

Sztuczna Inteligencja i Inżynieria Wiedzy

Laboratorium

Zadanie 2

Problemy spełniania ograniczeń

Opracowanie: Jan Jakubik, Artur Zawisza

Spis treści

1. Etap 1 – sformułowanie problemu, przeszukiwanie z nawrotami	3
1.1. Problemy spełniania ograniczeń – definicja	3
1.2. Sudoku	3
1.3. Jolka	4
1.4. Zmienne, dziedziny, ograniczenia	5
1.5. Przeszukiwanie z nawrotami	6
1.6. Heurystyki	7
1.7. Ocena etapu	8
2. Etap 2 – Metoda sprawdzania w przód	9
2.1. Metoda sprawdzania w przód	9
2.2. Ocena etapu	10
3. Etap 3 – badania	11
3.1. Zbiory danych	11
3.2. Ocena etapu	11
Materiały pomocnicze	12

1. Etap 1 – sformułowanie problemu, przeszukiwanie z nawrotami

Celem ćwiczenia jest zapoznanie studentów z problemami spełniania ograniczeń – ich formułowaniem oraz metodami rozwiązywania – na przykładzie wybranych łamigłówek logicznych.

1.1. Problemy spełniania ograniczeń – definicja

Problemy spełniania ograniczeń (ang. Constraint Solving Problems, Constraint Satisfaction Problems, CSP) są klasą problemów składających się z trzech podstawowych elementów:

- Zmiennych,
- Dziedzin,
- Ograniczeń.

Celem w problemach CSP jest znalezienie takiego **przypisania wartości** z dziedzin **do zmiennych**, aby wszystkie **ograniczenia były spełnione**.

Dokładny opis elementów CSP zawarto po przedstawieniu łamigłówek (sekcja 1.4).

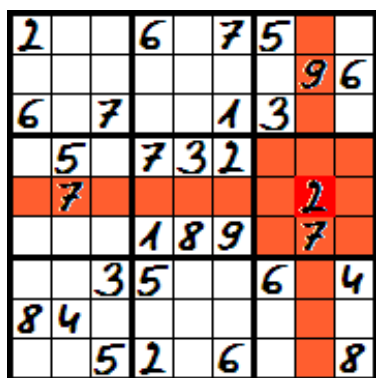
1.2. Sudoku

Sudoku jest łamigłówką liczbową, w której celem jest wypełnienie wszystkich pustych pól na planszy 9x9 cyframi od 0 do 9 tak, aby w każdym wierszu, każdej kolumnie oraz każdym większym kwadracie 3x3 dana cyfra wystąpiła dokładnie raz. Przykładową planszę Sudoku pokazano na rysunku 1.

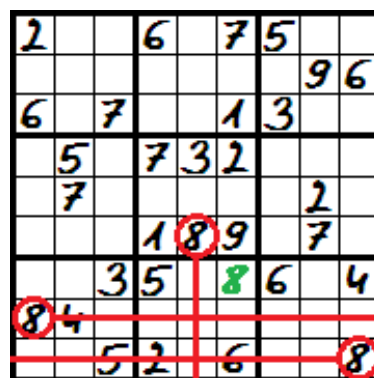
2			6		7	5		
							9	6
6		7			1	3		
	5		7	3	2			
	7						2	
			1	8	9		7	
		3	5			6		4
8	4							
		5	2		6			8

Rysunek 1 Przykładowa plansza Sudoku,
źródło: <https://pl.wikipedia.org/wiki/Sudoku>

Na rysunku 2 pomarańczowym kolorem wskazano obszar, w którym nie można wpisać cyfry 2 ze względu na występowanie tej cyfry (oznaczonej kolorem czerwonym) w zasięgu. Rysunek 3 przedstawia jak, na podstawie występowania cyfr, ustalono pozycję cyfry 8 w danym polu.



