Disjoint databases applicati al gioco del 15

Marco Benelli

Febbraio 2020

1 Introduzione

Per risolvere il gioco del quindici con A* possiamo molte euristiche, fra cui una che è stata introdotta per la prima volta da Korf e Felner nel 2002. [2] Questa nuova euristica è data dalla somma di altre euristiche. Ciò può sembrare strano a primo avviso, visto che preferiamo avere euristiche consistenti (o per lo meno ammissibili), ma la somma di due euristiche consistenti non è obbligatoriamente ammissibile. Tuttavia, per come sono definite le euristiche consistenti di partenza, possiamo sommarle ed ottenere una nuova euristica a sua volta consistente.

2 Le euristiche più semplici

Di seguito sono descritte delle euristiche che venivano utilizzate storicamente per risolvere il gioco del quindici e le sue varianti (in particolare il gioco dell'otto). Tutte queste sono state descritte da Russell e Norvig. [1]

2.1 Misplaced heuristic

Un'euristica molto semplice è data dal numero di celle che non sono al posto corretto. Questa euristica è ammissibile, visto che tutte le celle che non sono al proprio posto dovranno essere mosse almeno una volta per arrivare all soluzione. Inoltre è anche ammissibile, visto che facendo una mossa, il numero di celle al posto sbagliato non può aumentare di più di 1 (che è il costo di ogni mossa).

Questa euristica, come è facile immaginare, tende a sottostimare di molto il numero di mosse necessario per arrivare alla soluzione: in molti casi il numero effettivo di mosse è più del doppio del valore dato dall'euristica. Questo significa che trovare una soluzione usando questa euristica in tempi ragionevoli è realistico solamente per il gioco dell'otto.

2.2 Manhattan heuristic

Per calcolare un'euristica migliore della precedente possiamo trovare la somma delle distanze Manhattan di ogni cella dalla posizione corrente alla posizione

finale. La distanza Manhattan fra due celle (i_1, j_1) e (i_2, j_2) è data da $|i_1 - i_2| + |j_1 - j_2|$. Questa euristica è ammissibile, visto che se consideriamo un rilassamento del problema in cui i pezzi del puzzle possono muoversi indipendente gli uni dagli altri, si avrebbe che la distanza dalla soluzione sarebbe data dal valore di questa euristica. Inoltre questa è anche consistente poiché il valore dell'euristica dopo aver fatto una mossa può aumentare al massimo di 1.

È facile notare che questa euristica domina l'euristica precedente e visto che il costo per calcolarla, se n è la lunghezza del lato del puzzle, è $O(n^2)$ (lo stesso di misplaced heuristic), non ci sono svantaggi nell'usarla. Con questa euristica diventa realistico risolvere il gioco del quindici anche se i tempi sono comunque lunghi.

2.3 Linear conflict heuristic

Questa euristica si basa su un miglioramento dell'euristica di Manhattan, per spiegarla partiamo da un esempio. Supponiamo che la prima riga inizi con le caselle 2 e 1, in questo caso la euristica di Manhattan dice che la casella 2 deve essere spostata almeno 2 volte, mentre la 1 almeno 0. Tuttavia, riusciamo a intuire che oltre alle 2 mosse orizzontali per portare il 2 nella posizione obiettivo, dobbiamo fare anche 2 mosse verticali poiché se non spostiamo l'1 il 2 non può portarsi alla sua destra.

Anche questa euristica è molto veloce da calcolare ed è consistente e a sua volta domina quella Manhattan.

3 Disjoint databases

Supponiamo di voler trovare delle euristiche in altri modi. Invece di definirle come distanza dalla soluzione di un rilassamento del problema, proviamo a guardare dei sottoproblemi. Un sottoproblema del gioco del quindici potrebbe essere portare al posto giusto le caselle da 1 a 7, mentre un altro quello di risolverlo per le caselle da 8 a 15. Abbiamo dunque due euristiche consistenti, ma non possiamo sommarle per ottenerne una ancora consistente, possiamo solamente prenderne il massimo.

Per poterle sommare dobbiamo fare delle piccole modifiche ai sottoproblemi. Prendiamo per esempio il problema di posizionare al posto corretto le caselle da 1 a 7, però, anziché avere come costo di ogni mossa 1, prendiamo come costo 1 se abbiamo mosso una delle caselle interessate e 0 altrimenti. In questo modo, visto che ogni euristica tiene conto solamente delle mosse che interessano il proprio sottoinsieme di caselle, possiamo essere certi che la somma delle euristiche sia a sua volta un'euristica consistente.

Questa euristica domina sicuramente la euristica Manhattan, ma soffre di un problema: il tempo impiegato per calcolarla è troppo grande, se dovessimo calcolarla per ogni stato durante la ricerca, sarebbe peggiore delle altre euristiche viste fino ad ora. Siamo fortunati, però: possiamo calcolare le due euristiche per

euristica	nodi	secondi	nodi/sec
Manhattan distance	299379854	27654	10826
linear conflicts	31789159	5923	5367
disjoint database	102156	31	3295
disjoint + reflected	31168	18	1732

Table 1: I risultati sperimentali

ogni sottoproblema possibile e memorizzarle in due database per non calcolarle più.

È importante notare che avremmo potuto scegliere una qualunque partizione di caselle. Maggiore il numero di sottoinsiemi e peggiore sarà l'euristica finale (con un sottoinsieme per ogni casella torneremmo alla euristica Manhattan), ma più veloce sarà il calcolo dei database (e più piccoli saranno i database).

Una volta deciso il numero di sottoinsiemi, dobbiamo scegliere quali caselle mettere in ciascuno. Per avere risultati migliori dobbiamo fare in modo che molte celle "vicine" appartengano allo stesso sottoinsieme (questo è il motivo per cui abbiamo scelto come sottoinsiemi $\{1, ..., 7\}$ e $\{8, ..., 15\}$.

3.1 Un ulteriore miglioramento

Se pensassimo di avere un'altra partizione fatta da $\{1, 4, 5, 8, 9, 12, 13\}$ e $\{2, 3, 6, 7, 10, 11, 14, 15\}$, possiamo ricavare i valori dell'euristica relativa dal database dell'altra partizione. Questo è possibile perché le due partizioni sono l'una la simmetrica dell'altra rispetto alla diagonale principale.

4 Conclusione

Come possiamo vedere dalla tabella 1, i risultati dimostrano che il metodo di usare disjoint databases è il migliore e può essere velocizzato ancora di più sfruttando le simmetrie di cui abbiamo parlato prima.

References

[1] Stuart Russell Peter Norvig. *Artificial Intelligence: A Modern Approach.* 1995.

[2] Ariel Felner Richard E. Korf. Disjoint pattern database heuristics. 2002.