Sztuczna inteligencja. Problem spełnialności więzów

Paweł Rychlikowski

Instytut Informatyki UWr

21 marca 2018

A^* – powtórzenie

- Algorytm przeszukiwania, który wykorzystuje wiedzę o problemie (heurystyka h)
- Jeżeli heurystyka spełnia pewne właściwości, to algorytm jest optymalny i zupełny.
- Potrafi w wielkim stopniu zredukować przestrzeń poszukiwań

A^* – kontury

Uwaga

Pewną intuicję odnośnie działania algorytmu daje pojęcie konturów (czyli zbiorów węzłów o tej samej wartości f)

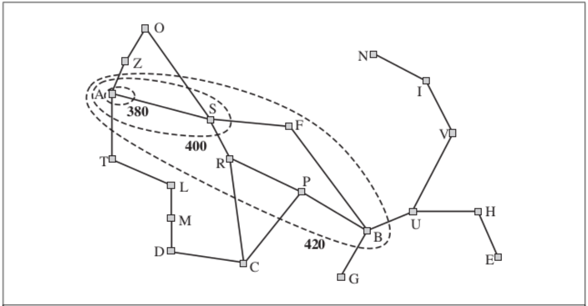


Figure 3.25 Map of Romania showing contours at $f = 380$, $f = 400$, and $f = 420$, with Arad as the start state. Nodes inside a given contour have f-costs less than or equal to the contour value.

Pytanie

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

Uczenie się heurystyk

- Stan może być charakteryzowany przez pewne parametry. Przykładowo dla ósemki:
  - wartości h_1, h_2 i h_3
  - liczba par sąsiednich kafelków, które są sąsiednie w tym stanie, a nie są sąsiednie w rozwiązaniu.
  - Nazwijmy te cechy $x_1, x_2, \ldots, x_n$
- Możemy zdefiniować

$$h(s) = \sum_{i=1}^{n} c_i x_i(s)$$

- Wyznaczając c_i w sposób minimalizujący błąd dla wygenerowanych przykładów (z prawdziwymi wartościami odległości)

Będziemy się tym jeszcze zajmować, rozważając zagadnienia uczenia maszynowego.

Heurystyki niedopuszczalne

- Heurystyki uczone na poprzednim slajdzie mogą być niedopuszczalne
- Oczywiście tracimy wówczas (w teorii i praktyce) gwarancje optymalności.
- Ale można otrzymać istotnie szybsze wyszukiwanie (zob. P2)

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.

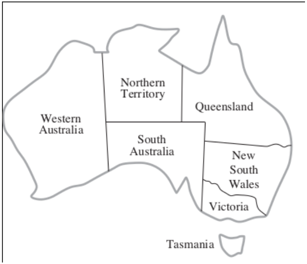
Przykład

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

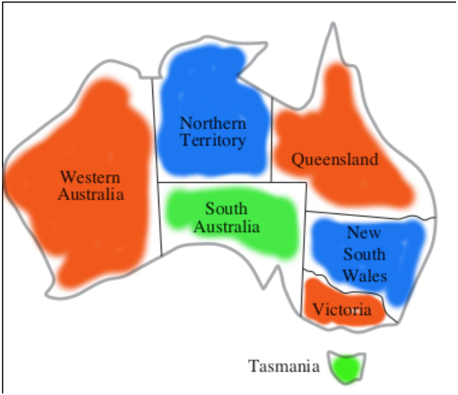


- Mamy pokolorować mapę Australii, za pomocą 3 kolorów: (R, G, B)
- Sąsiadujące prowincje muszą mieć różne kolory.

