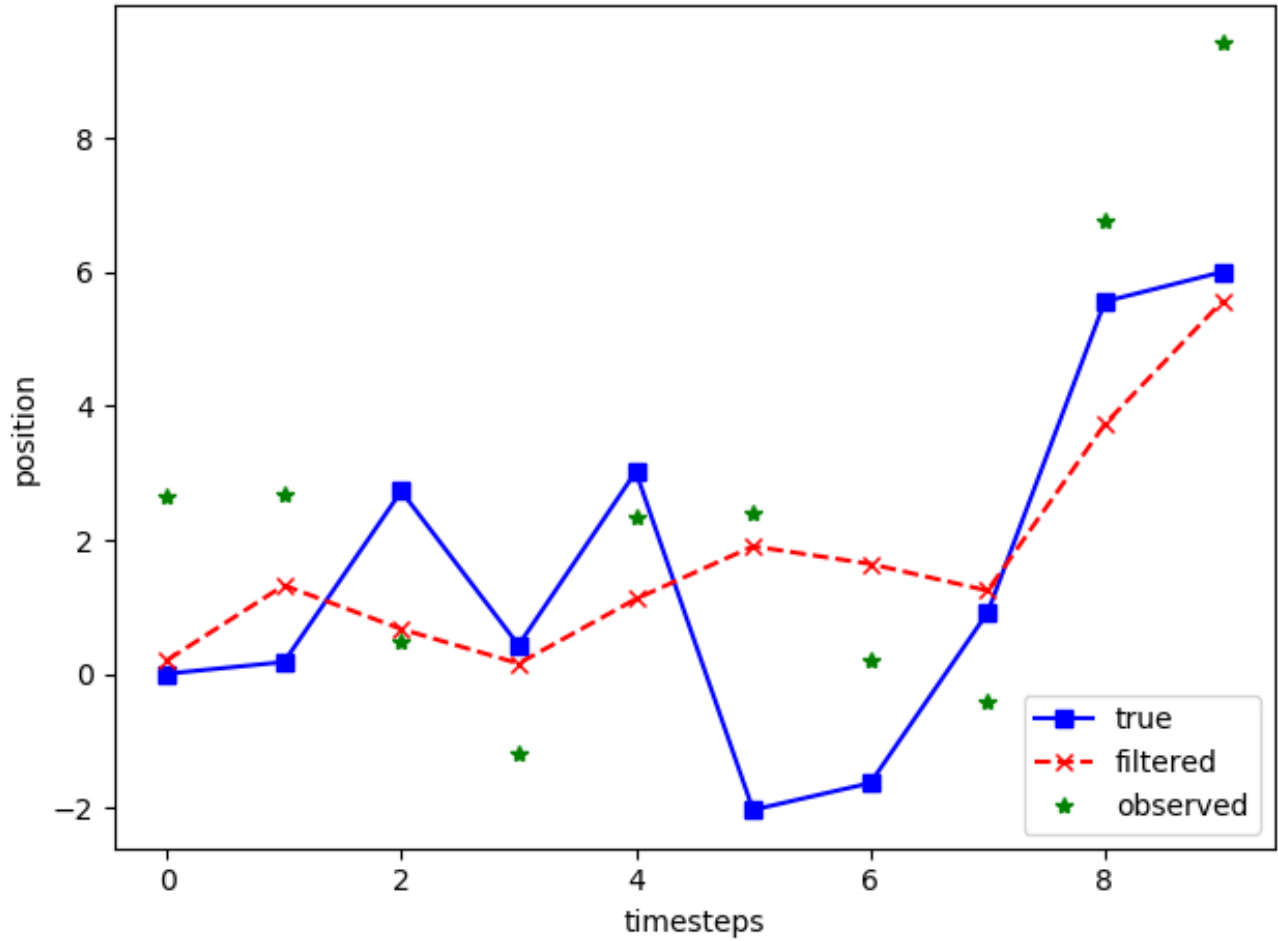


Figura 4.15: Rumore osservazioni  $\sigma_z = 5$  e rumore transizioni  $\sigma_x = 5$ , generati con una gaussiana  $N(\mu = 5, \sigma = 5)$ . Partenza perfetta dallo stato iniziale e 10 iterazioni

In questo caso di forte rumore né le previsioni, né le osservazioni sono affidabili, quindi il filtro tenderà a discostarsi sia dalle osservazioni che dal percorso reale. Il Kalman Gain calcolato mantiene un fattore medio di circa 0.55 (prendendo sempre in considerazione i valori relativi alla posizione) indicando un alto grado di correzione delle previsioni. L'errore medio di previsione risulta essere di circa 1.5, indicando che mediamente è stato necessaria una correzione pesante della previsione.

