

Risoluzione puzzle gioco del 15 tramite algoritmi di ricerca

Presciani Davide, matricola 1035189 - Università degli studi di Bergamo

Sommario

Questo progetto si pone come obbiettivo l'implementazione di algoritmi di ricerca al fine di risolvere il puzzle "gioco del 15".

1. INTRODUZIONE

Il gioco del 15, consiste in una griglia 4x4 contenente numeri che vanno da 1 a 15; L'obbiettivo di questo gioco, è il posizionare in ordine crescente i numeri, spostando le caselle utilizzando lo spazio vuoto presente nella griglia.

Questo gioco può essere trovato anche in varianti con una griglia di dimensioni diverse, in generale, una griglia di dimensioni $N \times N$ conterrà numeri che vanno da 1 a $N^2 - 1$.

Questo gioco può essere rappresentato in modo grafico considerando la configurazione dello schema come un nodo e il movimento di una tessera come arco. Una volta rappresentato questo gioco in modo grafico, è possibile utilizzare gli algoritmi di ricerca su grafo per risolvere il gioco, inoltre, con specifici algoritmi si può trovare una soluzione che presenti il minor numero di mosse da eseguire al fine di risolvere lo schema dato.

L'utilizzo di algoritmi di ricerca è necessario dato che ogni schema ha una soluzione ottima che richiede tra le 0 e le 80 mosse [1]; questo con un branching factor medio di 3 porta a dover esplorare circa 10^{38} stati (10^{13} se vengono considerate le configurazioni un'unica volta).

L'implementazione degli algoritmi è avvenuta tramite *python*; in particolare lo schema è rappresentato come un array di 16 elementi, ognuno di questi elementi è una delle tessere (il numero 0 viene utilizzato per rappresentare la cella vuota).

2. Algoritmo di ricerca non informata: ricerca a profondità limitata

Come primo algoritmo, si è scelto di utilizzare un algoritmo di ricerca a profondità limitata, in modo da

poter valutare il numero di stati che si dovrebbero esplorare, utilizzando un algoritmo di ricerca non informata. Inoltre, si vuole analizzare il tempo necessario per esplorare tutte le configurazioni fino ad una determinata profondità. Per effettuare queste misure, è stato generato uno schema che necessita di un numero di spostamenti per arrivare alla soluzione maggiore alla profondità massima fissata.

Profondita' massima	Partenza angolo	Partenza lato	Partenza centro
1	2	3	4
2	7	11	17
3	30	50	50
4	66	87	66
5	415	164	415
6	421	590	421
7	811	811	811
8	3227	3227	3227
9	3887	3887	3887
10	19475	19475	19475
11	39056	39056	39056
12	173553	173553	173553
13	342136	342136	342136
14	1606975	1606975	1606975
15	2113660	2113660	2113660
16	4109565	4109565	4109565
17	37864245	37864245	37864245

Tabella 1. Numero nodi esplorati in relazione della posizione della cella vuota

Nella tabella 1, sono riportati il numero di nodi esplorati, in funzione della profondità massima e della posizione di partenza dello spazio vuoto. In particolare, si può notare come, fino ad una profondità pari a 6, il numero di nodi da esplorare è influenzato dalla posizione di partenza, dato che l'algoritmo non ha ancora espanso tutte le celle presenti nello schema. Si può inoltre notare come il numero di nodi da esplorare aumenta in modo simile ad un esponenziale.

Nella tabella seguente, sono riportate le medie e le varianze dei tempi per l'esplorazione di tutte le configurazioni ad una determinata profondità massima; si può

notare, come ad una profondità massima pari a 16, la varianza dei tempi è pari al 20% della media di questi ultimi. Questa varianza, si presenta benché l'ordine degli stati esplorati non varia da iterazione ad iterazione dell'algoritmo; inoltre, lo schema sul quale è eseguito l'algoritmo è invariato tra le iterazioni.

