

***Sztuczna inteligencja***

**Sprawozdanie z projektu końcowego**

Temat:Prezentacja możliwości biblioteki JUNG w projektach informatycznych, wymagających użycia metod przeszukiwania z wykorzystaniem heurystyki.

Wykonujący projekt: **Michał Kierzkowski, Paweł Halicki**

Studia dzienne

Kierunek: Informatyka

Semestr: IV

Grupa zajęciowa: PS1

Prowadzący ćwiczenie: **mgr inż. Dariusz Jankowski**

06.06.2018

Data wykonania ćwiczenia

…………...............................................

Data i podpis prowadzącego

Spis treści

**Wprowadzenie3**

**Podstawowe pojęcia teorii grafów3**

**Pojęcie heurystyki3**

**Zarys historyczny4**

**Algorytm a heurystyka5**

**Funkcje heurystyczne w przeszukiwaniu6**

**Charakterystyka przeszukiwania heurystycznego7**

**Klasyczne strategie przeszukiwań**

Strategia z powracaniem (backtracking)**7**

Strategia wszerz (breadth-first)**7**

Strategia zachłanna (hill-climbing)**8**

Strategia "najpierw najlepszy" (best-first)**8**

Strategia A\***8**

**Reprezentacja zadań i przestrzeni poszukiwań9**

**Bibliotek JUNG10**

**Opis programu i przykładu testowego10**

**Główne części programu i wynikix**

**Wnioskix**

**Bibliografiax**

Wprowadzenie

Metody heurystyczne należą do jednych z najważniejszych narzędzi sztucznej inteligencji. Są one różnie interpretowane i wykorzystywane na wiele różnych sposobów.

W swoim projekcie chciałbym przedstawić pojęcie heurystyki, jej podstawowe algorytmy, zastosowanie oraz na podstawie założonego problemu pokrótce przedstawić możliwości biblioteki Java Universal Network/Graph Framework w tworzeniu projektów informatycznych, wymagających użycia metod przeszukiwania z wykorzystaniem heurystyki.

Podstawowe pojęcia teorii grafów

* Korzeń grafu

Stan, od którego zaczynamy przeszukiwanie grafu (drzewa) – początkowy stan problemu (instancja problemu)

* Wierzchołek końcowy (terminalny)

Stan, który ma określoną wartość z punktu widzenia wyniku danego zadania (np. porażka, zwycięstwo, remis w grze)

* Liść

Dowolny stan w grafie, w którym zatrzymujemy proces przeszukiwania i przypisujemy mu ocenę heurystyczną

* Wierzchołek „wewnętrzny”

Każdy inny stan, którego wartość zależy od jego poprzedników lub następników

* Głębokość przeszukiwania (aktualna)

Liczba przejść stanów (ruchów) od korzenia grafu do stanu aktualnego

* Branching factor

Średnia liczba następników stanu (śr. liczba ruchów w stanie)

* Drzewa przeszukiwania/DAG

Większość grafów przeszukiwania to DAG (acykliczne grafy skierowane), część z nich to drzew [1]

Pojęcie heurystyki

Heurystyka z greckiego *heuresis* (odnaleźć, odkryć) jest to metoda znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego, a często nawet prawidłowego. Rozwiązań tych używa się np. wtedy, gdy pełny algorytm jest z przyczyn technicznych zbyt kosztowny lub gdy jest nieznany (np. przy przewidywaniu pogody lub przy wykrywaniu niektórych zagrożeń komputerowych, takich jak wirusy lub robaki). Metody używa się też często do znajdowania rozwiązań przybliżonych, na podstawie, których później wylicza się ostateczny rezultat pełnym algorytmem. To ostatnie zastosowanie szczególnie dotyczy przypadków, gdy heurystyka jest wykorzystywana do nakierowywania pełnego algorytmu ku optymalnemu rozwiązaniu, aby zmniejszyć czas działania programu w typowym przypadku bez poświęcania, jakości rozwiązania (np. algorytm A\*).

