Sztuczna Inteligencja

Problem przeszukiwania

Dr hab. inż. Grzegorz Dudek
Wydział Elektryczny
Politechnika Częstochowska

Projekt finansowany w ramach programu Ministra Nauki i Szkolnictwa Wyższego pod nazwą „Regionalna Inicjatywa Doskonałości" w latach 2019 - 2022 nr projektu 020/RID/2018/19 kwota finansowania 12 000 000 PLN

ROZWIĄZYWANIE PROBLEMÓW JAKO PRZESZUKIWANIE

- Istotną rolę podczas rozwiązywania problemu ma wybór sposobu postępowania prowadzącego do uzyskania określonych wyników.

- Wśród różnych technik postępowania tzw. metody przeszukiwania są jednymi z częściej stosowanych podczas procesu rozwiązywania zadań.

- Wymagane jest określenie zbioru stanów przestrzeni rozwiązywanego problemu, zbioru operatorów przekształcających te stany, stanu początkowego i zbioru stanów końcowych. Rozwiązanie polega na określeniu ciągu operatorów przekształcających stan początkowy w stan końcowy.

- Metody przeszukiwania dzielimy na:

    - ślepe (*blind search*) – nie wykorzystujące informacji o dziedzinie rozwiązywanego problemu

    - heurystyczne – wykorzystujące informacje o dziedzinie rozwiązywanego problemu

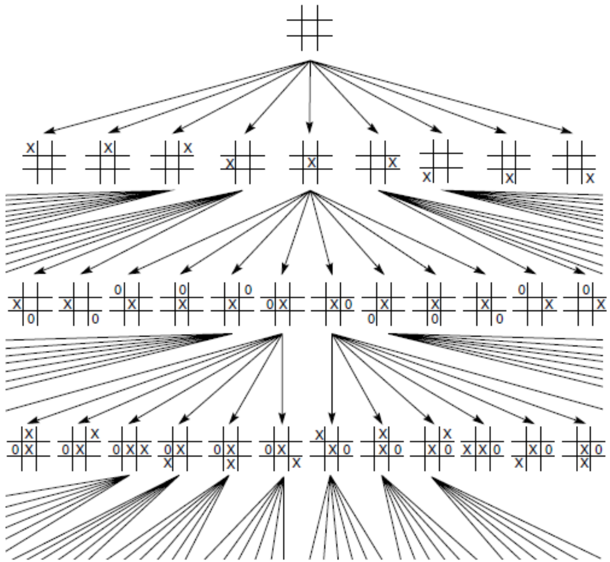
REPREZENTACJA PROBLEMU W PRZESTRZENI PRZESZUKIWAŃ

Rozwiązanie zadania jest często wyznaczane przez przeszukiwanie zbioru wszystkich możliwych stanów, zwanych przestrzenią przeszukiwania. Kompletne przebadanie takiej przestrzeni zawodzi w praktyce przy dużych przestrzeniach. Efektywną metodą rozwiązywania dużych zadań jest generowanie stanów za pomocą operatorów według określonych zasad i badanie ich właściwości.

Ze względu na opis zadania istotne są trzy podstawowe wymagania:

- sposób reprezentacji każdego ze stanów przestrzeni przeszukiwania, tzw. kod,
- metody obliczeniowe umożliwiające wygenerowanie kodu kolejnego stanu na podstawie kodu danego stanu, czyli operatory,
- metody wyboru operatorów spośród zestawu możliwych do zastosowania, czyli strategie sterowania.

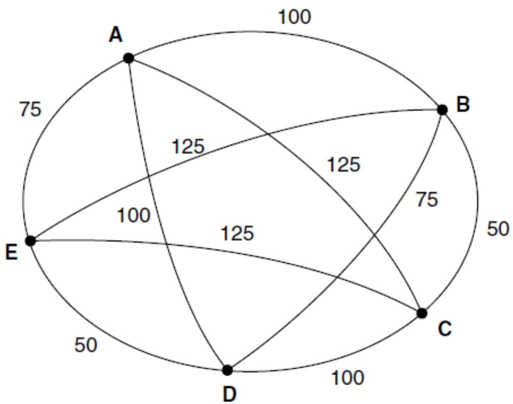
Wśród najbardziej pożądanych cech kodu stanów należy wymienić jednoznaczność i uwzględnienie struktury zadania. Taki sposób reprezentacji umożliwia efektywne przekształcenia zbioru za pomocą operacji rozszczepiania. Polega ona na podziale problemu reprezentowanego przez dany stan na podproblemy, odrzucaniu części stanów i badaniu jedynie najbardziej obiecujących. Kolejne rozszczepienia mogą sprowadzić problem początkowy do problemu łatwiejszego do rozwiązania. Strategia ta osi nazwę rozszczepiania i odrzucania (*split-and-prune*) lub generowania i testowania (*generate-and-test*).