Questa varianza, si può ipotizzare essere dovuta, dalla presenza di altri processi in esecuzione sul calcolatore utilizzato per eseguire l'algoritmo. Bisogna inoltre far notare, che questi risultati variano molto in base alle specifiche della macchina sul quale viene eseguito l'algoritmo. Vista dunque l'alta variabilità dei tempi di esecuzione, su di un algoritmo che non presenta randomicità, non verranno riportate altre analisi temporali sugli algoritmi utilizzati d'ora in avanti visto che questi conterranno elementi di randomicità.

Profondità massima	Media tempi	Varianza tempi
1	9,81E-05	9,62E-08
2	0,00050	2,79E-07
3	0,00160	7,06E-07
4	0,00220	1,77E-07
5	0,00460	7,06E-07
6	0,01798	7,34E-06
7	0,02378	1,04E-05
8	0,09225	6,79E-05
9	0,10305	4,75E-05
10	0,51579	0,00115
11	1,07553	0,00380
12	4,80080	0,07402
13	9,29271	0,23402
14	43,43936	5,93188
15	57,66461	9,23124
16	113,69547	22,02420

Tabella 2. Medie e varianze dei tempi di esecuzione dell'algoritmo di ricerca a profondità limitata

3. Descrizione delle euristiche utilizzate

Per ridurre il numero di stati esplorati, si utilizzano algoritmi di ricerca informata; questi algoritmi sono composti da una funzione di ricerca e di una funzione di conoscenza (funzione euristica).

Il compito della funzione euristica, è lo stimare il numero di passi necessari, da una determinata configurazione, per raggiungere una soluzione (che può non essere ottima). Vengono dette euristiche ammissibili quelle euristiche che non sovrastimano il costo effettivamente necessario per raggiungere l'obiettivo [2, p. 129].

Le euristiche utilizzate sono:

- *Numero di celle fuori posto (Distanza di Hamming);*
- *Distanza di Manhattan;*
- *Distanza di Manhattan con inversioni.*

La prima euristica (Distanza di Hamming), fornisce una stima della distanza dalla soluzione, calcolando per ogni schema, il numero di celle che non sono in posizione corretta. Questa euristica è ammissibile in dato che il numero di mosse da eseguire per raggiungere la soluzione, sarà almeno uguale al numero di celle fuori posto[4].

La seconda euristica utilizzata (distanza di Manhattan), stima la distanza della soluzione calcolando il numero di mosse che ogni tessera deve compiere, per raggiungere la posizione corretta. Questa euristica è ammissibile perché è sempre minore o uguale al numero di volte per cui ogni tessera dovrà essere mossa[4].

L'ultima delle euristiche utilizzate, è una modifica all'euristica distanza di Manhattan. Oltre a calcolare la distanza di Manhattan, si calcolano le inversioni che vanno effettuate sulla stessa linea o colonna e, per ogni inversione, si aumenta la distanza di 2. Questa euristica rappresenta l'effettivo numero di movimenti, che è necessario eseguire per invertire la posizione di due tessere che sono sulla stessa linea, o colonna ed è quindi ammissibile.

4. Comparazione delle euristiche

Per comparare le euristiche utilizzate sono stati generate 100 diversi schemi applicando ad una griglia risolta 15 mosse pseudocasuali; possiamo dunque aspettarci che la soluzione ottima per ogni schema sia attorno alle 15 mosse.

4.1. Distanza di Hamming VS distanza di Manhattan

Le prime euristiche che si è deciso di comparare sono la distanza di Hamming, che non presenta un'euristica particolarmente intelligente al fine di risolvere questo puzzle, e la distanza di Manhattan.

In entrambi i casi, le due euristiche vengono utilizzate per scegliere la prossima configurazione da esplorare da un algoritmo di ricerca greedy. In caso vi siano più configurazioni rilevate dall'euristica, l'algoritmo ne espande una scegliendo randomicamente tra quelle trovate dall'euristica.