- Rumore osservazione e transizione bassi

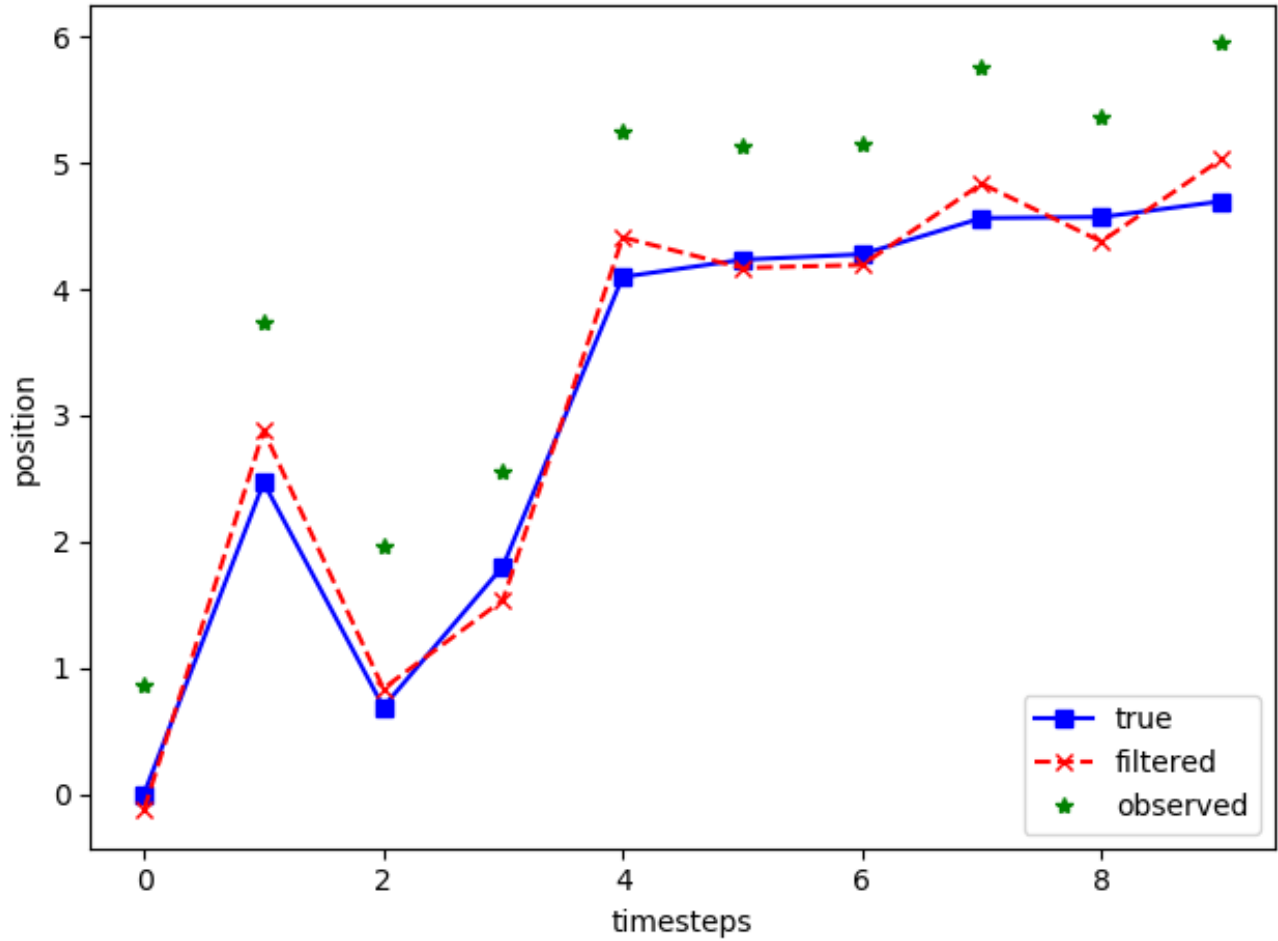


Figura 4.16: Rumore osservazioni  $\sigma_z = 0.4$  e rumore transizioni  $\sigma_x = 0.5$ , generati con una gaussiana con media  $\mu = 0$ . Partenza perfetta dallo stato iniziale e 10 iterazioni

In questo caso si può notare come, avendo rumore di osservazione e transizione bassi, il filtro si avvicina decisamente di più alla traiettoria reale rispetto al primo caso. Si osserva anche che si discosta in modo notevolmente più ridotto dalle osservazioni, credendo quindi di più ai sensori rispetto al primo caso. Il Kalman Gain presenta dei valori più bassi mantenendosi nelle varie iterazioni tra 0.2 e 0.3 mostrando quindi che il grado di correzione è diminuito rispetto al primo caso. L'errore medio calcolato è di 0.22, confermando quindi quanto mostrato sul grafico.

- Rumore osservazione alto, rumore transizione basso:

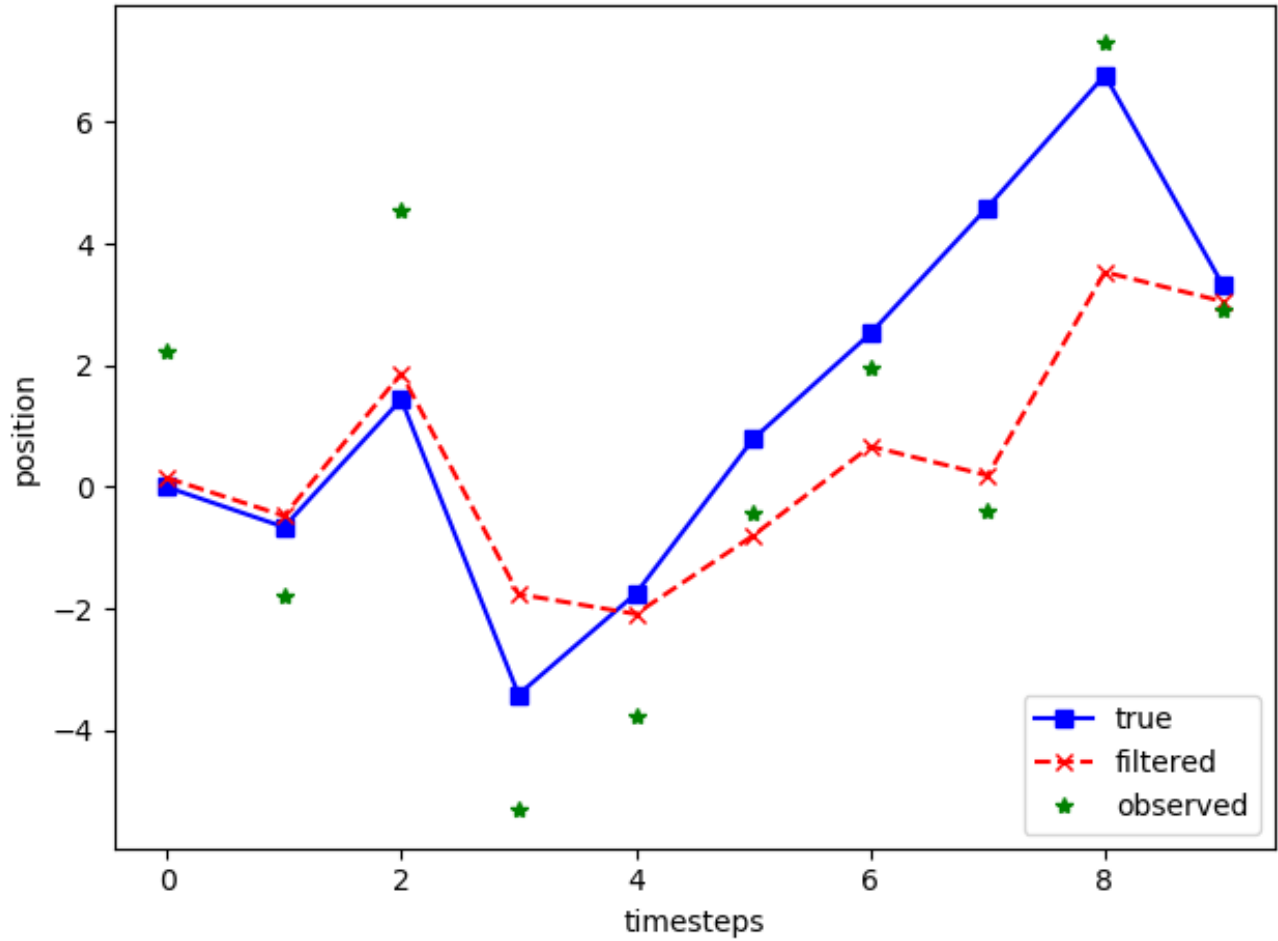


Figura 4.17: Rumore osservazioni  $\sigma_z = 5$  e rumore transizioni  $\sigma_x = 0.1$ , generati con una gaussiana con media  $\mu = 5$ . Partenza perfetta dallo stato iniziale e 10 iterazioni

Come si può notare dal grafico, il rumore delle osservazioni alto distacca notevolmente il filtro dalla traiettoria reale a partire dal timestep 5, mentre all'inizio si avvicina molto aiutato dalla partenza perfetta dove  $P_0 = X_0$ . Il Kalman Gain mantiene un fattore oltre lo 0.5 che va ad aumentare con il passare delle iterazioni, indicando, infatti, che il filtro necessita di un grado di correzione maggiore con il passare delle iterazioni. L'errore medio raggiunge quasi il valore presentato con entrambi i rumori alti, ovvero di circa 1.4.

- Rumore osservazione basso, rumore transizione alto:

