



UNIVERSITÀ DEGLI STUDI
DI PERUGIA

Tesina di

Algoritmi e Strutture di dati

Corso di Laurea in Ingegneria Informatica ed Elettronica – A.A. 2020-2021

DIPARTIMENTO DI INGEGNERIA

docente

Prof. Emilio DI GIACOMO

**Implementazione e analisi dell'algoritmo A* per
problemi di ricerca del cammino minimo**

studenti

316649	Francesca	Nocentini	francesca.nocentini@studenti.unipg.it
312294	Paolo	Speziali	paolo.speziali@studenti.unipg.it

0. Indice

1 Problema Affrontato	2
-----------------------	---

1. Problema Affrontato

Il progetto si basa sull'implementazione dell'algoritmo di ricerca A^* in linguaggio Javascript V8.

Si tratta di un algoritmo di tipo greedy best-search, cioè utilizza un'euristica che permette di stimare, data una rete di percorsi, quello che è il cammino migliore verso un determinato traguardo. Tale algoritmo si avvale dell'utilizzo di grafi pesati (con pesi non negativi) e lo scopo è quello di partire da un vertice di partenza (*start*) per giungere a un vertice di arrivo (*goal*) impiegando il minor costo totale.

Nell'ambito delle problematiche di ricerca dei cammini minimi il primo pensiero può andare verso una soluzione implementata con il classico algoritmo di Dijkstra o con altri che svolgono una ricerca in ampiezza. Tuttavia la conoscenza di maggiori informazioni, nel nostro caso la posizione del punto di arrivo, ci permette di sfruttare questa soluzione decisamente meno onerosa dal punto di vista computazionale.

L'algoritmo A^* può essere considerato un algoritmo analogo a quello di Dijkstra con la differenza che, mentre quest'ultimo esplora tutti i possibili cammini, tra i nodi connessi, con un raggio d'azione circolare che va espandendosi fino a trovare il traguardo e su questi definisce quale è il migliore, A^* cerca direttamente il cammino minimo usando la funzione euristica e quindi dando priorità ai nodi che sono stimati essere migliori di altri. Ovviamente A^* funziona bene solo quando l'euristica che viene definita per i nodi è ammissibile, cioè non sovrastima mai la distanza effettiva verso la meta. Nel nostro caso come euristica ammissibile abbiamo utilizzato la distanza effettiva in linea d'aria tra i nodi e il goal.

A^* funziona in maniera ottimale se l'euristica è anche consistente, cioè tale che per ogni nodo N e per ogni suo vicino P del grafo risulta che

$$h(N) \leq h(P) + c(N, P)$$

dove $c(N, P)$ è il peso del grafo tra N e P .

La nostra euristica soddisfa tale proprietà in quanto è possibile costruire sempre un triangolo con i vertici N, P e il goal G e, sapendo che la lunghezza del lato NP è sempre minore della somma degli altri due, tale uguaglianza vale.

Possiamo dire che A* è sicuramente più efficace ed efficiente rispetto a Dijkstra, per grafi molto ampi, se si vuole calcolare un percorso ottimo tra due punti in maniera rapida, a scapito ovviamente di un maggiore utilizzo della memoria dovuto alla presenza dell'euristica. Se si hanno a disposizione le informazioni che ci permettono di definire l'euristica, A* è preferibile.

Dijkstra è sicuramente migliore per problemi di pathfinding in quanto viene trovato, se esiste, il percorso migliore, confrontandone molti tra loro, a discapito della rapidità con cui questo viene elaborato.