

Algorytmy numeryczne

Zadanie 3

Iwona Jaśtak | Arkadiusz Dąbrowski
Tester programista - grupa 1

09.12.2018

Population Protocols

Majority/Consensus problem

Poniższe sprawozdanie dotyczy analizy problemu przeprowadzenia głosowania większościowego, przedstawionego za pomocą modelu obliczeniowego znanego jako protokoły populacyjne, w którym obliczenia są prowadzone przez poruszających się agentów o bardzo ograniczonych możliwościach. W naszym przypadku agenci przyjmują jeden z trzech określonych stanów (*TAK*, *NIE*, *niezdecydowany*), który ulega zmianie w zależności od reguł konkretnego algorytmu.

W istocie powyższy problem sprowadza się do wyznaczenia prawdopodobieństwa zmiany stanu wszystkich agentów na *TAK* po przejściu całego algorytmu, w zależności od początkowej ilości agentów na *TAK* oraz agentów na *NIE*.

Poprawność implementacji

Aby rozstrzygnąć w/w problem wygenerowany został układ równań przy wykorzystaniu wzoru:

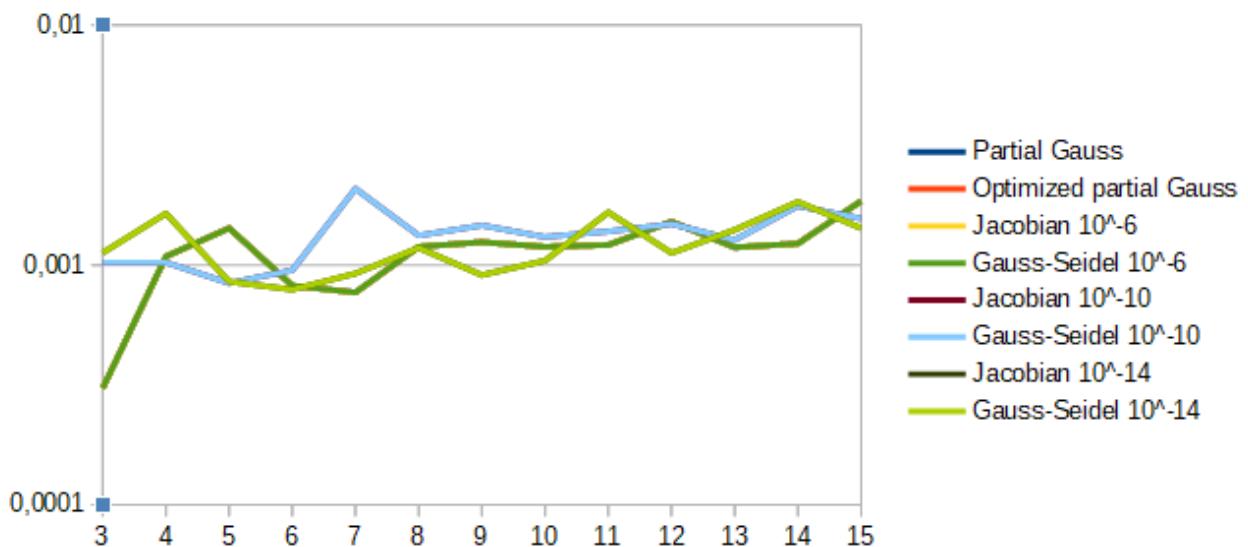
$$P_{X,Y} = \frac{X \cdot U}{n} P_{X+1,Y} + \frac{Y \cdot U}{n} P_{X,Y+1} + \frac{X \cdot Y}{n} P_{X-1,Y-1} + \frac{n - (X \cdot U + Y \cdot U + X \cdot Y)}{n} P_{X,Y}$$

gdzie N, X, Y, U, n oznaczają odpowiednio ilość wszystkich agentów, ilość agentów na tak, ilość agentów na nie, ilość niezdecydowanych agentów ($N - (X + Y)$) oraz ilość połączeń między nimi, wyliczaną z $\binom{N}{2}$. Następnie rozwiązanie takiego układu równań, zapisanego w postaci macierzy kwadratowej, zostało wyznaczone przy użyciu poniższych metod:

- metoda Gaussa z częściowym wyborem elementu podstawowego,
- metoda Gaussa z częściowym wyborem elementu podstawowego zoptymalizowana dla macierzy rzadkich,
- metoda iteracyjna Jacobiego w postaci $x_i^{(k+1)} = \frac{-\sum_{j=1}^n a_{ij}x_j^{(k)} + b_i}{a_{ii}}$,
- metoda iteracyjna Gaussa-Seidela w postaci $x_i^{(k+1)} = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} + b_i}{a_{ii}}$,

natomiast sam wynik tych obliczeń został zweryfikowany metodą Monte-Carlo (wielokrotna symulacja przebiegu procesu). W naszym przypadku wykonano 500000 symulacji. By udowodnić poprawność implementacji, na poniższym wykresie przedstawiono maksymalne błędy bezwzględne między kolejnymi metodami a wynikiem uzyskanym z metody Monte Carlo. Błędy uzyskane w wynikach osiągają dokładność rzędu około 10^{-3} , co potwierdza skuteczność zaimplementowanych algorytmów.

Program implementujący w/w problem, metody, symulację oraz wszelkie testy został napisany w języku *Java* przy wykorzystaniu typu podwójnej precyzji: *double*.



Stosowanie metod iteracyjnych

Zgodnie z wynikami przedstawionymi na wykresie powyżej oraz z wnioskami przedstawionymi w dalszej części sprawozdania możemy wprost stwierdzić, że każda z testowanych metod iteracyjnych jest jak najbardziej odpowiednia dla postawionego problemu. Można to zauważyć poprzez nakładanie się kolejnych wykresów na siebie. Dodatkowo sposób określenia końca iteracji, tj. $\|X^k - X^{k-1}\| < p$ daje nam możliwość podania własnej, żądanej dokładności p , dzięki czemu mamy kontrolę nad błędami. Jednakże istnieje również minus tych metod. Jest to ilość iteracji, które trzeba wykonać aby uzyskać wynik dla wyższych żądanych dokładności, gdzie są to całkiem duże liczby.

Porównanie wyników

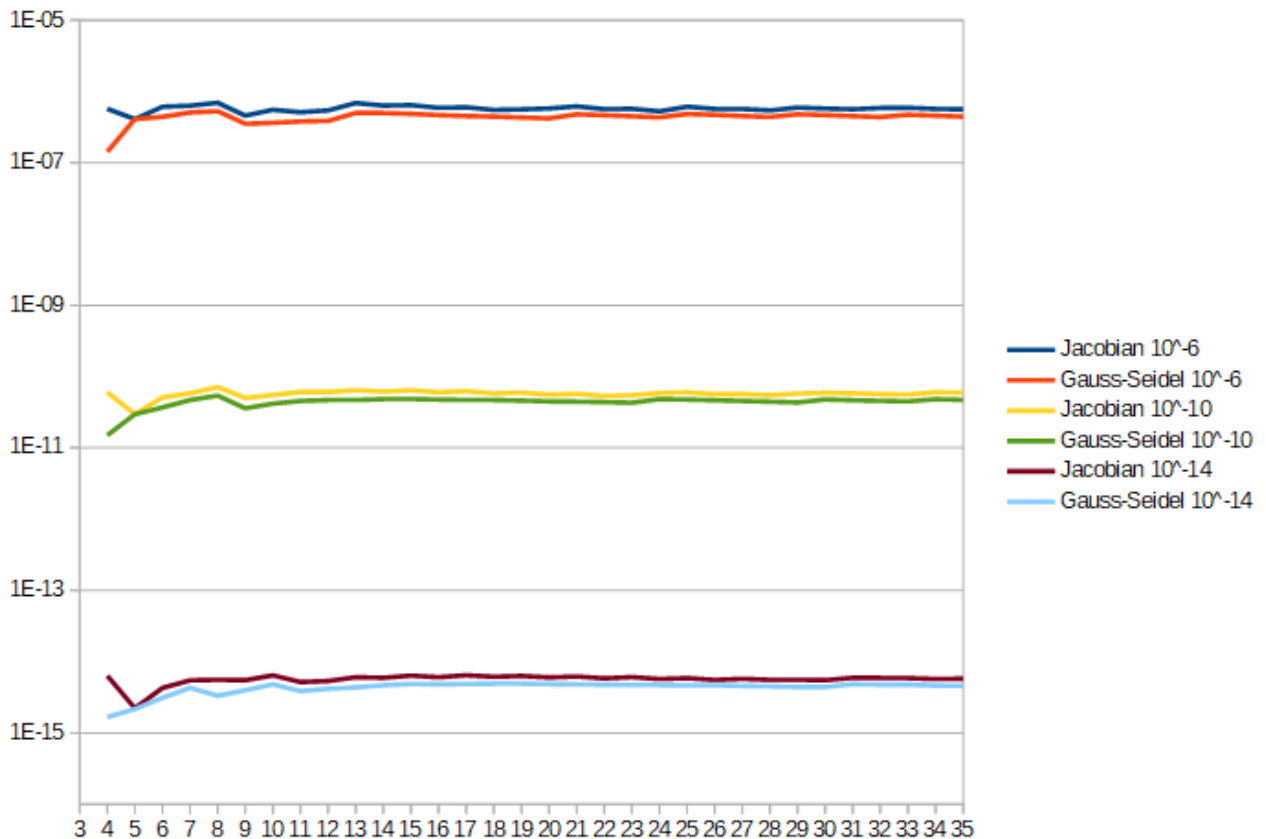
By porównać wyniki wszystkich metod, przeprowadzone zostały testy dla liczby agentów z przedziału $[3, 35]$. Dla danych wartości wyliczone zostały równania prawdopodobieństw z których stworzyliśmy macierz A oraz wektor B . Wektor X wyliczony zostaje odpowiednią metodą. Następnie wyliczamy B' ze wzoru $A \cdot X = B'$ oraz następnie wyliczamy maksymalny błąd bezwzględny między B , a B' .

