Sztuczna inteligencja. Przeszukiwanie: heurystyki i więzy

Paweł Rychlikowski

Instytut Informatyki UWr

20 marca 2019

Algorytm A*. Przypomnienie

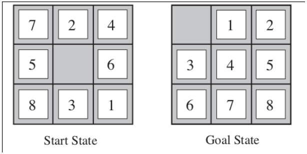
Definicje

- $g(n)$ – koszt dotarcia do węzła n
- $h(n)$ – szacowany koszt dotarcia od n do (najbliższego) punktu docelowego ($h(s) \geq 0$)
- $\mathrm{f(n)} = \mathrm{g(n)} + \mathrm{h(n)}$

Algorytm

Przeprowadź przeszukanie, wykorzystując $f(n)$ jako priorytet węzła (czyli rozwijamy węzły od tego, który ma najmniejszy f).

Heurystyki dla ósemki (przypomnienie)



Pomysł 1

Jak coś jest nie na swoim miejscu, to musi się ruszyć o co najmniej 1 krok. Zliczajmy zatem, ile kafelków jest poza punktem docelowym ($h_1(s) = 8$)

Pomysł 2

Jak coś jest nie na swoim miejscu, to musi pokonać cały dystans do punktu docelowego. Zliczajmy zatem, ile kroków od celu jest każdy z kafelków ($h_2(s) = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$)

Relaksacja

Kiedy możliwy jest ruch w łamigłówce ósemka? Docelowe pole jest: (koniunkcja warunków):

a) sąsiadujące

b) wolne

Możemy rezygnować z części (lub wszystkich) warunków, otrzymując łatwiejsze łamigłówki.

Uwaga

Liczba ruchów w łatwiejszym zadaniu od startu do punktu docelowego jest często sensowną heurystyką w zadaniu orygialnym.

Heurystyka h_1

Ruch możliwy jest **zawsze**.

Heurystyka h_2

Ruch możliwy jest **gdy pole jest obok** (niekoniecznie puste).

Heurystyka h_3

Ruch możliwy jest **gdy pole jest puste** (niekoniecznie obok).

Relaksacja na mapie

Relaksacja w zadaniu poszukiwania w labiryntach lub przy podróży samochodem drogami:

Relaksacja w zadaniu poszukiwania w labiryntach lub przy podróży samochodem drogami:

1. W labiryncie: pomijanie ścian, czyli odległość taksówkową

Relaksacja na mapie

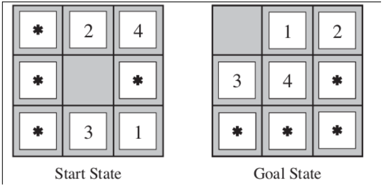
Relaksacja w zadaniu poszukiwania w labiryntach lub przy podróży samochodem drogami:

1. W labiryncie: pomijanie ścian, czyli odległość taksówkową
2. W atlasie drogowym: pomijanie dróg, czyli ogległość euklidesową (helikopterem)

Bazy wzorców

Heurystyki możemy budować korzystając z baz wzorców, zapamiętujących koszty rozwiązań podproblemów danego zadania.

Przykład:



Działanie bazy wzorców

- Znajdujemy wszystkie podproblemy dla danego stanu (które mamy w bazie)
- A następnie bierzemy maksimum kosztów jako wartość heurystyki
- Możemy do tego maksimum dołożyć jakieś proste heurystyki (typu h_2).

Pytanie

A czy nie moglibyśmy użyć sumowania, zamiast maksimum?

Działanie bazy wzorców (2)

- Niestety suma daje niedopuszczalne heurystyki (bo pewne ruchy liczymy wielokrotnie, gwiazdki w jednym wzorcu są istotnymi kafelkami w innym)
- **Pytanie:** Jak temu zapobiec?
- **Odpowiedź:** stosując „rozłączne" wzorce (nic się nie powtarza) i w każdym wzorcu liczyć tylko ruchy kafelków z liczbami.
  To to są te najefektywniejsze heurystyki dla 8-ki

Uwaga

Niemniej warto wiedzieć, że czasem rezygnuje się z optymalności i stosuje niedopuszczalne heurystyki (które czasem przeszacowują odległość), ze względu na szybkość działania.

Własności A^*

Plan

Spróbujemy dowieść następujących rzeczy:

1) A^* zwraca najkrótszą drogę
2) A^* jest zupełny.

Dowód optymalności

Potrzebujemy dwóch faktów:

F1. Jeżeli h jest spójna, wówczas na każdej ścieżce wartości f są niemalejące.
D-d (n' jest następnikiem n):

$$f(n') = g(n') + h(n') = g(n) + c(n, n') + h(n') \geq g(n) + h(n) = f(n)$$

F2. Zawsze, gdy algorytm bierze węzeł do rozwinięcia, to koszt dotarcia do tego węzła jest optymalny (najmniejszy możliwy).

