

Projekty z przedmiotu **Optymalizacja 1**

**Małgorzata Łukomska**  
**Olga Waszczuk**

# PROJEKT 1

Funkcje

Wyniki testów

Obserwacje

# Funkcje

- \* **largest\_coefficient:** zwraca zmienną, przy której stoi największy współczynnik w funkcji celu
- \* **maximal\_objective:** zwraca zmienną na podstawie wzrostu funkcji celu: zmienna wychodząca jest dobierana do zmiennej wchodzącej tak, aby ich zamiana w programie dawała najwyższy wzrost funkcji celu spośród wszystkich możliwych par zmiennych wchodzących i wychodzących
- \* **bland\_rule:** zwraca zmienną wchodzącą o najniższym indeksie, a zmienną wychodzącą dobieraną do zmiennej wchodzącej w zbiorze możliwych zmiennych wychodzących o najmniejszym indeksie
- \* **random\_edge:** zwraca zmienną wybraną losowo z prawdopodobieństwem jednostajnym
- \* **smallest\_coefficient:** zwraca zmienną, przy której w funkcji celu stoi najmniejszy współczynnik.
- \* **max\_bounds\_difference:** zwraca zmienną dla której wartość bezwzględna różnicy współczynnika stojącego przy zmiennej w pierwszej funkcji ograniczającej, i sumy współczynników stojących przy zmiennej w pozostałych funkcjach ograniczających jest największa
- \* **lexicographical\_min:** zwraca najmniejszą zmienną wśród możliwych zmiennych wchodzących i wychodzących do programu, gdzie ułożenie zmiennych podlega pod porządek leksykograficzny.
- \* **lexicographical\_max:** zwraca największą zmienną wśród możliwych zmiennych wchodzących i wychodzących do programu, gdzie ułożenie zmiennych podlega pod porządek leksykograficzny.

# Wyniki testów

Funkcje\testy	routes	paint	transportation	furniture	Whiskas 1	Whiskas 2	Hetmani 5	Hetmani 3	Exercise	profit
largest_Coefficient	5	2	2	2	2	2	16	9	2	5
maximal_objective_value	9	2	2	2	2	7	inf	15	2	5
bland_rule	7	3	2	2	2	11	29	10	3	4
random_edge	6	3	3	2	2	8	14	10	3	6
smallest_Coefficient	6	3	3	2	2	8	20	11	3	7
max_bounds_difference	7	2	2	2	2	10	31	10	2	3
Lexicographical_min	7	3	2	2	2	11	29	10	3	4
lexicographical_max	8	2	3	2	2	2	31	15	2	5

# Obserwacje

Reguła	Średnia liczba kroków
largest_Coefficient	4,7
bland_rule	7,3
random_edge	5,7
smallest_coefficient	6,5
max_bounds_difference	7,1
Lexicographical_min	7,3
lexicographical_max	7,2

- \* reguła **largest\_coefficient**
  - średnio dała najmniej krokówNawet w najbardziej złożonych problemach takich jak hetmani oraz whiskas 2 dała znacząco mniej kroków niż przy pozostałych regułach
- \* reguły **bland\_rule** i **lexicographical\_min**
  - średnio dały najwięcej kroków

# PROJEKT 3

Ogólne o algorytmie

Problem liniowy

Funkcje opisujące struktury

Funkcje wypisujące

# Ogólnie o algorytmie

Prezentacja każdej organizacji w formie drzewa

Połączenie dwóch drzew reprezentujących firmy



Sieć przepływów pomiędzy drzewami



Wygenerowanie problemu liniowego

# Problem liniowy

- \* **Funkcja celu:** Szukamy maksimum po sumie zwolnionych osób
- \* **Ograniczenia:**
  - \* **wydajności (Capacity constraints)** - ograniczenia liczby zwolnionych pracowników w danym departamencie
  - \* **Równowagi (Flow conservation equations)** - suma zwolnień podwładnych dodana do informacji, czy szef jest zwolniony, czy nie, musi być równa informacji idącej od tego szefa do jego przełożonego. W przypadku gdy szef jest korzeniem (nie ma przełożonego) suma ta nie może przekroczyć maksimum zwolnień, dla tego szefa.
- \* **Bounds:**
  - \* Zmienne pierwszego typu  $x_{i,j}$  mogą przyjmować wartości 0 lub 1 odpowiednio pracownik zostaje w firmie lub jest zwolniony
  - \* Zmienne drugiego typu  $x_{i,j,0}$  i  $x_{i,j,1}$  są nieujemne i przyjmują dowolną liczbę naturalną  $n$ , nieprzekraczającą maksimum możliwej liczby zwolnionych podwładnych pracownika  $j$  - tego (wraz z nim).
- \* **Generals:** Wypisanie wszystkich zmiennych