- metoda Gaussa z częściowym wyborem elementu podstawowego w dwóch wariantach: z dodatkową optymalizacją dla macierzy rzadkich i bez

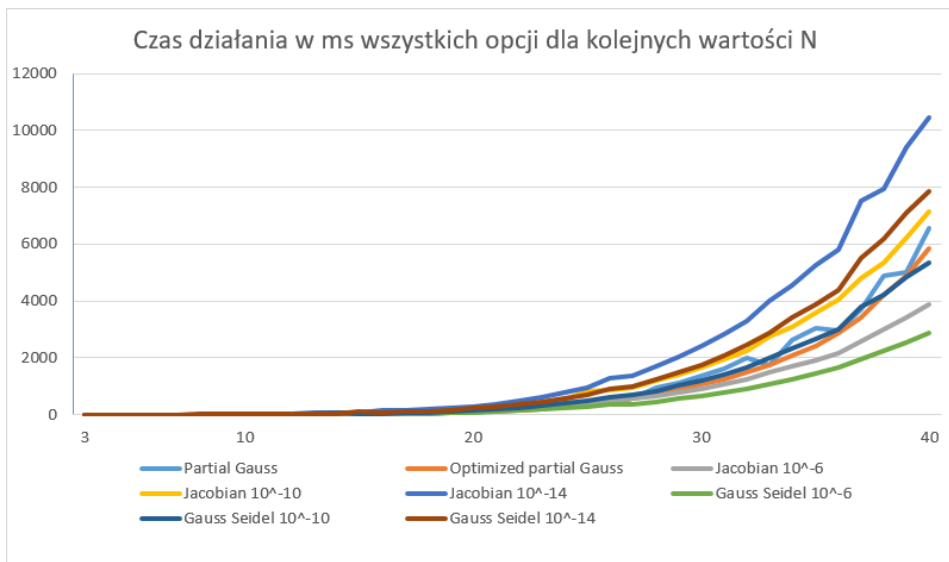
```
Number of agents: 5 Partial Gauss 5.551115123125783E-17 Optimized Partial Gauss 5.551115123125783E-17 Difference 0.0
Number of agents: 6 Partial Gauss 5.551115123125783E-17 Optimized Partial Gauss 5.551115123125783E-17 Difference 0.0
Number of agents: 7 Partial Gauss 5.551115123125783E-17 Optimized Partial Gauss 5.551115123125783E-17 Difference 0.0
Number of agents: 8 Partial Gauss 6.938893903907228E-17 Optimized Partial Gauss 6.938893903907228E-17 Difference 0.0
Number of agents: 9 Partial Gauss 1.1102230246251565E-16 Optimized Partial Gauss 1.1102230246251565E-16 Difference 0.0
Number of agents: 10 Partial Gauss 1.3877787807814457E-16 Optimized Partial Gauss 1.3877787807814457E-16 Difference 0.0
Number of agents: 11 Partial Gauss 8.326672684688674E-17 Optimized Partial Gauss 8.326672684688674E-17 Difference 0.0
Number of agents: 12 Partial Gauss 1.1102230246251565E-16 Optimized Partial Gauss 1.1102230246251565E-16 Difference 0.0
Number of agents: 13 Partial Gauss 1.6653345369377348E-16 Optimized Partial Gauss 1.6653345369377348E-16 Difference 0.0
Number of agents: 14 Partial Gauss 1.1102230246251565E-16 Optimized Partial Gauss 1.1102230246251565E-16 Difference 0.0
Number of agents: 15 Partial Gauss 1.1102230246251565E-16 Optimized Partial Gauss 1.1102230246251565E-16 Difference 0.0
```

