

Sztuczna inteligencja

Ćwiczenia 3

Pierwsze zajęcia po pracowniach P3

Każde zadanie warte jest 1 punkt. Zadanie z gwiazdką nie wlicza się do maksimum.

Zadanie 1. (1p) Opisz poniższe algorytmy (możesz użyć nazw, jeżeli znasz):

- a) local beam search dla $k = 1$,
- b) local beam search z jednym początkowym stanem i bez limitu na liczbę zachowanych stanów po generacji następnika,
- c) symulowane wyżarzanie z $T = 0$ przez cały czas działania algorytmu,
- d) symulowane wyżarzanie z $T = \infty$ przez cały czas działania algorytmu,
- e) algorytm genetyczny z populacją wielkości 1.

Odpowiedzi uzasadnij.

Zadanie 2. (1p) Heurystyczne algorytmy można łączyć ze sobą w dość dowolne kombinacje. Zaproponuj sensowne¹ połączenie:

- a) algorytmów ewolucyjnych i hill climbing,
- b) A^* oraz local beam search,
- c) symulowanego wyżarzania i algorytmów ewolucyjnych (inaczej niż w podpunkcie a, z symulowanego wyżarzania powinniśmy wziąć jedynie ogólną ideę),
- d) taboo search z algorytmami ewolucyjnymi.

W razie potrzeby krótko wyjaśnij, jak działają oryginalne algorytmy.

Zadanie 3. (1p*) Przeczytaj i opowiedz o tym, czym są algorytmy mrówkowe (ant colony algorithms). Powiedz, jak zastosować je do rozwiązywania problemu komiwojażera.

Zadanie 4. (1p) Podobnie jak burze, obrazki logiczne można również rozwiązać za pomocą więzów w Prologu. Opisz, jak modelować obrazki logiczne używając więzów dostępnych w SWI-Prologu (W5 i początek W6, możesz też skonsultować się z dokumentacją modułu clpfd w SWI-Prologu).

Zadanie 5. (1p) Sprawdź, jak działa łamigłówka statki (<https://www.puzzle-battleships.com/>). W razie potrzeby, krótko wyjaśnij jej zasady. Powiedz, jak ją rozwiązać, wykorzystując podejście z P3 (z zadań 3 i 4).

Zadanie 6. (1p) Rozważamy uproszczone zadanie układania planu lekcji, w którym (między innymi) nie przejmujemy się dostępnością sal.

- Mamy pewną liczbę zajęć do rozmieszczenia.
- Każde zajęcia mają następujące parametry: klasa (w znaczeniu „grupa uczniów”) oraz nauczyciel.
- Każdemu zajęciu należy przypisać termin: liczbę od 1 do 50. Liczby od 1 do 10 oznaczają poniedziałek, od 11 do 20 – wtorek, itd.
- Żadna klasa nie może mieć „okienka”, czyli przerwy w zajęciach. Czyli każdego dnia przychodzi na jakąś godzinę, ma zajęte wszystkie kolejne godziny, a potem idzie do domu i do szkoły przychodzi następnego dnia.

Przedstaw to zadanie jako problem więzowy, który da się wyrazić za pomocą konstrukcji dostępnych w SWI-Prologu. Przy definiowaniu warunków na bezokienkowość planu możesz potrzebować dodatkowych zmiennych.

¹Trudno zdefiniować precyzyjnie sensowność. Na potrzeby tego zadania przyjmujemy, że sensowne oznacza przekonanie autora, że warto takie połączenie sprawdzić w pewnych zadaniach – i że może ono zadziałać lepiej niż każdy ze składników kombinacji

Zadanie 7. (1p) Przedstaw zadanie układania planu zajęć w Instytucie Informatyki jako problem spełnialności więzów, w którym oprócz twardych wymagań, określających jakie plany są dopuszczalne², istnieje funkcja określająca jakość dopuszczalnego planu i tym samym jesteśmy zainteresowani znalezieniem dopuszczalnego planu maksymalizującego tę jakość. Najważniejszą częścią tego zadania jest przedstawienie propozycji takiej funkcji, czyli próba precyzyjnego opisu, co sprawia, że jakiś plan jest lepszy od innego. Powinieneś uwzględnić wyniki głosowania.

Zadanie 8. (1p) Wybierz jedną z następujących gier: lis i gęsi, breakthrough (board game), pentago, skoczki (gra planszowa). Zaproponuj dla wybranej gry heurystyczną funkcję oceniającą sytuację na planszy. Wyraźnie wskaż, jakie ta funkcja ma parametry.

Uwaga: zapytanie: `<gra-X> wikipedia` zawsze prowadzi do odpowiedniego opisu. Bądź przygotowany na krótkie opisanie zasad gry (w razie potrzeby).

Zadanie 9. (1p), ★ Jak wyżej, tylko inna gra. Możesz też zaproponować grę spoza powyższej listy.

Zadanie 10. Rozważamy grę „kółko i krzyżyk z buczeniem”. Mamy zwykłą planszę do kółka i krzyżyka, dwóch graczy i serwer gry. Gracze zgłaszają propozycje ruchu do serwera gry (gracz A nie widzi propozycji gracza B i odwrotnie) i serwer albo przyjmuje ten ruch (jeżeli zaproponowane pole jest puste), albo buczy i gracz ma prawo do kolejnej próby (dowolnie wiele razy). Buczenie jest słyszalne dla obu graczy, niedozwolone jest powtarzanie jakiegoś ruchu (tzn. jak przy ruchu przeciwnika usłyszałem buczenie, to mogę założyć, że zdobył on dodatkową wiedzę, a nie że próbuje mnie zmylić). Oprócz tego to normalne kółko i krzyżyk (tzn. celem jest ustawienie 3 swoich znaków w wierszu, kolumnie lub na przekątnej).

Zaproponuj agenta, który grałby w tę grę (możliwie dobrze). Oczywiście pierwszy gracz ma dużą przewagę (dlaczego?), więc zakładamy, że rozgrywka składa się z $2K$ partii, w których gracze rozpoczynają na zmianę.

Zadanie 11. Dla algorytmów Alpha-Beta-Search oraz MCTS odpowiedz na pytania:

- a) W jaki sposób można wykorzystać czas poświęcony na obliczenia najlepszego *poprzedniego* ruchu do obliczenia najlepszego *bieżącego* ruchu (zakładamy, że rozgrywamy tylko jedną partię).
- b) W jaki sposób wykorzystać możliwość równoległego wykonywania kodu w celu poprawy jakości gry?

Zadanie 12. (2p, ★) Przeglądaj artykuł: <https://dke.maastrichtuniversity.nl/m.winands/documents/BestReplySearch.pdf>
Odpowiedz na pytania:

- a) Co to jest algorytm \max^n ?
- b) Czym jest *paranoid assumption*?
- c) Jakie problemy łączą się z tymi dwoma algorytmami?
- d) Jak działa algorytm Best Reply Search, jakie są jego siły i słabości?
- e) Czym jest gra Rolit?
- f) Przedstaw dokładniej wybrany eksperyment z pracy i opisz jego wyniki.

Jeżeli będziesz prezentował to rozwiązanie postaraj się, by wszyscy mogli się jak najwięcej dowiedzieć z Twojej prezentacji, przy założeniu, że nie zajmiesz więcej niż 10 minut zajęć.

Pytanie dodatkowe, które można pominąć: jak wyobrażałbyś sobie grę Rolit w wersji pudełkowej (postaraj się powstrzymać od korzystania z wyszukiwarki przed wymyśleniem odpowiedzi)?

²W znacznym stopniu analogicznych do przedstawionych w poprzednim zadaniu.