

Sztuczna inteligencja i inżynieria wiedzy

laboratorium

Ćwiczenie 2. Problem spełniania ograniczeń

Opracowanie: J. Jakubik, W. Walentynowicz

Cel ćwiczenia

Zapoznanie się z podstawowymi algorytmami stosowanymi do rozwiązywania problemów spełniania ograniczeń (ang. *Constraint Satisfaction Problem*, CSP) przez własną implementację i zbadanie właściwości algorytmów na wytworzonym prototypie.

Realizacja ćwiczenia

- Zapoznanie się z działaniem algorytmów: sprawdzenia wprzód (ang. *forward checking*), przeszukiwania przyrostowego z powracaniem (ang. *backtracking*).
- Rozwiązanie dwóch problemów CSP: Futoshiki, oraz Skyscraper lub innej skonsultowanej z prowadzącym łamigłówki jako drugim problemie.
- Sformułowanie wybranych problemów jako problemów spełniania ograniczeń, poprzez podanie zmiennych, ich dziedzin i ograniczeń.
- Implementacja algorytmów sprawdzenia wprzód oraz przeszukiwania przyrostowego z powracaniem.
- Zaproponowanie i implementacja heurystyk: wyboru kolejnej zmiennej do wartościowania, wyboru kolejnej do przypisania.
- Porównanie czasów przetwarzania i liczby iteracji zaimplementowanych metod.
- Zbadanie wpływu zastosowanych heurystyk na czas przetwarzania.
- Prezentacja najciekawszych (zdaniem studenta) wyników.
- Dyskusja otrzymanych wyników.
- Przygotowanie sprawozdania zawierającego powyższe punkty.

Problem 1. Futoshiki

Futoshiki to logiczna łamigłówka rozwiązywana na kwadratowej planszy N na N pól. Żeby rozwiązać łamigłówkę, planszę należy wypełnić liczbami całkowitymi od 1 do N . Ograniczeniem tego wypełnienia jest reguła, że w każdej kolumnie i każdym rzędzie może znajdować się tylko jedno wystąpienie danej liczby. Oprócz tego, między niektórymi kratkami planszy (patrz rys. 1) znajduje się znak nierówności. Jest to dodatkowe ograniczenie, które jest spełnione tylko jeżeli liczby z tych kratek spełniają daną nierówność. Przypadki niespełnienia ograniczeń pokazane są na rysunkach 2 i 3. W przypadku z rysunku 2 ograniczenie nie jest spełnione, ponieważ pomiędzy kratkami zachodzi zależność *górna kratka zawiera liczbę większą od tej w dolnej kratce*, a wypełnienie jest odpowiednio 1 dla górnej kratki i 2 dla dolnej kratki (dlatego relacja jest zaznaczona kolorem czerwonym). W przypadku rysunku 3 ograniczenie nie jest spełnione, ponieważ powtórzono liczbę 3 w pierwszym rzędzie i w drugiej kolumnie (powtarzające się liczby zaznaczono kolorem czerwonym).

Rysunek 1	Rysunek 2	Rysunek 3

Należy zdefiniować problem wypełnienia planszy do Futoshiki w zależności od jej wypełniania początkowego oraz od zaznaczenia relacji między polami, jako problem CSP. Następnie należy rozwiązać ten problem wykorzystując algorytmy sprawdzenia wprzód oraz przeszukiwania przyrostowego z powracaniem dla różnych (udostępnionych w paczce z danymi) plansz. Należy porównać czasy przetwarzania dwóch metod dla różnych plansz (porównać też wartości otrzymywane wraz ze wzrostem rozmiaru problemu), a także zaproponować heurystyki wyboru kolejnej zmiennej do wartościowania i wartości do przypisania i zbadać ich wpływ na czas przetwarzania.

Problem 2. Skyscraper

Skyscraper to logiczna łamigłówka rozwiązywana na kwadratowej planszy N na N pól. Żeby rozwiązać łamigłówkę, należy ją wypełnić liczbami całkowitymi od 1 do N , które reprezentują liczbę pięter budynku. Ograniczeniem tego wypełnienia jest reguła, że w każdej kolumnie i każdym rzędzie może znajdować się tylko jedno wystąpienie danej liczby. Oprócz tego na bokach planszy (patrz rys. 4) znajduje się informacja ile budynków widać w danym rzędzie/kolumnie patrząc od tej strony planszy. Budynek o większej liczbie pięter zasłania budynek o mniejszej ilości pięter.

Dla przykładu, gdy rząd będzie zawierał następujące liczby: $[2, 3, 1, 4]$. Patrząc od lewej jego krawędzi widzimy trzy wieżowce $[2 < 3 > 1 < 4]$, wieżowiec o dwóch piętrach NIE zasłania wieżowca o trzech piętrach. Wieżowiec o trzech piętrach zasłania wieżowiec o jednym piętrze. Wieżowiec o trzech piętrach NIE zasłania wieżowca o czterech piętrach. Wieżowiec o jednym piętrze został pominięty w porównywaniu z wieżowcem o czterech piętrach ponieważ został zasłonięty przez wieżowiec o trzech piętrach. Patrząc od prawej strony widzimy jeden wieżowiec, bo wieżowiec o 4 piętrach zasłania wszystkie zasłonięte sobą.