Dowód F2

- Bierzemy nie wprost węzeł n, do którego kolejne dotarcie daje mniejszy koszt niż dotarcie pierwsze.
- Wartość f dla tego węzła drugi raz jest mniejsza (h takie same, g mniejsze)
- Na ścieżce od początku do n-a drugiego mamy widziany w pierwszym przebiegu węzeł n' (tuż po rozgałęzieniu)
- Z własności F1 mamy: $f(n') < f(n_1)$

Zatem algorytm powinien wybrać n' a nie n_1. **Sprzeczność.**

Optymalność

Popatrzmy na pierwszy znaleziony węzeł docelowy (n_{end})

- $f(n_{\text{end}}) = g(n_{\text{end}}) + h(n_{\text{end}}) = g(n_{\text{end}})$ (bo h jest rozsądna)
- Każdy kolejny węzeł docelowy jest nielepszy, bo dla niego $g(n) \geq g(n_{\text{end}})$

Zupełność

Niech C^* będzie kosztem najtańszego rozwiązania ($g(n_{\text{end}})$)

- Algorytm ogląda wszystkie węzły, dla których $f(n) < C^*$
- Być może oglądnie również pewne węzły z konturu $f(n) = C^*$, przed wybraniem docelowego n, t.że $f(n) = g(n) + 0 = C^*$

Uwaga

Skończona liczba węzłów o $g(n) \leq C^*$ gwarantuje to, że algorytm się skończy. Do skończoności z kolei wystarczy założyć, że istnieje $\varepsilon > 0$, t.że wszystkie koszty są od niego większe bądź równe.

Lepsze heurystyki

Uwaga

A^* nie rozwija węzłów t.że $f(n) > C^*$. Zatem im większa h (przy założeniu spełniania warunków dobrej heurystyki), tym lepsza.

Konsekwencja

Mając dwie spójne (optymistyczne) heurystyki h_1 i h_2, możemy stworzyć $h_3(n) = \max(h_1(n), h_2(n))$, która będzie lepsza od swoich składników (szczegóły na ćwiczeniach).

Pytanie

Co się będzie działo, jeżeli nasza funkcja h będzie liczyła dokładną odległość od celu?

Heurystyki niedopuszczalne

- Heurystyki mogą być niedopuszczalne (w szczególności, jeżeli są wynikiem uczenia się heurystyk)
- Oczywiście tracimy wówczas (w teorii i praktyce) gwarancje optymalności.
- Ale można otrzymać istotnie szybsze wyszukiwanie (o czym, mam nadzieję, przekonamy się na pracowni 2)

Problem spełnialności więzów

Uwaga

Między ósemką a hetmanami jest istotna różnica (mimo, że oba można przedstawiać jako problemy przeszukiwania.

- W ósemce interesuje nast droga dotarcia do celu, który jest dobrze znany (i tym samym mało ciekawy)
- W hetmanach interesuje nas, jak wygląda cel – droga do niego może być dość trywialna (dostawianie po kolei poprawnych hetmanów, przestawianie hetmanów z losowego ustawienia).

Problem spełnialności więzów (2)

Problemy takie jak hetmany są:

a) Bardzo istotne (ze względu na ich występowanie w rzeczywistym świecie)

b) Na tyle specyficzne, że warto dla nich rozważać specjalne metody.

Problemy spełnialności więzów. Definicja

Definicja

Problem spełnialności więzów ma 3 komponenty:

1. Zbiór zmiennych $X_1, \ldots, X_n$
2. Zbiór dziedzin (przypisanych zmiennym)
3. Zbiór więzów, opisujących dozwolone kombinacje wartości jakie mogą przyjmować zmienne.

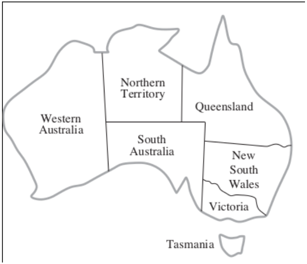
Przykład

Zmienne: X, Y, Z
Dziedziny: $X \in \{1, 2, 3, 4\}, Y \in \{1, 2\}, Z \in \{4, 5, 6, 7\}$
Więzy: $X + Y \geq Z, X \neq Y$

Komentarz do definicji

- Powyższy przykład to były więzy na dziedzinach skończonych, jeden z najważniejszych przypadków więzów.
- Ale można rozważać inne dziedziny:
  a) liczby naturalne,(trochę boli, że to nierozstrzygalny problem)
  b) liczby wymierne,
  c) ciągi elementów, napisy
  d) krotki
- Więzy określają relacje, często da się je wyrazić wzorem, ale nie jest to wymagane.

Kolorowanie Australii



- Mamy pokolorować mapę Australii, za pomocą 3 kolorów: (R, G, B)
- Sąsiadujące prowincje muszą mieć różne kolory.