Wyszukiwaniem informacji nazywamy proces przeszukiwania określonego zbioru dokumentów odnoszących się do tematu czy przedmiotu wskazanego w zapytaniu lub zawierających konieczne dla użytkownika fakty. Proces ten nie został jednak precyzyjnie i skończenie określony przez wzory, normy czy algorytmy i w dużej mierze opiera się na heurystykach w tym wypadku definiowanych, jako zbiór reguł oraz wskazówek, które mogą, lecz nie muszą, prowadzić do właściwego rozwiązania. [2]

Zarys historyczny

Jako pierwszy heurystyką zajmował sie Pappus z Aleksandrii, matematyk żyjący na przełomie III i IV wieku, usystematyzował i podsumował on wszystkie prace i osiągnięcia matematyków greckich. Uważał, że to Euklides, Apoloniusz z Pergi oraz Aristaeus Starszy byli twórcami heurystyki. Pappus, w swoich wielu pracach, zajmował sie przede wszystkim analiza i syntezą. Za podstawę analizy uważał zadany wynik zadania, a sama analiza nazywał proces rozkładania wyniku na elementy, z których da sie go osiągnąć. Według Pappusa, synteza była procesem odwrotnym do analizy, rozpoczynała się ona od elementów otrzymanych w wyniku analizy i wyprowadzających wynik. W jego ujęciu, analiza była układaniem planu rozwiązania, natomiast synteza wykonaniem tego planu. Zasługą Pappusa, z punktu widzenia współczesnej heurystyki było spostrzeżenie, że przystępując do rozwiązywania zadania (analizy), zakładamy prawdziwość lub spełnienie wyniku. Matematyk ten zajmował się także metodami dowodzenia twierdzeń, dobierania zadań pomocniczych i innymi problemami.

Prace Pappusa, nad metodami rozwiązywania problemów, kontynuował filozof i matematyk Rene Descartes (1596-1650). W swoich dwóch pracach: „Rozprawa o metodzie.” i niedokończonej: „Reguły kierowania umysłem.” starał się przedstawić zasady dokonywania odkryć, rozwiązywania nowych problemów. Starał się poznać i podać metodę myślenia całkowicie niezawodną, uniwersalną i użyteczną w praktyce. W rozważaniach opierał się głównie na wzorcach rozumowania.

Gottfried Leibniz (1646-1716) w „Sztuce dokonywania odkryć” i wielu innych swoich pracach podkreślał znaczenia wyjaśnienia źródeł odkryć i wyprowadzenia odkryć. Jego ideałem dochodzenia do wiedzy był również matematyczny sposób myślenia. Leibniz stworzył nawet projekt uniwersalnego języka ideograficznego.

Bernard Bolzano (1781-1848) określił w bezpośredni sposób zadania heurystyki. Polegały one na sformułowaniu reguł i metod badania problemów dokonanym na podstawie analizy metod ich rozwiązywania. Zasługą Bolzano było także oddzielenie psychologii od logiki i wprowadzenie pojęcia rozumienia zadań wynikającego z reguł logiki matematycznej.

Gyorgy Polya w 1945 roku opublikował swoją pracę pt. „Jak to rozwiązać”, którą uznano za jedna z najważniejszych w literaturze dotyczącej sztucznej inteligencji. Poświecona jest ona przede wszystkim nauczaniu matematyki, a dotyczy nauczania myślenia. „Krótki słownik heurystyczny” to najważniejsza część tej rozprawy. Zawiera on podstawowe pojęcia heurystyki. Polya potraktował heurystykę, jako przedmiot badań i zastosowań praktycznych, podkreślając jej podstawy doświadczalne. Według niego, heurystyka jest formalizacją doświadczeń zebranych przy rozwiązywaniu zadań oraz obserwacji rozwiązywania zadań, a także próbą znalezienia wspólnych cech różnych metod rozwiązywania. Takie podejście pedagogiczne sprowadziło zadania heurystyki do lepszego rozumienia procesu myślenia w rozwiązywaniu problemów i nauczaniu. Poszczególne hasła słownika sygnalizowały problemy prawdopodobieństwa uzyskania rozwiązania, jego użyteczności oraz nieścisłość operacji myślowych a także ich uwarunkowania psychologiczne.

