

Zasady podstawowe:

Każde **zadanie** składa się zawsze z dwóch **operacji**.

Operacje każdego zadania muszą być uszeregowane (zrealizowane) na osobnych maszynach.

Operacja 1 danego zadania musi się w całości wykonać, aby operacja druga tego samego zadania mogła się rozpocząć.

Flowshop – operacja nr 1 zawsze na I maszynie, operacja nr 2 zawsze na II maszynie

Jobshop – generator instancji problemu przypisuje w ramach każdego zadania maszyny do operacji.

Problem 1

Flowshop, liczba maszyn $m=2$, liczba zadań n ,

operacje niewznawialne,

dla pierwszej i drugiej maszyny po k okresów przestoju (na każdą maszynę), o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), okresy te nie mogą się na siebie nakładać (tj. czas przestoju na I maszynie nie może nakładać się z czasem dowolnego przestoju na II maszynie), $k \geq n/10$,

minimalizacja sumy czasów zakończenia wszystkich operacji

Problem 2

Jobshop, liczba maszyn $m=2$, liczba zadań n ,

operacje wznawialne z karą +25% czasu dla operacji wznawianej,

dla pierwszej i drugiej maszyny po k okresów przestoju (na każdą maszynę) o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), $k \geq n/10$,

minimalizacja całkowitego czasu wykonania wszystkich operacji

Problem 3

Jobshop, liczba maszyn $m=2$, liczba zadań n ,

operacje niewznawialne,

dla drugiej maszyny k okresów przestoju o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), $k \geq n/8$,

czas gotowości dla każdej operacji nr 1 każdego zadania, nieprzekraczający $\frac{1}{4}$ sumy czasów wszystkich operacji.

minimalizacja całkowitego czasu wykonania wszystkich operacji

Problem 4

Jobshop, liczba maszyn $m=2$, liczba zadań n ,

operacje niewznawialne,

dla pierwszej i drugiej maszyny k okresów przestoju o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), okresy te nie mogą się na siebie nakładać (tj. czas przestoju na I maszynie nie może pokrywać się z czasem dowolnego przestoju na II maszynie), $k \geq n/5$,

minimalizacja sumy czasów zakończenia wszystkich operacji

Problem 5

Flowshop, liczba maszyn $m=2$, liczba zadań n ,

operacje wznawialne z karą +20% czasu dla operacji,

dla pierwszej maszyny k okresów przestoju o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), $k \geq n/5$,

czas gotowości dla każdej operacji nr 1 każdego zadania, nieprzekraczający połowy sumy czasów wszystkich operacji dla maszyny I

minimalizacja całkowitego czasu wykonania wszystkich operacji

Problem 6

Flowshop, liczba maszyn $m=2$, liczba zadań n ,

operacje niewznawialne,

dla pierwszej maszyny k okresów przestoju o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), $k \geq n/4$,

czas gotowości dla każdej operacji nr 1 każdego zadania, nieprzekraczający połowy sumy czasów wszystkich operacji dla maszyny I

minimalizacja sumy czasów zakończenia wszystkich operacji

Dla wszystkich problemów:

Czas wykonania pojedynczej operacji: najlepiej różne przypadki w ramach klas instancji problemu; od 1 do 20, od 1 do 100 (200), mieszane, np. połowa zadań od 1 do 20, połowa od 1 do 200, itd.

Czas przestoju losowe, ale najlepiej z pewnego zakresu, liczba okresów czasu kiedy maszyna nic nie robi – różne przypadki, np. instancje mające $0.1n$ (n = liczba zadań), $0.5n$, itd.

