

Metody Optymalizacji – zestaw 1

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

Wrocław, 13 listopada 2011

1 Zadanie 1

1.1 Model

Model został zdefiniowany w treści zadania.

1.2 Wyniki

Dla $n=6$ otrzymano:

```
// solution (optimal) with objective 7.83852813852814
Error = 4.738847969956563e-010

x = [1 1 1 1 1 1];
```

Natomiast dla $n=7$:

```
// solution (optimal) with objective 9.22187258054863
Error = 0.6405973665768481

x = [1.0004 0.98316 1.1623 0.36869 2.1574 0 1.3283];
```

Wykonane eksperymenty pokazują, że *CPLEX Optimization Studio* ma problemy z rozwiązaniem zagadnienia programowania liniowego dla macierzy Hilberta wymiaru ≥ 7 .

2 Zadanie 2

2.1 Model

2.1.1 Dane

- f – liczba firm
- a – liczba lotnisk
- $S[i]$ – możliwość dostarczenia paliwa przez firmę i [galony], $i = 1 \dots f$
- $D[j]$ – niezbędna ilość paliwa do funkcjonowania lotniska j [galony], $j = 1 \dots a$
- $P[j][i]$ – cena kupna i dostarczenia galonu paliwa przez firmę i na lotnisko j [\$/galon], $i = 1 \dots f, j = 1 \dots a$

2.1.2 Zmienne decyzyjne

$C[j][i]$ – ilość galonów, jaką należy zakupić dla lotniska j od firmy i [galony], $i = 1 \dots f, j = 1 \dots a$

2.1.3 Funkcja celu

Minimalizujemy wydane pieniądze [dolary]:

$$\sum_{j=1}^a \sum_{i=1}^f C[j][i] \cdot P[j][i]$$

2.1.4 Ograniczenia

Każde lotnisko musi zakupić co najmniej tyle paliwa, ile potrzeba do jego funkcjonowania:

$$(\forall j \in 1 \dots a) \left(\sum_{i=1}^f C[j][i] \right) \geq D[j]$$

Jednakże, żadna firma nie może sprzedać więcej paliwa niż tyle ile jest w stanie dostarczyć:

$$(\forall i \in 1 \dots f) \left(\sum_{j=1}^a C[j][i] \right) \leq S[i]$$

2.2 Wyniki

Otrzymano następujący wynik:

```
// solution (optimal) with objective 8525000
pucharse = [[0 1.1e+5 0]
             [1.65e+5 55000 0]
             [0 0 3.3e+5]
             [1.1e+5 0 3.3e+5]];
```

Oznacza to, że:

- lotnisko 1 musi kupić 110000 galonów paliwa od firmy 2,
- lotnisko 2 musi kupić 165000 galonów paliwa od firmy 1 i 55000 galonów paliwa od firmy 2,
- lotnisko 3 musi kupić 330000 galonów paliwa od firmy 3,
- lotnisko 4 musi kupić 110000 galonów paliwa od firmy 1 i 300000 galonów paliwa od firmy 3.

Całkowity koszt zakupu wyniesie \$ 8,525,000.

3 Zadanie 3

3.1 Model

3.1.1 Dane

- m – liczba miesięcy
- $W[i]$ – liczba roboczogodzin potrzebnych do funkcjonowania firmy [godziny], $i = 1..m$
- s_1 – miesięczna płaca stewardesy [dolary]
- s_2 – miesięczna płaca praktykantki [dolary]
- l_1 – maksymalny czas pracy stewardessy miesięcznie [godziny]
- l_2 – długość stażu, jaki musi odbyć praktykantka w czasie jednego miesiąca [godziny]
- b – ilość zatrudnionych stewardes na początku pierwszego miesiąca

3.1.2 Zmienne decyzyjne

- $F_1[i]$ – ilość stewardes zatrudnionych w miesiącu i , $i = 1 \dots m$
- $F_2[i]$ – ilość praktykantek zatrudnionych w miesiącu i , $i = 1 \dots m$

