Università degli Studi di Udine
Dipartimento di Matematica e Informatica

Un piccolo problema di planning: Brain Twist

Andrea Viel

Anno accademico 2015-2016

Introduzione

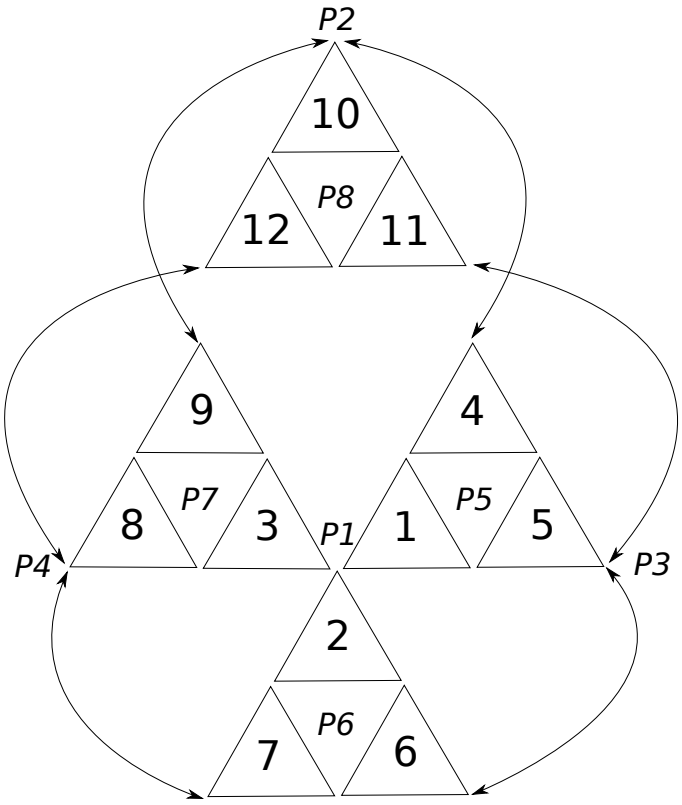
- L'obiettivo del gioco è quello di avere un colore uniforme su tutte le facce (alternativamente sulle punte).
- E' simile a un cubo di Rubik, solo piramidale.

Il gioco

- E' possibile ruotare i vertici in senso orario o antiorario.
- Le tessere hanno un colore esterno e interno, dunque il gioco può aprirsi (flippare) e mostrare il colore che prima era nascosto all'interno.

Demo

Rappresentazione del gioco



- Ci sono 12 tessere per 12 posizioni.
- Ci sono 4 vertici in una configurazione e altri 4 che appaiono dopo il flip.

Rappresentazione del gioco

```
%passi del gioco
time(1..n).

%le tessere colorate di cui è composto il puzzle
%hanno un colore esterno e uno interno
tess(rc). tess(bc). tess(fc).
tess(rg). tess(ag). tess(fg).
tess(as). tess(bs). tess(fs).
tess(rv). tess(av). tess(bv).

%colori esterni
rosso(rc;rg;rv).
blu(bc;bv;bs).
fucsia(fg;fc;fs).
arancione(ag;as;av).

%colori interni
celeste(rc;bc;fc).
verde(rv;av;bv).
giallo(rg;ag;fg).
smeraldo(as;bs;fs).
```

Rappresentazione del gioco

```
%se è flippato oppure no, da cui dipende se è visibile il
    colore interno o esterno
flip(0..1).

% 12 posizioni per i 12 tasselli
pos(1..12).

% 8 vertici ruotabili, 4 per configurazione. La mossa 0 è il
     flip
mossa(-8..8). %negativo significa rotazione antioraria

% una configurazione esistente per ogni istante
%all'istante Ti la tessera X sta nella posizione P
1{on(Ti,X,P):pos(P)}1:-time(Ti),tess(X).

%all'istante Ti il gioco è flippato o no
1{on(Ti,f,F):flip(F)}1:-time(Ti).
```

La scelta

```
%sceglie una singola azione non deterministicamente a ogni
    istante in base al flip
scelto(X,Ti) :- time(Ti), mossa(X), possibile(X,Ti), not
    diff_scelto(X,Ti).
diff_scelto(X,Ti) :- time(Ti), mossa(X), mossa(Y), possibile
    (X,Ti),possibile(Y,Ti), scelto(Y,Ti), X!=Y.

possibile(0,Ti):- time(Ti). %flip è sempre possibile

%se flip=0 ho accesso ai vertici 1..4
possibile(X,Ti):- mossa(X), time(Ti), on(Ti,f,0), X >= -4, X
    !=0, X <= 4.

% e viceversa flip=1 accedo ai vertici 5..8
possibile(X,Ti):- mossa(X), time(Ti), on(Ti,f,1), X >= 5.
possibile(X,Ti):- mossa(X), time(Ti), on(Ti,f,1), X <= -5.
```

