Przegląd modeli analitycznych stosowanych w systemach zarządzania, wprowadzenie do modelowania, klasyfikacja modeli
W5

dr inż. Janusz Granat

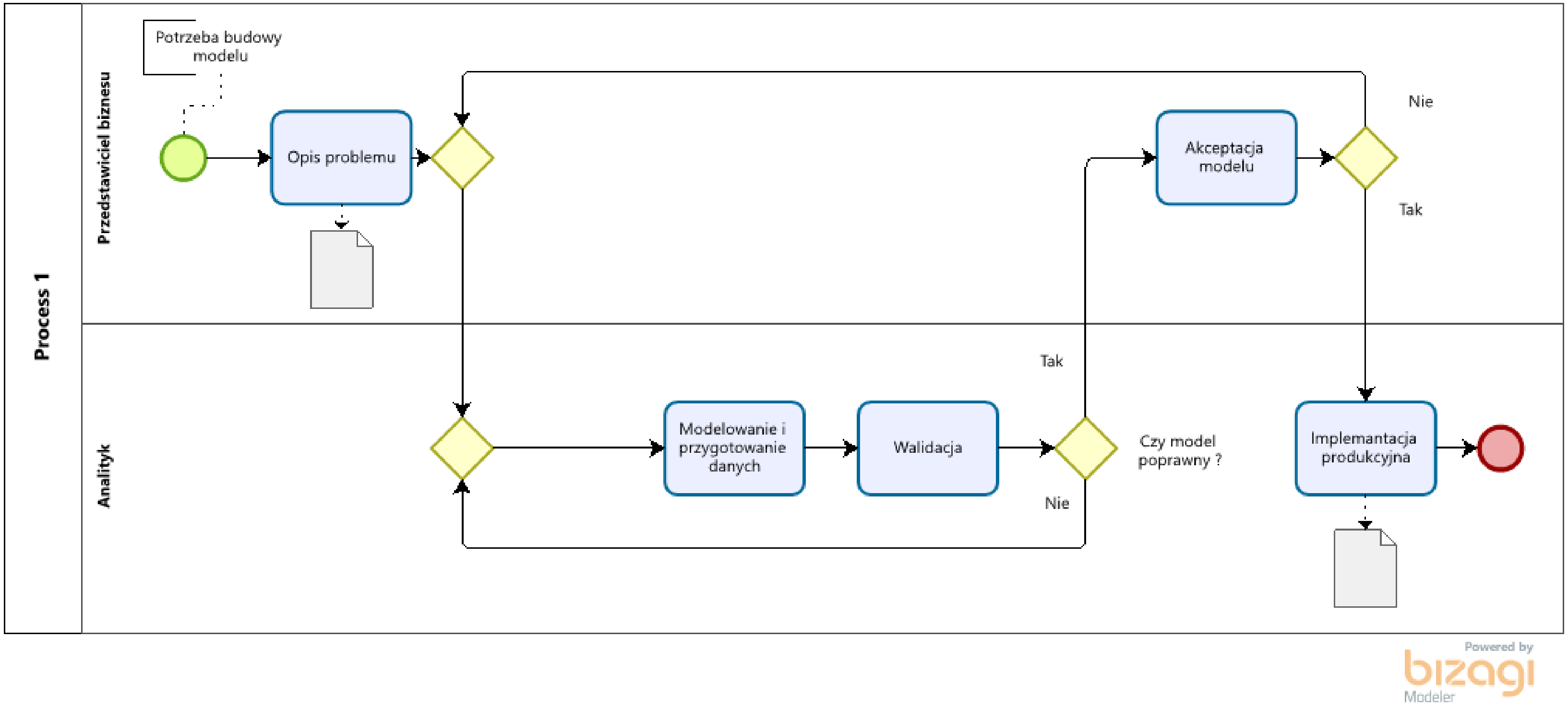
Plan wykładu

- Definicja modelu
- Proces modelowania
- Zadania analizy modelu
- Sformułowania modeli optymalizacji
- Zastosowania w zarządzaniu produkcją i usługami

Model - definicja

- Pojęcie „**model**” jest wykorzystywane w różnych dziedzinach. Niniejszy wykład jest wprowadzeniem do zagadnień dotyczących wykorzystania modeli matematycznych (algebraicznych) w systemach zarządzania
- **Model matematyczny** jest odzwierciedleniem fragmentu rzeczywistości w formie zależności matematycznych

Proces modelowania matematycznego



Model rzeczowy

$$\boldsymbol{y} = f(\boldsymbol{x}, \boldsymbol{p})$$ *- model rzeczowy*

$$f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p}) = \boldsymbol{0}$$ *(alternatywny zapis)*