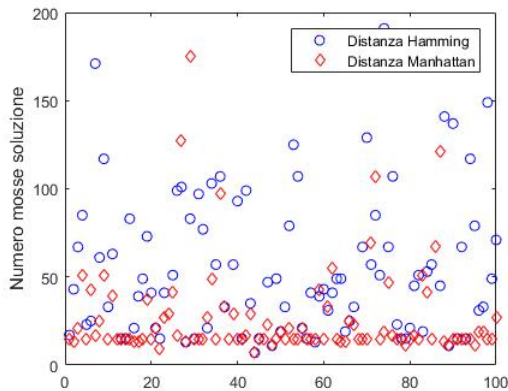


Figura 1. Confronto lunghezze soluzioni

Lunghezza soluzione		
	Hamming	Manhattan
Media	352,0	26,82
10° percentile	15	13
50° percentile	49	15
90° percentile	139	51

Tabella 3. Media e percentili delle lunghezze delle soluzioni

Come possiamo vedere nella tabella 3, utilizzando l'euristica distanza di Manhattan, otteniamo soluzioni con un minor numero di mosse da eseguire. Inoltre, come mostrato in figura 2 e nella tabella 4, l'euristica distanza di Manhattan esplora un numero molto minore di stati rispetto alla distanza di Hamming.

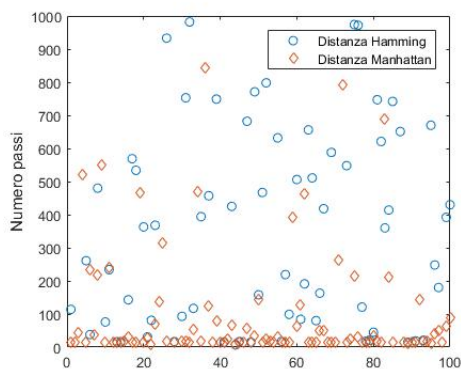


Figura 2. Confronto configurazioni esplorate

Numero di stati esplorati		
	Hamming	Manhattan
Media	1106,1	142,16
10° percentile	18,5	14,5
50° percentile	558,5	18
90° percentile	3236,5	467

Tabella 4. Media e percentili del numero di configurazioni esplorate

Come riportato nella tabella 4, si può notare che la distanza di Manhattan inoltre tende ad avere outliers più contenuti rispetto alla distanza di Hamming. Ne risulta quindi, che la distanza di Manhattan è un'euristica caratterizzata da una consistenza maggiore rispetto all'euristica distanza di Hamming.

4.2. Distanza di Manhattan VS distanza di Manhattan con inversioni

Passiamo ora a comparare l'euristica distanza di Manhattan con una sua versione raffinata, ossia una versione che considera anche i passi necessari per invertire di posto due celle sulla stessa linea o colonna.

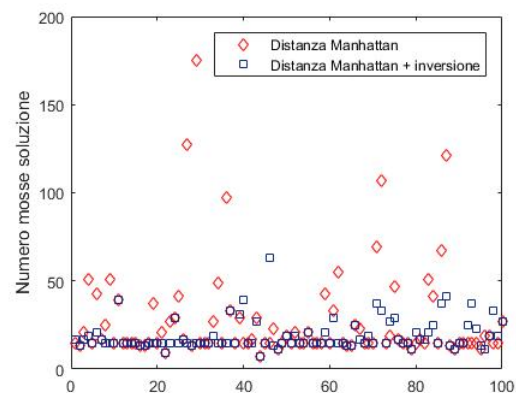


Figura 3. Confronto lunghezze soluzioni

Lunghezza soluzione		
	Manhattan	Manhattan + inversioni
Media	26,82	19
10° percentile	13	13
50° percentile	15	15
90° percentile	51	32

Tabella 5. Media e percentili delle lunghezze delle soluzioni

Comparando i risultati, si può osservare che l'introduzione di informazioni aggiuntive, ha portato ad avere un'euristica che si avvicina al numero di passi ottimo. Analizzando la tabella 6 e la figura 4, si può inoltre affermare che l'aggiunta di informazioni ha velocizzato il processo di ricerca riducendo il numero di stati esplorati.

