Ostatnia modyfikacja: 18.10.2018

Mini słownik:
okres bezczynności = maintenance = okres przestoju
algorytm wyszukiwania = metaheurystyka

Problem 1

- Flow-shop, liczba maszyn m = 2, liczba zadań *n*, liczba operacji w ramach zadania = m = 2
- Operacje niewznawialne (bez znaczenia w tym problemie, patrz punkt następny)
- Na pierwszej maszynie okresy bezczynności maszyny *ustalane przez algorytm wyszukiwania* (czas trwania – losowy, ale minimum 3x średni czas operacji dla M1). Każda operacja poza pierwszą w uszeregowaniu na M1 oraz poza pierwszą po każdym okresie bezczynności na M1 ma realny czas wykonywania większy o 10% (kumulacja!). Okres bezczynności resetuje tę karę.
    - Kumulacja = +0%, +10%, +20%, +30%, itd. aż do okresu bezczynności na M1, potem znowu +0%, +10%, itd.)
- Minimalizacja czasu wykonania uszeregowania

Problem 2

- Flow-shop, liczba maszyn m = 2, liczba zadań *n*, liczba operacji w ramach zadania = m = 2
- Operacje niewznawialne (bez znaczenia w tym problemie, patrz punkt następny)
- Na drugiej maszynie okresy bezczynności maszyny *ustalane przez algorytm wyszukiwania* (czas trwania – losowy, ale minimum 1.5 x średni czas operacji dla M2). Każda operacja poza pierwszą w uszeregowaniu na M2 oraz poza pierwszą po każdym okresie bezczynności na M2 ma realny czas wykonywania większy o 10% (kumulacja!). Okres bezczynności resetuje tę karę.
    - Kumulacja = +0%, +10%, +20%, +30%, itd. aż do okresu bezczynności na M2, potem znowu +0%, +10%, itd.)
- Minimalizacja sumy czasów zakończenia wszystkich operacji

Problem 3

- Flow-shop, liczba maszyn m = 2, liczba zadań *n*, liczba operacji w ramach zadania = m = 2
- Operacje niewznawialne
- Dla **pierwszej** maszyny k okresów bezczynności maszyny (o losowym czasie rozpoczęcia* i trwania, k to minimum 20% z n)
- Na pierwszej maszynie zaczynając od drugiej operacji w uszeregowaniu, każde kolejne trwa o 5% krócej (zaokrąglenie w górę) – **kumulacyjnie do maksymalnie 25% krótszego czasu trwania**.
    - Przerwa techniczna (= przypisany okres bezczynności na M1) resetuje tą wartość – czyli np. zaraz po przerwie na M1 pewna operacja trwa tyle ile wynosi jest teoretyczny czas, następna trwa o 5% krócej, kolejna już 10% krócej, kolejna 15%, itd. aż do następnego min. -25% do kolejnego maintenance'u.
- Minimalizacja czasu wykonania uporządkowania

Problem 4

- Job-shop, liczba maszyn m = 2, liczba zadań *n*, liczba operacji w ramach zadania = m = 2
- Operacje wznawialne z karą +50% czasu dla operacji wznowionej (po okresie bezczynności)
- Dla **pierwszej i drugiej** maszyny k okresów bezczynności maszyny (o losowym czasie rozpoczęcia* i trwania, k to minimum 20% z n)
- Minimalizacja sumy czasów zakończenia wszystkich operacji

Problem 5

- Job-shop, liczba maszyn m = 2, liczba zadań *n*, liczba operacji w ramach zadania = m = 2
- Operacje wznawialne z karą +50% czasu dla operacji wznowionej (po okresie bezczynności)
- Dla **pierwszej** maszyny k okresów bezczynności maszyny (o losowym czasie rozpoczęcia* i trwania, k to minimum 20% z n)
- Dla operacji nr 1 każdego zadania nadany jest losowy czas gotowości do uszeregowania (*ready time*, wynoszący od 0 do maksymalnie około 50% z sumy czasów trwania wszystkich operacji nr 1 na M1).
- Minimalizacja sumy czasów zakończenia wszystkich operacji

Zasady podstawowe dla problemów do wyboru (będą one bardzo dokładnie wyjaśniane na drugich laboratoriach)