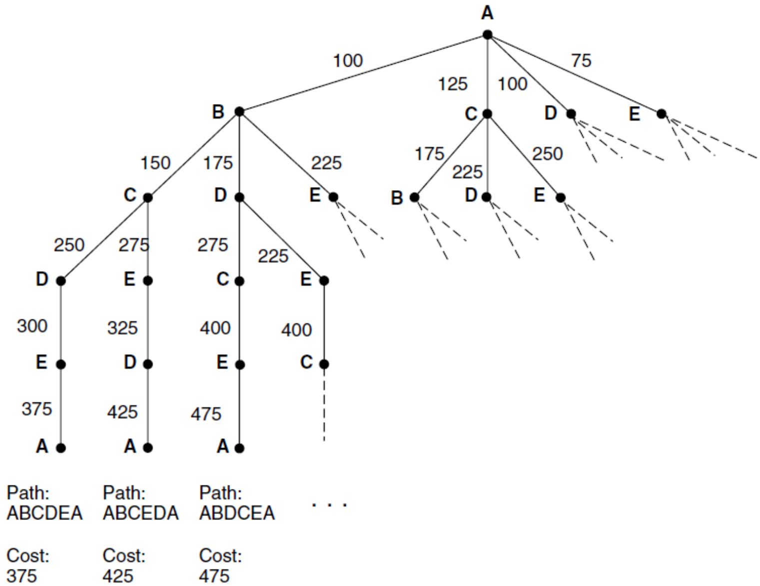


REPREZENTACJA PROBLEMU W PRZESTRZENI PRZESZUKIWAŃ – PRZYKŁAD

Problem komiwojażera

Znajdź najkrótszą ścieżkę przechodzącą przez wszystkie punkty, zaczynająca się i kończącą w tym samym punkcie (A)