Metoda heurystyczna jest także uważana za narzędzie planowania sposobów rozwiązywania, podobnie do analizy opisywanej przez Pappusa. W tym sensie każda procedura jest heurystyczna, jeżeli tylko jej celem jest odkrycie nieznanego rozwiązania danego problemu. O wyborze metody rozwiązywania ma decydować właśnie heurystyka.

Pojmowanie heurystyki współcześnie jest coraz bardziej odległe od koncepcji Polya. Jego metody dotyczyły głównie odkrywania, wymyślania rozwiązań, dopuszczały także „działania po omacku”. Rozwiązanie u Polya, z metodologicznego punktu widzenia, było wynikiem ogólnych rozważań, które dopasowywano do konkretnego problemu. Przy heurystycznym rozwiązywaniu zadań duże znaczenie ma obecnie stosowanie metod ściśle związanych ze specyfikacja dziedziny, do której należy problem. Kierunkiem dominującym przy wprowadzaniu i odkrywaniu nowych heurystyk jest poprawianie metod i strategii już istniejących. Dąży się także do formalnego zdefiniowania struktur i przestrzeni obiektów rozwiązywanych problemów. [3]

Algorytm a heurystyka

Wielu autorów proponuje połączenie pojęcia heurystyki i algorytmu. Algorytm według nich jest rodzajem generatora rozwiązań, spełniających następujące warunki:

* zastosowanie algorytmu wyznacza skończony ciąg stanów;
* ciąg stanów ma stan początkowy (zbiór wejściowy, opis sytuacji);
* każdy stan (poza końcowym) ma jednoznaczny następnik;
* stan końcowy jest rozwiązaniem lub sygnałem, ze problem nie ma rozwiązania.

Algorytm jest, więc działaniem według ściśle wyznaczonego schematu; od stanu początkowego do stanu końcowego. Natomiast heurystyka może rozwiązywać dany problem, lecz nie daje gwarancji znalezienia rozwiązania. Działa jak matematyk uzasadniający twierdzenie nie wiedząc przy tym czy jest ono prawdziwe i czy jest metoda postępowania, która zakończy sie sukcesem. W tym zestawieniu postępowanie heurystyczne jest bardzo twórcze. Wiele kontrowersji wzbudza takie przeciwstawianie pojęć algorytmu i heurystyki. [3]

Funkcje heurystyczne w przeszukiwaniu

Z punktu widzenia efektywności przeszukiwania istotne jest uwzględnienie następujących czynników:

* niepewność wyniku,
* niekompletność dostępnej wiedzy,
* poprawienie uzyskanego wyniku.

Wykorzystanie heurystyki w procesie konstruowania rozwiązania zadania zwiększa niepewność otrzymania wyniku. Niepewność wyniku związana jest z wykorzystaniem wiedzy „nieformalnej” (praw, reguł, intuicji), której użyteczność nie jest do końca znana. Z tego względu metody heurystyczne wykorzystywane są tam, gdzie algorytmy wyznaczają niezadowalające rozwiązania lub nie gwarantują rozwiązania zadania. Maja one szczególne znaczenie przy rozwiązywaniu zadań (problemów) o dużej złożoności (gdzie dokładny algorytm zawodzi), a zwłaszcza w zadaniach związanych z rozpoznawaniem mowy, obrazów, w robotyce, konstruowaniu strategii gier. Prosty sposób modelowania dziedziny problemów za pomocą obiektów i operatorów umożliwia zastosowanie metod heurystycznego przeszukiwania.

