Algorytmy numeryczne - zadanie 2

Oskar Gawryszewski

14 kwietnia 2024

1 Wstęp

Naszym celem jest analiza losowych spacerów w parku, gdzie alejki krzyżują się w punktach $v_1, v_2, ..., v_n$. Wędrowiec, poruszając się, wybiera losowo kolejną drogę na skrzyżowaniu. Interesuje nas prawdopodobieństwo, że wędrowiec bezpiecznie wróci do domu, w przypadku gdy istnieje jedno lub więcej wyjść. Zastosujemy algorytmy Monte Carlo do symulacji spacerów oraz algorytmy numeryczne do budowy i rozwiązania układów równań, aby zweryfikować poprawność rozwiązań i weryfikować hipotezy dotyczące dokładności i wydajności algorytmów.

2 Opis problemu

Celem tego eksperymentu jest symulacja ruchu wędrowca w parku z wykorzystaniem losowo wygenerowanego grafu reprezentującego ścieżki w parku. Park składa się z wielu alejek, które krzyżują się w różnych punktach. Każdy punkt krzyżowania jest traktowany jako skrzyżowanie, a początek i koniec każdej alejki są uważane za skrzyżowania.

Na wygenerowanym grafie wierzchołki oznaczone kolorem zielonym reprezentują wyjścia, co odpowiada sukcesowi w dotarciu do celu. Wierzchołki czerwone symbolizują punkty, które należy unikać - wyjście z parku, co oznacza niepowodzenie. Natomiast wierzchołki żółte oznaczają punkty startowe dla wędrowca.

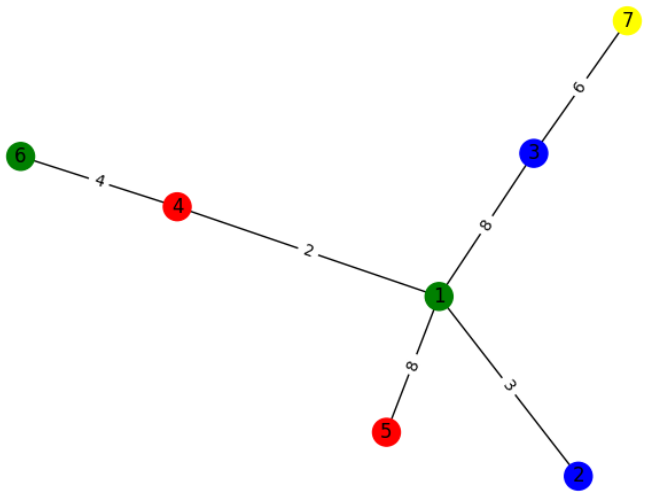

Figure 1: Losowo wygenerowany graf obrazujący problem

Każda ścieżka na grafie będzie miała przypisaną wagę, reprezentującą odległość lub trudność przejścia. Wagi te symbolizują długość alejki wzdłuż której może poruszać się wędrowca. Ruch wędrowca odbywa się zgodnie z określonymi zasadami, polegającymi na wyborze losowego sąsiedniego skrzyżowania.

Symulacja ruchu wędrowca ma na celu zbadanie prawdopodobieństwa bezpiecznego powrotu do domu. Zastosowanie metody Monte Carlo pozwoli na przeprowadzenie wielokrotnych prób symulacji, co umożliwi ocenę efektywności wybranej strategii poruszania się w parku.

3 Opis rozwiązywania problemu

3.1 Przygotowanie danych

Nasz pierwszy krok to przygotowanie danych do analizy. Tworzymy graf, który będzie reprezentował strukturę alejek i skrzyżowań w parku. Przykładowa struktura danych grafu wygląda następująco:

```
data1 = {
    "alleys": 4,
    "intersections": 5,
    "osk": {1: [1]},
    "exit": {1: [2]},
    "start": {1: [3]},
    "routes": [[1, 2, 2], [1, 3, 2], [1, 4, 2], [1, 5, 2]],
```

```
    "trash_cans": {2: [4, 5]},
}
```

Następnie na podstawie danych dotyczących tras (routes), rozszerzamy graf o długość alejek, tworząc mniejsze ścieżki pomiędzy skrzyżowaniami. Przekształcamy dane tak, że każde skrzyżowanie i jego połączenia z innymi skrzyżowaniami są reprezentowane w postaci grafu. Dla przykładowego grafu będzie wyglądać to następująco:

```
{
1: [6, 7, 8, 9], 2: [6], 3: [7], 4: [8],
5: [9], 6: [1, 2], 7: [1, 3], 8: [1, 4], 9: [1, 5]
}
```

