PTSZ - Zadanie 1 - Problem $Q5|r_j|F$

Dariusz Max Adamski 136674 (grupa I9, godzina 8:15)

dariusz.adamski@student.put.poznan.pl

Data oddania: 18 grudnia 2020

Wstęp

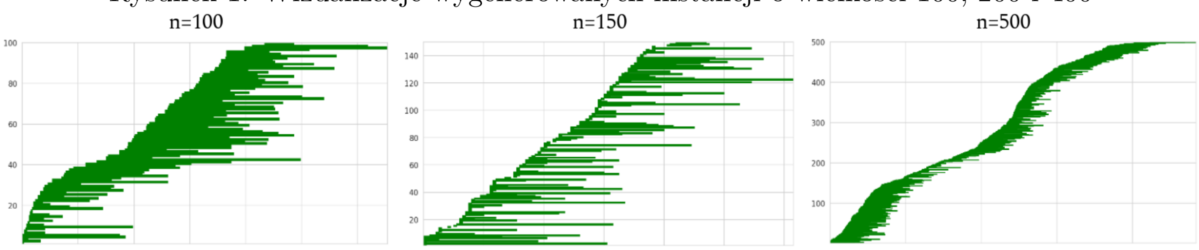
W tym sprawozdaniu opisane jest podejście rozwiązujące problem szeregowania zadań na pięciu równoległych maszynach o różnych prędkościach, z czasami startowymi i minimalizacją średniego czasu przepływu.

1 Generator instancji

Algorithm 1: Algorytm generatora instancji dla problemu

$t \coloneqq 0$
for $i \in 1 \ldots n$ **do**
    if $i = 0 \vee \mathit{Bernoulli}(0.05)$ **then**
        $\mu_p \coloneqq \mathit{Uniform}(10, 100)$
        $\sigma_p \coloneqq \mathit{Uniform}(5, 20)$
        $\mu_s \coloneqq \mathit{Uniform}(1, 4)$
    $t \coloneqq t + \lfloor \mathit{Normal}(\mu_s, 0.5\mu_s) \rfloor$
    $r_i \coloneqq \max\{0, \lfloor \mathit{Normal}(t, 0.5 * \mu_p) \rfloor\}$
    $\underline{p_i} \coloneqq \mathit{clip}(\lfloor \mathit{Normal}(\mu_p, \sigma_p) \rfloor, 1, 200)$
Potasuj losowo zadania

Rysunek 1: Wizualizacje wygenerowanych instancji o wielkości 100, 200 i 400



1.1 Opis algorytmu

Funkcja generująca instancje działa na zasadzie symulacji śledzącej aktualny czas t. Generowane jest n zadań z czasem rozpoczęcia r_i i czasem przetwarzania p_i. Przed wygenerowaniem każdego zadania, czas symulacji jest zwiększany o krok losowany z rozkładu normalnego wyśrodkowanego na parametrze μ_s z odchyleniem standardowym $0.5\sigma_s$. Czas przetwarzania jest losowany z rozkładu normalnego o średniej wartości μ_p i odchyleniu standardowym σ_p, przy czym minimalną wartością jest 1 a maksymalną 200. Następnie generowany jest czas rozpczęcia

r_j, który jest losowany z rozkładu normalnego wyśrodkowanego na aktualnym czasie symulacji t, z odchyleniem standardowym równym połowie aktualnego średniego czasu przetwarzania $0.5\mu_p$, przy czym minimalną wartością r_j jest 0. Po wygenerowaniu wszystkich zadań, zadania są tasowane. Szybkości maszyn b_m są ustawione na 1, 0.25, 0.4, 0.65 i 0.8.

