

Sprawozdanie 2
Problem spełniania ograniczeń

Szymon Woźniak, 235040

12.04.2019

1 Wstęp teoretyczny

1.1 Problem spełniania ograniczeń

Problemy spełniania ograniczeń można zdefiniować poprzez trzy terminy:

- zmienne: V_1, V_2, \dots, V_k
- dziedziny zmiennych: D_1, D_2, \dots, D_k
- ograniczenia: C_1, C_2, \dots, C_L

W każdym z ograniczeń jest może być uwikłanych od 1 do n zmiennych. Celem zadania spełniania ograniczeń, jest znalezienie takiego przypisania wartości v_i każdej zmiennej V_i , że $v_i \in D_i$ i żadne z ograniczeń C_j nie jest złamane. Bardziej ogólnie, celem takiego zadania może być również znalezienie wszystkich tego typu przypisań i to właśnie zostanie rozpatrzone w tej pracy.

1.2 Rozwiązywane problemy

1.2.1 Wspólne cechy

Zmienne

Oba rozwiązywane w tej pracy problemy, są łamigłówkami logicznymi przedstawianymi jako kwadratowa plansza, składająca się z $n \times n$ pól. W terminach problemów spełniania ograniczeń każde pole jest zmienną o dziedzinie składającej się z liczb całkowitych $1..n$.

Ograniczenia

Dla obu problemów definiuje się również takie samo ograniczenie, mówiące że w żadnej z kolumn i w żadnym z wierszy planszy nie mogą wystąpić powtarzające się wartości. W każdym tego typu problemie występuje zatem co najmniej $2n$ ograniczeń, a w każdym z nich uwikłanych jest n zmiennych.

Rozmiar przestrzeni rozwiązań

Tak jak już wcześniej wspomniano, w obu problemach występuje $n \times n$ zmiennych, a dziedzina każdej z nich składa się z n wartości. Oznacza to więc, że rozmiar pełnej przestrzeni rozwiązań wynosi:

$$S = n^{n^2} \tag{1}$$

Rozmiar przestrzeni rośnie w zastraszającym tempie. Dla $n = 5$ jest to już $S \approx 2,98 \cdot 10^{17}$. W oczywisty sposób świadczy to o tym, że szukanie rozwiązań metodą przeglądu zupełnego nie może się sprawdzić dla tej klasy problemów.

1.2.2 Futoshiki

Futoshiki jest łamigłówką logiczną, w której przykładowa plansza może wyglądać następująco:



Rysunek 1: Przykładowa plansza Futoshiki

Oprócz wspomnianego już wcześniej ograniczenia nakładanego na kolumny i wiersze, w tym problemie definiuje się jeszcze jedno. Na obrazku 1 jest pokazane jako znaki nierówności pomiędzy dwoma sąsiadującymi polami. Oznacza to, że wartości przypisane tym zmiennym muszą spełniać daną nierówność. Przykład niespełnienia tego typu ograniczenia jest został przedstawiony na rysunku 2.



Rysunek 2: Niepoprawnie uzupełniona plansza Futoshiki

1.2.3 Skyscraper

Skyscraper jest łamigłówką logiczną, w której wartości przypisywane zmiennym reprezentują wysokość wieżowca stawianego na danym polu. W tym problemie definiuje się dodatkowe ograniczenie, poza tym nakładanym na kolumny i wiersze. Na każdym z boków planszy, obok każdego pola tego boku, może być napisana liczba, mówiąca ile wieżowców powinno być widać w wierszu lub kolumnie, jeżeli spojrzeć na planszę z tego miejsca. Jeżeli wyższy budynek znajduje się przed niższym, to go zasłania. Przykładowa plansza jest przedstawiona na rysunku 3, a przykład niespełnienia ograniczenia na rysunku 4.

	3	1	2	3	2	5	
2							3
2							2
3							3
1							3
2							2
3							1
	3	5	2	3	3	1	

Rysunek 3: Przykładowa plansza do łamigłówki Skyscraper

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

Rysunek 4: Przykład niespełnienia ograniczenia w łamigłówce Skyscraper

1.3 Proponowane rozwiązania

1.3.1 Algorytm przeszukiwania przyrostowego z powracaniem (*Backtracking*)

Algorytm ten ma na celu ograniczyć liczbę sprawdzanych rozwiązań. Wybierana jest zmienna, a następnie jest jej przypisywana wartość z dziedziny. Jeżeli po takim przypisaniu żadne ograniczenie nie zostało złamane, to algorytm przechodzi do przypisania wartości następnej zmiennej. Jeżeli w dziedzinie aktualnie rozpatrywanej zmiennej nie istnieje wartość niełamająca żadnego z ograniczeń, następuje powrót do poprzedniej zmiennej i próba przypisania kolejnej wartości z jej dziedziny.