3.2 Budowa i rozwiązywanie układów równań

Kolejnym krokiem jest budowa i rozwiązanie układu równań na podstawie grafu. Na początku tworzymy macierz i wektor na podstawie przekształconego grafu. Macierz reprezentuje prawdopodobieństwo ruchu pomiędzy skrzyżowaniami, gdzie 1 reprezentuje pozycję, w której znajduje się wędrowiec, a pozostałe wartości to prawdopodobieństwa ruchu w daną stronę. Wektor reprezentuje miejsca wyjścia (exit), gdzie wartość 1 oznacza, że wędrowiec jest na skrzyżowaniu wyjściowym.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1.0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1.0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1.0 \\ -0.5 & -0.5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.5 & 0 & -0.5 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.5 & 0 & 0 & -0.5 & 0 & 0 & 0 & 1 & 0 \\ -0.5 & 0 & 0 & 0 & -0.5 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

3.3 Budowa i rozwiązywanie układów równań używając wybranych algorytmów

Zaczynamy od rozwiązania układu równań stosując **algorytm eliminacji Gaussa bez wyboru elementu podstawowego**. Metoda eliminacji Gaussa bez wyboru elementu podstawowego polega na eliminacji niewiadomych w kolejnych równaniach, aż do uzyskania trójkątnej postaci macierzy współczynników. Wyniki wyglądają następująco:

$$\text{Solution 1:}[0.0, \mathbf{1.0}, 0.0, 0.0, 0.0, \mathbf{0.5}, 0.0, 0.0, 0.0]$$

Następnie robimy to samo, tym razem stosując **algorytm Gaussa z częściowym wyborem elementu podstawowego**. Metoda eliminacji Gaussa z częściowym wyborem elementu podstawowego polega na wyborze największego elementu w kolumnie jako elementu podstawowego, co zapewnia większą stabilność numeryczną. Wyniki wyglądają następująco:

$$\text{Solution pivotal 1:}[0.0, \mathbf{1.0}, 0.0, 0.0, 0.0, \mathbf{0.5}, 0.0, 0.0, 0.0]$$

Analogicznie robimy to samo, tym razem stosując **metodę iteracyjną Gaussa-Seidela**. Metoda iteracyjna Gaussa-Seidela polega na iteracyjnym rozwiązaniu układu równań, gdzie każde równanie jest rozwiązywane niezależnie, a następnie aktualizowane są wartości zmiennych. Wyniki wyglądają następująco:

$$\text{Solution seidal 1:}[0.0, \mathbf{1.0}, 0.0, 0.0, 0.0, \mathbf{0.5}, 0.0, 0.0, 0.0]$$

Oznacza to, że w prawdopodobieństwo wyjścia jest równe 1, dla startu na wierzchołku 2 oraz 0.5, dla startu na wierzchołku 6 (powstałego poprzez rozszerzenie grafu o tworzenie mniejszych ścieżek między skrzyżowaniami na podstawie długości alejek).

3.4 Weryfikacja rozwiązań metodą Monte Carlo

Ostatnim krokiem jest weryfikacja rozwiązań za pomocą metody Monte Carlo. Testujemy prawdopodobieństwo wyjścia dla różnych ilości próbek $n = 1000$ z różnych miejsc startowych. W wyniku otrzymujemy sukcesywnie wyniki, które pokazują, jakie prawdopodobieństwo wyjścia zostało osiągnięte dla każdego miejsca startowego i liczby próbek. Wyniki zobrazuje na histogramie:

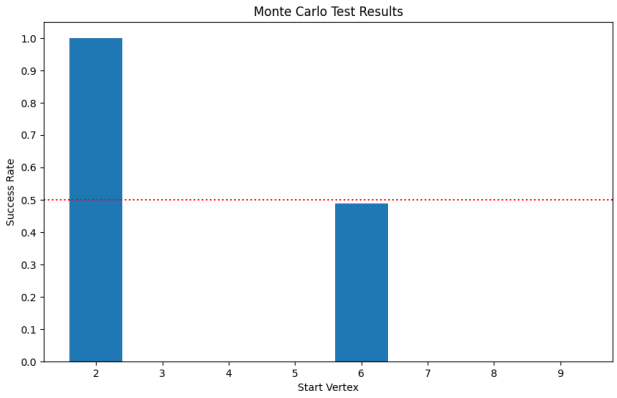

Figure 2: Wyniki testu Monte Carlo

Jak widać test Monte Carlo poprawnie weryfikuje nasze wyniki, dla startu na wierzchołku 2 otrzymujemy prawdopodobieństwo równe 1, a dla startu na wierzchołku 6 otrzymujemy prawdopodobieństwo zbliżone do 0.5.