Vincoli Aggiuntivi

```
% evitare mosse ridondanti
:- scelto(X,Ti+1),scelto(X,Ti),mossa(X),time(Ti). %fare due
    volte la stessa mossa non ha senso se è un flip ma
    nemmeno se è una rotazione perchè è come farne una sola
    ma in senso opposto
:- scelto(X,Ti+1),scelto(-X,Ti),mossa(X),time(Ti), X != 0. %
    non ha senso fare e disfare una rotazione

% vincoli aggiuntivi, non strettamente necessari ma
    velocizzano la ricerca
1{scelto(X,Ti):mossa(X)}1:-time(Ti).          %vincolo 1
8{diff_scelto(X,Ti):mossa(X)}8:-time(Ti).     %vincolo 2
9{possibile(X,Ti):mossa(X)}9:-time(Ti).       %vincolo 3
```

Le mosse

Flip

```
%flippare cambia solo il flip ma non lo stato delle tessere
on(Ti+1,f,0):-
scelto(0,Ti),on(Ti,f,1),time(Ti).
on(Ti+1,f,1):-
scelto(0,Ti),on(Ti,f,0),time(Ti).
on(Ti+1,X,P):-
scelto(0,Ti),on(Ti,X,P),time(Ti),tess(X),pos(P).

%viceversa ogni altra mossa non cambia il flip
on(Ti+1,f,F):-
scelto(R,Ti),on(Ti,f,F),time(Ti),flip(F),mossa(R), R != 0.
```

Le mosse

Numerazione delle posizioni

- Un'ottimizzazione consiste nel considerare la posizione 1, in ogni istante, quella in cui si trova la tessera **rc** e le altre di conseguenza.
- Complica la codifica delle rotazioni dei vertici ma semplifica la ricerca eliminando le simmetrie.
- Le configurazioni possibili finali diventano solamente due al posto di 24.

Le mosse

Rotazione

```
%la posizione 1 è sempre occupata da rc
on(Ti,rc,1):-time(Ti).

%la tessera in posizione 2 è sostituita nelle rotazioni 5 e
    6
on(Ti+1,X,2):-
scelto(5,Ti),on(Ti,X,9),time(Ti),tess(X). %significa che con
     la rotazione 5 la tessera che stava in posizione 9 va
    in posizione 2
on(Ti+1,X,2):-
scelto(6,Ti),on(Ti,X,7),time(Ti),tess(X).
on(Ti+1,X,2):-
scelto(-6,Ti),on(Ti,X,6),time(Ti),tess(X).
on(Ti+1,X,2):-
scelto(-5,Ti),on(Ti,X,11),time(Ti),tess(X).

%con le altre rotazioni non si muove
on(Ti+1,X,2):-
scelto(R,Ti),on(Ti,X,2),time(Ti),tess(X), mossa(R), R != 5,
    R != 6, R != 0, R != -5, R != -6.
```

Goal

```
goal :- on(n + 1 ,X2,2), on(n + 1 ,X3,3), on(n + 1 ,X4,4),
on(n + 1 ,X5,5), on(n + 1 ,X6,6), on(n + 1 ,X7,7), on(n + 1
    ,X8,8),
on(n + 1 ,X9,9), on(n + 1 ,X10,10), on(n + 1 ,X11,11), on(n
    + 1 ,X12,12),
%facce stesso colore
out_color(rc,X4,X5), out_color(X2,X6,X7), out_color(X3,X8,X9
    ), out_color(X10,X11,X12),
in_color(rc,X2,X3), in_color(X4,X9,X10), in_color(X5,X6,X11)
    , in_color(X7,X8,X12),
%punte stesso colore
%in_color(rc,X4,X5), in_color(X2,X6,X7), in_color(X3,X8,X9),
    in_color(X10,X11,X12),
%out_color(rc,X2,X3), out_color(X4,X9,X10), out_color(X5,X6,
    X11), out_color(X7,X8,X12),
tess(X2), tess(X3), tess(X4), tess(X5), tess(X6),
tess(X7), tess(X8), tess(X9), tess(X10), tess(X11), tess(X12
    ).
```

Fine