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

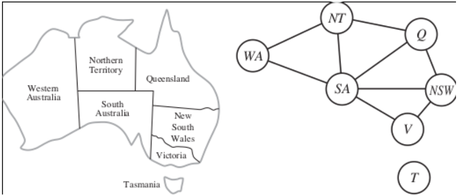


Arność węzłów

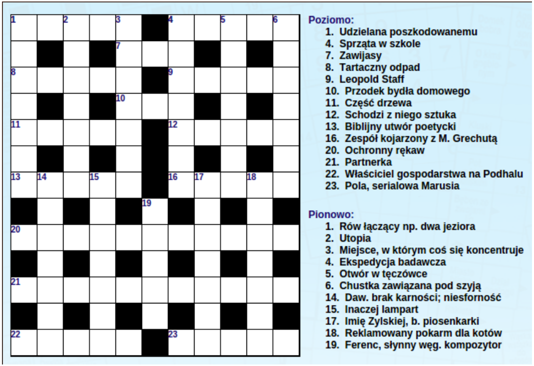
- Więzy (jako relacje) mogą mieć różną arność.
- Unarne – można potraktować jako modyfikację dziedziny (Tasmania nie jest czerwona) i zapomnieć.
- Binarne – jak w naszym przykładzie z kolorowaniem
- Mogą mieć też inną arność, w zasadzie dowolną (w praktyce spotyka się więzy o arności np. kilkaset)

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.



Problemy dualne



Pytanie

Co powinno być zmienną w zadaniu rozwiązywania krzyżówki?

Krzyżówka (podstawowa)

Pomijamy (na trochę) 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, jeśli potrzeba]

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.

Algorytm AC-3

Algorytm zapewnia spójność łukową sieci więzów.

Idea

1. Zarządzamy kolejką więzów,
2. Usuwamy niepasujące wartości z dziedzin, analizując kolejne więzy z kolejki,
3. Po usunięciu wartości z dziedziny B, sprawdzamy wszystkie zmienne X, które występują w jednym więzie z B

Algorytm AC-3

```
function AC-3(csp) returns false if an inconsistency is found and true otherwise
   inputs: csp, a binary CSP with components (X, D, C)
   local variables: queue, a queue of arcs, initially all the arcs in csp

   while queue is not empty do
      (X_i, X_j) ← REMOVE-FIRST(queue)
      if REVISE(csp, X_i, X_j) then
         if size of D_i = 0 then return false
         for each X_k in X_i.NEIGHBORS - {X_j} do
            add (X_k, X_i) to queue
   return true
```

```
function REVISE(csp, X_i, X_j) returns true iff we revise the domain of X_i
   revised ← false
   for each x in D_i do
      if no value y in D_j allows (x,y) to satisfy the constraint between X_i and X_j then
         delete x from D_i
         revised ← true
   return revised
```

Algorytm AC-3. Uwagi

- Zwróćmy uwagę na niesymetryczność funkcji Revise (oznacza ona konieczność dodawania każdej pary zmiennych dwukrotnie)
- Wersja na slajdzie tylko sprawdzała spójność.
- Użyteczna implementacja zwracałaby również nowe wartości dziedzin.

Złożoność algorytmu AC-3

- Mamy n zmiennych, dziedziny mają wielkość $O(d)$. Mamy c więzów.
- Obsługa więzu to $O(d^2)$
- Każdy więz może być włożony do kolejki co najwyżej d razy.

Złożoność

Złożoność wynosi zatem $O(cd^3)$ (raczej pesymistycznie)

AC-3 a obrazki logiczne

- Obrazki logiczne da się zapisać jako sieć więzów, z więzami typu:

$$\text{wiersz}_{[2,3,3]}(B_1, \ldots, B_n)$$

dla wierszy i kolumn.

- Ale te więzy mają dużą arność, a my chcemy więzów binarnych.

Opcje

a) Utworzyć problem dualny (całkiem sensowna)

b) Połączyć zmienne z problemu dualnego i oryginalne.

AC-3 i obrazki logiczne

- Mamy zmienne B_i odpowiadające poszczególnym kratkom,
- Mamy zmienne K_i odpowiadające kolumnom i W_i odpowiadające wierszom
- Zwróćmy uwagę, że za definicję zadania w zasadzie odpowiadają więzy unarne na zmiennych K oraz W.
- Musimy powiązać kratki, wiersze i kolumny:

$$B_{ij} \text{ jest-elementem-j } W_i$$

oraz

$$B_{ij} \text{ jest-elementem-i } K_j$$

Uwaga

Binarna sieć więzów, zatem można stosować AC-3.