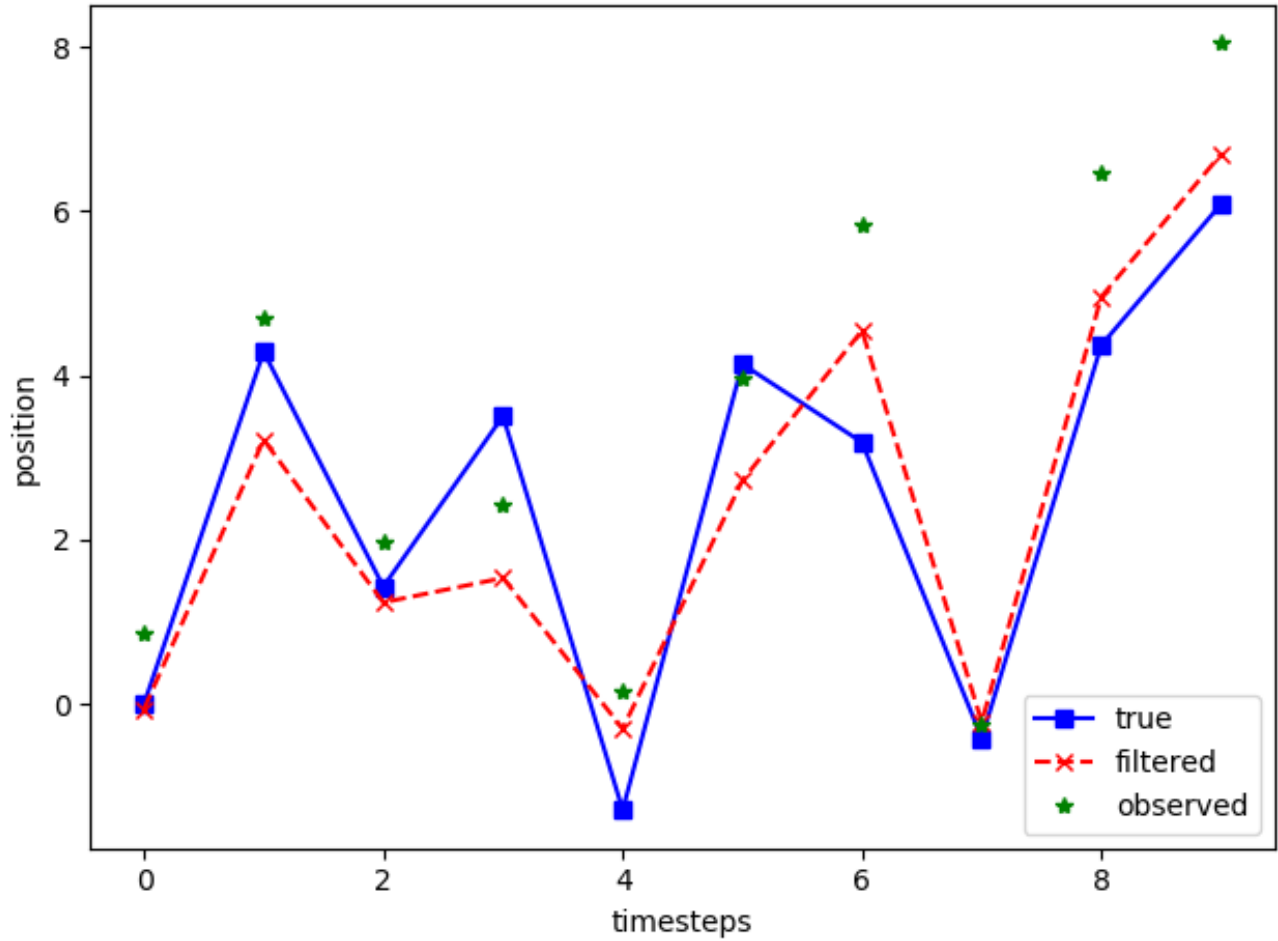


Figura 4.18: Rumore osservazioni  $\sigma_z = 0.1$  e rumore transizioni  $\sigma_x = 5$ , generati con una gaussiana con media  $\mu = 5$ . Partenza perfetta dallo stato iniziale e 10 iterazioni

Il filtro in questo caso tende a fidarsi maggiormente dei sensori delle osservazioni. Il Kalman Gain risulta mantenere dei valori alti di circa 0.8 indicando che nelle diverse iterazioni il grado di correzione è stato alto, mentre l'errore stimato del processo risulta di circa 0.85 che risulta essere medio alto rispetto agli altri esperimenti,

- Rumore osservazione e transizione nulli:

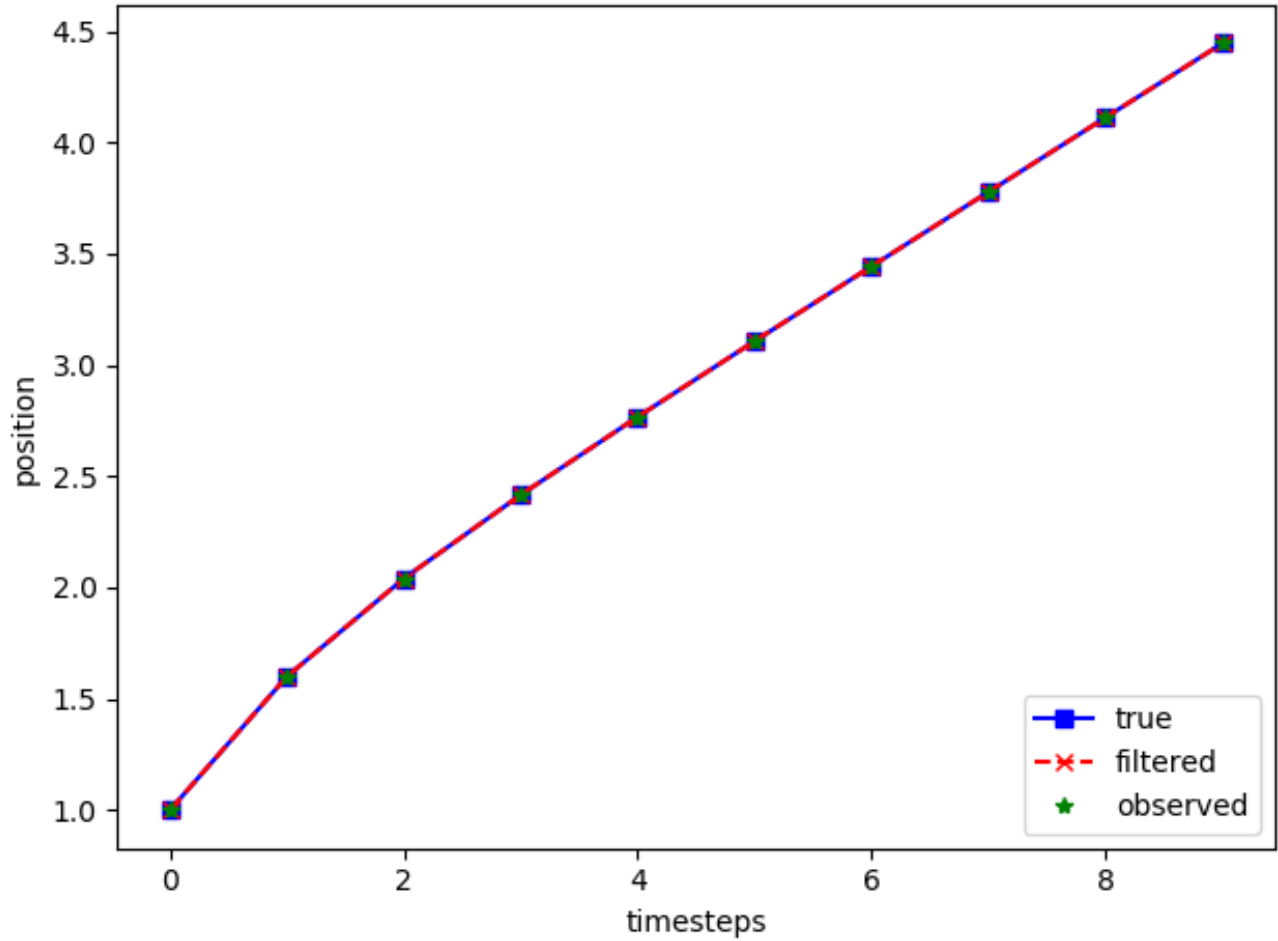


Figura 4.19: Rumore osservazioni  $\sigma_z = 0$  e rumore transizioni  $\sigma_x = 0$ , generati con una gaussiana con media  $\mu = 0$ . Partenza perfetta dallo stato iniziale e 10 iterazioni

Annullando completamente sia il rumore delle osservazioni sia il rumore delle transizioni, il filtro segue esattamente la traiettoria reale credendo totalmente alla predizione, il Kalman Gain infatti sarà pari a 0 indicando che il grado di correzione è stato nullo per ogni iterazione. L'errore è anch'esso nullo.

- Partenza perfetta:  $P0 \neq X0$ : Analogamente ai casi precedenti si sono svolti i medesimi esperimenti, ma andando ad aggiungere del rumore gaussiano alla matrice di covarianza iniziale (prima inizializzata per default alla matrice identità)
  - Rumore osservazione e transizione bassi