Przy rozwiązywaniu problemów często korzysta się z informacji niepewnych i nieprecyzyjnych, szczególnie wtedy, gdy przetwarzana jest informacja pochodząca ze świata rzeczywistego. Niepewność polega na tym, że nie można jednoznacznie określić czy są one prawdziwe czy fałszywe. Ocena badanych obiektów jest więc często subiektywna, zależy ona od reguł wypracowanych doświadczalnie (opinia ekspertów). Nieprecyzyjność przejawia sie w tym, że nie możliwe jest jednoznaczne umieszczenie informacji w obrębie przyjętej skali wartości.

Za pomocą metod heurystycznych można w naturalny sposób wykorzystać informacje niepewna i nieprecyzyjna. Przy definiowaniu heurystyk opierających się na doświadczeniu, duże znaczenie ma zrozumienie i uwzględnienie uwarunkowań rozwiązywanego zadania.

Główne zadanie heurystyki polega na usprawnieniu algorytmu rozwiązywania danego problemu. Najważniejszym jest eliminowanie z dalszych rozważań tej części obiektów, które nie zostały jeszcze sprawdzone. Jakość zastosowanej heurystyki ma wpływ na złożoność rozwiązania problemu. Natomiast za najważniejszą funkcję heurystyki uważa sie kierowanie procesem decyzyjnym bezpośrednio (przez wskazanie najlepszych kierunków poszukiwania rozwiązania) lub pośrednio (przez eliminowanie najmniej obiecujących kierunków).

Cecha, która budzi najwięcej kontrowersji (spośród wymienionych), jest ta, która mówi, że heurystyka nie gwarantuje znalezienia rozwiązania. Mimo to, to właśnie ta cecha dla wielu autorów ma zalety. Twierdzą oni, że postęp w dziedzinie rozwiązywania problemów jest opóźniany przez nadmierne wykorzystywanie zupełnych reguł decyzyjnych, tzn. takich, które nie eliminują z rozwiązań żadnych kierunków prowadzących do rozwiązania (o ile takie istnieje). Najnowsze metody przeszukiwania heurystycznego maja stanowić pomost pomiędzy zupełnością algorytmów a ich optymalną złożonością. Strategie modyfikowane są w kierunku wyznaczenia rozwiązania quasi-optymalnego (zamiast optymalnego) ze znaczna redukcja kosztów. [3]

Charakterystyka przeszukiwania heurystycznego

Wykorzystanie heurystyki, jako strategii kontrolnej lub zarządzającej wyborem najbardziej obiecujących kierunków przeszukiwania (bezpośrednio lub pośrednio) jest najważniejsze we wszelkich zadaniach przeszukiwania. Przy rozstrzygnięciach heurystycznych może być wykorzystana wiedza „nieformalna”, której zastosowanie nie daje całkowitej pewności jej użyteczności. Czynniki te zostały jednak uwzględnione podczas formowania definicji (określenia) przeszukiwania heurystycznego.

„Heurystyczne”, w zadaniach przeszukiwania, określa się wszelkie prawa, kryteria, zasady i intuicje (również takie, których skuteczność nie jest całkowicie pewna), które pozwalają jednak wybrać najbardziej efektywne kierunki działania z punktu widzenia osiągnięcia zamierzonego celu.

