Metody Optymalizacyjne
Laboratorium 3

09.06.2025

Adrian Herda
Informatyka Algorytmiczna, Politechnika Wrocławska

1. Wprowadzenie

Celem zadania było zaimplemetowanie algorytmy 2-aproksymującego dla problemu szeregowania zadań na niezależnych maszynach z kryterium minimalizacji długości uszeregowania.

Zaimplementowany algorytm jest oparty na algorytmie 2-aproksymującym dla problemu szeregowania zadań na maszynach równoległych z kryterium minimalizacji długości uszeregowania, który jest opisany w książce Approximation Algorithms (2001, V. V. Vazirani) [1].

Implementacja wykonana została w języku Julia z wykorzystaniem pakietu JuMP [2], oraz solvera HiGHS.

2. Opis algorytmu

Algorytm składa się z 5 kroków:

1. okreslenie zakresu T,
2. wyszukania za pomocą binary search minimalnej wartości T^* z możliwym rozwiązaniem $LP(T)$,
3. znalezienia x rozwiązania bazowego dopuszczalnego dla T^*,
4. zaokrąglenia ułamkowego x za pomocą doskonałego dopasowania w grafie dwudzielnym,
5. wyliczenia długości szeregowania.

2.1. Zakres wartości T

Zakres wartości T to przedział $\left[\left\lceil\frac{\alpha}{m}\right\rceil, \lceil\alpha\rceil\right]$ w którym szukamy rozwiązania, gdzie α to suma czasów wykonania wszystkich zadań na wszystkich maszynach na których te zadania wykonywane są najszybciej, czyli

$$\alpha = \max_{1\leq j\leq m} \sum_{i\in\left\{k:\, \min_{m\in[1,n]}\{p_{m,j}\}=p_{k,j}\right\}} p_{i,j}$$

2.2. Relaksacja do programowania liniowego

2.2.1. Dane wejściowe

- T - zakres czasu, w którym szukamy rozwiązania,
- n - liczba zadań,
- m - liczba maszyn,
- $p_{i,j}$ - czas wykonania zadania i na maszynie j, dla $1 \leq i \leq n$ oraz $1 \leq j \leq m$.

2.2.2. Zmienne

- $x_{i,j} \in [0,1]$ - zmienna, która przyjmuje wartość 1, jeśli zadanie i jest przypisane do maszyny j, dla $1 \leq i \leq n$ oraz $1 \leq j \leq m$.

2.2.3. Ograniczenia

- każde zadanie musi być przypisane do dokładnie jednej maszyny:

$$\sum_{j=1}^{m} x_{i,j} = 1, \text{ dla } \ 1 \leq i \leq n$$

- maszyny moigą pracować conajwyżej w czasie T:

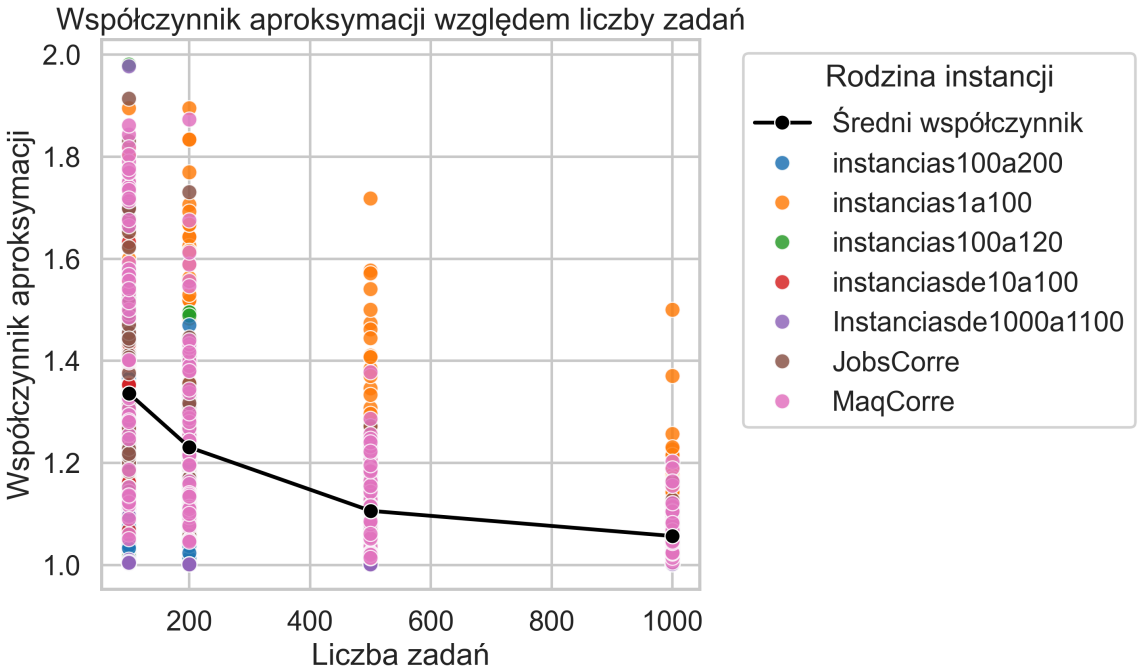
$$\sum_{i=1}^{n} p_{i,j} \cdot x_{i,j} \leq T, \text{ dla } 1 \leq j \leq m$$

2.2.4. Cel

Celem jest zadecydwanie czy istnieje takie przypisanie zadań do maszyn, że całkowity czas wykonania wszystkich zadań nie przekracza T.