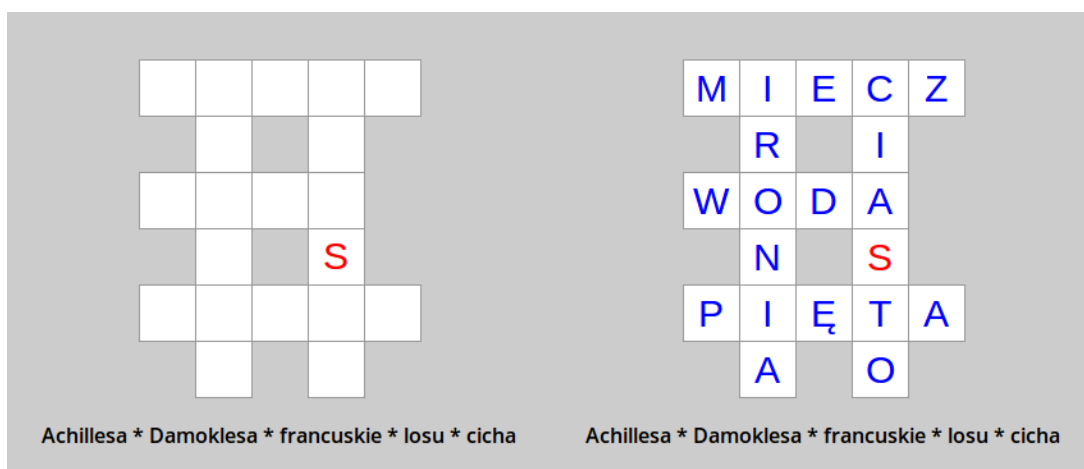
Rysunek 2 Obszar wykluczenia cyfry 2



Rysunek 3 Przypisanie cyfry 8 do pola na podstawie wykluczeń

1.3. Jolka

Jolka jest rodzajem krzyżówki, polegającej na dopasowaniu wyrazów do diagramu, bez znajomości ich dokładnego położenia. Podobnie jak w klasycznych krzyżówkach, słowa w jolkach są ukryte za opisami, ale dany opis nie jest powiązany z konkretnymi polami. Rysunek 4 przedstawia przykładową jolkę.



Rysunek 4 Przykładowa jolka wraz z rozwiązaniem,
źródło: [https://pl.wikipedia.org/wiki/Jolka_\(krzy%C5%BC%C3%B3wka\)](https://pl.wikipedia.org/wiki/Jolka_(krzy%C5%BC%C3%B3wka))

W ramach zadania krzyżówka zostanie ograniczona do samego dopasowywania słów do diagramu.

1.4. Zmienne, dziedziny, ograniczenia

Jak zostało powiedziane w sekcji 1.1 problemy CSP składają się z trzech elementów – zmiennych, dziedzin i ograniczeń.

Zmienna V_i jest **nazwą (etykietą)**, do której przypisywane są **wartości**. Każdej zmiennej odpowiada **dziedzina** D_i będąca **zbiorem** wszystkich możliwych **wartości** danej zmiennej.

Przykładowo, w Sudoku za zmienne można przyjąć wszystkie pola diagramu. W takim przypadku, dziedziną każdej ze zmiennych $V_{i,j}$ (i-ty wiersz, j-ta kolumna) będzie zbiór cyfr $D_{i,j} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, o ile dane pole jest puste. Pola z zadanymi cyframi będą miały dziedziny jednoelementowe. Dla diagramu z rysunku 1 $D_{1,1} = \{2\}$ (pole 1, 1 ma przypisaną wartość 2), a $D_{1,2} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (pole 1, 2 jest puste).

Warto zwrócić uwagę, że definicja zmiennych i dziedzin nie musi być jednoznaczna. Dla Sudoku za zmienne można równie dobrze przyjąć kolejne wiersze (lub kolumny). Wtedy dziedziny będą zawierać wszystkie permutacje ciągu $(1, 2, 3, 4, 5, 6, 7, 8, 9)$. Decyzja o tym, która reprezentacja będzie przyjęta zależy od rozwiązywanego problemu.