Powyższy zrzut ekranu prezentuje maksymalne błędy bezwzględne dla metody Gaussa z optymalizacją oraz bez oraz różnice między ich wynikami. Testy zostały przeprowadzone dla różnej liczby agentów. Obie metody mają identyczną dokładność, co jest oczekiwanym wynikiem. Jest tak ponieważ wersja z optymalizacją nie wpływa na otrzymany wynik (pomijane są niektóre iteracje).

- metody iteracyjne Jacobiego i Gaussa-Seidela



Powyższy wykres przedstawia maksymalne błędy bezwzględne dla metod iteracyjnych: Jacobiego i Gaussa-Seidela. Testy zostały przeprowadzone dla dokładności 10^{-6} , 10^{-10} , 10^{-14} . Oba algorytmy mają bardzo podobną dokładność, z minimalną przewagą dla algorytmu Gaussa-Seidela.



Przetestowano czas działania dla wszystkich wariantów. Maksymalny rozmiar testowanych macierzy wyniósł $N = 40$, gdzie iteracje zaczynały się od $i = 3$, które w kolejnych krokach było inkrementowane.

Aby rozstrzygnąć kwestię optymalnego doboru metody musimy zwrócić uwagę na dwie kluczowe rzeczy: rozmiar planszy oraz oczekiwana dokładność obliczeń.

Jak łatwo można zauważyć dla każdego testowanego wariantu czas działania algorytmu rośnie wykładniczo wraz ze wzrostem wartości N , chociaż najko-

rzystniejsze wyniki były osiągane dla algorytmu iteracyjnego Gaussa-Seidela.

Podsumowując powyższe rozważania i dane z wykresu, możemy dojść do wniosku, że dla odpowiednio małych wartości N dobór metody w praktyce nie ma wpływu na czas obliczeń, więc jedynym kryterium którym powinno się kierować jest dokładność obliczeń. Jednak dla większych wartości zdecydowaną przewagę ma algorytm Gaussa-Seidela dla każdej żądanej dokładności, co czyni go najbardziej optymalną metodą rozwiązania tego problemu.

Podział pracy

Iwona Jaśtak	Arkadiusz Dąbrowski
przeprowadzenie symulacji	generowanie układu równań oraz macierzy
implementacja metody Monte-Carlo	zoptymalizowany algorytm eliminacji Gaussa z częściowym wyborem elementu podstawowego
algorytm iteracyjny Jacobiego	algorytm iteracyjny Gaussa-Seidela
testowanie błędów bezwzględnych oraz porównanie wyników we wszystkich metodach	testowanie czasu działania oraz analiza optymalnego doboru metody
przygotowanie sprawozdania	