

Szeregowanie równoległych zadań online

Prezentujący: Adam Polak

Autorzy: Grzegorz Gutowski, Adam Polak

Problem.

W rozważanym problemie mamy do dyspozycji m jednakowych maszyn, na których będziemy szeregować zadania. Z i -tym zadaniem związany jest jego czas ogłoszenia r_i , czas jaki zajmuje jego wykonanie t_i , oraz liczba maszyn wymaganych do równoległego wykonania m_i . W żadnym momencie czasu nie mogą być równocześnie zaszeregowane zadania, których łączne wymaganie przekracza m . Celem jest minimalizacja czasu zakończenia ostatniego zadania (tego, które zakończy się jako ostatnie).

Problem rozważamy w wersji online, to znaczy algorytm dowiaduje się o zadaniu w momencie jego ogłoszenia i musi na bieżąco podejmować decyzje o tym, które zadania zaszeregować. Jak to zwykle bywa w przypadku algorytmów online, miarą jakości algorytmu jest współczynnik kompetytywności, czyli maksymalny stosunek wyniku uzyskanego przez algorytm do wyniku optymalnego (możliwego do uzyskania zakładając znajomość przyszłości).

Problem występuje w dwóch wariantach: dopuszczającym wywłaszczanie i niedopuszczającym. W pierwszym wariantcie wykonywanie zadania można w dowolnym momencie przerywać i przywracać. Łączny czas przez jaki zadanie się wykonuje musi być równy jego czasowi wykonania t_i . W drugim wariantcie algorytm może co prawda przerwać wcześniej zaszeregowane zadanie, ale po ponownym zaszeregowaniu będzie musiał je wykonywać od początku.

Naroska i Schwiegelshohn [NS02] pokazali, że każdy algorytm *szeregujący z list* – czyli intuicyjnie po prostu taki, który nigdy nie marnuje czasu nie szeregując zadania, które jeszcze by się zmieściło – osiąga współczynnik kompetytywności co najmniej 2. Wynik ten jest prawdziwy zarówno dla wariantu bez wywłaszczeń jak i z wywłaszczeniami. Same algorytmy szeregujące z list jednak nigdy zadań nie wywłaszczają. Można mieć więc nadzieję, że zwłaszcza dla wariantu dopuszczającego wywłaszczenia istnieją algorytmy o niższym współczynniku kompetytywności.

Johannes [Joh06] pokazała, że żaden deterministyczny algorytm nie może osiągnąć współczynnika kompetytywności poniżej $\frac{6}{5}$ w wariantcie z wywłaszczeniami. W pracy znajdują się również przykłady pokazujące, że algorytmy szeregujące zadania zachłannie (w każdym przedziale czasowym niezależnie) w kolejności zarówno od największej wymaganej liczby maszyn, jak i najdłuższego pozostałego czasu wykonania mają współczynnik kompetytywności równy dokładnie 2.

Kolejnym naturalnym kandydatem na algorytm o współczynniku kompetytywności poniżej 2 jest algorytm zachłanny szeregujący zadania w kolejności od największej pozostałej do wykonania „objętości” (iloczyn czasu i liczby maszyn). Dla obu wyżej wspomnianych przykładów algorytm ten działa optymalnie. Jaki jest współczynnik kompetytywności tego algorytmu?

REFERENCES

- [Joh06] Berit Johannes. Scheduling parallel jobs to minimize the makespan. *Journal of Scheduling*, 2006.
- [NS02] Edwin Naroska and Uwe Schwiegelshohn. On an on-line scheduling problem for parallel jobs. *Inf. Process. Lett.*, 2002.