Wybór zmiennych i dziedzin wpływa na ostatni element – ograniczenia.

Ograniczenie P_i jest **predykatem** tzn. funkcją zwracającą **wartości logiczne** prawda lub fałsz $P_i: X_i \rightarrow \{true, false\}$, gdzie X_i jest iloczynem kartezjańskim dziedzin **wybranych** zmiennych. Innymi słowy, **ograniczenie określa**, czy **wartościowanie wybranych** zmiennych jest **akceptowalne**, czy też nie. Ograniczenia dotyczące jednej zmiennej nazywamy **unarnymi**, dwóch – **binarnymi**, a n zmiennych **n-arnymi**. Postać i liczba ograniczeń zależy od wyboru zmiennych i ich dziedzin.

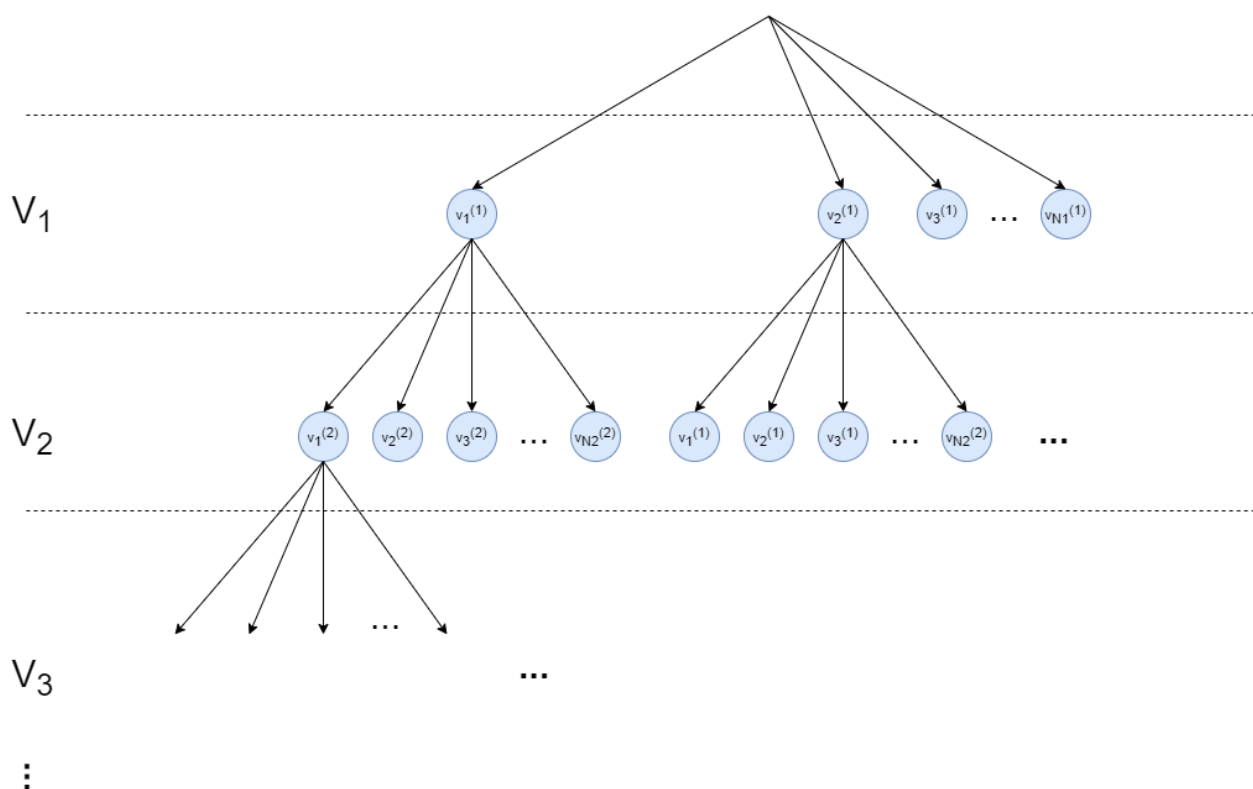
Przykładowo, przyjmując reprezentację pole – zmienna, jedno z ograniczeń przyjmie postać: $P(v_{1,2}, v_{2,3}) = (v_{1,2} \neq v_{2,3})$, wskazujące na to, że wartości $v_{1,2}$ oraz $v_{2,3}$ leżą w tym samym dużym kwadracie, a więc nie mogą być identyczne. Warto zwrócić uwagę, że przykładowe ograniczenie dotyczy jedynie dwóch zmiennych (ograniczenie binarne), a nie wszystkich występujących w problemie. Dla Sudoku ograniczeń będzie znacznie więcej i opisywać będą związki między wartościami w wierszach, kolumnach i dużych kwadratach.