3.1.3 Funkcja celu

Minimalizujemy wydane pieniądze [dolary]:

$$\sum_{i=1}^m F_1[i] \cdot s_1 + F_2[i] \cdot s_2$$

3.1.4 Ograniczenia

Stewartesy muszą wypracować wymaganą liczbę godzin, aby samoloty mogły latać, muszą też wyszkolić praktykantki (zakładamy, że nie spełniają wówczas swoich obowiązków):

$$(\forall i \in 1 \dots m) \quad F_1[i] * l_1 - F_2[i] * l_2 \geq W[i]$$

Na początku mamy b stewartes:

$$F_1[1] = b$$

Kontrolujemy sytuację, która zmienia się z miesiąca na miesiąc:

$$(\forall i \in 2 \dots m) \quad F_1[i] \approx \frac{9}{10} F_1[i-1] + F_2[i-1]$$

3.2 Wyniki

Dla modelu zmiennoprzecinkowego otrzymano następujący wynik:

```
// solution (optimal) with objective 1021744.59876543
fullTimeFlightAssistant = [60 63.445 62.269 69.444 66.667 70];
internshipFlightAssistant = [9.4452 5.1678 13.403 4.1667 10 0];
```

Oznacza to, że w styczniu należy zatrudnić 9.4452 praktykantki, w lutym 5.1678, w marcu 13.403, w kwietniu 4.1667 i w maju 10. W czerwcu praktykantki będą zbędne. Łączne pensje wyniosą około \$ 1,021,744.60.

Przechodząc na model ciągły, wyniki nabierają większego sensu:

```
// solution (optimal) with objective 1027000
fullTimeFlightAssistant = [60 64 63 70 67 70];
internshipFlightAssistant = [10 5 13 4 10 0];
```

Należy zatem w styczniu zatrudnić 10 praktykantek, w lutym 5, w marcu 13, w kwietniu 4, w maju 10 i żadnej w czerwcu. Płace wyniosą \$ 1,027,000.

Obserwując slack variables z ograniczenia mówiącego o konieczności wypracowania odpowiedniej liczby godzin można wywnioskować, w którym miesiącu stewartesy będą miały nadmiar godzin.

↓ Months (size 6)	Values	
	Constraint	Slack
1	fullTimeFlightAssistant[1]*1...Assistant[1]*(-100) >= 8000	0
2	fullTimeFlightAssistant[2]*1...Assistant[2]*(-100) >= 9000	-100
3	fullTimeFlightAssistant[3]*1...Assistant[3]*(-100) >= 8000	-150
4	fullTimeFlightAssistant[4]*1...Assistant[4]*(-100) >= 10000	-100
5	fullTimeFlightAssistant[5]*1...Assistant[5]*(-100) >= 9000	-50
6	fullTimeFlightAssistant[6]*1...Assistant[6]*(-100) >= 10500	0

W tym przypadku najmniej pracy przypada na marzec.

4 Zadanie 4

4.1 Model

4.1.1 Dane

- d – ilość dni w tygodniu
- s – ilość ćwiczeń z każdego z przedmiotów
- c – ilość przedmiotów
- $(b, e, a) \in T[i][j] \Leftrightarrow$ ćwiczenie i z przedmiotu j zaczynają się o godzinie b ,
kończą się o godzinie e w dzień a , $i = 1 \dots s, j = 1 \dots c$,
 $a \in \{1 \dots d\}, b, e \in \mathbf{R} \cap [0, 24)$
- $L[i][j]$ – preferencja ćwiczeń i z przedmiotu j , $i = 1 \dots s, j = 1 \dots c$

4.1.2 Zmienne decyzyjne

- $X[i][j]$ – 1 gdy wybieramy i ćwiczenia z j przedmiotu, 0 w przeciwnym przypadku,
 $i = 1 \dots s, j = 1 \dots c$