Celem poprawienia efektywności systemu rozwiązującego dany problem dołącza się do niego heurystykę. Wynikiem zastosowania praktycznych reguł (często intuicyjnych i empirycznych), ściśle związanych z dziedzina, z której pochodzi problem daje lepszą jakość. [3]

Klasyczne strategie przeszukiwania

* **Strategia z powracaniem (backtracking)**

Strategia ta jest modyfikacją algorytmu przeszukiwania w głąb. Najważniejsza różnica polega na tym, że ekspansja badanego węzła (generowanie wszystkich potomków), jest zastąpiona jego rozszerzeniem (generowaniem jednego potomka). Dla wybranego węzła jest generowany tylko jeden potomek i w przypadku, gdy nowy węzeł nie spełnia kryterium celu lub końcowego, to jest dalej rozszerzany. Jeśli po kolejnych rozszerzeniach otrzymany węzeł spełnia kryterium końca przeszukiwanego grafu lub nie można dla niego wygenerować nowego potomka, to następuje powrót do najbliższego przodka, dla którego jest możliwe wygenerowanie potomków. Główną zaletą tej strategii jest oszczędność pamięci. Większa niż przy strategii w głąb. Strategia z powracaniem gwarantuje, że po jej zakończeniu wszystkie wygenerowane węzły są przetestowane. Strategia w głąb nie posiada tej właściwości, w jej przypadku część węzłów otrzymanych w wyniku kolejnych ekspansji może w ogóle nie być testowana w czasie wykonywania algorytmu. Oszczędność pamięci odbywa się kosztem komplikacji strategii. Testowanie krawędzi każdego rozszerzanego węzła wymaga dodatkowego nakładu obliczeń.

* **Strategia wszerz (breadth-first)**

Strategia wszerz bada kolejno poziomy grafu o jednakowej głębokości, i przyznaje wyższy priorytet węzłom o mniejszej głębokości. Algorytm wszerz wyznacza węzeł celu o najmniejszej głębokości (w porównaniu z innymi węzłami celu). Strategia ta daje gwarancję, że dla lokalnie skończonych grafów (każdy węzeł ma skończoną liczbę potomków) osiągnie się węzeł celu, jeżeli istnieje. Strategia wszerz ma, więc bardzo pożądaną cechę zbieżności. Dodatkowo, jako pierwsze wyznacza rozwiązanie optymalne pod względem długości ścieżki rozwiązania. Strategia wszerz ma jednak poważne wady. Analizowane są wszystkie węzły o głębokości mniejszej od głębokości wyznaczonego węzła celu. Zamiast jednej ścieżki w pamięci są przechowywane wszystkie węzły o danej głębokości przed wygenerowaniem jakiegokolwiek węzła o głębokości o jeden większej, co powoduje duże wymagania pamięciowe. W każdym kroku strategii następują powroty do węzłów wygenerowanych i przechowywanych w pamięci od wielu kroków.

* **Strategia zachłanna (hill-climbing)**

Główną operacją strategii zachłannej jest ekspansja węzłów. Gdy zostaje wykonana są badane nowe węzły i najbardziej obiecujący z nich jest wybierany do dalszej ekspansji. Strategia wykorzystuje lokalną optymalizację i nie pozwala na powroty do żadnego przodka aktualnie badanego węzła. Jest strategią nieodwracalną i niesystematyczną.  
Strategia ta odznacza się jednak prostym algorytmem obliczeniowym. Wadą jest brak możliwości powrotu do kierunków przeszukiwania, które na pewnym etapie były lokalnie gorsze. Może to prowadzić do badania drogi prowadzącej do węzła końcowego nie spełniającego kryterium celu lub do penetrowania drogi nieskończonej.

* **Strategia "najpierw najlepszy" (best-first)**

Wykorzystuje ona pewną informację heurystyczną związaną z rozwiązywanym problemem, w celu zminimalizowania kosztów przeszukiwania. Stosuje się, więc pewną funkcję heurystyczną, która wyraża ocenę węzła ze względu na następujące kryteria:

* zbieżność, czyli osiągnięcia celu;
* najmniejszego kosztu drogi wyznaczanej od węzła początkowego, przez węzeł w, do węzła końcowego;
* najmniejszej złożoności obliczeniowej procesu przeszukiwania.

