**Università degli Studi di Torino**

Dipartimento di Informatica

Corsi di Laurea Magistrale in Informatica



**Relazione del progetto di Intelligenza**

**Artificiale e Laboratorio**

Salvatore Coluccia

a.a. 2018/2019

Indice

[1. Prolog 3](#_Toc528587092)

[1.1 Iterative Deepening 3](#_Toc528587093)

[1.2 A\* 3](#_Toc528587094)

[1.3 IDA\* 3](#_Toc528587095)

[1.4 Benchmark e considerazioni 3](#_Toc528587096)

[2.ASP 3](#_Toc528587097)

[2.1 Calendario torneo sportivo 3](#_Toc528587098)

[2.2 Trasporto Aereo 3](#_Toc528587099)

# Prolog

Per la parte del progetto relativa al linguaggio Prolog ho implementato 3 algoritmi di ricerca: Iterative Deepening, A\* e IDA\*.

Iterative Deepening appartiene alla famiglia degli algoritmi di ricerca non informata e quindi non utilizza alcun tipo di conoscenza del problema al di là della definizione del problema stesso.

A\* e IDA\* appartengono invece alla famiglia degli algoritmi di ricerca informata che quindi utilizzano una conoscenza aggiuntiva (in questo caso un’euristica) in modo da espandere prima i nodi che soddisfano determinate condizioni.  
La funzione euristica utilizzata è stata la distanza di Manhattan (*L1(P1,P2) = |x1-x2|+|y1-y2|*) perché soddisfa le condizioni di **ammissibilità** e **consistenza**.  
Un’euristica è ammissibile se non sbaglia mai per eccesso la stima del costo per arrivare all’obiettivo. Un’euristica è consistente (o monotona) se, per ogni nodo *n* e ogni successore *n0* di *n*, il costo stimato per raggiungere l’obiettivo partendo da *n* non è superiore al costo di passo per arrivare a *n0* sommato al costo stimato per andare da lì all’obiettivo.



***Figura 1.1:*** *La linea verde rappresenta la distanza euclidea mentre le altre indicano alcune possibili rappresentazioni geometriche della distanza di Manhattan (tutte equivalenti).*

I domini utilizzati per testare questi due algoritmi sono: dominio del *labirinto* e dominio del *percorso della Metropolitana*.

## Iterative Deepening

L’algoritmo di Iterative Deepening si basa sull’algoritmo di ricerca in profondità limitata. L’idea sulla quale si basa è quella di eseguire iterativamente l’algoritmo di ricerca in profondità limitata incrementando il limite ad ogni iterazione fino a trovare la soluzione (se esiste).

La ricerca a profondità limitata permette di non espandere rami infiniti che non portano ad alcuna soluzione in quanto al raggiungimento del limite imposto l’algoritmo termina di espandere altri nodi.

L’iterative deepening è un algoritmo completo ed ottimo in quanto …..

Nel caso in cui non esista alcuna soluzione non è in grado di accorgersene perché aumenterebbe all’infinito il limite di profondità.

Nell’implementazione si è deciso di tenere traccia dei nodi già espansi per evitare di ripercorrere rami già percorsi. Questo aumenta lievemente l’ammontare della memoria necessaria durante l’esecuzione dell’algoritmo ma riduce il tempo necessario ad eseguire l’intero algoritmo.

## A\*

## IDA\*

## Benchmark e considerazioni

# 2.ASP

## 2.1 Calendario torneo sportivo

## 2.2 Trasporto Aereo