Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

Sterowanie Procesami Dyskretnymi

Oliwier Woźniak, Dzmitry Mandrukevich

kierunek studiów: Automatyka i robotyka

specjalność: Robotyka

Opis i analiza działania lagorytmu INSA dla problemu $J||C_{max}$

Prowadzący: dr inż. Radosław Grymin

Wrocław 4 czerwca 2024

Spis treści

1 Opis problemu

Problemem do rozwiązania jest gniazdowy problem szeregowania zadań. Polega on na wyznaczeniu kolejnośći ułożenia wielu zadań na wielu maszynach, gdzie jest narzucona kolejność przejścia zadań pomiędzy maszynami. Wynikiem działania algorytmu powinno być wypisanie kolejności w jakiej maszyny powinny wykonywać poszczególne etapy zadań.

Problem ten jest bardzo złożony i żaden algorytm nie zagwarantuje optymalnego rozwiązania.

4	3				
1	4	3	2	2	5
2	2	3	1	1	6
1	4	2	1	3	2
3	2	1	1	2	5

Tabela 1: Przykładowy zestaw danych dla rozwiązywanego problemu

Jak widać na powyższej tabeli ilość danych nawet dla prostych problemów może być zatrważająca. Przechodząc do opisu znaczenia każdej kolejnej cyfry zacznę od pierwszego rzędu: Pierwsza cyfra (w tym przypadku 4) oznacza ilość zadań do wykonania, druga zaś na ilu maszynach będą wykonywane. W zestawie danych dostarczonym przez dr. Makuchowskiego znajduje się jeszcze trzecia cyfra, informująca o sumarycznej ilości zadań elementarnych (nie wszystkie zadania muszą być wykonywane na wszystkich maszynach).

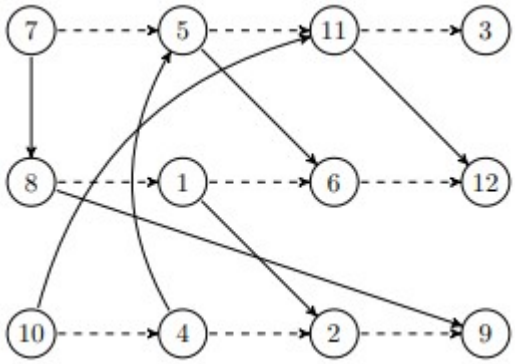
Następnie patrząc na rzędy rozróżniamy te parzyste, oraz nieparzyste. Tworzą one pary, gdzie nieparzyste oznaczają maszynę na której zadanie elementarne ma być wykonywane, zaś parzysta oznacza czas przez jaki będzie wykonywane. Zadania elementarne w ramach każdego zadania muszą być wykonywane w określonej kolejności. Poniżej znajduje się wizualizacja ustawienia działania, po wykonaniu na nim algorytmu.

Rysunek 1: Wizualizacja ułożenia zadań po uruchomieniu algorytmu

Każde zadanie na wizualizacji ma osobny kolor, jak widać nie każda z maszyn jest zajęta przez cały czas. Zadania można rozkodować na podstawie ich ułożenia i czasu działania zadań elementarnych. Brązowa linia oznacza wynik po wczytaniu, zaś zielona po optymalizacji (niestety nie ma możliwości wizualizacji tego wyniku).

1.1 Algorytm INSA

Algorytmem który pozwala na optymalizację ułożenia zadań jest algorytm INSA, który jest oparty o graf opisany przez dwie ścieżki. Jedna z nich opisuje kolejność wykonywania zadań na maszynach (łuk maszynowy), a druga kolejność maszyn przez jakie musi przejść zadanie (łuk technologiczny). Obie te kolejki muszą spełniać określone wymagania, co pozwala na spełnienie ograniczeń narzuconych przez problem. Poniższy graf wizualizuje przykładowe ułożenie zadań w algorytmie, gdzie przerywane linie to łuki maszynowe, a ciągłe to łuki technologiczne.



Rysunek 2: Przykładowy graf dla algorytmu INSA

Algorytm najpierw szereguje zadania zgodnie z czasem ich wykonywania, czyli najdłuższe na początku, a następnie są szeregowane w taki sposób, aby minimalizować funkcję celu. Poniżej jest pseudokod dla algorytmu.