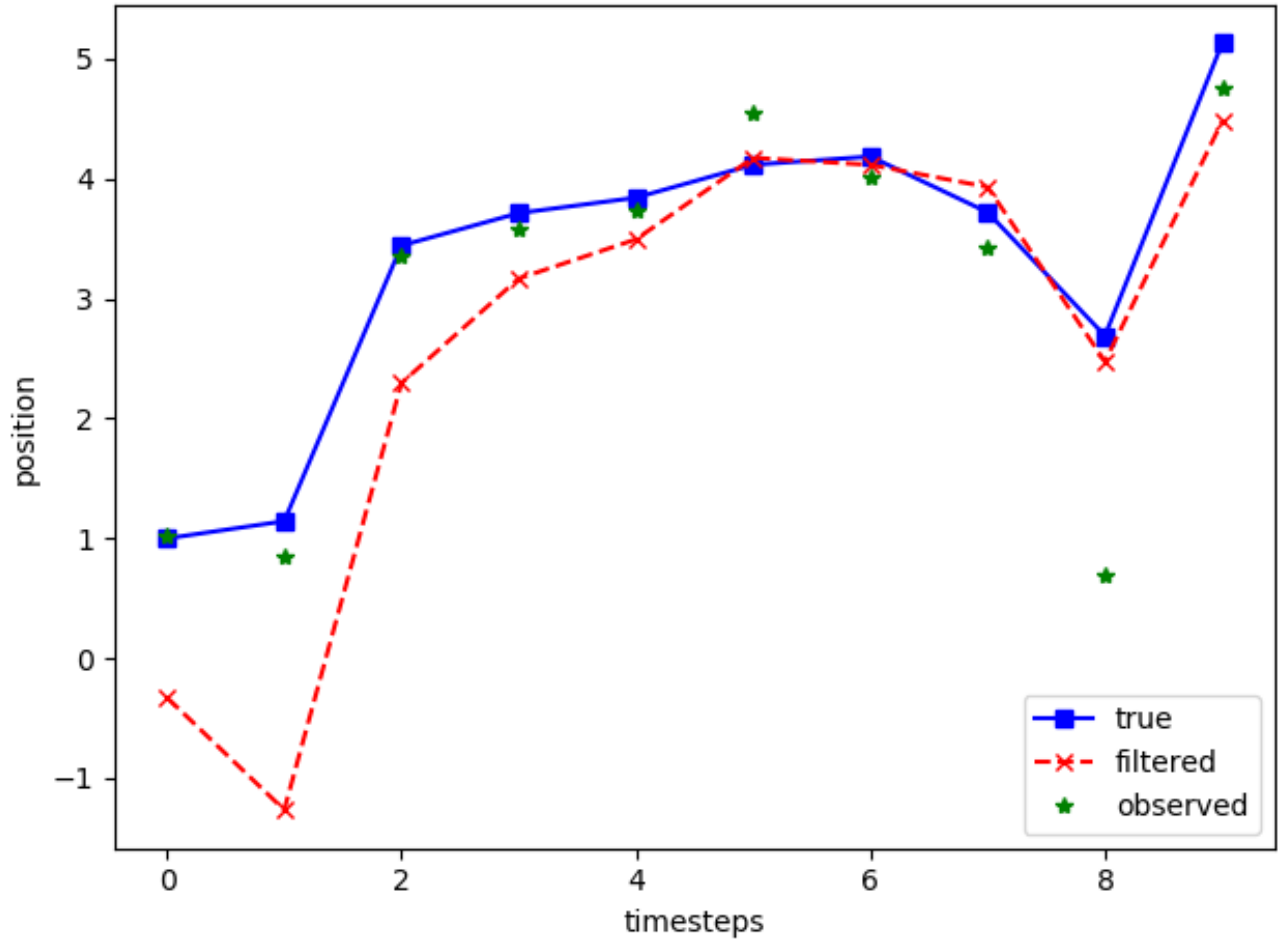


Figura 4.20: Rumore osservazioni  $\sigma_z = 0.1$  e rumore transizioni  $\sigma_x = 0.1$ , generati con una gaussiana con media  $\mu = 1$ . Partenza soggetta a rumore  $\sigma = 0.8\mu = 2$  e 10 iterazioni

Si può notare come anche partendo da uno stato iniziale non corrispondente alla realtà, il filtro di Kalman, con il procedere delle iterazioni e in condizioni di rumore delle osservazioni e delle transizioni basso, ritrovi la strada corretta e si avvicini alla traiettoria reale. Il Kalman Gain è elevato, con valori compresi tra 4 e 6 soprattutto nelle prime iterazioni, indicando il fatto che il grado di correzione era elevato inizialmente ed è diminuito con le iterazioni, come evidenziato dal grafico. L'errore medio commesso in questo caso è di 0.66, che risulta essere un buon risultato considerando l'elevato errore iniziale.

- Considerazioni sugli altri test con  $P_0 \neq X_0$ : in generale si è notato che, quando le osservazioni e il modello di transizione sono molto affidabili, lo stato iniziale non influisce in modo eccessivamente negativo sull'andamento del filtro rispetto alla traiettoria reale, ovvero il filtro viene aiutato notevolmente dalle osservazioni e dal modello di transizione per ritornare sul percorso corretto. Effettuando ulteriori test, aumentando il rumore sui sensori e sulle transizioni si è notato che, anche se inizialmente oscilla notevolmente intorno alla traiettoria reale, dopo un certo numero di iterazioni riesce a ritornare in prossimità del percorso corretto. Infine si è provato ad usare valori di rumore, sia delle

osservazioni e delle transizioni sia sullo stato iniziale, estremamente alti, ottenendo il seguente risultato