Wartości początkowe zmiennych można reprezentować dwojako – poprzez dziedziny, lub poprzez ograniczenia. Przykładowo, dla diagramu z rysunku 1, wartość 2 pola 1, 1 można zapisać albo poprzez dziedzinę: $D_{1,1} = \{2\}$, albo poprzez ograniczenie: $P(v_{1,1}) = (v_{1,1} = 2)$ i przyjęcie domyślnej dziedziny ($D_{1,1} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$). Obydwa podejścia są równoważne. W przypadku pierwszego zaletą jest to, że część dziedzin będzie mała oraz nie wprowadzimy dodatkowych ograniczeń. W przypadku drugiego zaletą jest to, że wszystkie dziedziny będą identyczne, kosztem wprowadzenia dodatkowych ograniczeń. Wybór jednego, bądź drugiego może mieć wpływ na efektywność działania metod rozwiązywania problemów CSP.

1.5. Przeszukiwanie z nawrotami

Jak rozwiązywać problemy CSP? Najprostsze, naiwne podejście może polegać na losowaniu wartości przypisywanych poszczególnym zmiennym i sprawdzaniu, czy wszystkie ograniczenia są spełnione. Oczywiście jest, że taki schemat postępowania nie prowadzi donikąd, gdyż w przypadku problemów o restrykcyjnych ograniczeniach szansa na wylosowanie poprawnego rozwiązania jest praktycznie zerowa. Potrzebne jest podejście systematyczne.

Algorytm przeszukiwania z nawrotami (ang. backtracking algorithm) jest systematyczną metodą rozwiązywania problemów CSP polegającą na stopniowym przypisywaniu wartości z dziedzin do kolejnych zmiennych i sprawdzaniu, czy są one zgodne z ograniczeniami. W przypadku złamania któregoś z ograniczeń należy „wycofać” decyzję o przypisaniu wartości i spróbować dokonać innego przypisania. Wszystkie możliwe przypisania (wartościowania) zmiennych można przedstawić w postaci drzewa. Każdy z poziomów drzewa poszukiwań odpowiada jednej ze zmiennych, a każdy węzeł na danym poziomie zawiera jedną z wartości z dziedziny tej zmiennej. Zejście w dół drzewa wiąże się z przypisaniem nowej wartości do kolejnej zmiennej. Wycofanie się zawsze wiąże się z przejściem w górę drzewa, czyli z nawrotem, skąd metoda bierze swoją nazwę. Przykład drzewa poszukiwań, dla abstrakcyjnego problemu CSP, przedstawiono na rysunku 5.