Zmienna pierwszego rodzaju:

"eb\_(indeks pracownika)e\_(indeks pracownika)"

Zmienna drugiego rodzaju:

„eb\_(indeks szefa)e\_(indeks podwładnego)\_(numer organizacji)”



# Funkcje opisujące struktury

\* **count**(tree, index\_pracownika, macierz\_dane\_wejsciowe)

Zlicza liczbę osób zatrudnionych w departamencie pracownika o pobranym indeksie przeszukując macierz z danymi wejściowymi

\* **get employees under boss**(index\_pracownika, tree, matrix\_dane wejsciowe)

zwraca listę podwładnych dla pracownika o pobranym indeksie przeszukując macierz zawierającą dane wejściowe

\* **get employees lists**(tree, matrix\_dane\_wejsciowe)

Na podstawie macierzy zawierającej dane wejściowe zwraca listę opisującą strukturę danej (tree) organizacji, każdy pracownik opisany jest następująco:

\* "Self" - jego własny indeks

\* "Count" - liczba jego podwładnych

\* "ToFire" - maksymalna liczba jego podwładnych (łącznie z nim), która może zostać zwolniona

\* "Tree" - organizacje dla której opisywana jest hierarchia

\* "Boss" - indeks szefa tego pracownika

\* "Employees" - lista podwładnych tego pracownika (bez niego)

# Funkcje wypisujące

- \* **getSelfObjectiveFunction()**  
zwraca tekst, wypisuje sumę wszystkich zmiennych pierwszego typu
- \* **getCapacityConstraint()**  
zwraca tekst, wypisuje nierówności - ograniczenia wydajności
- \* **getConservationConstraint()**  
zwraca tekst, wypisuje ograniczenia równowagi
- \* **getSelfBounds()**  
zwraca tekst, wypisuje ograniczenia dla zmiennych pierwszego typu
- \* **getBounds()**  
zwraca tekst, ograniczenia dla zmiennych drugiego typu
- \* **getSelfGenerals()**  
zwraca tekst, wypisuje zmienne pierwszego typu
- \* **getGenerals**  
zwraca tekst, wypisuje zmienne drugiego typu

```
Maximize
eb_0e_0 + eb_1e_1 + eb_2e_2 + eb_3e_3 + eb_4e_4

Subject To
eb_0e_0 <= 0
eb_1e_1 + eb_1e_0_0 + eb_1e_4_0 <= 2
eb_2e_2 + eb_2e_1_0 + eb_2e_3_0 <= 3
eb_3e_3 <= 1
eb_4e_4 <= 1
eb_0e_0 + eb_0e_1_1 <= 3
eb_1e_1 + eb_1e_2_1 + eb_1e_3_1 <= 2
eb_2e_2 <= 1 eb_3e_3 + eb_3e_4_1 <= 1
eb_4e_4 <= 1

eb_0e_0 - eb_1e_0_0 = 0
eb_1e_1 + eb_1e_0_0 + eb_1e_4_0 - eb_2e_1_0 = 0
eb_2e_2 + eb_2e_1_0 + eb_2e_3_0 <= 3
eb_3e_3 - eb_2e_3_0 = 0
eb_4e_4 - eb_1e_4_0 = 0
eb_0e_0 + eb_0e_1_1 <= 3
eb_1e_1 + eb_1e_2_1 + eb_1e_3_1 - eb_0e_1_1 = 0
eb_2e_2 - eb_1e_2_1 = 0
eb_3e_3 + eb_3e_4_1 - eb_1e_3_1 = 0
eb_4e_4 - eb_3e_4_1 = 0

Bounds
0 <= eb_0e_0 <= 1
0 <= eb_1e_1 <= 1
0 <= eb_2e_2 <= 1
0 <= eb_3e_3 <= 1
0 <= eb_4e_4 <= 1

Generals
eb_0e_0
eb_1e_1
eb_2e_2
eb_3e_3
eb_4e_4
eb_1e_0_0
eb_1e_4_0
eb_2e_1_0
eb_2e_3_0
eb_0e_1_1
eb_1e_2_1
eb_1e_3_1
eb_3e_4_1
End

0 <= eb_1e_0_0 <= 0
0 <= eb_1e_4_0 <= 1
0 <= eb_2e_1_0 <= 2
0 <= eb_2e_3_0 <= 1
0 <= eb_0e_1_1 <= 2
0 <= eb_1e_2_1 <= 1
0 <= eb_1e_3_1 <= 1
0 <= eb_3e_4_1 <= 1
```