1. Każde **Zadanie** składa się zawsze z dwóch **Operacji.**
2. Jako punkt wyjściowy do sprawozdania należy przyjąć *n*=50
3. Operacje każdego zadania muszą być uszeregowanie (zrealizowane, wstawione, itp.) na **różnych maszynach** (zawsze liczba operacji w każdym zadaniu jest równa liczbie maszyn)
4. Operacja 1 danego zadania X **musi się w całości wykonać, zanim operacja druga tego samego zadania X może się rozpocząć** na alternatywnej maszynie *(niezależnie od modelu szeregowania z punktu 4).*
5. Dwa modele uszeregowania operacji:
    a. **Flow-shop –** operacja nr 1 zawsze na M1, operacja nr 2 zawsze na M2
    b. **Job-shop –** generator instancji problemu przypisuje **niezależne ramach każdego zadania z listy zadań** odpowiednie operacje do maszyn (czyli albo op1-> M1 i op2-> M2 albo odwrotnie: op1-> M2 a op2-> M1)
        i. *Uwaga: nawet jeśli op1 jest na M2, to całe op1* ***musi się najpierw zakończyć na M2****, aby op2 na M1 mogło się w ogóle rozpocząć, patrz: punkt 4 powyżej)*
6. Wszelkie ułamki wynikające na przykład z procentu doliczanej kary zaokrąglamy **w górę**, tj. 30% kary dla operacji o długości 1, zrobi z niej operację o długości 2, 50% kary dla zadania o dł. 9 robi z niego zadanie o dł. 14 (9 + 4.5 (czyli 5 po zaokrągleniu) = 14)
7. Operacje wstrzymane przez przerwę techniczną muszą się wykonać do końca zaraz po zakończeniu przerwy (jeśli problem dopuszcza wznawialność)
8. Jeśli (jakimś cudem) dana operacja wznawialna będzie np. dwukrotnie przerywana przerwami technicznymi (okres bezczynności, *maintenance*), to kara naliczana do jej czasu trwania jest naliczana tylko za pierwszym razem.
9. Czas rozpoczęcia przerw technicznych (* przy opisie w problemie): np. jeśli przerwy mają być na M1, wtedy zliczamy sumę czasów wszystkich **operacji** przewidzianych w instancji problemu dla M1. Niech suma ta będzie oznaczona jako **sum**. Wtedy przerwy techniczne na M1 mają mieć wylosowany czas rozpoczęcia z przedziału od 0 do **sum**. Analogicznie dla M2 (jeśli dotyczy).

Generator instancji:
- określa liczbę zadań (choć można przyjąć, że n=50 jest stałe)
- dla każdego zadania losuje z pewnego zakresu czasy trwania operacji (np. z przedziału od 5 do 20)
- jeśli model szeregowania to job-shop, to określa, które zadania mają op1->M1/op2->M2 a które odwrotnie
- jeśli występują czasy gotowości dla operacji nr 1 zadań to też je losuje

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 linijkach, 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**, gdzie pierwsza liczba to nr operacji (1 lub 2), druga to numer zadania do której operacja należy
  - -par1: czas startu uszeregowania dla operacji
  - -par2: długość operacji wg instancji
  - -par3: długość **rzeczywista dla operacji w uszeregowaniu** (o ile np. dla danego problemu jest doliczana kara, tj. operacje są wznawialne, czas się wydłuża (problemy 1 i 2) lub skraca (3))

- **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.

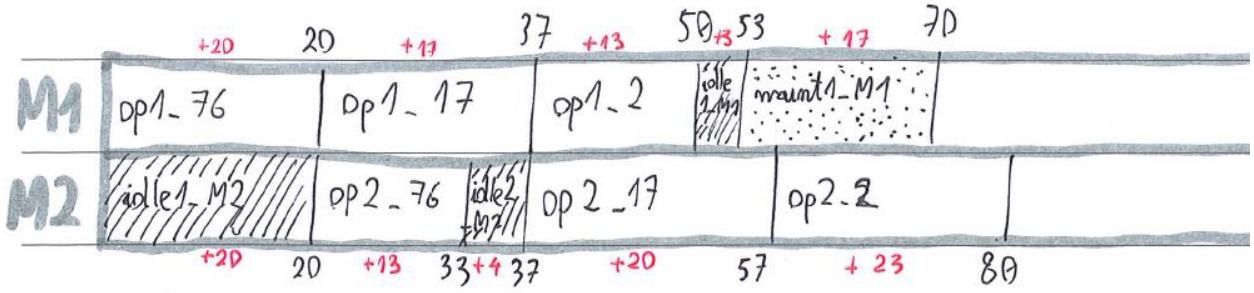
Przykład drugiego pliku:

****** 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 były tysiące (w co, przyznaję nauczony 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 :)