Miron Czech, Jakub Pryc, Ivan Zarzhitski

Algorytmy mrówkowe przy problemie max flow

Wstęp

- Algorytmy mrówkowe stanowią alternatywne podejście do rozwiązywania problemów, które czerpie inspirację z zachowań rzeczywistych mrówek. W kontekście problemów maksymalnego przepływu, algorytmy mrówkowe oferują efektywne narzędzie do optymalnego wykorzystania sieci przepływowej.
- Tradycyjne metody rozwiązywania tego problemu, takie jak algorytm Forda-Fulkersona czy metoda Edmondsa-Karpa, opierają się na wykorzystaniu przepływu powiększającego. Jednakże, w przypadku dużych i złożonych sieci przepływowych, te metody mogą być czasochłonne i generować duże obciążenie obliczeniowe.

Metodologia

- Przeprowadzając testy różnych algorytmów mrówkowych porównywaliśmy wyniki jakościowe i czasowe z algorytmem Forda-Fulkersona.
- Zaimplementowaliśmy 3 algorytmy mrówkowe przyjmujące różne podejścia do algorytmów mrówkowych.
- Stworzyliśmy generator acyklicznych grafów skierowanych w celu skutecznego testowania naszych rozwiązań

Algorytm 1

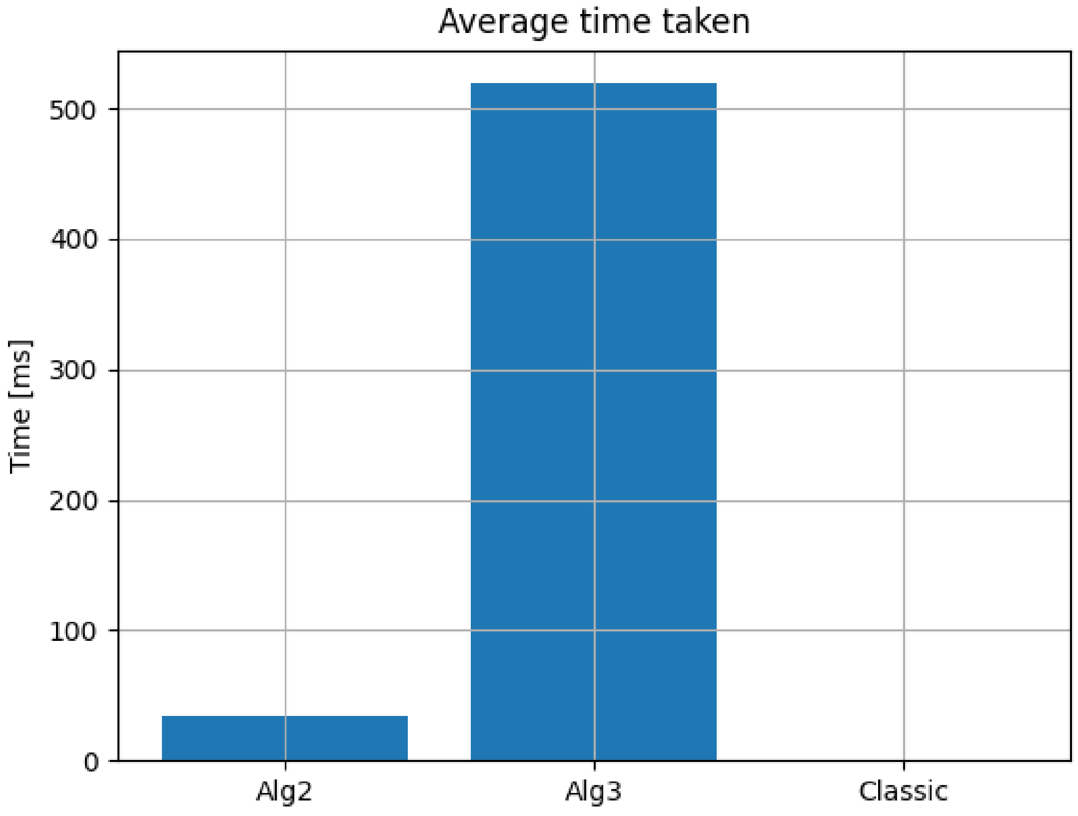
- Każdą krawedź wychodzącą ze źródła (zbiór S) dzielimy na m+1 krawędzi, gdzie m - maksymalny przepływ na tej krawędzi.
- Kazdej z nowopowstałych krawędzi przypisujemy liczbę tau z wartością początkową 1.
- W kolejnych iteracjach dla każdej krawędzi z S losujemy po jednej krawędzi zgodnie z prawdopodobieństwem tau[i]/sum(tau) i następnie dla tych krawędzi próbujemy poprowadzić klasyczny algorytm Forda-Fulkersona.
- Jeżeli się udało - mnożymy użyte tau przez pewne ustalone ro (pewna stała), a następnie dodajemy wartość przepływu podzieloną przez ustalone Q (pewna stała)
- Jeśli się nie udało - tylko mnożymy przez ro