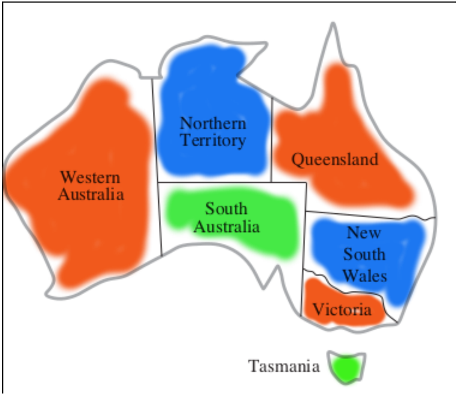
Kolorowanie jako problem więzowy

Zmienne: WA, NT, Q, NSW, V, SA, T

Dziedziny: {R,G,B}

Więzy: $SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW, SA \neq V, WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V$

Przykładowe kolorowanie

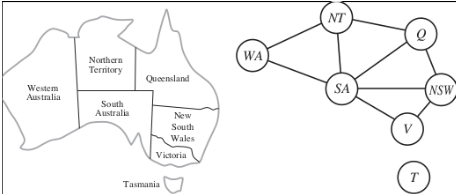


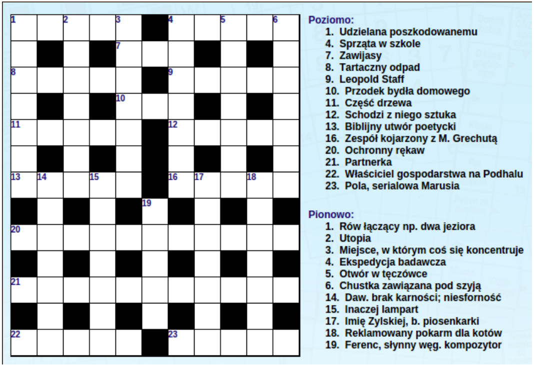
Arność węzłów

- Więzy (jako relacje) mogą mieć różną arność.
- Unarne – można potraktować jako modyfikację dziedziny (Tasmania nie jest czerwona) i zapomnieć.
- Binarne – jak w naszym przykładzie z kolorowaniem
- Mogą mieć też inną arność, w zasadzie dowolną (w praktyce spotyka się więzy o arności np. kilkaset)

Graf węzłów

- Dla więzów binarnych możemy stworzyć graf, w którym krawędź oznacza, że dwie zmienne są powiązane więzem.
- Więzy binarne są istotną klasą więzów, wiele algorytmów działa przy założeniu binarności więzów.





Pytanie

Co powinno być zmienną w zadaniu rozwiązywania krzyżówki?

Krzyżówka (podstawowa)

Pomijamy (chwilowo?) kwestie zgodności hasła z definicją.

- Zmienne odpowiadają kratkom, dziedziną są znaki
- Więzy (fragment):
  jest-słowem-7(A,B,C,D,E,F,G), jest-słowem-3(B,H,I), ...

Krzyżówka (dualna)

- Zmienne to słowa (dziedziną jest słownik przycięty do określonej długości)
- Mamy więz dla każdej pary krzyżujących się słów. Jaki?

[na tablicy]

Problemy dualne (2)

- Zwróćmy uwagę, że to, co zrobiliśmy z krzyżówką stosuje się do dowolnych więzów.
- Więzy w problemie prymarnym zmieniają się na zmienne w problemie dualnym (z dziedziną będącą dozwolonym zbiorem krotek)
- Dodatkowo potrzebujemu więzów, które mówią, że i-ty element jednej krotki jest j-tym elementem drugiej (te więzy są binarne!)

Propagacja więzów

Uwaga 1

To co odróżnia CSP od zadania przeszukiwania jest możliwość wykorzystania dodatkowej wiedzy o charakterze problemu do przeprowadzenia wnioskowania.

Uwaga 2

Podstawowym celem wnioskowania jest zmniejszenie rozmiaru dziedzin (a tym samym zmniejszenie przestrzeni przeszukiwań).

Wnioskowanie. Przykład

Przykład

Zmienne: X, Y, Z
Dziedziny: $X \in \{1, 2, 3, 4\}, Y \in \{1, 2\}, Z \in \{5, 6, 7, 8\}$
Więzy: $X + Y \geq Z, X \neq Y$

Czy możemy nie tracąc żadnego rozwiązania skreślić jakieś wartości z dziedzin?

Dodatkowo skreślił nam się 1 więz (zawsze to jakiś zysk...)

Spójność więzów

- Spójność (intuicyjnie) rozumiemy jako niemożność wykreślenia żadnej wartości z dziedziny.
- Mamy różne rodzaje spójności:
  1. Węzłowa (każda wartość z dziedziny spełnia więzy unarne dla zmiennych)
  2. Łukowa: jak dwie zmienne są połączone więzem, to dla każdej wartości z dziedziny X jest wartość w dziedzinie Y, t.że dla tych wartości więz jest spełniony.

Uwaga

Są jeszcze inne rodzaje spójności. Więcej na ćwiczeniach.