Przypadek niespełnienia ograniczeń pokazany jest na rysunku 5. W tym przypadku ograniczenia nie są spełnione, bo:

- dla drugiego rzędu patrząc od strony lewej widać trzy wieżowce, zamiast dwóch $[1 < 2 < 4]$,
- dla trzeciej kolumny patrząc od dołu widać dwa wieżowce, zamiast dwóch $[3 > 2 < 4]$.

	1	3	4	2	2	3	
1	6	2	1	5	4	3	4
2	4	3	2	1	6	5	2
2	5	6	4	3	1	2	4
4	1	4	3	2	5	6	1
3	2	5	6	4	3	1	4
3	3	1	5	6	2	4	2
	3	3	2	1	4	2	

Rysunek 4

		2	1	
3	2	3	1	4
2	1	2	4	3
2	3	4	2	1
	4	1	3	2
	1	3	3	

Rysunek 5

Należy zdefiniować problem wypełnienia planszy do Skyscraper w zależności od informacji na bokach planszy, jako problem CSP. Następnie należy rozwiązać ten problem wykorzystując algorytmy sprawdzenia wprzód oraz przeszukiwania przyrostowego z powracaniem dla różnych (udostępnionych w paczce z danymi) plansz. Należy porównać czasy przetwarzania dwóch metod dla różnych plansz (porównać też między sobą wartości wraz ze wzrostem rozmiaru problemu), oraz zaproponować heurystyki wyboru kolejnej zmiennej do wartościowania i wartości do przypisania, a także zbadać ich wpływ na czas przetwarzania.

Przykładowe heurystyki

Przykładowa heurystyka wyboru **zmiennej**: *najbardziej ograniczona*. Wybierz tę zmienną, która ma najmniej możliwych wartości do przypisania. Przykładowa heurystyka wyboru **wartości**: *najmniej ograniczająca*. Wybierz tę wartość, która wyklucza najmniej wartości możliwych do przypisania w powiązanych zmiennych.

Problem zaproponowany przez studenta

Zamiast problemu Skyscraper, można konsultując uprzednio wybór z prowadzącym, wybrać inną łamigłówkę logiczną, możliwą do rozwiązania jako problem CSP, np. kakuro, kryptarymy, etc. Wybrane zagadnienie musi umożliwiać sterowanie parametrem określającym rozmiar problemu!

Ocena ćwiczenia

2 pkt	Sformułowanie wybranych problemów jako CSP.
2 pkt	Implementacja algorytmów sprawdzenia wprzód i przeszukiwania przyrostowego z powracaniem dla wybranych problemów.
2 pkt	Przeprowadzenie badania czasów przetwarzania zaimplementowanych metod w rozwiązywaniu łamigłówki Futoshiki dla różnych plansz i ich rozmiarów. [W zbiorze danych pliki oznaczone przedrostkiem 'test_'.]
2 pkt	Przeprowadzenie badania czasów przetwarzania zaimplementowanych metod w rozwiązywaniu łamigłówki Skyscraper* dla różnych plansz i ich rozmiarów. [W zbiorze danych pliki oznaczone przedrostkiem 'test_'.]
2 pkt	Zbadanie wpływu zastosowania heurystyk na czas przetwarzania.

* w przypadku samodzielnie zaproponowanego zagadnienia przez studenta, student może konsultować z prowadzącym laboratorium, jakie parametry w badaniach zastosować.

WAŻNE! Podczas realizacji zadania należy zbadać zarówno czas wykonywania jak i liczbę instrukcji, które wykonują algorytmy. Należy również znaleźć wszystkie możliwe rozwiązania łamigłówek. Zalecane jest użycie profilera w celu optymalizacji kodu prototypów. Sugeruje się też, aby rozwiązania łamigłówek prezentować przy użyciu plików wyjściowych w formacie HTML.

Podział realizacji zadania

Zajęcia 1 – implementacja łamigłówek jako CSP.

Zajęcia 2 – implementacja metod sprawdzenia wprzód i przeszukiwania przyrostowego z powracaniem oraz heurystyk własnych studenta.

Zajęcia 3 – przeprowadzenie badań i wykonanie sprawozdania.

Materiały pomocnicze

- 1) Notatki z wykładu.
- 2) Wojna A., Przeszukiwanie przestrzeni stanów – problemu z więzami
https://www.mimuw.edu.pl/~awojna/SID/wyklady/przesz_z_wiezami.pdf
- 3) Strona z łamigłówką Futoshiki
<https://www.futoshiki.org/>
- 4) Strona z łamigłówką Skyscraper
<https://www.conceptispuzzles.com/index.aspx?uri=puzzle/skyscrapers/rules>