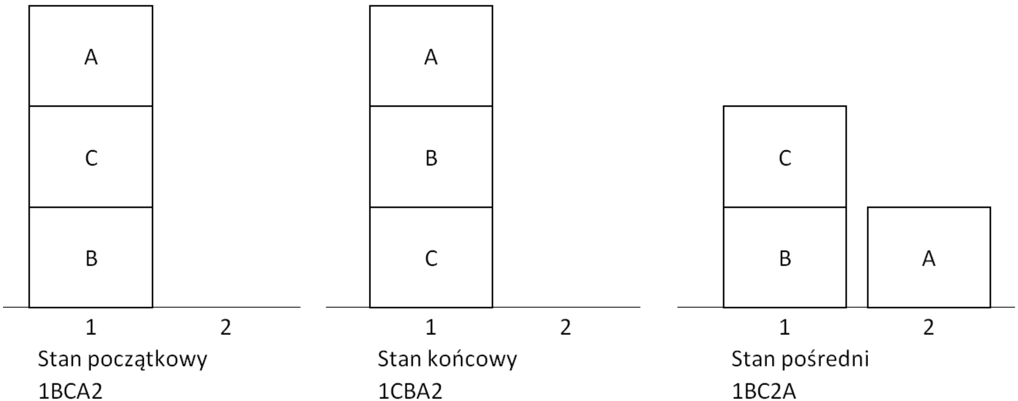


REPREZENTACJA PROBLEMU W PRZESTRZENI PRZESZUKIWAŃ – PRZYKŁAD

Świat klocków

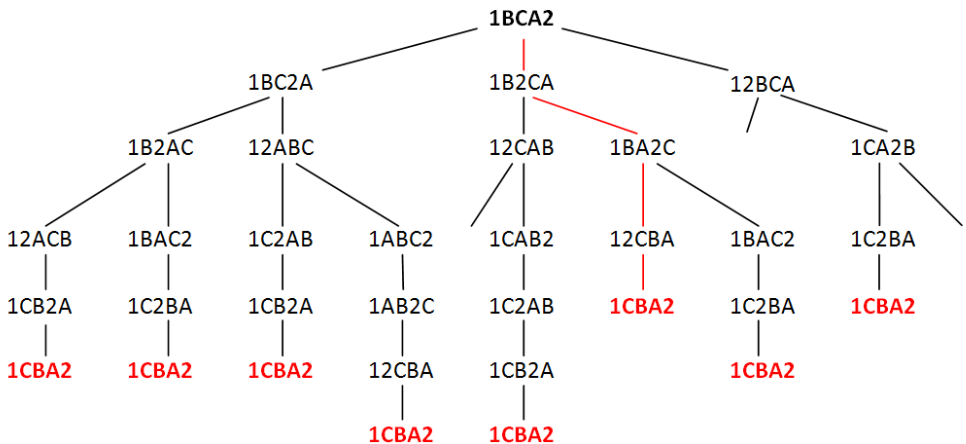
Pewna początkowa konfiguracja ustawienia na sobie trzech klocków (stan początkowy) ma być przestawiona, np. przy użyciu ramienia robota, w inną konfigurację będącą stanem końcowym (celem). Osiągnięcie celu jest zatem zadaniem zaplanowania sekwencji działania robota mogącego przestawiać klocki pojedynczo, po dwa i po trzy na dwóch określonych pozycjach na stole (oznaczone cyframi 1 i 2).

Przyjmijmy stan początkowy 1BCA2 (na pozycji 1 znajdują się klocki w kolejności od dołu: B, C i A, a na pozycji 2 nie ma żadnego klocka). Przyjmijmy stan końcowy: 1CBA2.



REPREZENTACJA PROBLEMU W PRZESTRZENI PRZESZUKIWAŃ – PRZYKŁAD

Przejścia pomiędzy stanami obrazuje drzewo (graf) przeszukiwań:



Z przedstawionego drzewa przestrzeni stanów wynika, że istnieje kilka dróg dojścia do celu i drogi te nie są równe. W celu przyspieszenie dojścia do celu stosuje się heurystyki. Dla powyższego przykładu, znając stan końcowy i operatory (możliwości przestawiania klocków), można określić pewne reguły postępowania przyśpieszające dojście do celu:

- klocek C powinien być ustawiany bezpośrednio na stole w pozycji 1 lub 2,
- klocek A powinien być ustawiany na B,
- klocek B powinien być ustawiany na C.

Stosując te reguły w pierwszym kroku preferowany jest stan 1B2CA, a pozostałe dwa stany będą pominięte. Redukuje się w ten sposób przestrzeń przeszukiwań do jednej trzeciej całego drzewa. Dalej z dwóch stanów otrzymanych z 1B2CA, wybrany będzie 1BA2C, ponieważ klocek C leży na stole, a klocek A leży na B. Pominięcie poddrzewa wychodzącego z 12CAB w dalszym ciągu redukuje przestrzeń stanów. Po rozszczepieniu węzła 1BA2C wybrany będzie węzeł 12CBA, ponieważ odpowiada przyjętym regułom postępowania. Z tego stanu wygenerowany będzie stan końcowy. Pomijanie gałęzi ze stanami mniej obiecującymi nazywane jest przycinaniem gałęzi drzewa przestrzeni stanów.

Strategie przeszukiwania grafów

Przestrzeń stanu zwykle zawiera ogromną liczbę stanów. Stany generowane i analizowane w kolejnym kroku przeszukiwania możemy przedstawić w postaci drzewa (grafu) przeszukiwania.

Strategie przeszukiwania mają na celu wybór operatorów, określających gałęzie grafu przestrzeni stanów możliwe do wyboru podczas rozwiązywania danego problemu. Parametrami charakteryzującymi przeszukiwane grafy są: głębokość węzła (poziom) i stopień rozgałęzienia węzła (liczba potomków, rząd).