Inne uwagi ogólne raz jeszcze:

- *Dla dowolnego zadania w ramach instancji dowolnego problemu obowiązują zasady:*
 - *Każde zadanie ma zawsze 2 operacje*
 - *Operacja nr 1 zadania X (op1X) **musi** się w całości zakończyć, aby op2X **mogła** się na innej maszynie rozpocząć (ale nie musi od razu).*
 - *W żadnym wypadku ta sama maszyna **nie ma prawa przetworzyć 2 operacji tego samego zadania (czyli na raz całego zadania) w ramach jakiegokolwiek nawet tymczasowego uszeregowania.***
 - *W przypadku używania w programie zmiennych całkowitoliczbowych, wszelkie ułamki wynikające z procentu doliczanej kary zaokrąglamy **W GÓRĘ**, min. o wartość 1. Tj. 30% kary dla operacji o długości 1, robi z niej operację o długości 2, 30% kary dla zadania o dł. 9 robi z niego zadanie o dł. 12, itd.*
 - *Operacje wstrzymane przez przerwę techniczną muszą się wykonać do końca zaraz po zakończeniu przerwy.*
 - *Jeśli (jakimś cudem) dana operacja wznawialna będzie np. dwukrotnie przerywana przerwami technicznymi, to wciąż kara naliczana do czasu jest **TYLKO** raz.*
-

Format danych wejściowych (jeśli dany element w problemie nie występuje jako ustalony, w ogóle jego dane się nie pojawiają w pliku wejściowym)

Oznaczenia:

op1_2 (czyli *operacja1_2*) – czyli w ogólności opX_Y : X to numer operacji zadania Y, czyli op1_2 to pierwsza operacja drugiego zadania, op1_76 to pierwsza operacja 76-ego zadania, itd. Czyli X zawsze z przedziału <1,2>, Y z przedziału od 1 do n, gdzie n to liczba zadań
readyTime – dla całego zadania, czyli w praktyce **tylko dla jego pierwszej operacji**, jeśli problem nie ma go zdefiniowanego, to wtedy tego pola w pliku też nie ma

**** NR INSTANCJI PROBLEMU ****

liczba_zadań

czas_operacji1_1; czas_operacji2_1; nr_maszyny_dla_op1_1; nr_maszyny_dla_op1_2; readyTime(op1_only);
czas_operacji1_2; czas_operacji2_2; nr_maszyny_dla_op2_1; nr_maszyny_dla_op2_2; readyTime(op1_only);
czas_operacji1_3; czas_operacji2_3; nr_maszyny_dla_op2_1; nr_maszyny_dla_op2_2; readyTime(op1_only);
<itd. aż do ostatniego zadania>

nr_przerwy; **nr_maszyny**; czas_trwania_przerwy; czas_startu_przerwy

<itd. aż do wszystkich przerw>

*** EOF ***

Podstawowy format pliku wyjściowego algorytmu rozwiązującego dany problem:

**** NR INSTANCJI PROBLEMU ****

maksymalny_czas_uszeregowania; **czas_początkowy** (lub inne kryterium optymalizacyjne w zależności od problemu)

Następnie, w dwóch liniach, wymieniamy uszeregowane elementy na pierwszej maszynie (linia I pliku) oraz na drugiej maszynie (linia II pliku) wg schematu:

M1: **id/nazwa_elementu**, par1, par2, par3; **id/nazwa_elementu**, par1, par2, par3; itd.

M2: **id/nazwa_elementu**, par1, par2, par3; **id/nazwa_elementu**, par1, par2, par3; itd.

łączna_liczba_przerw_konserwujących_M1, ich_sumaryczny_czas_trwania_na_M1

łączna_liczba_przerw_konserwujących_M2, ich_sumaryczny_czas_trwania_na_M2

łączna_liczba_przerw_typu_idle_M1, ich_sumaryczny_czas_trwania_na_M1

łączna_liczba_przerw_typu_idle_M2, ich_sumaryczny_czas_trwania_na_M2

*** EOF ***

id/nazwa_elementu: możliwe są trzy rodzaje tego wpisu, każdy ze swoim zestawem parametrów (par1 do maks. par3)

- rodzaj I: operacja (oznaczenie: np. o1_1, o2_14, o1_99)

-par1: czas startu uszeregowania dla operacji

-par2: długość operacji wg instancji

-par3: długość **rzeczywista dla operacji w uszeregowaniu** (dla problemu II i IV – z doliczoną karą) dla problemów I i III: par3 nie występuje (*poprawiono, wcześniej było napisano dokładnie na*

odwrot)

- rodzaj II: przerwa konserwująca (oznaczenia: maint1_M1, maint2_M1, maint8_M2, itd. –czyli tzw.

okresy przestoju)

-par1: czas startu przerwy

-par2: czas trwania przerwy

- rodzaj III: przerwa gdzie nic się nie dzieje (*ale nie rodzaj II*) (oznaczenie: idle1_M1, idle21_M2, itd.)