3. Wyniki

Współczynnik aproksymacji jest obliczany jako stosunek długości uszeregowania otrzymanego przez algorytm do długości uszeregowania optymalnego.

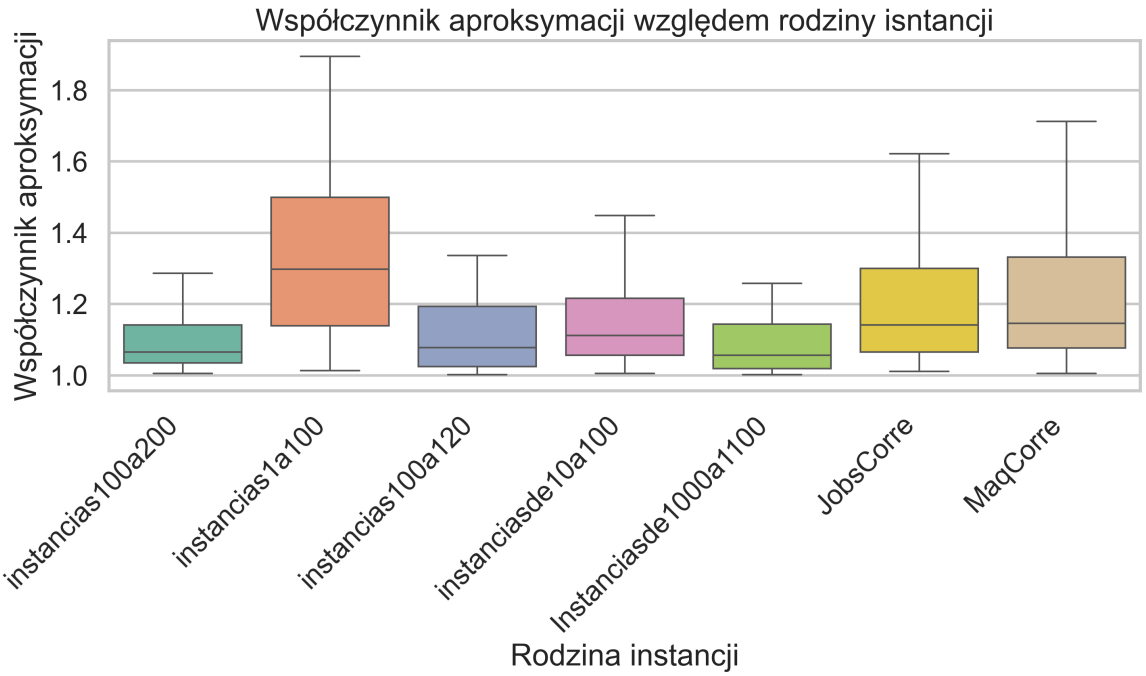


Rysunek 1: Wykres ilustrujący zależność między liczbą zadań a współczynnikiem aproksymacji algorytmu.

Na rysunku 1. widoczny jest trend polepszania się współczynnika aproksymacji wraz ze wzrostem liczby zadań.

Liczba zadań	Średni współczynnik aproksymacji	Maksymalny współczynnik aproksymacji
100	1.336283	1.980198
200	1.230766	1.894737
500	1.106448	1.717949
1000	1.056992	1.500000

Dla mniejszych instancji wpsółczynniki aprokymacji mimo nie tak dużej średniej osiągją wartości nawet bliskie 2, podczas gdy dla większych instancji maksymalne współczynniki nie przekraczają wartości 1.5.



Rysunek 2: Wykres ilustrujący zależność między liczbą zadań a współczynnikiem aproksymacji algorytmu.

Rodzina instancji	Średni współczynnik aproksymacji	Maksymalny współczynnik aproksymacji
Instanciasde1000a1100	1.110597	1.976802
JobsCorre	1.206766	1.913462
MaqCorre	1.242718	1.872549
instancias100a120	1.122442	1.980198
instancias100a200	1.110965	1.540670
instancias1a100	1.331844	1.894737
instanciasde10a100	1.152753	1.632911

Na wykresie 2. widoczny jest rozkład współczynników aproksymacji dla różnych rodzin instancji. Widać, że dla rodzin instancji z wyższymi zakresami czasów wykonywania zadań współczynniki są mniejsze i nie rozrzucone aż tak.Dla rodziny „instancias1a100”, gdzie czasy wykonywania zadań są najmniejsze i mieszczą się w zakresie $1, 100$, współczynniki są największe i dochodzą az do wartości bardzo bliskich wartości 2.

Bibliografia

[1] V. V. Vazirani, *Approximation algorithms*, t. 1. Springer, 2001.

[2] M. Lubin, O. Dowson, J. Dias Garcia, J. Huchette, B. Legat, i J. P. Vielma, „JuMP 1.0: Recent improvements to a modeling language for mathematical optimization", *Mathematical Programming Computation*, 2023, doi: 10.1007/s12532-023-00239-3.