$\boldsymbol{x} \in \boldsymbol{X}$ $-$ *wektor zmiennych decyzyjnych*

$\boldsymbol{p} \in \boldsymbol{P}$ $-$ *wektor parametrów*

$\boldsymbol{y} \in \boldsymbol{Y}$ $-$ *wektor zmiennych wyjściowych*

$\boldsymbol{X}$ $-$ *zbiór decyzji dopuszczalnych*

$\boldsymbol{Y}$ $-$ *zbiór ograniczeń na zmienne wyjściowe*

$\boldsymbol{P}$ $-$ *zbiór ograniczeń na parametry*

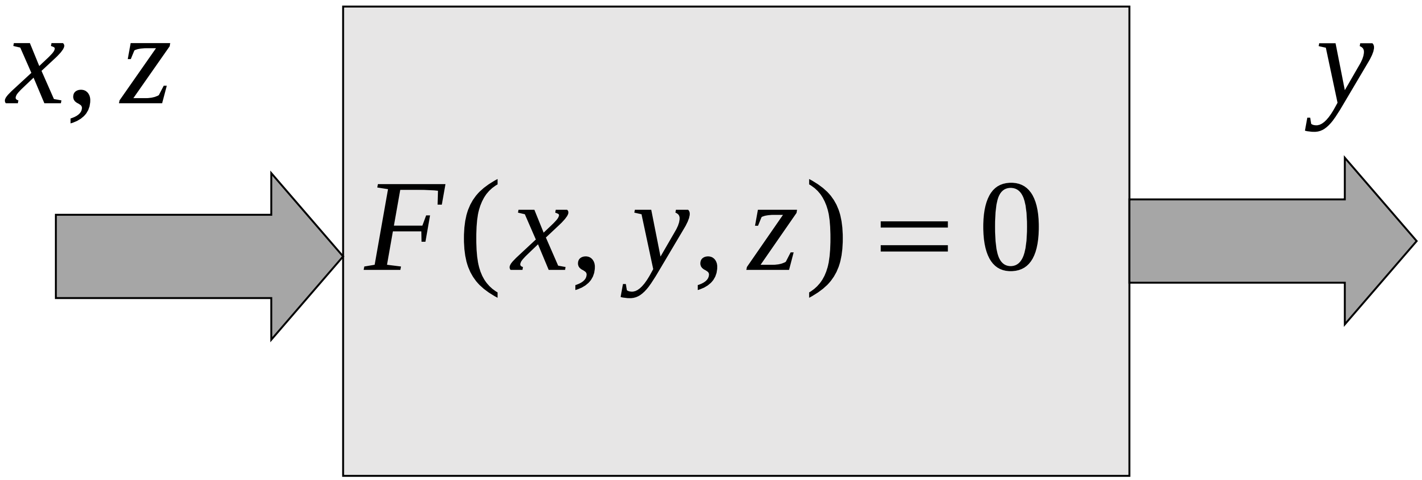
Inne modele

- Modele prognozowania
- Modele symulacji dyskretnej (Event based simulation)
- …