Algorytm 1 Pseudokod dla algorytmu INSA.

procedure INSA($\mathcal{M}, \mathcal{N}, \mathcal{O}$)
  $\alpha \leftarrow (\alpha(1), \alpha(2), \ldots, \alpha(o))$ takie, że: $p_{\alpha(1)} \geqslant p_{\alpha(2)} \geqslant \cdots \geqslant p_{\alpha(o)}$
  $\pi \leftarrow (\pi_1, \pi_2, \ldots, \pi_m)$, gdzie $\pi_{v_{\alpha(1)}} = (\alpha(1))$ oraz $\pi_l = ()$ dla $l \in \mathcal{M} \setminus \{v_{\alpha(1)}\}$
  for $i = 2$ **to** o **do**
    $j \leftarrow \alpha(i)$, $h \leftarrow v(j)$
    $k \leftarrow \arg \min_{a=1,2,\ldots,|\pi_h|+1} \{d_{\pi^a}(a) : G(\pi^a)$ acykliczny$\}$ przy czym $\pi^a = (\pi^a_1, \pi^a_2, \ldots, \pi^a_m)$, $\pi^a_w = \pi_w$ dla $w \in \mathcal{M} \setminus \{h\}$, natomiast $\pi^a_h = \{\pi_h(1), \pi_h(2), \ldots, \pi_h(a-1), j, \pi_h(a+1), \ldots, \pi_h(|\pi_{v_j}|)\}$
    $\pi \leftarrow \pi^k$
  end for
  return π
end procedure

Rysunek 3: Pseudokod dla algorytmu INSA

Działanie, oraz dokładna logika idąca za algorytmem zdecydowanie lepiej opisana jest w [1], więc nie będę dalej rozwijał tematu w tym opracowaniu.

2 Przedstawienie naszego rozwiązania

Nasza implementacja algorytmu INSA jest prymitywna co sprawia, że nie znajduje poprawnego rozwiązania dla każdego problemu. Jednak zachowuje wszystkie ograniczenia związane z samym problemem.

Ze względu na rozległość napisanego kodu nie zostanie on umieszczony w tym sprawozdaniu. Opiszemy jedynie jakie struktury, oraz metody zostały użyte do wykonania zadania.

Graf rozpinający zadania zrealizowaliśmy na dwuwymiarowej tablicy przechowującej wszystkie zadania. Podobnie jak jest to zaprezentowane w zestawach danych dr. Makuchowskiego. Każde z zadań elementarnych otrzymuje dodatkowo własne ID, oraz numer zadania do którego jest przypisane i maszynę na której ma być wykonywany. Podobnie jak w poprzednich przypadkach, wykorzystujemy sortowanie bąbelkowe, aby ustawić zadania od największego czasu wykonywania do najmniejszego.

Nasza implementacja kodu utrudnia realizację operacji grafowych, które są niezbędne do poprawności działania. W związku z tym wszystkie następne operacje są bardzo skomplikowane i prawdopodobnie wykonane niepoprawnie, w wyniku czego kod nie zawsze daje odpowiedni wynik. Niestety dopiero po wykonaniu zadania zorientowaliśmy się jaki błąd popełniliśmy, a zamiana wymagałaby napisania wszystkiego od nowa, lub ingerencję w bardzo skomplikowane funkcje przekształcające tablice przechowujące zadania. Jednak kod w zamyśle działa zgodnie z algorytmem, niekoniecznie jednak w poprawny sposób.

3 Podsumowanie

Ze względu na niepoprawne założenia przy rozpoczęciu projektu nie byliśmy w stanie zaimplementować poprawnie działający algorytm INSA. Obecny kod jest zbyt rozbudowany, żeby wprowadzać w nim poważne zmiany, więc implementacja poprawnego algorytmu wymagałaby zmianę założeń, czyli napisanie programu po raz kolejny.

Jednak jak można zauważyć w pliku wynikowym dostępnym na naszym Githubie, wszystkie zestawy zadań są rozwiązywane w skończonym czasie i ułożone są zgodnie z regułami (funkcja C_{MAX} dba o to). Niestety nie spełniają wymogów postawionych w zadaniu przez dr. Makuchowskiego.

Bibliografia

[1] Czeslaw Smutnicki Eugeniusz Nowicki. “A Fast Taboo Search Algorithm for the Job Shop Problem, Volume 42, Issue 6”. 1996. URL: https://doi.org/10.1287/mnsc.42.6.797.

[2] Andrzej Gnatowski. “Lab. 06: Algorytm INSA”. 2015. URL: http://andrzej.gnatowski.staff.iiar.pwr.wroc.pl/SterowanieProcesamiDyskretnymi/lab06_insa/instrukcja/lab06.pdf.