Drzewo przeszukiwania konstruuje się wykorzystując strategie iteracyjne. Strategie te mają na celu w jak największym stopniu zmniejszyć liczbę badanych węzłów oraz zoptymalizować obszar pamięci przeznaczony na zapamiętanie stanów przejściowych. Najważniejszymi kryteriami przy porównywaniu strategii są:

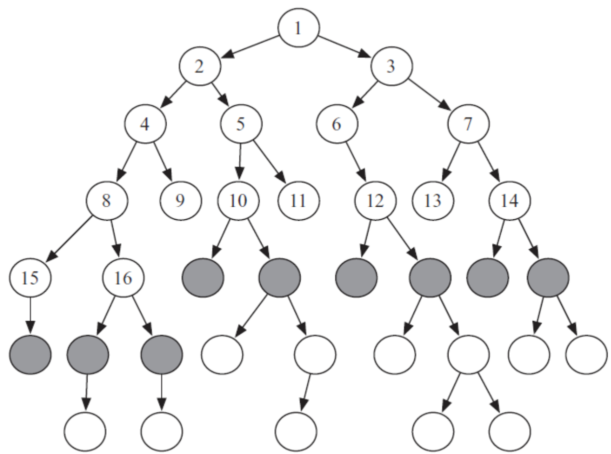
- jakość wyznaczonego rozwiązania,
- koszt obliczeniowy i
- obszar pamięci wymagany przez strategię.

STRATEGIA ŚLEPA PRZESZUKIWANIA WSZERZ

W strategiach ślepych nie wykorzystuje się informacji o zadaniu, dzięki czemu mają one charakter uniwersalny. Porządek przeszukiwania zależy wyłącznie od informacji dostarczanych przez już zbadane węzły grafu i sam proces przeszukiwania.

Strategia przeszukiwania wszerz (*breadth-first*) podkreśla kolejność przeszukiwania węzłów grafu. Startując od korzenia (węzeł początkowy) generujemy kolejne węzły "poziom po poziomie". Graf budowany jest to momentu znalezienia stanu końcowego.

Numery na rysunku wskazują porządek odwiedzanych węzłów. Węzły z numerami to węzły już rozgałęzione (zamknięte). Węzły szare to węzły w kolejce do rozgałęzienia (otwarte). W przypadku przeszukiwania wszerz jest to kolejka FIFO.

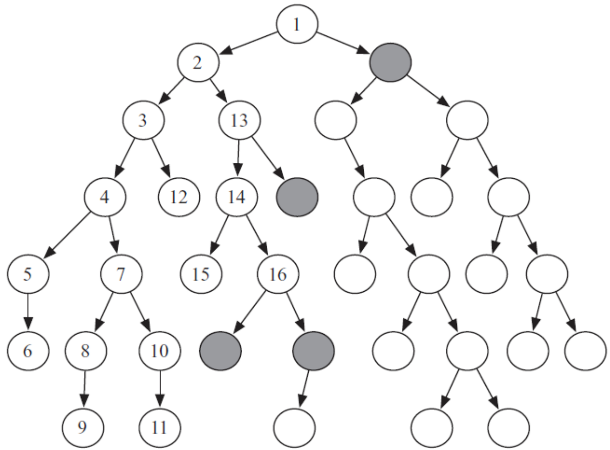


STRATEGIA ŚLEPA PRZESZUKIWANIA W GŁĄB

W strategii przeszukiwania w głąb (*depth-first*) dla węzła z poziomu *j* generuje się potomka z poziomu *j*+1, dla którego generuje się potomka z poziomu *j*+2 itd. Po dojściu do liścia (węzeł który nie ma potomków), cofamy się do jego rodzica, z którego generujemy innego potomka i znowu schodzimy w głąb drzewa.

Przeszukiwanie w głąb wydaje się rozsądniejsze od przeszukiwania wszerz, gdy graf jest szeroki i płytki (węzły mają dużo potomków).

