25 ZUM – Dijkstrův algoritmus, heuristiky pro odhad ceny cesty, hladové prohledávání, A* algoritmus.

Matej Choma

June 3, 2019

V tejto otázke budeme hľadať čo najkratšie cesty v stavovom priestore s ohodnotenými akciami. Tento priestor si môžeme predstaviť ako neorientovaný graf s ohodnotenými hranami.

Definícia 1 (Stavový priestor s ohodnotenými akciami) *Nech (S, A) je stavový priestor, kde S je množina javov a A je množina akcií. Trojicu (S, A, c), kde c je funkcia $A \to \mathbb{R}_0^+$, nazývame stavový priestor s ohodnotenými akciami. Hodnota $c(a)$ priraďuje cenu každej akcií $a \in A$.*

Všetky spomenuté algoritmy pracujú s expanziou stavu. Pri expanzií pridáme do poolu možných nasledujúcich stavov $OPEN$, všetkých susedov expanzovaného stavu, ktorých sme ešte neobjavili.

1 Dijkstrov algoritmus

Dijkstrov algoritmus nájde najkratšiu cestu z počiatočného stavu do všetkých ostatných v čase $\mathcal{O}(|A| + |S| \cdot \log(|S|))$.

Myšlienka algoritmu spočíva v udržovaní si najkratšej známej cesty $g(s)$ z počiatočného vrcholu do vrcholu s pre všetky vrcholy $s \in OPEN$, ktoré sme už objavili. Vždy sa expanduje stav $s^* \in OPEN$ s najkratšou cestou z počiatočného stavu

$$s^* \in \arg \min_{s \in OPEN} g(s).$$

Pokiaľ sme expanziou našli kratšiu cestu do nejakého stavu $s \in OPEN$, dĺžku cesty upravíme. Rozmyslite si, že v momente expanzie stavu už nemôže existovať kratšia cesta do tohto stavu ako tá, ktorú poznáme. Vizualizácia behu algoritmu sa dá nájsť na `https://www.cs.usfca.edu/~galles/visualization/Dijkstra.html`.

2 Informované prehľadavánie stavového priestoru

Dijkstra je ako BFS berúce odhad na dĺžku cesty od počiatočnéhos tavu. Nájde síce optimálne riešenie, no expanduje príliš veľa vrcholov na všetky strany a teda je v praxi pomalý. Jeden so spôsobov ako expandovať vrcholy inteligentnejšie smerom ku cielu je **heuristika**.

Definícia 2 (Heuristika) *Nech (S, A, c) je stavový priestor s ohodnotenými hranami a $G \subseteq S$ je množina koncových stavov pre úlohu prehľadávania tohto priestoru.* ***Heuristika*** *h je ľubovoľná funkcia*

$$h : S \to \mathbb{R}_0^+$$

taká, že $h(s)$ udáva odhad ceny cesty zo stavu $s \in S$ ku najbližšiemu $s_g \in G$ a pre všetky $s_g \in G$ platí $h(s_g) = 0$.

2.1 Greedy search

Algoritmus, ktorý v každom kroku expanduje vrchol s^* s najmenšou hodnotou heuristickej funkcie (teoreticky najbližšie ku cielu):

$$s^* \in \arg \min_{s \in OPEN} h(s).$$

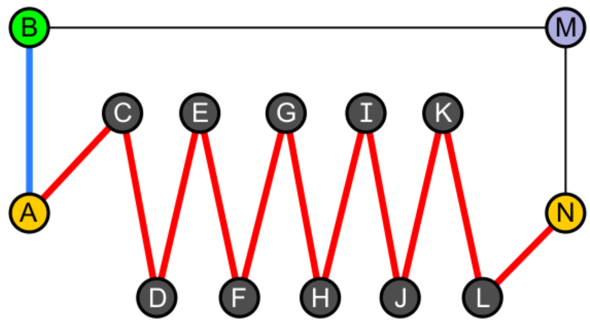

Figure 1: Greedy search nenájde najkratšiu cestu.

Greedy search má niekoľko problémov:

- algoritmus nenájde najkratšiu cestu (Fig. 1)
- rýchlosť a kvalita výsledku závisí od rýchlosti výpočtu a kvality heuristiky

2.2 Algoritmus A*

Algoritmus A* kombinuje oba zmienené postupy, čím sa snaží eliminovať problémy. Označme $g(s)$ cenu cesty od počiatočného vrcholu vypočítanú pomocou Dijkstru a $h(s)$ hodnotu heuristiky. Potom A* expanduje vrchol s^*, ktorý minimalizuje súčet týchto dvoch vzdialeností:

$$s^* \in \arg\min_{s \in OPEN} \big(h(s) + g(s)\big).$$

Vizualizácia sa nachádza na `http://www.jsgl.org/doku.php?id=pathfinder`.

2.3 Štúdium heuristík

Definícia 3 (Optimálna heuristika) ***Optimálnou heuristikou*** *h^* nazývame takú heuristiku, ktorá vracia skutočnú cenu cesty do cieľového stavu. V praxi je z výpočetných dôvodov nesplniteľným snom – vieme ju vypočítať hrubou silou, to nám ale algoritmus používajúci túto heuristiku nezrýchli. Má dôležktý teoretický význam – lepšia heuristika sa nedá skonštruovať.*

Definícia 4 (Prípustná heuristika) *Nech (S, A, c) je stavový priestor s ohodnotenými hranami a h je heuristika. Heuristiku h nazývame* ***prípustnou*** *pokiaľ*

$$\forall s \in S : h(s) \le h^*(s).$$

Definícia 5 (Monotónna heuristika) *Nech (S, A, c) je stavový priestor s ohodnotenými hranami a h je heuristika. Heuristiku h nazývame* ***monotónnou*** *pokiaľ*

$$\forall x, y \in S, \forall (x, y) \in A : h(x) \le h(y) + c(x, y).$$

Neformálne, heuristika stavu $s \in S$ je vždy menšia rovná heuristike každého zo susedných stavou plus cena cesty do daného stavu.

Definícia 6 (Dominujúca heuristika) *Nech (S, A, c) je stavový priestor s ohodnotenými hranami a h_1 a h_2 sú prípustné heuristiky. Heuristika h_1* ***dominuje*** *heuristiku h_2 pokiaľ*

$$\forall s \in S : h_1(s) \le h_2(s).$$

Heuristika h^* dominuje všetky prípustné heuristiky. Neformálne, ak heuristika h_1 dominuje heuristiku h_2, znamená to, že h_1 lepšie odhaduje optimálnu heuristiku.