Rozwijany jest "najlepszy" węzeł spośród węzłów rozpatrywanych do tej pory, niezależnie od ich położenia w grafie. Zakłada się, że węzeł najbardziej obiecujący ma najmniejszą wartość funkcji heurystycznej.

Operacją odróżniającą omawianą strategię od strategii wcześniej opisanych, jest uporządkowanie listy (zawierającej węzły grafu przestrzeni stanów z nie wykorzystanymi krawędziami) według wartości funkcji heurystycznej. Rozszerzenie węzłów jest dokonywane, podobnie jak w strategii w głąb, przez ekspansję, czyli generowanie wszystkich potomków.

* **Strategia A\***

Kolejną metodą heurystycznego przeszukiwania jest strategia A\*. Celem tej strategii jest wyznaczenie najtańszej drogi w grafie. Funkcja heurystyczna f(w) w strategii A\* jest sumą dwóch składników.

**f(w) = h(w) + g(w)**

To wyrażenie oznacza, że dla danego węzła w jest wyznaczana najpierw w sposób heurystyczny estymacja h(w) kosztu drogi łączącej węzeł w z węzłem celu. Następnie wyznacza się dla węzła w koszt drogi łączącej węzeł początkowy p z węzłem w, co reprezentuje składnik g(w). Odmianą omawianej strategii jest algorytm A\* z tzw. iteracyjnym pogłębianiem. Służy on do znajdowania najkrótszej drogi w grafie zaczynając od węzła początkowego do węzła stanowiącego cel lub węzła najbardziej do tego cel zbliżonego. Pomocą w przeszukiwaniu drzewa stanów jest tutaj również funkcja heurystyczna, określająca, jakość aktualnego węzła. Funkcja heurystyczna powinna uwzględniać w swojej ocenie zarówno przybliżoną odległość od węzła celu, koszt drogi od węzła początkowego do węzła aktualnego, jak i złożoność procesu obliczeniowego. Dla tego algorytmu najważniejsze są długość drogi w grafie oraz przybliżona odległość od celu. Algorytm A\* z iteracyjnym pogłębianiem służy do znajdowania najlepszego istniejącego rozwiązania. Algorytm A\* z iteracyjnym pogłębianiem realizuje schemat algorytmu w głąb dla kolejnych iteracji z przyjętą wartością progu, dotyczącego wartości funkcji heurystycznych. Jeżeli wartość funkcji heurystycznej badanego węzła jest większa od zadanego progu, to następuje powrót. W jednej iteracji są badane wszystkie węzły z wartością funkcji heurystycznej mniejszą od zadanego progu; w kolejnych iteracjach algorytmu próg ten jest zwiększany. Zaletą algorytmu A\* z iteracyjnym pogłębianiem w porównaniu z tradycyjnym algorytmem A\* jest zmniejszenie wymagań dotyczących pamięci komputera. [4]

Reprezentacja zadań i przestrzeni poszukiwań

Rozwiązanie zadania jest często realizowane przez przeszukiwanie zbioru wszystkich możliwych stanów (przestrzeni przeszukiwania). Całkowite przebadanie przestrzeni jest mało efektywne, jednak często realizowane praktycznie, ale możliwe jedynie w przypadku małych przestrzeni. Efektywną metodą rozwiązywania dużych zadań jest generowanie stanów za pomocą operatorów według określonych zasad i badanie ich właściwości.  
 Biorąc pod uwagę opis zadania należy uwzględnić trzy podstawowe wymagania:

* sposób reprezentacji każdego ze stanów przestrzeni przeszukiwania(kod),
* metody obliczeniowe (operatory) pozwalające na wygenerowanie kodu kolejnego stanu na podstawie kodu danego stanu,
* metody wyboru operatorów (strategii sterowania) spośród zestawu możliwych do zastosowania.