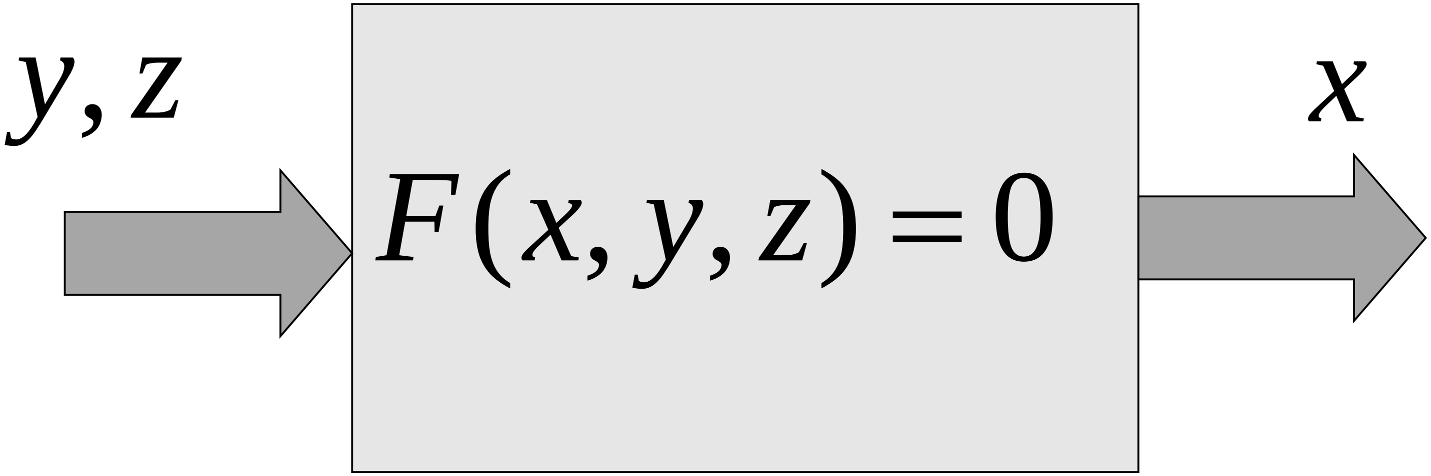
Zadania analizy modelu

- Symulacja
- Symulacja odwrotna
- Optymalizacja jednokryterialna
- Analiza wielokryterialna

Symulacja



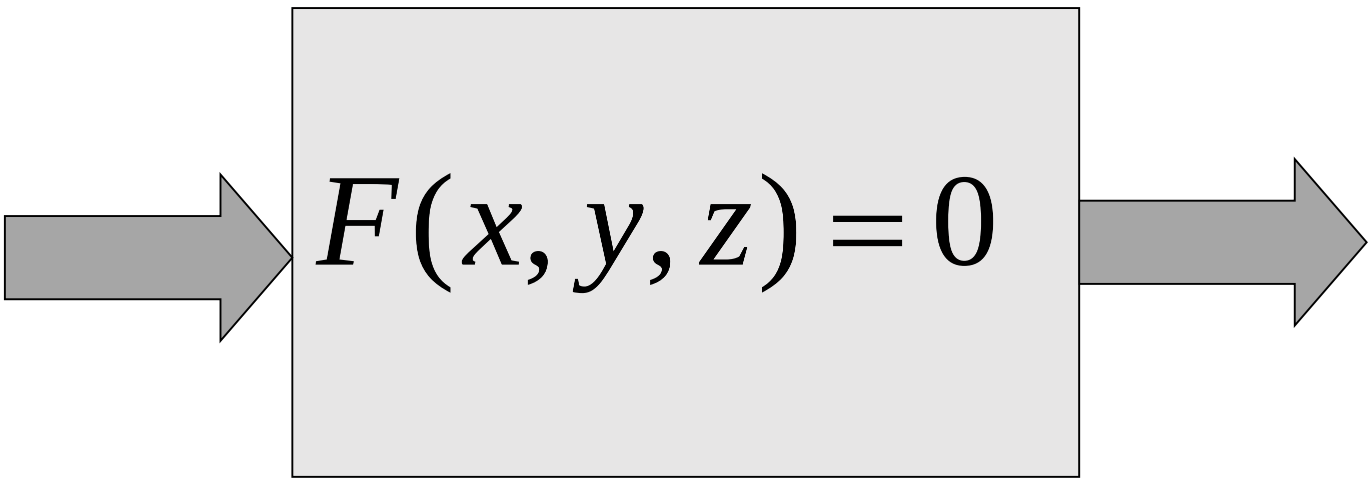
Symulacja odwrotna



Pomocnicze zadanie optymalizacji:

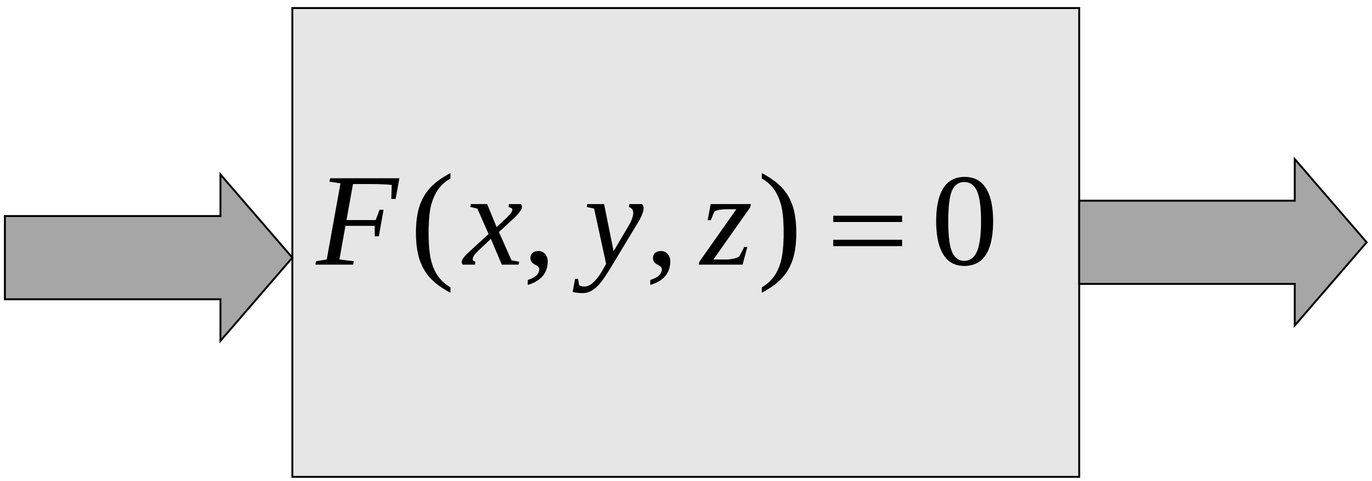
$$\min_{x \in X} \| y - \widetilde{y} \| + \varrho \| x - \tilde{x} \|$$