-par1: czas startu przerwy w szeregowaniu

-par2: czas trwania

Zaokrąglanie: w górę, czasy wszystkiego: liczby całkowite.

Nazwy plików muszą umożliwiać łatwe rozpoznanie który plik wyjściowy zawiera rozwiązanie dla którego pliku instancji – czyli np. numer/seria/id instancji powinny być w ramach nazw plików usystematyzowane.

Przykład drugiego pliku:

**** 117 ****

11789, 14311

M1: op1_76, 0, 20; op1_17, 20, 17; op1_2, 37, 13; idle1_M1, 50, 3; maint1_M1, 53, 17; <itd.>

M2: idle1_M2, 0, 20; op2_76, 20, 13; idle2_M2, 33, 4; op2_17, 37, 20; op2_2, 57, 23; <itd.>

270

0

120

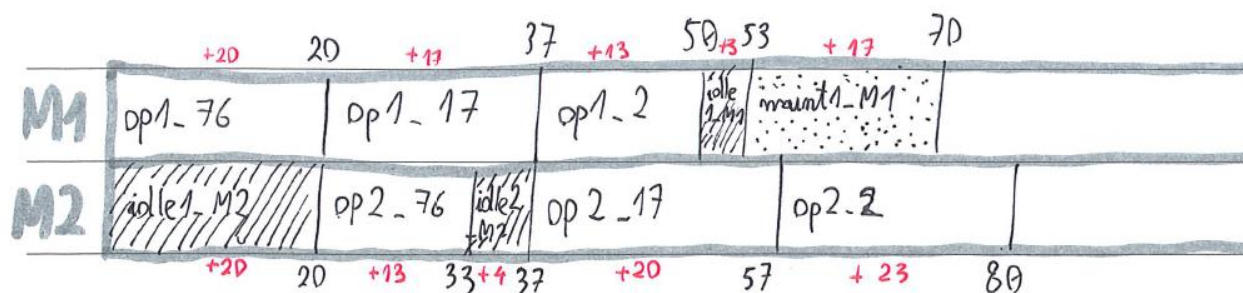
243

*** EOF ***

pierwsza linia: nr instancji problemu (117), aby można było odnaleźć dla uszeregowania odpowiedni plik instancji –**proszę o wysłanie paczki plików na których były prowadzone testy –par plików: instancja-uszeregowanie wystarczy**. To będą pliki tekstowe, czyli nawet gdyby tego było tysiące (w co, przyznaję nauczone doświadczeniem poprzednich lat, nie wierzę :)) to spakowane zajmą pewnie mniej niż plik exe.

druga linia: 11789 to czas po optymalizacji uszeregowania, 14311 to czas uszeregowania początkowego ułożonego generatorem rozwiązań losowych

trzecia i czwarta: patrz rysunek poniżej:



ostatnie cztery linie: 270 – suma maintenance'ów na M1 ; 0 – suma maintenance'ów na M2; 120 – suma przerw gdy maszyna M1 po prostu nie ma co robić (suma bloków idleX_M1), 243 – suma przerw dla M2 gdy ona z kolei nie ma co szeregować (czyli jest to suma wykropkowanych na rysunku bloków idleX_M2). Jakby się ktoś zastanawiał czemu te sumy są w przykładzie takie duże – bo pewnie gdzieś dalej na rysunku byłoby więcej takich prostokątów, tylko nie chciało mi się rysować tak dużego uszeregowania :)