Najbardziej pożądane cechy kodu stanów to: jednoznaczność i uwzględnienie struktury zadania. Ten sposób reprezentacji pozwala na efektywne przekształcenia zbioru za pomocą operacji rozszczepiania, która polega na podziale problemu reprezentowanego przez dany stan na podproblemy, odrzuceniu części stanów i badaniu jedynie najbardziej obiecujących. Kolejne podziały mogą przekształcić problem początkowy na problem łatwiejszy do rozwiązania. Przedstawiony wyżej sposób rozwiązywania zadań jest nazywany metodą rozszczepiania i odrzucania (split-and-prune). W sztucznej inteligencji z powodu dużych rozmiarów zadań, są stosowane podobne reguły postępowania, nazywane metodą generowania i testowania (generate-and-test). Zamiast odrzucania stanów z pewnego zbioru, generuje się nowe stany, z których część wykorzystuje się do dalszego badania.  
 Optymalne wykonywanie przekształcania, generowania i testowania wymaga spełnienia przez kody stanów pewnych dodatkowych warunków. Przyjmujemy, że kod danego stanu zawiera dodatkowe informacje, definiujące wprost pod problem. Kod o takich cechach jest nazywany kodem stanu i pozwala na przyspieszenie procesu przeszukiwania przestrzeni stanów. [4]

Bibliotek JUNG

JUNG - Java Universal Network / Graph Framework - jest biblioteką oprogramowania, która zapewnia wspólny i rozszerzalny język do modelowania, analizy i wizualizacji danych, które mogą być reprezentowane jako wykres lub sieć. Został napisany w języku Java, dzięki czemu aplikacje oparte na JUNG mogą korzystać z rozbudowanych funkcji API Java, a także z innych istniejących bibliotek Java innych firm.

Architektura JUNG została zaprojektowana w celu obsługi różnorodnych reprezentacji podmiotów i ich relacji, takich jak wykresy ukierunkowane i nieukierunkowane, wykresy multimodalne, wykresy z równoległymi krawędziami i hipergrafy. Zapewnia mechanizm opisywania wykresów, encji i relacji z metadanymi. Ułatwia to tworzenie narzędzi analitycznych dla złożonych zestawów danych, które mogą badać relacje między podmiotami, a także metadane związane z każdą jednostką i relacją.

Obecna dystrybucja JUNG obejmuje implementacje szeregu algorytmów z teorii grafów, eksploracji danych i analizy sieci społecznościowych, takich jak procedury klastrowania, dekompozycji, optymalizacji, generowania wykresów losowych, analizy statystycznej i obliczania odległości sieci, przepływów i ważne miary (centralność, PageRank, HITS itp.).

JUNG zapewnia także strukturę wizualizacji, która ułatwia konstruowanie narzędzi do interaktywnej eksploracji danych sieciowych. Użytkownicy mogą korzystać z jednego z dostarczonych algorytmów układu lub użyć frameworku do tworzenia własnych układów niestandardowych. Ponadto zapewniono mechanizmy filtrujące, które umożliwiają użytkownikom skupienie uwagi lub ich algorytmów na określonych częściach wykresu. Jako biblioteka typu open source JUNG zapewnia wspólną strukturę do analizy / wizualizacji grafów/sieci.  [5]

Aktualna wersja biblioteki: 2.0.1

Dokumentacja: http://jung.sourceforge.net/doc/api/index.html3

Wprowadzenie: http://informatyka.wroc.pl/node/310?page=0,0

Opis programu i przykładu testowego

Program ma na celu wyznaczenie optymalnej ścieżki między dwoma wierzchołkami grafu, za pomocą algorytmu BFS, DFS, Dijkstra i dostępnej w bibliotece JUNG, przy czym każda węzeł posiada swoją określoną wagę.

Przykład testu:

Student lubi podróżować, chce w jak najtańszy sposób dotrzeć z miejsca X do miejsca Y za pomocą serwisu BlaBlaCar, niestety często nie znajduje oferty bezpośredniego przejazdu, jedynie krótsze przejazdy do wybranych miejscowości.