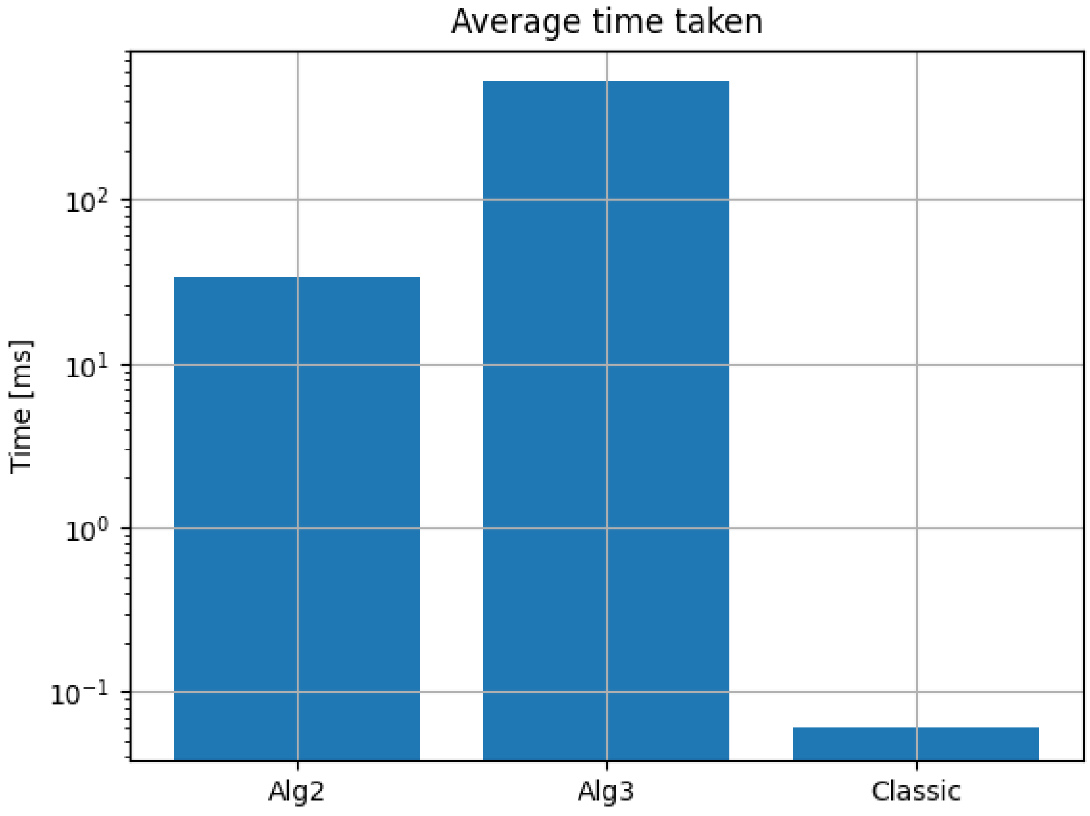
Algorytm 2

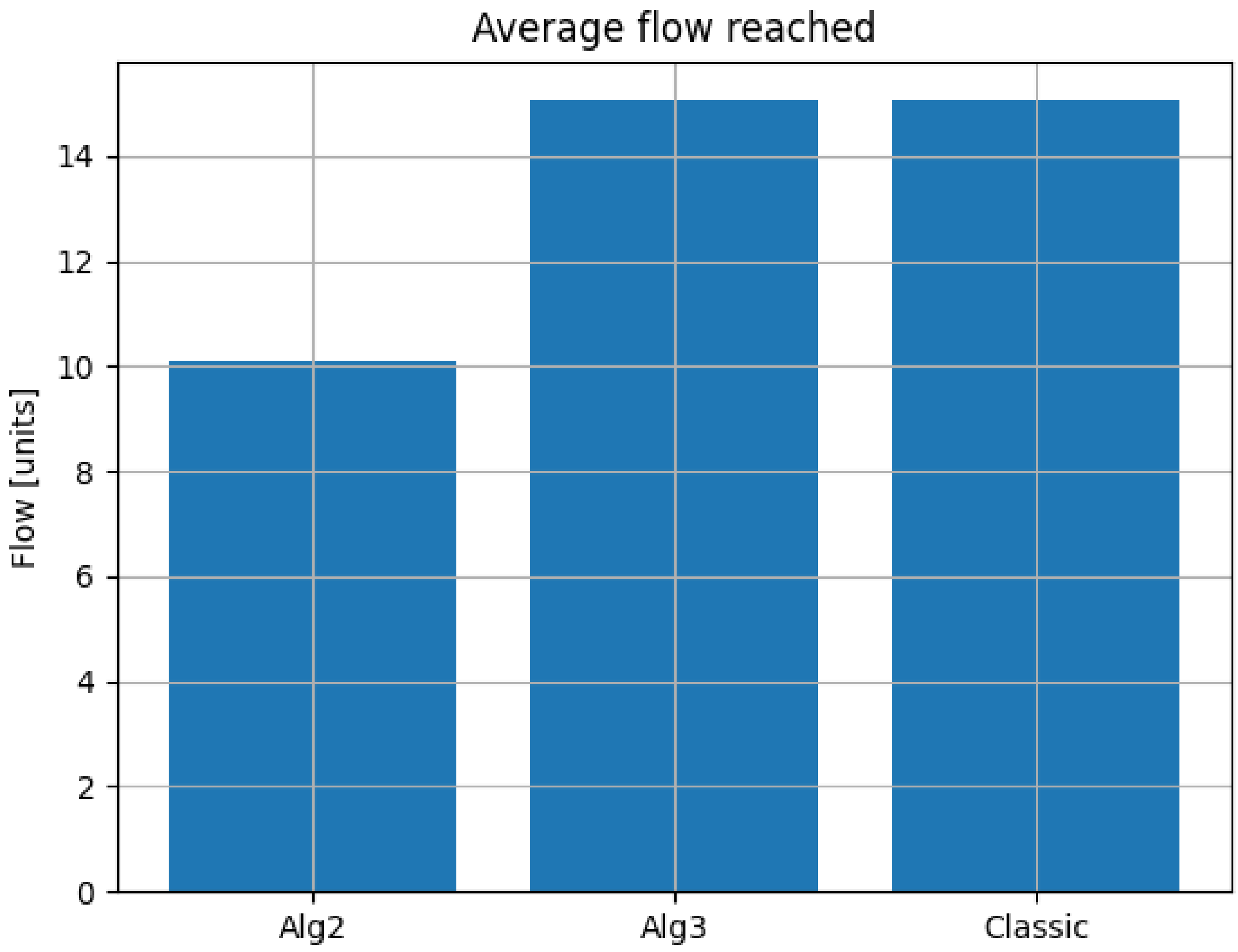
- Każdą losową krawędź dzielimy na m+1 krawędzi, gdzie m - maksymalny przepływ na tej krawędzi.
- Każdej z nowopowstałych krawędzi przypisujemy liczbę tau z wartością początkową 1.
- Dzielimy krawędzie na dwa zbiory: „losowalne” i „nielosowalne”.
- W kolejnych iteracjach dla każdej losowalnej krawędzi losujemy po jednej krawędzi zgodnie z prawdopodobieństwem tau[i]/sum(tau) i następnie za pomocą nielosowalnych krawędzi próbujemy zrobić poprawny przepływ.
- Jeżeli się udało - mnożymy użyte tau przez pewne ustalone ro (pewna stała), a następnie dodajemy wartość przepływu podzieloną przez ustalone Q (pewna stała)
- Jeśli się nie udało - tylko mnożymy przez ro