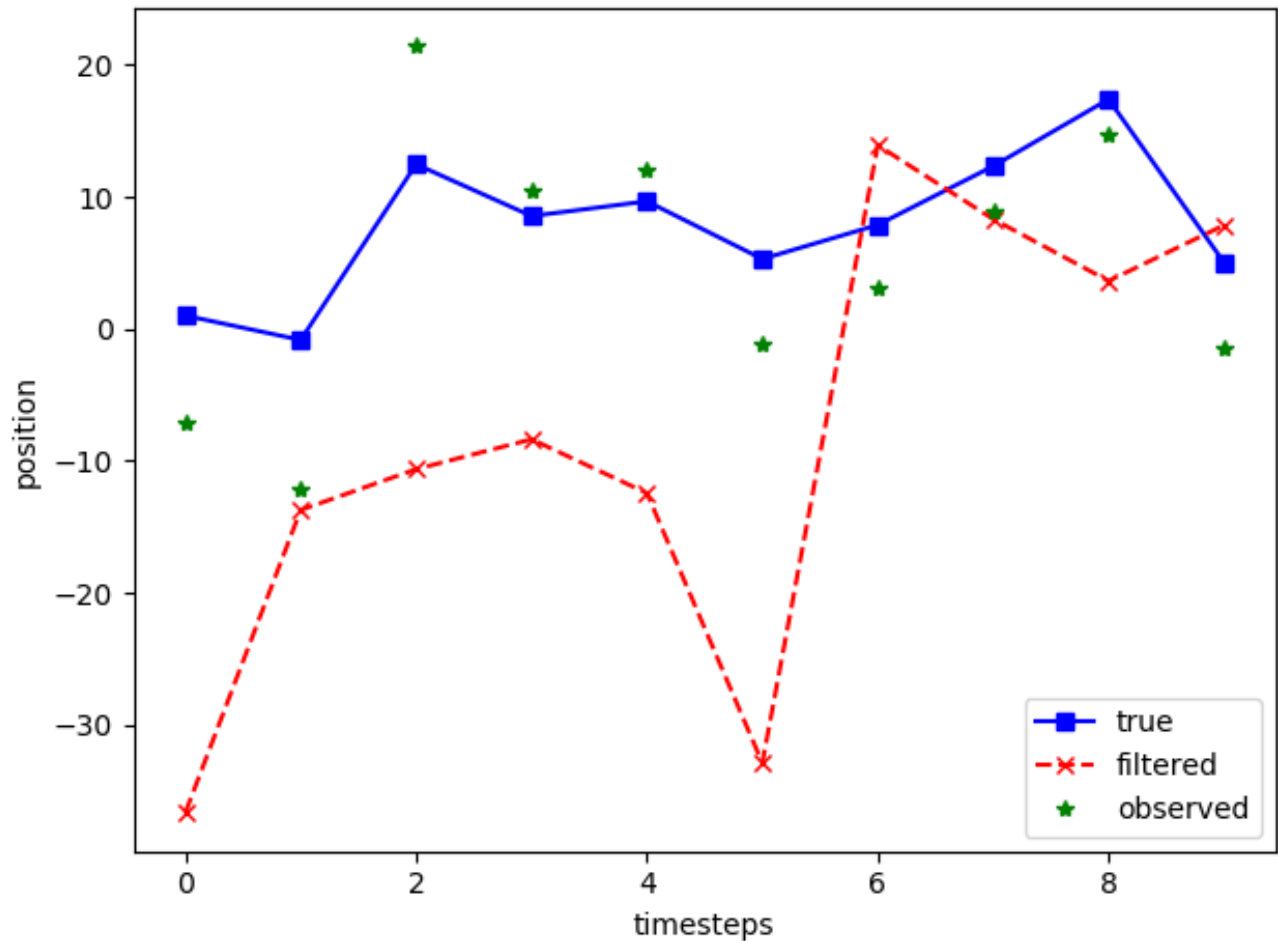


Figura 4.21: Rumore osservazioni  $\sigma_z = 60$  e rumore transizioni  $\sigma_x = 70$ , generati con una gaussiana con media  $\mu = 20$ . Partenza soggetta a rumore  $\sigma = 80\mu = 20$  e 10 iterazioni

Sorprendentemente, in sole 10 iterazioni il filtro si avvicina nuovamente al modello di transizione.

## 4.2.2 Comportamento con modello di transizione non lineare

Sappiamo che per quanto riguarda funzioni lineari e il prediction step si ha che:

$$\mathbf{P}(\mathbf{X}_t | \mathbf{e}_{1:t})$$

è Gaussiana, allora

$$\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t}) = \int_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{x}_t) P(\mathbf{x}_t | \mathbf{e}_{1:t}) d\mathbf{x}_t$$

è Gaussiana e che quindi l'aggiornamento della distribuzione:

$$\mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t+1}) = \alpha \mathbf{P}(\mathbf{e}_{t+1} | \mathbf{X}_{t+1}) \mathbf{P}(\mathbf{X}_{t+1} | \mathbf{e}_{1:t})$$

produce una Gaussiana.

Non possiamo dire lo stesso però per le funzioni non lineari, infatti se si dà in input ad una Gaussiana una funzione non-lineare, l'output non sarà più una Gaussiana e applicando il filtro di Kalman si otterrà che la distribuzione a posteriori crescerà in modo illimitato per  $t \Rightarrow \text{inf}$ , dove  $t$  sono i timesteps. Per questo test è stata generata una sequenza di stati che seguono una distribuzione esponenziale e delle osservazioni soggette ad un rumore basso per poter verificare come si comporta il filtro in situazioni normali. Il risultato ottenuto è il seguente:

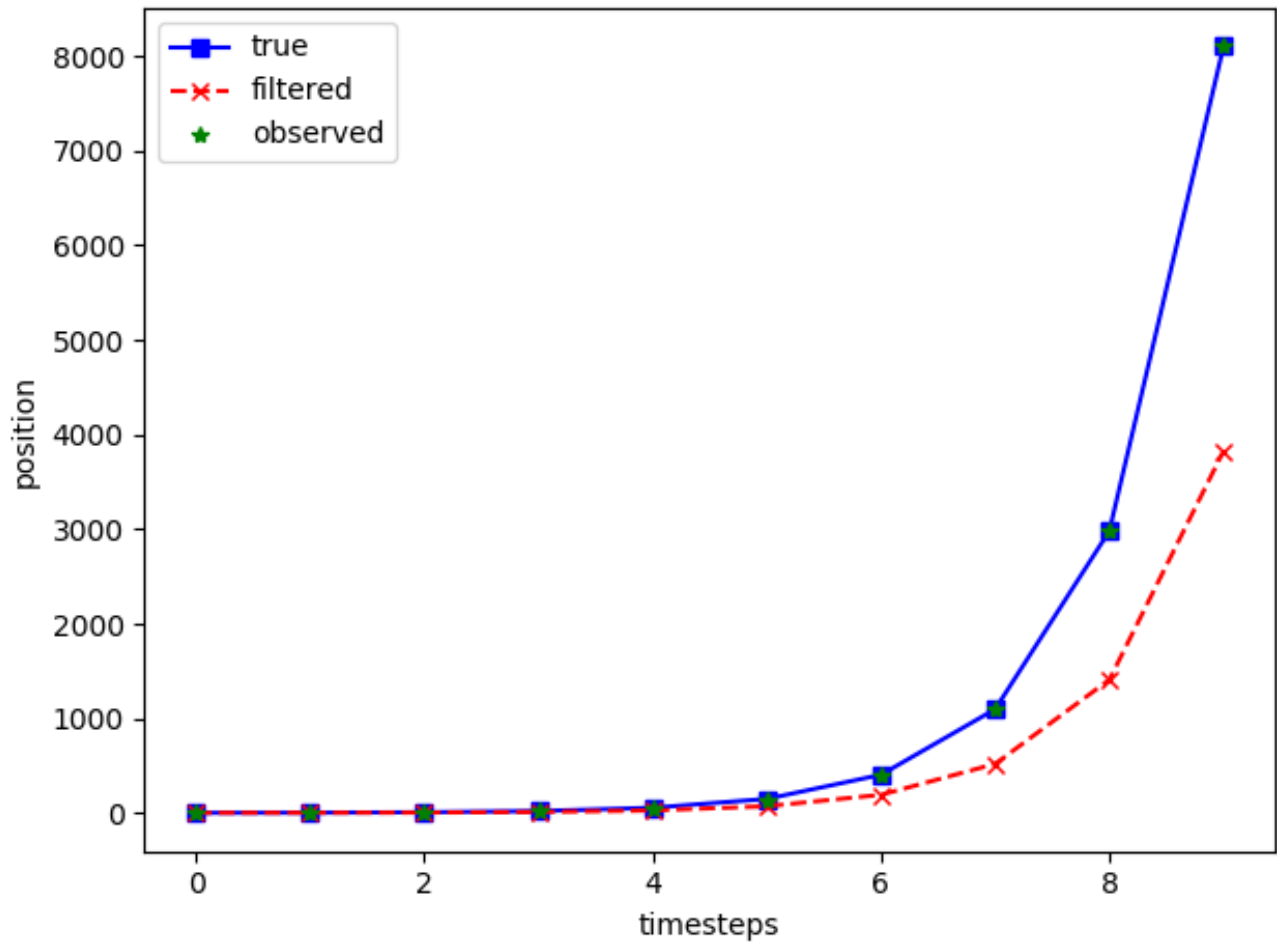


Figura 4.22: Test su distribuzione esponenziale. 10 iterazioni

Osservando il grafico si nota subito come a partire dal timestep 5 il filtro inizi ad allontanarsi sempre di più dalla traiettoria reale commettendo un errore sempre più grande e dimostrando quanto detto in precedenza.



```
Estimated error at time step 0 is 0.09020918830108193
Estimated error at time step 1 is 0.7588957708754394
Estimated error at time step 2 is 3.1194616172064524
Estimated error at time step 3 is 9.738979674585824
Estimated error at time step 4 is 27.879308835940666
Estimated error at time step 5 is 77.33173571128802
Estimated error at time step 6 is 211.89981313102666
Estimated error at time step 7 is 577.8365698478281
Estimated error at time step 8 is 1572.6986856075728
Estimated error at time step 9 is 4277.1571754810875
Mean error is 675.8510834865713
```

Figura 4.23: Errore del filtro applicato a funzione non lineare per ogni timestep ed errore medio in 10 iterazioni

Il filtro di Kalman standard quindi non risulta adatto a trattare funzioni non lineari, ma a questo scopo è possibile utilizzare l'Extended Kalman Filter che permette di ottenere un'approssimazione lineare delle funzioni non-lineare sfruttando lo sviluppo in serie di Taylor.

### 4.2.3 Note

La modifica della libreria è stata fatta alla riga 1168 e 1178 della funzione *filter* contenuta nel file `..\Lib\site-packages\pykalman-0.9.5-py3.6.egg\pykalman\standard.py`. In particolare, il valore di ritorno relativo al Kalman Gain nella funzione privata `_filter` veniva ignorato ("`_`" al posto di "`kalman_gain`" alla riga 1168). Si è deciso quindi di assegnargli una variabile e di restituirla insieme alle medie e le covarianze filtrate.

```
1168     (_, _, kalman_gain, filtered_state_means,
1169         filtered_state_covariances) = (
1170         _filter(
1171             transition_matrices, observation_matrices,
1172             transition_covariance, observation_covariance,
1173             transition_offsets, observation_offsets,
1174             initial_state_mean, initial_state_covariance,
1175             z
1176         )
1177     )
1178     return (filtered_state_means, filtered_state_covariances, kalman_gain)
```