4.1.3 Funkcja celu

Maksymalizujemy preferencje zajęć, na które się zapisujemy:

$$\sum_{i=1}^s \sum_{j=1}^c L[i][j] \cdot X[i][j]$$

4.1.4 Ograniczenia

Wybieramy dokładnie jedno ćwiczenie z danego przedmiotu:

$$(\forall j \in 1 \dots c) \left(\sum_{i=1}^s X[i][j] \right) = 1$$

Nie pozwalamy na kolizje:

$$\begin{aligned} & (\forall k \in 1 \dots d) (\forall i_1 \in 1 \dots s) (\forall j_1 \in 1 \dots c) \\ & \sum_{\substack{i_2 \in 1 \dots s \\ j_2 \in 1 \dots c \\ \text{takie, że:} \\ \neg(i_1 = i_2 \wedge j_1 = j_2) \\ (b_1, e_1, k) \in T[i_1][j_1] \\ (b_2, e_2, k) \in T[i_2][j_2] \\ e_1 > b_2 \wedge e_2 > b_1}} X[i_1][j_1] + X[i_2][j_2] \leq 1 \end{aligned}$$

Nie chcemy więcej, niż 4 godziny zajęć dziennie:

$$\begin{aligned} & \forall (k \in 1 \dots d) \\ & \sum_{\substack{i \in 1 \dots s \\ j \in 1 \dots c \\ \text{takie, że:} \\ (b, e, k) \in T[i][j]}} X[i][j](e - b) \leq 4 \end{aligned}$$

Chcemy również mieć możliwość zjedzenia obiadu. Wprowadzamy więc warunek mówiący: W każdym dniu musimy mieć wolne między 12 a 13 lub między 13 a 14. Implikuje to warunek podany w treści zadania:

$$\begin{aligned} & \forall (k \in 1 \dots d) \\ & \sum_{\substack{i \in 1 \dots s \\ j \in 1 \dots c \\ \text{takie, że:} \\ (b, e, k) \in T[i][j] \\ e > 12 \wedge 13 > b}} X[i][j] + \sum_{\substack{i \in 1 \dots s \\ j \in 1 \dots c \\ \text{takie, że:} \\ (b, e, k) \in T[i][j] \\ e > 13 \wedge 14 > b}} X[i][j] \leq 1 \end{aligned}$$

Chcemy również trenować naszą ulubioną dyscyplinę sportu 1, 2 lub 3 razy w terminach ustalonych w treści zadania. Zatem: zatem:

$$\begin{aligned} & \sum_{\substack{i \in 1 \dots s \\ j \in 1 \dots c \\ \text{takie, że:} \\ ((b, e, 1) \in T[i][j] \wedge e > 13 \wedge 15 > b) \vee \\ ((b, e, 3) \in T[i][j] \wedge e > 11 \wedge 13 > b) \vee \\ ((b, e, 3) \in T[i][j] \wedge e > 13 \wedge 15 > b)}} X[i][j] \leq 2 \end{aligned}$$

4.2 Wyniki

Zamodelowanie i uruchomienie powyższego modelu w *CPLEX Optimization Studio* dało następujące wyniki:

```
// solution (optimal) with objective 37
Day: 1 Course: Chemia mineralow Start: 8 End: 10
Day: 1 Course: Chemia organiczna Start: 10.5 End: 12
Day: 2 Course: Analiza Start: 10 End: 12
Day: 3 Course: Algebra Start: 10 End: 12
Day: 4 Course: Fizyka Start: 17 End: 20
```

Osiągamy sumę preferencji równą 37 przy spełnionych ograniczeniach zdefiniowanych w treści zadania.

Aby dowiedzieć się, czy można plan utworzyć taki, żeby na zajęcia chodzić jedynie w poniedziałek, wtorek i czwartek oraz aby ocena żadnego z przedmiotów nie była mniejsza niż 5, należy zmodyfikować dane. Wystarczy wszystkim zajęciom środowym oraz takim, których poziom preferencji jest mniejszy niż 5, ustawić poziom preferencji na bardzo małą, ujemną liczbę (np. -1000). Otrzymujemy wówczas następujący wynik:

```
// solution (optimal) with objective 28
Day: 1 Course: Algebra Start: 13 End: 15
Day: 1 Course: Chemia organiczna Start: 10.5 End: 12
Day: 2 Course: Fizyka Start: 10 End: 13
Day: 4 Course: Chemia mineralow Start: 13 End: 15
Day: 4 Course: Analiza Start: 8 End: 10
```

Okazuje się więc, że można utworzyć plan również z tak zadanymi, dodatkowymi ograniczeniami. Wartość funkcji celu spada nam jednak do 28.