Metody Optymalizacji – zestaw 2

Bartosz Chodorowski `<166142@student.pwr.wroc.pl>`

Wrocław, 13 grudnia 2011

1 Zadanie 1

1.1 Model

1.1.1 Dane

M	–	maksymalna szerokość deski
N	–	liczba różnych desek, które można wyciąć
$c[i]$	–	i-ta szerokość deski, które można wyciąć [cale], $i = 1 \dots N$
$d[i]$	–	zapotrzebowanie na deski o szerokości $c[i]$ [sztuki], $i = 1 \dots N$

1.1.2 Dane pomocnicze (obliczone przez skrypt)

R	–	maksymalna liczba sposobów podzielenia deski w zadany sposób
$t[i][j]$	–	liczba desek o szerokości $c[i]$ które są wycięte w kombinacji j [sztuki], $i = 1 \dots N, j = 1 \dots R$
$o[j]$	–	szerokość powstałego odpadu przy cięciu deski na sposób j [cale], $j = 1 \dots R$

1.1.3 Zmienne decyzyjne

$X[j]$ – liczba desek, które trzeba podzielić w sposób j [sztuki], $j = 1 \dots R$

1.1.4 Funkcja celu

Minimalizujemy sumę szerokości odpadów [cale]:

$$\sum_{j=1}^{R} X[j] \cdot o[j]$$

1.1.5 Ograniczenia

Należy wyprodukować dokładnie tyle desek, jakie jest zapotrzebowanie:

$$(\forall i \in 1 \dots N)(\sum_{j=1}^{R} X[j] \cdot t[i][j]) \geqslant d[i]$$

1.2 Wyniki

Dla danych z treści zadania otrzymano następujący wynik:

```
// solution (optimal) with objective 18
1 podzialow deski: 3 desek 7-calowych, 0 desek 5-calowych, 0 desek 3-calowych,
1 podzialow deski: 0 desek 7-calowych, 1 desek 5-calowych, 0 desek 3-calowych,
19 podzialow deski: 1 desek 7-calowych, 3 desek 5-calowych, 0 desek 3-calowych,
44 podzialow deski: 2 desek 7-calowych, 1 desek 5-calowych, 1 desek 3-calowych,
9 podzialow deski: 0 desek 7-calowych, 2 desek 5-calowych, 4 desek 3-calowych,
```

2 Zadanie 2

2.1 Model

2.1.1 Dane

M – liczba cech populacji
N – liczba dysków
$q[i][j]$ – równe 1 gdy na dysku j znajdują się informacje o cesze populacji i, równe 0 w przeciwnym przypadku, $i = 1 \dots M, j = 1 \dots N$

2.1.2 Zmienne decyzyjne

$u[j]$ – równe 1 gdy należy użyć dysku j równe 0 w przeciwnym przypadku, $j = 1 \dots N$

2.1.3 Funkcja celu

Minimalizujemy liczbę używanych dysków:

$$\sum_{j=1}^{N} u[j]$$

2.1.4 Ograniczenia

Każda cecha musi być na co najmniej jednym użytym dysku:

$$(\forall i \in 1 \ldots M) \ \ (\sum_{j=1}^{N} q[i][j] \cdot u[j]) \geqslant 1$$

2.2 Wyniki

Dla danych:

```
M = 3;
N = 10;

q = [
        [1 0 0 1 0 0 1 0 0 1]
        [0 1 0 0 1 0 0 1 0 0]
        [0 0 1 0 0 1 0 1 1 0]
        ];
```

otrzymano następujący wynik:

```
// solution (optimal) with objective 2
u = [1 0 0 0 0 0 0 1 0 0];
```

Oznacza to, że należy użyć dysku pierwszego i ósmego.

3 Zadanie 3

3.1 Model

3.1.1 Dane

M	–	liczba funkcji
N	–	liczba podprogramów
$r[i][j]$	–	liczba komórek pamięci, którą potrzebuje podprogram j obliczania funkcji i, $i = 1 \ldots M, j = 1 \ldots N$
$t[i][j]$	–	ilość czasu, którą potrzebuje podprogram j obliczania funkcji i [jednostka niesprecyzowana], $i = 1 \ldots M, j = 1 \ldots N$
$I[i]$	–	równe 1, gdy zadane jest obliczenie funkcji i, równe 0 w przeciwnym przypadku, $i = 1 \ldots M$
R	–	liczba komórek pamięci, które są do dyspozycji

3.1.2 Zmienne decyzyjne

$X[i][j]$ – zmienna binarna równa 1 gdy należy użyć podprogramu j obliczania funkcji i, $i = 1 \ldots M, j = 1 \ldots N$

3.1.3 Funkcja celu

Minimalizujemy czas obliczenia zadanych funkcji [jednostka nieokreślona]:

$$\sum_{i=1}^{M}\sum_{j=1}^{N} X[i][j] \cdot t[i][j]$$

3.1.4 Ograniczenia

Należy obliczyć każdą zadaną funkcję:

$$(\forall i \in \{\iota : I[\iota] = 1\}) \quad (\sum_{j=1}^{N} X[i][j]) \geqslant 1$$