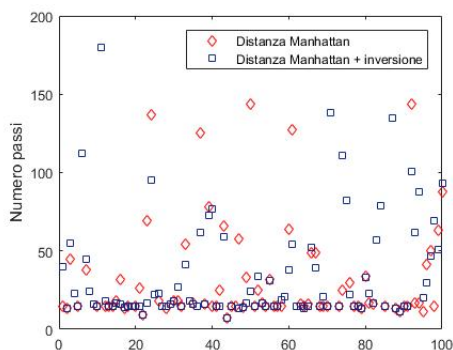


Figura 4. Confronto configurazioni esplorate

Numero di stati esplorati		
	Manhattan	Manhattan + inversioni
Media	142,16	40,22
10° percentile	14,5	14,5
50° percentile	18	17,5
90° percentile	467	94

Tabella 6. Media e percentili del numero di configurazioni esplorate

5. Algoritmo A*

L'algoritmo di ricerca A*, differisce dai precedenti algoritmi per via di come viene calcolata l'euristica: oltre ad usare il valore dell'euristica scelta, somma il numero di passi necessari a raggiungere la configurazione corrente. Intuitivamente, si può dimostrare che A* è ottimo rispetto ad altri algoritmi di ricerca ammissibili. A* stima ottimisticamente il costo del percorso, in particolare, quando A* termina la sua ricerca, per definizione, avrà trovato un percorso il cui costo attuale è più basso del costo stimato per ogni altro percorso attraverso tutte le altre configurazioni [3].

5.1. Comparazione distanza di Manhattan con inversioni e algoritmo A*

Analizziamo ora la lunghezza delle soluzioni trovate dall'algoritmo che usa l'euristica distanza di Man-

hattan con inversioni e l'algoritmo A* con la medesima euristica.

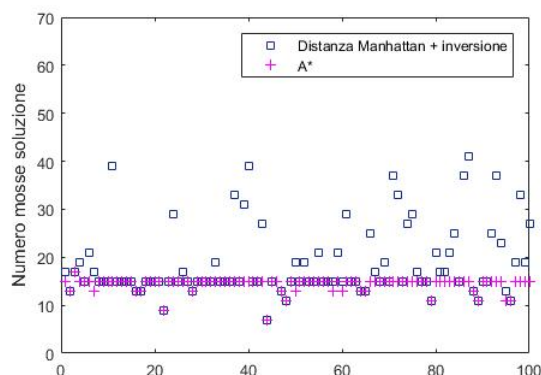


Figura 5. Confronto lunghezze soluzioni

Lunghezza soluzione		
	Manhattan + inversioni	A*
Media	19	14,44
10° percentile	13	13
50° percentile	15	15
90° percentile	32	15

Tabella 7. Media e percentili delle lunghezze delle soluzioni

Analizzando la figura 5, si può notare come l'algoritmo A* riesce a trovare una soluzione sempre più corta o uguale all'algoritmo di ricerca greedy che utilizza come euristica la distanza di Manhattan con inversioni. Di seguito sono riportati un'immagine ed una tabella che riportano il numero di stati esplorati dai due algoritmi.