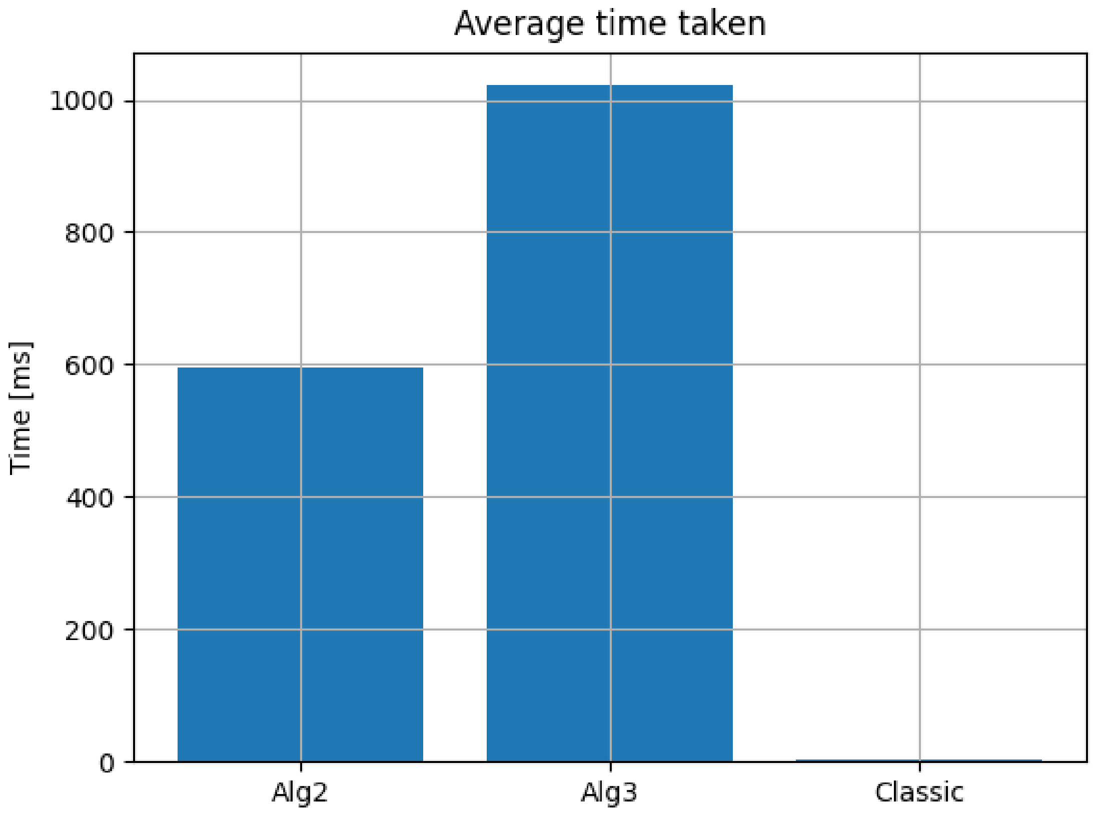
Algorytm 3

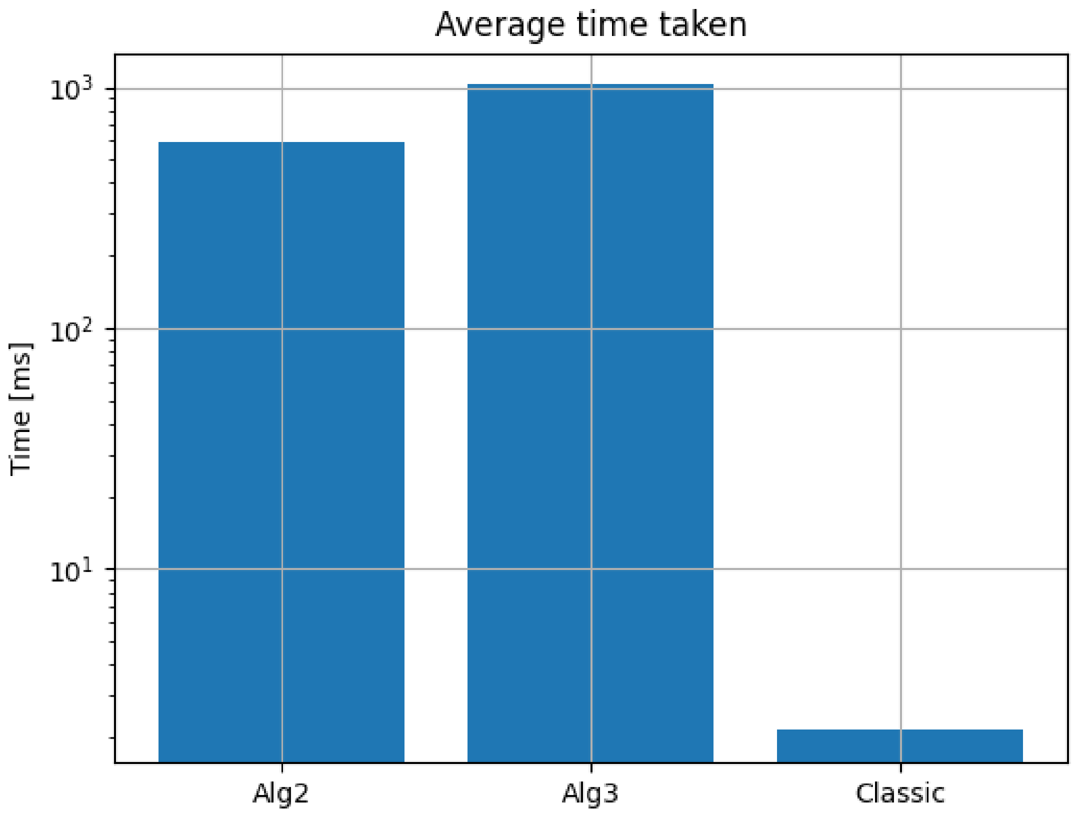
- Mamy n iteracji, w każdej z nich graf przechodzi m mrówek.
- W każdej iteracji tworzymy nową sieć rezydualną.
- Każda z mrówek startuje w źródle i na podstawie prawdopodobieństwa wynikającego z pojemności oraz feromonów danej krawędzi wybiera następny wierzchołek do odwiedzenia.
- Mrówka kontynuuje ten proces aż dojdzie do ujścia bądź nie będzie mogła pójść dalej.
- Jeżeli udało się osiągnąć ujście mrówka odpowiednio modyfikuje sieć rezydualną, sieć feromonów oraz zwiększa aktualny przepływ iteracji.
- Po zakończeniu trawersu grafu przez wszystkie mrówki uaktualniamy sieć feromonów.