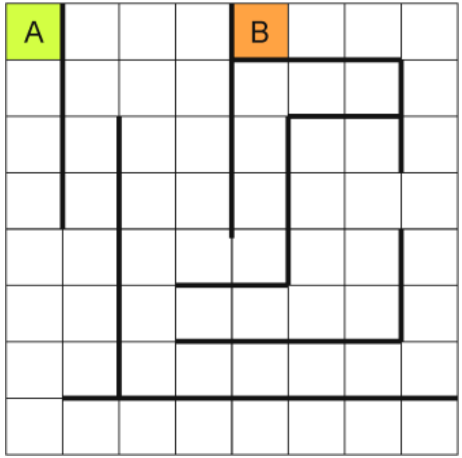
Na rysunkach na szaro zaznaczono węzły otwarte, znajdujące się w kolejce do rozgałęzienia. W przypadku przeszukiwania w głąb jest to kolejka LIFO (stos).



MODYFIKACJE PRZESZUKIWANIA W GŁĄB I WSZERZ

Modyfikacje strategii w głąb i wszerz:

- ograniczone przeszukiwanie w głąb – generujemy węzły tak jak w metodzie przeszukiwania w głąb, ale tylko do ustalonej głębokości *p*.
- przeszukiwanie pogłębiane iteracyjnie – rozszerzenie metody przeszukiwania ograniczonego. Jeśli nie znaleziono węzła końcowego dla głębokości *p*, zwiększamy głębokość do *p*+1, następnie *p*+2 itd.
- przeszukiwanie o jednolitym koszcie – zakładamy, że potrafimy oceniać koszt przejścia pomiędzy stanami. Możemy wyznaczyć koszt przejścia od stanu początkowego (korzenia) do węzła *n* – *g*(*n*). Wtedy stosujemy strategię wyboru węzłów do rozgałęzienia w kolejności od najniższego do najwyższego kosztu *g*(*n*).



Labirynt*

Załóżmy, że zamierzamy odnaleźć drogę w labiryncie, prowadzącą z kratki A do kratki B. Każda kratka ma co najwyżej czterech sąsiadów (brak sąsiada jest zaznaczony grubą linią – "ścianą" labiryntu).

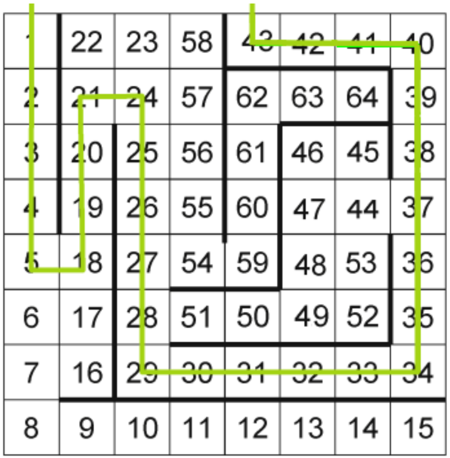
Kratki labiryntu będą węzłami przestrzeni przeszukiwań.

Poszukiwanie wyjścia polega na tym, że wpisujemy do kratek numery iteracji, w których były one węzłem rozwijanym przez algorytm przeszukiwania. Po odnalezieniu rozwiązania drogę będziemy odtwarzać, idąc od węzła B do węzła A w kierunku sąsiedniej kratki o najmniejszym numerze.

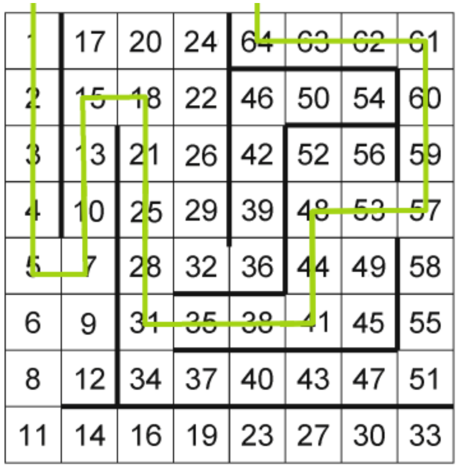
* Arabas J., Cichosz P.: Sztuczna inteligencja. Materiały do wykładu. http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja

Załóżmy, że obie metody przeszukiwania (w głąb i wszerz) rozważają kratki sąsiadujące w kolejności: góra, lewo, dół, prawo. Dostajemy wówczas dla obu metod kolejność odwiedzania kratek labiryntu jak na poniższych rysunkach.