Rysunek 5 Przykładowe drzewo przeszukiwań. V_i – i -ta zmienna, $v_j^{(i)}$ – j -ta wartość z D_i

Poniżej zamieszczono pseudokod przeszukiwania z nawrotami.

```
1  Wybierz kolejną zmienną do przypisania
2
3  JEŻELI brak kolejnej zmiennej
4      Znaleziono rozwiązanie
5      KONIEC
6  WPP
7      Wybierz kolejną wartość z dziedziny aktualnej zmiennej
8
9      JEŻELI brak kolejnej wartości
10         Wróć do poprzedniej zmiennej          // Nawrót
11
12         JEŻELI brak poprzedniej zmiennej
13             Brak rozwiązań problemu CSP
14             KONIEC
15         WPP
16         Przejdź do kroku 7
17     WPP
18     Przypisz wybraną wartość do aktualnej zmiennej
19
20     JEŻELI wszystkie ograniczenia są spełnione
21         Przejdź do kroku 1
22     WPP
23     Przejdź do kroku 7          // Nawrót
```

Powyższa wersja metody przeszukiwania z nawrotami kończy działanie przy znalezieniu pierwszego akceptowalnego wartościowania wszystkich zmiennych. Schemat można zmodyfikować w taki sposób, że metoda będzie odnajdywała wszystkie możliwe rozwiązania problemu.

1.6. Heurystyki

Schemat przeszukiwania z nawrotami posiada dwa niedomówienia – nigdzie nie jest sprecyzowane, którą ze zmiennych, ani którą z wartości przyjąć za „kolejną” w krokach 1 oraz 7. W związku z tym wyszczególnić można dwa rodzaje heurystyk – heurystyki wyboru zmiennych oraz heurystyki wyboru wartości.

Przykładowe **heurystyki wyboru zmiennych**:

- W kolejności definicji,
- Losowo,
- Zmienna najbardziej ograniczona (o najmniejszej dziedzinie),
- Inne,

Przykładowe heurystyki wyboru wartości:

- W kolejności definicji,
- Losowo,
- Wartość najmniej ograniczająca (wykluczająca najmniej wartości pozostałych zmiennych),
- Inne.

Należy pamiętać, że obydwa rodzaje heurystyk można wykorzystywać statycznie lub dynamicznie, tzn. decyzja o kolejności może być podjęta albo na początku działania metody, albo być podejmowana w trakcie przechodzenia po przestrzeni rozwiązań.

1.7. Ocena etapu

Za etap można otrzymać **maksymalnie 4 punkty**.

Sformułowanie łamigłówek jako CSP (definicja zmiennych, dziedzin i ograniczeń) wraz z implementacją, wczytywanie danych	1 pkt.
Implementacja metody przeszukiwania z nawrotami	1 pkt.
Implementacja jednej heurystyki wyboru zmiennej i jednej heurystyki wyboru wartości	1 pkt.
Zbadanie działania implementacji dla obu łamigłówek (po jednym pliku)	1 pkt.

Implementacja metody rozwiązywania problemów CSP **musi** odnajdywać **wszystkie** rozwiązania. Powinna też uwzględniać możliwość dodania kolejnych heurystyk w dalszych etapach realizacji ćwiczenia.

W badaniach należy podać:

- Czas do znalezienia pierwszego rozwiązania,
- Liczbę odwiedzonych węzłów drzewa do znalezienia pierwszego rozwiązania,
- Liczbę nawrotów wykonanych do znalezienia pierwszego rozwiązania,
- Całkowity czas działania metody,
- Całkowitą liczbę odwiedzonych węzłów drzewa,
- Całkowitą liczbę nawrotów,
- Liczbę rozwiązań.

Trzeba też być przygotowanym do wyświetlenia znalezionych rozwiązań. Sugeruje się wydruk na konsolę lub zapis wyników do pliku.

Ze względu na ograniczony czas na zajęciach, sugerowane jest, aby mniej istotne fragmenty programu takie jak np. ładowanie danych, czy zapis wyników, były zaimplementowane w ramach przygotowania do zajęć przed laboratorium.

2. Etap 2 – Metoda sprawdzania w przód

Przeszukiwanie z nawrotami będące tematem poprzedniego etapu daje pewność znalezienia rozwiązania, ale nie jest przy tym wolna od wad.