Propagacja w obrazkach

- Z wiersza (kolumny) do pola: pole B_{ij} musi mieć wartość b, bo wszystkie wartości W_i mają na j-tym polu b
- Z pola do wiersza (kolumny): skoro pole B_{ij} ma wartość b, to możemy wykreślić wszystkie układy z dziedziny K_j, które nie mają na i-tej pozycji b.

Dokładnie tak rozwiązują obrazki logiczne ludzie. Ale to nie zawsze doprowadzi do sukcesu...

Poszukiwanie z nawrotami dla problemów więzowych

przeszukiwanie z nawrotami = backtracking search

- Wariant przeszukiwania w głąb, w którym stanem jest niepełne podstawienie.
- Nie pamiętamy całej historii, ale potrafimy zrobić **undo**
- Po każdym przypisaniu wykonujemy jakąś formę wnioskowania, bo może da się zmniejszyć dziedziny...

Backtracking

```
function BACKTRACK(assignment, csp) returns a solution, or failure
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
      if value is consistent with assignment then
          add {var = value} to assignment
          inferences ← INFERENCE(csp, var, value)
          if inferences ≠ failure then
             add inferences to assignment
             result ← BACKTRACK(assignment, csp)
             if result ≠ failure then
                return result
      remove {var = value} and inferences from assignment
  return failure
```

Backtracking. Uwagi

1. Powyższe sformułowanie zakłada, że wnioskiem może być znalezienie wartości dla zmiennej lub failure.
2. Moglibyśmy to zmodyfikować, mówiąc że wnioskiem może być również usunięcie wartości z dziedziny.
3. Możliwy jest też taki wariant, że najpierw uruchamiamy AC-3, potem Backtracking z jakimś uproszczonym wnioskowaniem.

Parametry backtrackingu

1. Jak wybieramy zmienną do podstawienia (SelectUnassignegVariable)
2. W jakim porządku sprawdzamy dla niej wartości (OrderDomainValue)
3. Jak przeprowadzamy wnioskowanie (Inference)

Przykład. Plan lekcji

- Rozmieszczamy lekcje: zajęcia otrzymują termin
- Mamy naturalne więzy:
  - Jeżeli Z_1 i Z_2 mają tego samego nauczyciela (klasę, salę), wówczas $Z_1 \neq Z_2$
  - Nauczyciele nie mogą mieć zajęć o określonych porach (bo na przykład pracują w innych miejscach)
  - Wszystkie zajęcia klasy X danego dnia spełniają określone warunki: brak okienek, po jednej godzinie przedmiotu, itd.

Pytanie

W jakiej kolejności rozmieszcza zajęcia Pani Sekretarka?

Heurystyka: First Fail

Definicja

Wybieramy tę zmienną, która jest najtrudniejsza, co oznacza, że:

- ma najmniejszą dziedzinę,
- występuje w największej liczbie więzów.

Inne nazwy: Most Constrained First, Minimum Remaining Values (MRV)

Uzasadnienie

I tak będziemy musieli tę zmienną obsłużyć. Lepiej to zrobić, jak jeszcze inne zmienne są „wolne”

Wybór wartości

- Wybieramy tę wartość, która w najmniejszym stopniu ogranicza przyszłe wybory LCV, Least Contstraining Value.
- Przykład. W planie zajęć:
    1. Mamy teraz przydzielić termin zajęć panu A z klasą 1c
    2. Musimy później przydzielić zajęcia A z klasą 2a.
    3. Wcześniej przydzieliliśmy panią B z klasą 2a w czwartek na 8.
    4. Jest to argument za tym, żeby (A,1c) też była na ósmą w czwartek (bo nie stracimy żadnej możliwości dla (A,2a).

LCV a MRV

- W pierwszej chwili może dziwić przeciwne traktowanie wyboru zmiennych i wartości.
- Celem MRV jest agresywne ograniczanie przestrzeni poszukiwań.
- Celem LCV jest dążenie do jak najszybszego znalezienia pierwszego rozwiązania.

Musimy rozpatrzeć wszystkie zmienna, ale niekoniecznie wszystkie wartości!