Nie można przekroczyć pamięci:

$$(\sum_{i=1}^{M}\sum_{j=1}^{N} X[i][j] \cdot r[i][j]) \leqslant R$$

3.2 Wyniki

Dla danych:

```
M = 3;
N = 4;

r = [
        [1 4 2 5]
        [3 3 5 6]
        [2 4 7 7]
];

t = [
        [1.5 6.6 4.5 5.4]
        [0.6 4.5 6.4 5.3]
        [0.7 2.3 0.3 3.4]
];

I = [1 1 1];

R = 12;
```

otrzymano następujący wynik:

```
// solution (optimal) with objective 2
X =    [[1 0 0 0]
        [1 0 0 0]
        [0 0 1 0]];
```

Oznacza to, że do obliczenia funkcji 1 należy użyć podprogramu 1, do obliczenia funkcji 2 należy użyć 1 podprogramu oraz dla funkcji 3 należy użyć 3 podprogramu.

4 Zadanie 4

4.1 Model

4.1.1 Dane

$$\begin{aligned} M &\quad - \quad \text{liczba zadań} \\ N &\quad - \quad \text{liczba maszyn} \\ d[i][j] &\quad - \quad \text{czas wykonania zadania } i \text{ na } j\text{-tej maszynie, } i = 1 \dots M, j = 1 \dots N \end{aligned}$$

4.1.2 Zmienne decyzyjne

$$\begin{aligned} t[i][j] &\quad - \quad \text{czas rozpoczęcia } i\text{-tego zadania na } j\text{-tej maszynie, } i = 1 \dots M, j = 1 \dots N \\ m &\quad - \quad \text{czas zakończenia wszystkich zadań} \end{aligned}$$

4.1.3 Funkcja celu

Minimalizujemy m.

4.1.4 Ograniczenia

Moment rozpoczęcia i-tego zadania na $j+1$-szej maszynie musy być nie mniejszy niż moment zakończenia i-tego zadania na j-tej maszynie:

$$(\forall i \in 1 \dots M)(\forall j \in 1 \dots N) \;\; t[i][j+1] \geqslant t[i][j] + d[i][j]$$

Tylko jedno zadanie wykonywane jest w danym momencie na j-tej maszynie:

$$(\forall i, k \in 1 \dots M)(\forall j \in 1 \dots N) \quad (t[i][j] \geqslant t[k][j] + d[k][j] \quad \vee \quad t[k][j] \geqslant t[i][j] + d[i][j])$$

Ponadto, m ma równać się czasowi zakończenia wszystkich zadań na ostatniej maszynie, więc:

$$(\forall i \in 1 \dots M) \;\; t[i][N] + d[i][N] \leqslant m$$

4.2 Wyniki

Dla danych:

```
m = 7;
n = 3;

d =  [
        [3,       3,       2],
        [9,       3,       8],
        [9,       8,       5],
        [4,       8,       4],
```

```
        [6,       10,       3],
        [6,        3,       1],
        [7,       10,       3]
        ];
```

otrzymano następujący wynik (odrobinę skrócono długie przedziały, aby całość zmieściła się na papierze):

```
[4--4][7-----7][5-----------------5][2--------------------2][3---------------3]------[1----1][6------------------6]-------------------
------[4-----------4][7--------------------7][5--------------------5][2------2][3------------------3][1----1]------[6---------6]------
---------------------[4-----4]---------------[7------7]--------------[5------5][2------------------2][3----------------3][1---1][6--6]
```

5 Zadanie 5

5.1 Model

5.1.1 Dane

$$
\begin{aligned}
N & \quad - \quad \text{liczba zadań} \\
P & \quad - \quad \text{czas wykonania zadania [liczba okien czasowych]} \\
s[i] & \quad - \quad \text{czas w którym przybyło zadanie } i \text{ [okno czasowe]}, \; i = 1 \ldots N \\
w[i] & \quad - \quad \text{waga zadania } i, \; i = 1 \ldots N
\end{aligned}
$$

5.1.2 Zmienne decyzyjne

$$
t[i][j] \quad - \quad \text{czas w którym wykonywane jest „}j\text{-ta część" zadania } i \text{ [okno czasowe]}, \; i = 1 \ldots N, j = 1 \ldots P
$$

5.1.3 Funkcja celu

Minimalizujemy kryterium podane w treści zadania

$$
\sum_{i=1}^{N} \left(w[i] \cdot ((\max_{j \in 1 \ldots P} t[i][j]) + 1) \right)
$$

5.1.4 Ograniczenia

Nie możemy wykonywać 2 zadań w tym samym czasie:

$$
(\forall i_1, i_2 \in 1 \ldots N)(\forall j_1, j_2 \in 1 \ldots P) \quad (\neg(i_1 = i_2 \wedge j_1 = j_2) \rightarrow t[i_1][j_1] \neq t[i_2][j_2])
$$

Nie możemy zacząć zadania, zanim się ono pojawi:

$$
(\forall i \in 1 \ldots N)(\forall j \in 1 \ldots P) \quad t[i][j] \geqslant s[i]
$$

5.2 Wyniki

Dla danych z treści zadania otrzymano następujący wynik:

```
// solution (integer optimal, tolerance) with objective 2649
122211466644555333
```