Pierwszą główną wadą jest tendencja metody do **przeszukiwania dużych obszarów przestrzeni rozwiązań**. Nie stanowi to oczywiście dużego kłopotu dla małych problemów, jednak wraz ze **wzrostem liczby zmiennych** rozmiar przestrzeni poszukiwań ma tendencję do wzrostu **wykładniczego** (dlaczego?).

Drugą główną wadą metody jest „**lokalne wykorzystanie ograniczeń**”. Ograniczenia wykorzystywane są do sprawdzania poprawności **TYLKO** dla **zmiennych z przypisanymi wartościami**. Może się jednak zdarzyć sytuacja, że przypisanie danej wartości do zmiennej spowoduje złamanie ograniczenia z inną zmienną analizowaną znacznie później, niezależnie od wartości jaka zostanie przypisana do tej drugiej zmiennej. Ten problem po części rozwiązuje odpowiedni dobór heurystyki wyboru zmiennych, ale dla większości problemów nie da się dobrać takiej kolejności zmiennych, aby „dalekie zależności” między zmiennymi nie występowały.

2.1. Metoda sprawdzania w przód

Obydwa problemy przeszukiwania z nawrotami rozwiązuje „**globalne wykorzystanie ograniczeń**”. Jeżeli ograniczenie wiąże N zmiennych, a znane są wartości $N-1$ zmiennych, to można go użyć do **odfiltrowania wartości z dziedziny** ostatniej zmiennej, które **łamają ograniczenie**. Dzięki temu możemy wpłynąć na zmienną, do której przypisanie nastąpiłoby bardzo późno.

Przykładowo, dla zmiennych V_1 i V_2 o dziedzinach $D_1 = D_2 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ oraz ograniczenia $P(v_1, v_2) = (v_1 > v_2)$, jeżeli do zmiennej V_1 przypiszemy wartość $v_1 = 4$ to możemy wykorzystać ograniczenie $P(v_1 = 4, v_2) = (4 > v_2)$, aby odfiltrować wartości z D_2 otrzymując $D_2' = \{1, 2, 3\}$.

Dodatkowo warto zwrócić uwagę, że jeśli któraś z dziedzin po przefiltrowaniu stanie się **pusta**, to wiemy, że w danym poddrzewie nie znajdziemy rozwiązania, gdyż do tej zmiennej nie da się przypisać żadnej wartości. Z tej przyczyny można od razu takie poddrzewo pominąć.

Taki schemat postępowania stanowi podstawę **metody sprawdzania w przód** (ang. forward checking, FC), której pseudokod zamieszczono poniżej.

- 1 Wybierz kolejną zmienną do przypisania
- 2
- 3 JEŻELI brak kolejnej zmiennej
- 4 Znaleziono rozwiązanie
- 5 KONIEC
- 6 WPP
- 7 Wybierz kolejną wartość z dziedziny aktualnej zmiennej
- 8
- 9 JEŻELI brak kolejnej wartości

```

10     Wróć do poprzedniej zmiennej          // Nawrót
11
12     JEŻELI brak poprzedniej zmiennej
13         Brak rozwiązań problemu CSP
14     KONIEC
15     WPP
16     Przejdź do kroku 7
17 WPP
18     Przypisz wybraną wartość do aktualnej zmiennej
19     Wykorzystując ograniczenia odfiltruj dziedziny zmiennych bez wartości
20
21     JEŻELI któraś dziedzina jest pusta
22         Przejdź do kroku 7                // Nawrót
23     WPP
24     Przejdź do kroku 1

```

Powyższa wersja metody przeszukiwania z nawrotami kończy działanie przy znalezieniu pierwszego akceptowalnego wartościowania wszystkich zmiennych. Schemat można zmodyfikować w taki sposób, że metoda będzie odnajdywała wszystkie możliwe rozwiązania problemu.