4 Hipotezy badawcze

A1 Algorytm eliminacji Gaussa bez wyboru elementu podstawowego.

A2 Algorytm eliminacji Gaussa z częściowym wyborem elementu podstawowego.

A3 Metoda iteracyjna Gaussa-Seidela.

4.1 Algorytm A2 zwykle daje dokładniejsze wyniki niż A1. Różnica w dokładności rośnie wraz z rozmiarem macierzy i liczbą niezerowych współczynników

Nasz eksperyment polegał na porównaniu dokładności dwóch różnych algorytmów rozwiązywania układów równań: eliminacji Gaussa bez wyboru elementu podstawowego oraz eliminacji Gaussa z częściowym wyborem elementu podstawowego. Eksperyment został przeprowadzono na losowo wygenerowanych grafach o następujących danych:

Intersections	Alleys	Number of OSK	Number of Exits	Max Alley Length
10	20	2	2	5
20	40	4	4	10
30	80	6	6	15
40	160	8	8	20

Table 1: Dane do testu H1

Na podstawie tego grafu, tworzyliśmy macierz i wektor układu równań. Algorytm eliminacji Gaussa bez wyboru elementu podstawowego oraz algorytm eliminacji Gaussa z częściowym wyborem elementu podstawowego były używane do rozwiązania tego układu równań.

Po rozwiązaniu układu równań, obliczaliśmy błąd rozwiązania dla obu algorytmów. Błąd ten był obliczany jako norma różnicy między iloczynem macierzy a wektorem oraz wektorem prawych stron. Wyniki były zapisywane i analizowane w zależności od rozmiaru macierzy.

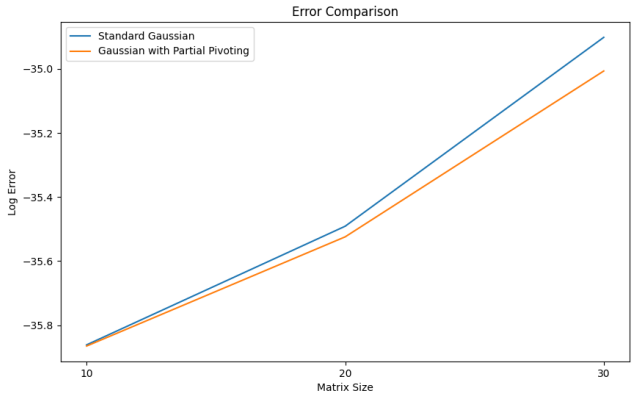

Figure 3: Porównanie błędów

Wykres ten przedstawia jak zmieniała się logarytmiczna wartość błędu, w zależności od rozmiaru danych, w zależności od użytego algorytmu.

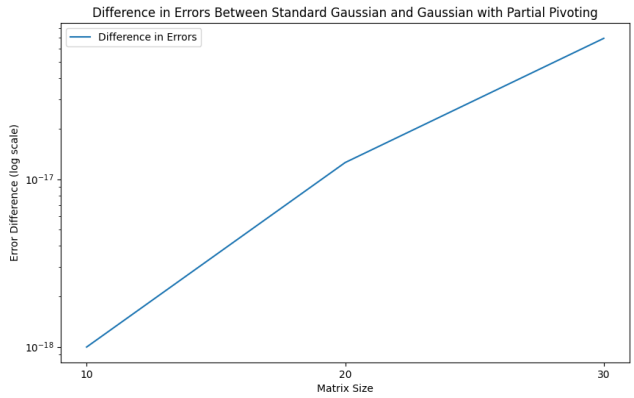

Figure 4: Porównanie błędów

Ten wykres przedstawia jak rosła różnica błędów, między tymi algorytmiami w zależności od rozmiaru macierzy.

Jak widać algorytm **A2** daje dokładniejsze wyniki niż algorytm **A1**.