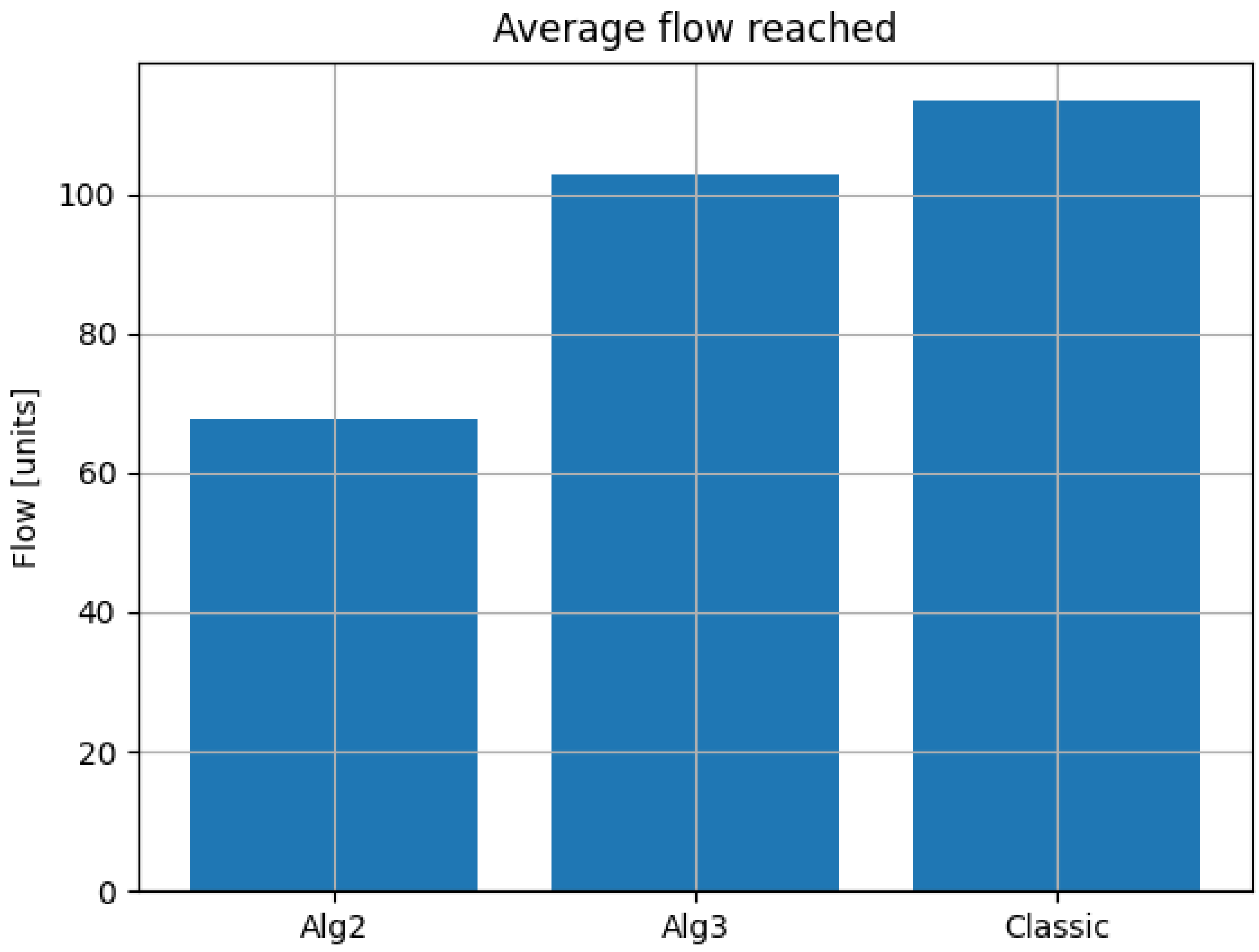

Average time taken
Time [ms]
0
100
200
300
400
500
Alg2
Alg3
Classic

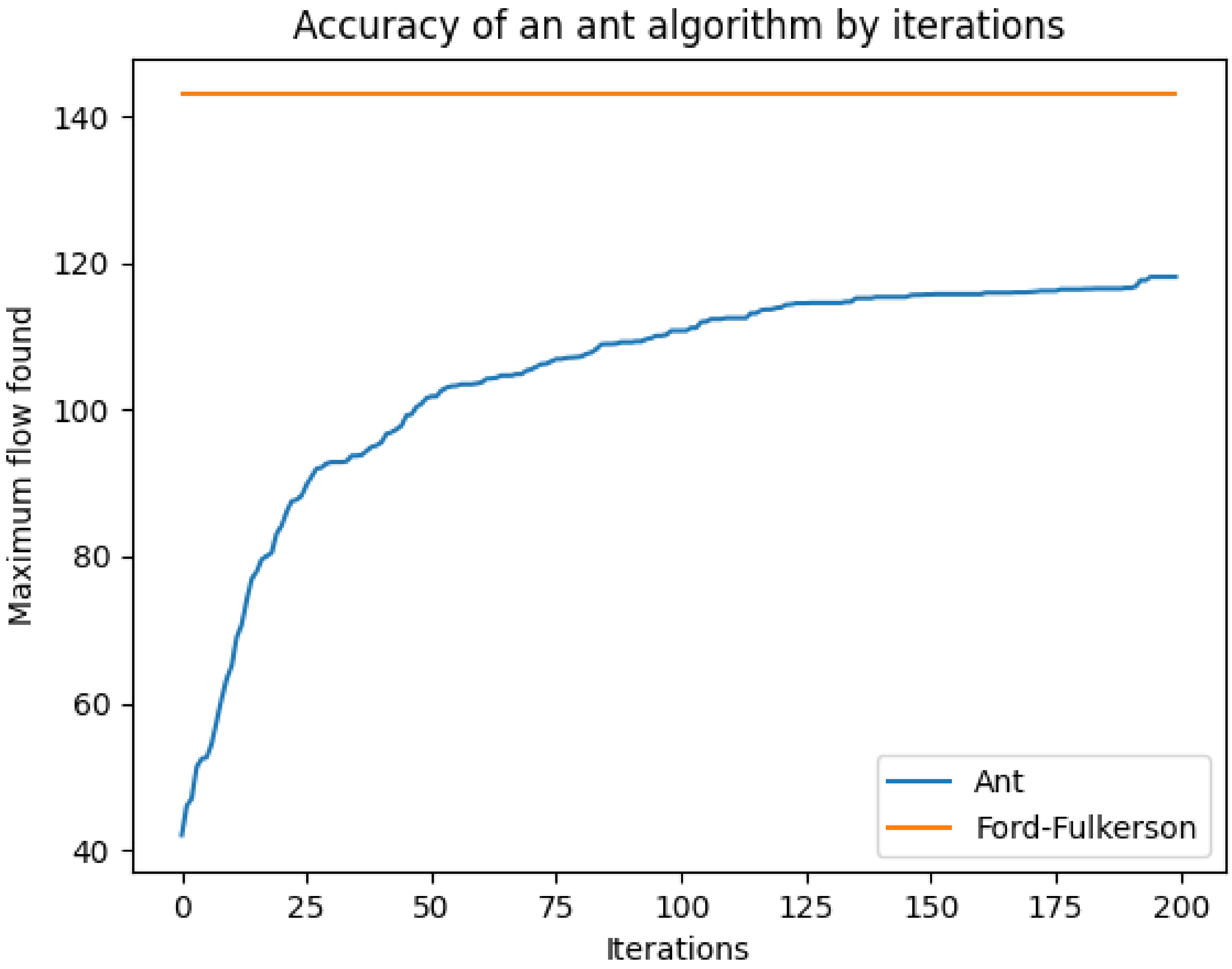

Average time taken
Time [ms]
10^{-1}
10^{0}
10^{1}
10^{2}
Alg2
Alg3
Classic


Average flow reached
Flow [units]
0
2
4
6
8
10
12
14
Alg2
Alg3
Classic


Average time taken
Time [ms]
0
200
400
600
800
1000
Alg2
Alg3
Classic


Average time taken
Time [ms]
10^1
10^2
10^3
Alg2
Alg3
Classic


Average flow reached
Flow [units]
0
20
40
60
80
100
Alg2
Alg3
Classic


Accuracy of an ant algorithm by iterations
Maximum flow found
140
120
100
80
60
40
0
25
50
75
100
125
150
175
200
Iterations
Ant
Ford-Fulkerson

Wnioski

- Wyniki nie są dokładne, lecz wraz z liczbą iteracji się do nich zbliżają.
- Algorytmy mrówkowe lepiej nadają się do problemów NP-trudnych.
- W przeciwieństwie do klasycznych algorytmów dokładność wyniku wymaga więcej czasu.

Dziękujemy za uwagę

Xie, M., Gao, L., Guan, H. (2008). Ant Algorithm Applied in the Minimal Cost Maximum Flow Problem. In: Huang, DS., Wunsch, D.C., Levine, D.S., Jo, KH. (eds) Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. ICIC 2008. Lecture Notes in Computer Science(), vol 5227. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-85984-0_25