Dana jest mapa wraz z kosztami przejazdu każdego odcinków między miastami. Za pomocą algorytmów BFS, DFS, Dijkstra wyznacz trasy, dzięki którym student dotrze do miejscowości Y jak najniższym kosztem.

Wszystkie dane wczytaj z pliku txt.

Główne części programu i wyniki

5. Opisanie danych wykorzystanych do testowania działania metody wybranej biblioteki  
6. Omówienie głównych części programu/skryptu i wyników.

Wnioski

Nie wszystkie własności strategii przeszukiwania można badać w sposób ogólny. Zależą one miedzy innymi od przeszukiwanego grafu (warunki zadania) oraz jakości zastosowanych metod (funkcje heurystyczne).

Przy porównywaniu różnych strategii wykorzystywane jest pojecie przewagi - dominacji (ang. dominance) strategii. Strategia S1 ma przewagę nad strategia S2, gdy każdy węzeł badany przez strategie S1 jest również zbadany przez S2 (strategia S2 analizuje co najmniej tyle samo węzłów co strategia S1). Strategia jest nazywana optymalna w pewnej klasie, gdy ma przewagę nad wszystkimi strategiami tej klasy.

Przy badaniu efektywności strategii przeszukiwania duże znaczenie odgrywa ich złożoność. Z praktycznego punktu widzenia na złożoność składają sie dwa zasadnicze składniki:

* koszt wykonania znalezionego rozwiązania (związany ze spełnianiem przez strategie warunku dopuszczalności);
* koszt wyznaczenia rozwiązania, na który składają się koszty generowania i sprawdzania węzłów oraz wykrywania niepożądanych kierunków przeszukiwania.

W wielu zadaniach praktycznych oba składniki kosztu i są równoważne.

Funkcje heurystyczne wpływają oczywiście na każdą z wymienionych cech strategii przeszukiwania - zbieżność, dopuszczalność i złożoność. Głównym zadanie heurystyki jest poprawa, jakości rozwiązania i zmniejszenie złożoności jego poszukiwania dzięki ograniczaniu algorytmowych kierunków przeszukiwania badanych strategii. Główna cecha

„dobrej” heurystyki jest wykorzystanie przez nią informacji specyficznych dla danego problemu (dopasowanie strategii do konkretnego zadania), a to oczywiście powoduje trudności w ogólnej analizie własności heurystycznych. Możliwe jest jednak wyodrębnienie pewnych cech ogólnych tych funkcji, pozwalających na klasyfikacje i charakterystykę.

W przedstawionych w projekcie przykładach ...???.....

Dzięki zaimplementowanych w bibliotece JUNG funkcjach można z łatwością przedstawić grafy i poszczególne strategie przeszukiwań. Jest ona idealnym narzędziem do wspomagania projektów informatycznych wymagających użycia metod przeszukiwania z wykorzystaniem heurystyki.

Bibliografia

Źródła internetowe:

1. http://www.cs.put.poznan.pl/amichalski/wsi/AI4&5.rgb.pdf
2. https://pl.wikipedia.org/wiki/Heurystyka\_(informatyka)
3. https://www.ii.pwr.edu.pl/~kwasnicka/tekstystudenckie/metodyheurystycznewgrach.pdf
4. http://www.ummon.org/Others/Documents/SystemyEkspertowe/08\_SystemyEkspertowe/03\_SE.html
5. http://jung.sourceforge.net/
6. http://www.neurosoft.edu.pl/media/pdf/jbartman/sztuczna\_inteligencja/SI\_2\_strategie\_poszukiwania.pdf

Literatura:

1. L. Bolc - „Metody przeszukiwania heurystycznego”, tom 1.
2. Bolc - „Metody przeszukiwania heurystycznego”, tom 2.
3. Antoszkiewicz - „Metody heurystyczne. twórcze rozwiazywanie problemów.”