4.2 Algorytm A3 działa dla postawionego zadania (jeśli nie działa, tzn. proces nie zawsze jest zbieżny do rozwiązania, to wskaż przykłady gdy jest rozbieżny)

Badając zbieżność algorytmu iteracyjnego w rozwiązywaniu układów równań liniowych, przeprowadziłem serię testów na losowych danych. Wykorzystując funkcję generującą dane losowe, przetestowano algorytm dla różnych zestawów parametrów.

$$\begin{gathered}\text{liczba skrzyżowań} = 20 * i, \\ \text{liczba alejek} = 40 * i, \\ \text{liczba osk} = 4 * i, \\ \text{liczba wyjść} = 4 * i \\ \text{maksymalna długość alejki} = 10 * i\end{gathered}$$

gdzie i przyjmuje wartości od 1 do 100. Ta formuła pozwoliła na wszechstronne zbadanie zbieżności algorytmu w zależności od różnych konfiguracji losowych grafów, co umożliwiło lepsze zrozumienie jego zachowania i działania w różnych warunkach.

Wyniki testów wykazały, że w każdym przeprowadzonym teście algorytm był zbieżny do rozwiązania układu równań liniowych. Nie odnotowano żadnego przypadku, w którym algorytm nie zbiegał do rozwiązania.

Na podstawie powyższych wyników można wysnuć wniosek, że algorytm iteracyjny jest zbieżny dla badanych przypadków układów równań liniowych.

4.3 Jeśli algorytm A3 jest zbieżny do rozwiązania, to wyniki otrzymujemy istotnie szybciej niż dla A1 i A2

Dla przetestowanych przeze mnie danych, nie udało mi się znaleźć przypadku, kiedy **A3** byłby najskuteczniejszym algorytmem.

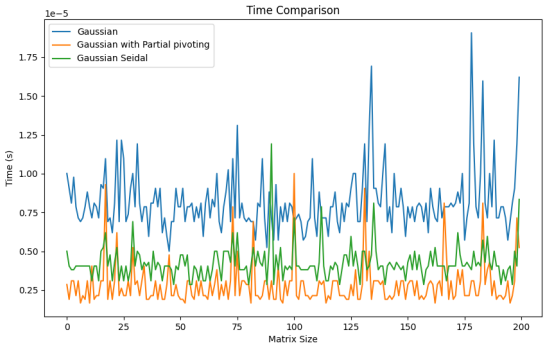

Figure 5: Porównanie błędów

Testowałem tutaj dane:

$$\begin{gathered}\text{liczba skrzyżowań} = 10 * i, \\ \text{liczba alejek} = 20 * i, \\ \text{liczba osk} = 2 * i, \\ \text{liczba wyjść} = 2 * i \\ \text{maksymalna długość alejki} = 5 * i\end{gathered}$$

gdzie i przyjmuje wartości od 200 do 400.

Metoda	Czas (s)
Gauss bez wyboru elementu podstawowego	0.00189948
Gauss z częściowym wyborem elementu podstawowego	0.00062490
Metoda iteracyjna Gaussa-Seidela	0.00093102

W tej tabelce zsumowałem wszystkie wyniki dla danej metody. Jak widać najgorzej wypada metoda Gaussa bez wyboru elementu podstawowego, a najlepiej Gauss z częściowym wyborem elementu podstawowego. Można powiedzieć, że metoda iteracyjna Gaussa-Seidela generuje podobne, a wraz ze wzrostem rozmiaru macierzy różnica również pozostaje na tym samym poziomie. Aby to zweryfikować, przeprowadziłem doświadczenie jeszcze raz, tym razem podstawiając wartości i w zakresie od 200 do 500. Wynikiem jest wykres, pokazujący różnice czasów wykonania się algorytmu. Jak widać, pomimo wzrostu rozmiaru macierzy, oscyluje ona cały czas wokół tych samych wartości.

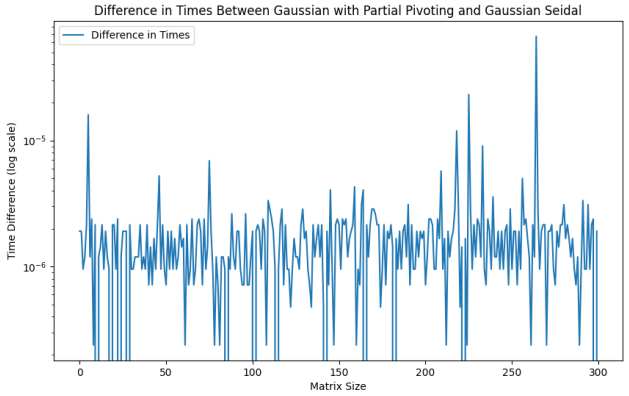

Figure 6: Porównanie błędów