Podobnie jak w przypadku metody przeszukiwania z nawrotami, metoda sprawdzania w przód nie precyzuje ani kolejności wyboru zmiennych, ani kolejności wyboru wartości - wymaga zdefiniowania heurystyk (por. sekcja 1.6).

2.2. Ocena etapu

Za etap można otrzymać **maksymalnie 3 punkty**.

Implementacja metody sprawdzania w przód	1 pkt.
Implementacja jednej heurystyki wyboru zmiennej i jednej heurystyki wyboru wartości	1 pkt.
Zbadanie działania implementacji dla obu łamigłówek (po jednym pliku)	1 pkt.

Implementacja metody rozwiązywania problemów CSP **musi** odnajdywać **wszystkie** rozwiązania.

Heurystyki implementowane w tym etapie **muszą** być **inne** niż te przygotowane w etapie poprzednim.

Wymagania odnośnie badań są takie same jak w poprzednim etapie (por. sekcja 1.7).

3. Etap 3 – badania

Celem etapu trzeciego jest **porównanie** obu metod rozwiązywania problemów CSP na przykładowych łamigłówkach.

3.1. Zbiory danych

Wraz z instrukcją dostarczone zostało archiwum ZIP zawierające pliki z danymi opisującymi instancje problemów.

W pliku Sudoku.csv zebrano 44 instancje łamigłówki **Sudoku** o różnych poziomach złożoności. Instancje o poziomie **złożoności 0** służą jedynie do **testowania kodu** i nie są objęte sprawozdaniem. Dla tych instancji podano rozwiązania.

Student ma wykonać badania na następującej liczbie plików:

- 1 plik z grupy problemów **łatwych** (**trudność 1 i 2**),
- 1 plik z grupy problemów **średnich** (**trudność 3 - 5**),
- 1 plik z grupy problemów **trudnych** (**trudność 6 i 7**),
- 1 plik z grupy problemów **mieszanych1** (**trudność 8**),
- 1 plik z grupy problemów **mieszanych2** (**trudność 9**).

Instancje łamigłówki **Jolka** zawarto w folderze Jolka (**5 instancji**). Każda instancja opisana jest dwoma plikami: puzzleX oraz wordsX. Pierwszy plik zawiera opis planszy, a drugi listę słów. Instancja **puzzle0** służy do **testowania kodu** i dla niej podano rozwiązanie – plik filled0.

Student ma przeprowadzić badania na **wszystkich** instancjach Jolki.

3.2. Ocena etapu

Za etap można otrzymać **maksymalnie 3 punkty**.

Badanie porównawcze metod rozwiązywania problemów CSP dla Sudoku	1 pkt.
Badanie porównawcze metod rozwiązywania problemów CSP dla Jolki	1 pkt.
Badanie wpływu doboru heurystyk na działanie metod	1 pkt.

Podczas badań należy podać te same **parametry wykonania**, co w **sekcji 1.7**. Wyniki należy uśrednić z wielu uruchomień. Jeżeli prowadzący nie powie inaczej, to proszę przyjąć **10** uruchomień.

Badanie heurystyk można ograniczyć do **trzech**, spośród **wskazanych pięciu**, instancji **Sudoku**.

Ze względu na potencjalną czasochłonność obliczeń sugerowane jest wykonanie większości badań w ramach przygotowania do zajęć.

Materiały pomocnicze

- 1) Notatki z wykładu,
- 2) Kasprzak Witold, „Inteligentne techniki obliczeniowe (studia magisterskie)”,
Politechnika Warszawska, rozdział 7:
<http://www.ia.pw.edu.pl/~wkasprza/PAP/ITO-2015.pdf>
- 3) Strona z łamigłówką Sudoku:
<https://sudoku.com/pl>
- 4) Strona z łamigłówką Jolka:
<https://szarada.net/rodzaj-zadania/jolka/> (sekcja „Kolekcje”)