

Sztuczna inteligencja

Zadanie 2 : CSP

Wrocław, 15.11.2019

Kacper Kozimor

CSP to określenie ogółu problemów, w których priorytetem jest spełnienie warunku umiejscowienia zmiennej z dziedziny w rozwiązaniu .

Implementacja:

Utworzona przeze mnie implementacja składa się z dwóch głównych klas, klasy Graph odpowiadającej za tworzenie rozwiązań poprzez umieszczanie w Węzłach (Klasa Node) kolejnej planszy z rozwiązaniem.

Klasa graph zawiera implementacje metod Forward Checking oraz Backtracking dla problemu N hetmanów oraz problemu rozkładu liczby naturalnej większej niż 3 na sumy elementów ciągu Fibonacciego.

Problem 1: Problem N hetmanów:

Problem N hetmanów jest problemem szachowo-matematycznym. Został on sformułowany w połowie XIX w. Przez Maksa Bezzela i Franza Naucka. Zadanie polega na rozmieszczeniu na szachownicy o wymiarach $N \times N$, N figur hetmana szachowego, w taki sposób aby, żadna figura się nie atakowała. W przypadku hetmana szachowego oznacza to, że żadne dwie figury nie mogą się znaleźć w tej samej kolumnie, w tym samym rzędzie i w rzędach ukośnych.

Pomiary:

Zmierzono czas jaki zajęło programowi znalezienie pierwszego rozwiązania oraz liczbę węzłów odwiedzonych (utworzonych) w celu znalezienia wszystkich rozwiązań.

| Algorytm | Liczba N | Znalezione rozwiązania | Węzły odwiedzone | Czas do pierwszego rozwiązania (ms) | Czas ogółem (s) |
|-----------------|----------|------------------------|------------------|-------------------------------------|-----------------|
| Forward Cheking | 4 | 2 | 64 | 1 | 0,001 |
| Backtracking | 4 | 2 | 608 | 1 | 0,001 |
| Forward Cheking | 5 | 10 | 876 | 1 | 0,002 |
| Backtracking | 5 | 10 | 16 100 | 1 | 0,003 |
| Forward Cheking | 6 | 4 | 7 843 | 8 | 0,042 |
| Backtracking | 6 | 4 | 267 156 | 7 | 0,047 |
| Forward Cheking | 7 | 40 | 209 932 | 1 | 1,5 |
| Backtracking | 7 | 40 | 8 994 489 | 1 | 1,7 |
| Forward Cheking | 8 | 92 | 5 148 009 | 261 | 54 |
| Backtracking | 8 | 92 | 300 929 904 | 256 | 72 |
| Forward Cheking | 9 | 84 | 168 876 381 | 84 | 2 400 |
| Backtracking | 9 | 84 | 12 610 490 508 | 91 | 3 600 |

| | | | | | |
|--|----|--|--|--|-------------------------------------|
| | 10 | | | | Przekroczono czas oczekiwania |
| | 11 | | | | Przekroczono czas oczekiwania |

Wnioski:

Czas obliczania znalezienia pierwszego rozwiązania nie jest zależny od rozmiaru problemu. Czas obliczania ogółem jest zależny od liczby węzłów odwiedzanych. Algorytm backtracking w celu znalezienia wszystkich rozwiązań przeszukuje dużo więcej (wszystkie) elementy grafu. Przez to czas działania algorytmu backtracking jest widocznie dłuższy.

Problem 2: Rozkład liczby naturalnej na sumę liczb Fibonacciego

Każdą liczbę naturalną N większą od 2 można przedstawić jako sumę przynajmniej dwóch różnych dodatnich liczb Fibonacciego. Liczby Fibonacciego tworzą ciąg liczb naturalnych, w którym każdy wyraz (oprócz dwóch pierwszych liczb) jest sumą dwóch poprzednich: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... Niektóre liczby naturalne mogą być przedstawione jako różne sumy liczb Fibonacciego.

UWAGA: Przed rozpoczęciem obliczeń włączano jeden z algorytmów celem określenia liczby rozwiązań ogółem. Pomiary z poniższych tabel obejmują obliczenia algorytmu do momentu znalezienia wszystkich rozwiązań. Potem algorytm przestaje liczyć.

Pomiary:

Zmierzono czas jaki zajęło programowi znalezienie pierwszego rozwiązania oraz liczbę węzłów odwiedzonych (utworzonych) w celu znalezienia wszystkich rozwiązań.

| Algorytm | Liczba N | Rozwiązania | Węzły | Czas do rozwiązania (ms) | Czas ogółem (s) |
|-----------------|------------|-------------|-------------|--------------------------|-----------------|
| Forward Cheking | 59 | 3 | 52 004 | 6 | 0,037 |
| Backtracking | 59 | 3 | 53 459 | 1 | 0,054 |
| Forward Cheking | 617 | 5 | 464 792 490 | 365 tys. | 383 |
| Backtracking | 617 | 5 | 465 022 381 | 223 tys. | 234 |
| Forward Cheking | 1447 | 11 | X | Ponad godzinę | Ponad godzinę |
| Backtracking | 1447 | 11 | X | Ponad godzinę | Ponad godzinę |
| Forward Cheking | 2137 | 30 | X | Ponad godzinę | Ponad godzinę |
| Backtracking | 2137 | 30 | X | Ponad godzinę | Ponad godzinę |

| | | | | | |
|--|--------|--|---|---------------|---------------|
| | 10177 | | X | Ponad godzinę | Ponad godzinę |
| | 104009 | | X | Ponad godzinę | Ponad godzinę |

Wpływ heurystyk na czas przeszukiwania (do znalezienia pierwszego rozwiązania).

Zaimplementowano heurystykę, która wybiera z dziedziny liczbę, która pozostawi najmniej po odjęciu tej liczby od liczby N.

Pomiary:

| Algorytm | Liczba N | Rozwiązania | Węzły | Czas do rozwiązania (ms) | Czas ogółem (s) |
|-----------------|----------|-------------|------------|--------------------------|-----------------|
| Forward Cheking | 59 | 3 | 52 tys. | 1 | 0,037 |
| Backtracking | 59 | 3 | 53 tys. | 1 | 0,022 |
| Forward Cheking | 617 | 5 | 347 | 1 | 0,001 |
| Backtracking | 617 | 5 | 799 | 1 | 0,001 |
| Forward Cheking | 1447 | 11 | 662 mln. | 37 | 771 |
| Backtracking | 1447 | 11 | 674 mln. | 1 | 472 |
| Forward Cheking | 2137 | 30 | 1,616 mln. | 17 | 1 899 |
| Backtracking | 2137 | 30 | 1,622 mln. | 1 | 1 217 |
| | 10177 | | X | Ponad godzinę | Ponad godzinę |
| | 104009 | | X | Ponad godzinę | Ponad godzinę |

Wnioski:

Heurystyka wpływa znacząco na czas znalezienia pierwszego jak i pierwszych n rozwiązań. Niestety, przy zbyt rozgałęzionym grafie przeszukiwań, nawet zastosowanie heurystyki nie pomogło znaleźć rozwiązań w rozsądnym czasie.