Strategia w głąb



Strategia wszerz



Strategie heurystyczne umożliwiają w pewnej mierze uwzględnianie informacji o niezbadanej jeszcze części grafu. Przy wyborze węzłów do rozgałęzienia może być przydatna wiedza o dziedzinie danego problemu oraz charakterystyce poszukiwanych węzłów celu.

Strategie heurystyczne stosują pewną funkcję heurystyczną, która wyraża ocenę węzła *n* ze względu na następujące kryteria:

- zbieżności, czyli osiągnięcia celu,
- najmniejszego kosztu drogi wyznaczanej od węzła początkowego, przez węzeł *n*, do węzła końcowego,
- najmniejszej złożoności obliczeniowej procesu przeszukiwania.

Do dalszego rozgałęzienia wybieramy "najlepszy" węzeł spośród wszystkich węzłów rozpatrywanych do tej pory, niezależnie od jego położenia w grafie. Przyjmujemy, że węzeł "najlepszy" ma najmniejszą wartość funkcji heurystycznej.

Przeszukiwanie "pierwszy najlepszy" (*best-first search*) wybiera do rozgałęzienia węzeł, który posiada najmniejszy szacunkowy koszt przejścia do stanu końcowego – $h(n)$. Strategia ta jest analogiczna do przeszukiwania o jednolitym koszcie, przy czym zamiast funkcji $g(n)$ stosuje się funkcję $h(n)$.

W strategii A* funkcja heurystyczna jest sumą dwóch składników:

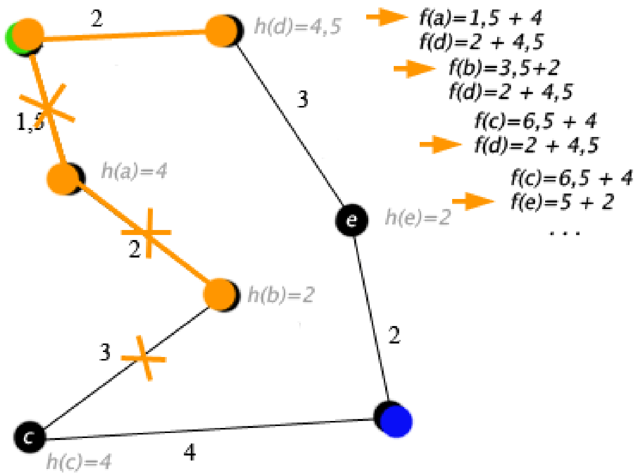
$$f(n) = h(n) + g(n)$$

Dla węzła n wyznacza się szacunkowy koszt drogi łączącej ten węzeł z węzłem celu $h(n)$ oraz koszt drogi łączącej węzeł początkowy z węzłem n – $g(n)$. Funkcję $f(n)$ oblicza się dla różnych węzłów n i do dalszej ekspansji wybiera się ten węzeł, dla którego $f(n)$ przyjmuje najmniejszą wartość.

Strategia A* jest analogiczna do przeszukiwania o jednolitym koszcie, przy czym zamiast funkcji $g(n)$ stosuje się funkcję $f(n)$.

Udowodniono, że strategia A* prowadzi do rozwiązania optymalnego, jeśli $h(n)$ jest nie większe niż rzeczywisty koszt drogi łączącej węzeł n z węzłem celu.

Oto przykład działania algorytmu A* dla grafu, którego węzłami są miasta, z gałęziami związane są odległości drogowe, a heurystyka *h*(*n*) jest odległością w linii prostej od węzła *n* do węzła celu (węzeł niebieski). Przykład pokazuje prostą sytuację, w której A* wykona nawrót (w węźle c) ze względu na niesłuszne przewidywania heurystyki.



STRATEGIA A* – PRZYKŁAD

Drzewo przeszukiwania dla układanki 3x3 utworzone za pomocą metody A*.

Wartości funkcji *f*(*n*) pokazano w kółkach.

- *g*(*n*) jest liczbą kroków od węzła startowego do *n*-tego
- *h*(*n*) jest liczba klocków na niewłaściwych pozycjach (innym pomysłem jest wyrażenie *h*(*n*) za pomocą sumy odległości każdego klocka od jego położenia docelowego)