Najciekawszą według mnie częścią algorytmu jest to, że ani krok symulacji, ani parametry zadań nie są losowane ze stałych rozkładów prawdopodobieństwa, ale z rozkładów których parametry losowo zmieniają się z prawdopodobieństwem 5%. Średni czas przetwarzania zadania μ_p jest na przykład losowany z rozkładu jednostajnego ciągłego z minimalną wartością 10 a maksymalną 100. Podobnie odchylenie standardowe czasu przetwarzania σ_p i średni krok czasowy μ_s są losowane z rozkładów jednostajnych. Takie podejście sprawia, że dane instancji są wielomodowe, co widać po okresowych zmianach właściwości zadań. Na rysunku 1. zamieściłem wizualizacje kilku wygenerowanych instancji, które pokazują tą charakterystykę.

2 Algorytm szeregowania

Algorithm 2: Algorytm dla problemu szeregowania

$h(j,m) = \alpha_1(r_j - t_m) + \alpha_2 \frac{p_j}{b_j}$
$\forall_{m \in M} t_m := 0$
$t := 0$
while $T \neq \varnothing$ **do**
    $T_{valid} := \{\langle j,m \rangle \in T \times M \mid t_m \leq t \wedge r_j \leq t\}$
    if $T_{valid} = \varnothing$ **then**
        if $\forall_j t < r_j$ **then**
            $t := \min_j r_j$
        else
            $t := \min_m t_m$
        continue
    $\langle j,m \rangle := T_{valid}[\operatorname{argmin}_i h(T_{valid,i})]$
    $t_m := \max\{r_j, t_m\} + \frac{p_j}{b_m}$
    Zaplanuj zadanie j na maszynie m
    $T := T \setminus \{T_j\}$

2.1 Oznaczenia

Zbiór T początkowo zawiera wszystkie zadania $T_1 \dots T_n$. Gdy zadanie jest dodane do uszeregowania, usuwane jest z tego zbioru. Zbiór M zawiera numery maszyn od 1 do 5. Zmienna t oznacza aktualny czas symulacji, a zmienne t_i oznacza aktualny czas symulacji na i-tej maszynie.

2.2 Opis algorytmu

Algorytm szeregowania działa w następujący sposób. Dla danego punktu w czasie symulacji t wybierana jest najlepsza według heurystyki para zadania i maszyny. Rozpatrywane są tylko maszyny, których czas jest mniejszy lub równy aktualnemu czasowi $t_m \leq t$ (do tej maszyny w tym momencie nic nie możemy przypisać), oraz zadania, których czas rozpoczęcia jest mniejszy lub równy aktualnemu czasowi $r_j \leq t$ (nie chcemy, żeby maszyna skoczyła do czasu r_j w przyszłości, bo zostawiłaby dziurę w uszeregowaniu, którą byśmy musieli później zapełnić).

Po wybraniu zadania j i maszyny m obliczany jest nowy czas maszyny t_m, z uwzględnieniem jej szybkości b_m, zadanie jest dodane do uszeregowania na tej maszynie, a następnie jest usuwane ze zbioru T.

Wybrana heurystyka priorytetyzuje najmniejszy czas wykonania zadania, w zależności od maszyny p_j/b_m i najbardziej spóźnione/najwcześniej dostępne zadania $r_j - t_m$. Heurystyka jest sparametryzowana wartościami α, które są losowane z zakresu (0, 1). Najlepsze parametry ustalane są przez uruchomienie algorytmu 20 razy. Tylko najlepszy wynik szeregowania jest zapisywany.

2.3 Analiza złożoności

Złożoność algorytmu to $O(n^2)$, gdzie n to wielkość instacji - liczba zadań. Złożoność wynika z tego, że zachłannie planujemy n zadań, a przy planowaniu każdego zadania rozważamy maksymalnie $5n$ zadań kandydatów, aby skonstruować zbiór T, Jeśli zbiór kandydatów jest pusty, to ustawiamy aktualny czas na najmniejszy czas maszyny t_m, dzięki czemu złożoność algorytmu nie zależy od czasów p_i, r_i. Na złożoność nie wpływa fakt, że stałą liczbę razy uruchamiam algorytm, żeby dostosować parametry heurystyki.