Zadanie jest rozwiązane w momencie kiedy wszystkie zmienne mają przypisane wartości i żadne ograniczenie nie jest złamane.

1.3.2 Algorytm sprawdzania w przód (*Forward checking*)

Algorytm ten postępuje podobnie do wspomnianego wcześniej. Różnica jest taka, że po każdym przypisaniu wartości do zmiennej, jeżeli ograniczenia są spełnione, to wykonywany jest dodatkowy krok. Z dziedziny wszystkich zmiennych uwikłanych w ograniczenia z aktualnie rozpatrywaną zmienną, usuwane są wartości, które powodowałyby złamanie jakiegось ograniczenia. Jeżeli po tym pro-

cesie dziedzina którejkolwiek ze zmiennych jest pusta, to aktualnej zmiennej przypisuje się kolejną wartość z dziedziny.

W znaczący sposób ogranicza to liczbę przeglądanych rozwiązań, ale procedura usuwania wartości z dziedziny zmiennych jest dość złożona, więc nie zawsze przekłada się to na zysk czasowy.

1.3.3 Heurystyki wyboru zmiennych i wartości

Wybór zmiennej do wartościowania

Do wyboru zmiennej do wartościowania została w tej pracy wybrana heurystyka *najmniejszej liczby pozostałych wartości* (*Minimum Remaining Values*). Jako następna zmienna do wartościowania wybierana będzie ta, której pozostało najmniej dostępnych wartości w dziedzinie. Heurystyka ta jest naturalnym uzupełnieniem algorytmu sprawdzania w przód. W przypadku algorytmu przeszukiwania przyrostowego z powrotami generuje ona spory narzut obliczeniowy, ponieważ wymaga wykonania obliczeń, których ten algorytm normalnie nie wykonuje.

Do rozwiązywania remisów (sytuacji gdy zmienne mają taką samą liczbę dostępnych wartości) zostanie użyta heurystyka wybierająca zmienną uwikłaną w najwięcej ograniczeń. Para ta będzie w dalszej części pracy nazywana w skrócie heurystyką *MRV*.

Niestatyczny wybór zmiennych do wartościowania potencjalnie oferuje duży zysk, ponieważ pozwala ustawić zmienne w kolejności generującej zdecydowanie mniejsze drzewo poszukiwań.

Wybór wartości dla zmiennej

Do wyboru następnej wartości dla zmiennej zostanie użyta heurystyka *najmniej ograniczającej wartości*. Wybiera ona dla zmiennej taką wartość, która eliminuje najmniej dopuszczalnych wartości z dziedzin powiązanych zmiennych.

Sformułowanie celu jako znalezienia wszystkich rozwiązań problemu, sprawia że heurystyka ta nie powinna mieć wpływu na liczbę sprawdzonych rozwiązań, ponieważ i tak należy sprawdzić wszystkie dopuszczalne wartości dla każdej ze zmiennych. Rzeczą którą można w tym przypadku badać jest czas i liczba rozwiązań sprawdzonych do momentu znalezienia pierwszego rozwiązania.

1.3.4 Usprawnienia wprowadzone do ograniczeń

Aby zmniejszyć liczbę sprawdzanych rozwiązań i potencjalnie przyspieszyć działanie algorytmów, do sposobu rozpatrywania spełnienia ograniczeń zostały wprowadzone niewielkie zmiany, wynikające wprost z ich logiki.

Futoshiki

Ograniczenie mówiące o nierówności wartości dwóch zmiennych można by rozpatrywać dopiero w momencie przypisania obu z nich. Znając jednak pełną dziedzinę zmiennych, można wyciągnąć następujące wnioski:

- jeżeli do "mniejszej" zmiennej została przypisana największa wartość z dziedziny, to ograniczenie nie jest spełnione,
- jeżeli do "większej" zmiennej została przypisana najmniejsza wartość z dziedziny, to ograniczenie nie jest spełnione.

Skyscraper

Ograniczenie "widzialności" wieżowców można rozszerzyć w następujący sposób:

- jeżeli do pierwszej zmiennej widzianej, z którejś ze stron została przypisana największa wartość z dziedziny, a z tej strony powinno być widać więcej niż 1 wieżowiec, to ograniczenie nie jest spełnione,
- jeżeli do pierwszej zmiennej widzianej, z którejś ze stron została przypisana najmniejsza wartość z dziedziny, a z tej strony powinno być widać tylko 1 wieżowiec, to ograniczenie nie jest spełnione.

2 Badania

2.1 Czas działania i liczba sprawdzanych rozwiązań

2.1.1 Futoshiki

2.1.2 Skyscraper

2.2 Wpływ heurystyki wyboru zmiennej

2.3 Wpływ heurystyki wyboru wartości