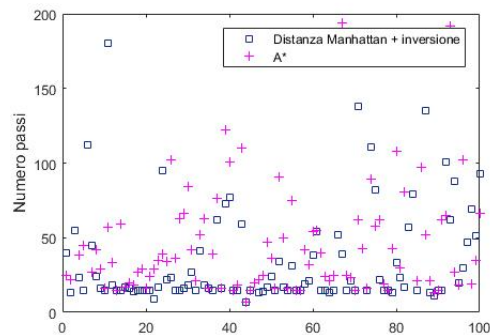


Figura 6. Confronto configurazioni esplorate

	Numero di stati esplorati	
	Manhattan + inversioni	A*
Media	40,22	71,5
10° percentile	14,5	15,5
50° percentile	17,5	35,5
90° percentile	94	109

Tabella 8. Media e percentili del numero di configurazioni esplorate

Rispetto ai casi precedenti, si può notare che l'algoritmo A* esplora mediamente un numero maggiore di stati rispetto ad un algoritmo greedy. Aumentando il numero di mosse per generare gli schemi partendo da uno schema risolto, si è notato come questo fenomeno si inverte e, quindi, l'algoritmo A* esplora mediamente un numero minore di stati rispetto a tutti gli altri algoritmi; in alcuni casi però l'algoritmo A* arriva ad esplorare circa 1000 configurazioni risultando più lento degli algoritmi che usando un'euristica basata sul calcolo della distanza di Manhattan.

6. Analisi ripetizione algoritmi su stessa configurazione

Come ultima analisi, vista la presenza di randomicità negli algoritmi, sono state eseguite 20 iterazioni degli algoritmi su di uno stesso schema, generato applicando 20 mosse pseudocasuali ad uno schema risolto, al fine di valutare come la randomicità influenzasse la lunghezza delle soluzioni trovate e il numero di stati esplorati.

	Lunghezza soluzione			
	Hamming	Manhattan	Manhattan + inversioni	A*
Media	39,6	16,6	17	15
10° per.	17	15	15	15
50° per.	17	17	17	15
90° per.	82	17	17	15

Tabella 9. Media e percentili delle lunghezze delle soluzioni

Analizzando i dati nelle tabelle 9 e 10 si può notare come tutti gli algoritmi escluso A* vengono influenzati dalla randomicità nella generazione dei risultati; inoltre come evidenziato nel capitolo precedente, A* esplora un numero maggiore di stati rispetto agli algoritmi che si basano sulla distanza di Manhattan.

	Numero di stati esplorati			
	Hamming	Manhattan	Manhattan + inversioni	A*
Media	471,7	57,2	27	97,5
10° per.	51,5	17	16,5	87
50° per.	162	26,5	22	97
90° per.	1297	120,5	45	108

Tabella 10. Media e percentili del numero di configurazioni esplorate

7. Conclusioni

In questo progetto sono stati implementati alcuni degli algoritmi visti a lezione per meglio comprenderne il funzionamento, i metodi per implementarli e per avere dei dati per riuscire a verificare quanto visto durante il corso. Nel confrontare i dati si è potuto inoltre constatare come l'algoritmo A*, che genera soluzioni ottime, non è l'algoritmo che esplora il minor numero di stati. Si potrebbe inoltre aumentare la velocità di esecuzione del codice utilizzando strutture diverse per la memorizzazione dei dati: gli algoritmi utilizzati sono stati implementati utilizzando degli array, strutture non ottimizzate per la ricerca, utilizzando strutture ottimizzate, si può dunque supporre che l'esecuzione diventi più rapida, visto che la ricerca di un elemento in un array ha costo $O(n)$, mentre utilizzando strutture ottimizzate (come ad esempio delle mappe) si può arrivare ad avere un costo per ogni ricerca pari a $O(1)$.

Riferimenti bibliografici

- [1] Brügger, A., Marzetta, A., Fukuda, K., and Nievergelt, J. (1999). The parallel search benchmark and its applications. *Annals of Operations Research*, 90:45–63.
- [2] Russell, S. J. and Norvig, P. (2005). *Intelligenza artificiale. Un approccio moderno*, volume 1. Pearson Italia Spa.
- [3] Wikipedia (2018a). Algoritmo a* — wikipedia, l'enciclopedia libera. [Online; in data 13-novembre-2018].
- [4] Wikipedia (2018b). Euristica ammissibile — wikipedia, l'enciclopedia libera. [Online; in data 13-novembre-2018].