Optymalizacja



$$\min_{x \in X} \; q_i$$

Analiza wielokryterialna



$$\min_{x \in X} \boldsymbol{q}$$

Klasyfikacja modeli

- Statyczne
  - liniowe
  - dyskretne
  - nieliniowe
- Dynamiczne
- Stochastyczne
- Inne (np. wykorzystujące zbiory rozmyte)

Zadanie liniowe

$f(x) = c_1 x_1 \;\; + c_2 x_2$ +…+$c_n x_n$ -> minimum/maximum

$a_{11}x_1 + a_{12}x_2$+…+$a_{1n}x_n \; (\geq, =, \leq) \; b_1$

$a_{21}x_1 + a_{22}x_1$+…+$a_{2n}x_n \; (\geq, =, \leq) \; b_2$

… … … … … … … … … … … … … … …

$a_{m1}x_1 + a_{m2}x_2$+…+$a_{mn}x_n \; (\geq, =, \leq) \; b_m$

$x_i \geq 0 \; \forall i = 1, \ldots, n$

Zadanie liniowe – zapis macierzowy

$$\min_{\mathbf{x} \in \mathbf{X}} \mathbf{c}^T \mathbf{x}$$

$$\boldsymbol{X} = \{\mathbf{x} \in \mathbb{R}^n: \quad \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \; \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

Zadanie całkowitoliczbowe

$$\min_{\mathbf{x} \in \mathbf{X}} \mathbf{c}^T \mathbf{x}$$

$$\boldsymbol{X} = \{\mathbf{x} \in \mathbb{R}^n: \quad \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \ \in \mathbb{Z}^{0+}\}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \ \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{mn} \end{bmatrix}$$

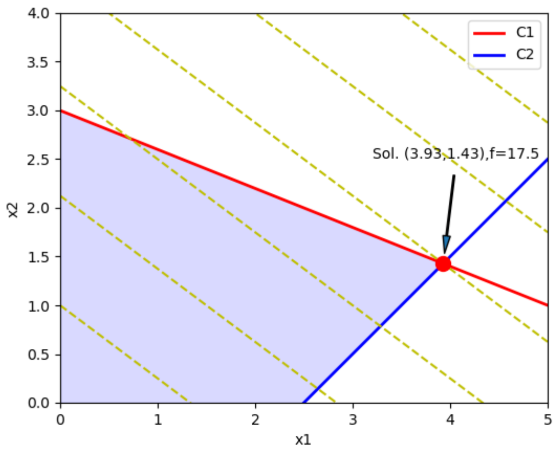
$$\mathbb{Z}^{0+} = \{0,1,2, \dots\}$$

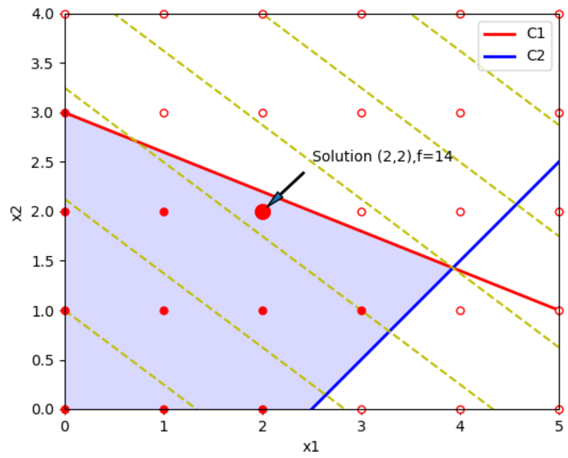
Przykład programowania liniowego i całkowitoliczbowego

$$\max_{x1,x2} 3x_1 + 4x_2$$

$$2x_1 + 5x_2 \leq 15$$

$$2x_1 - 2x_2 \leq 5$$





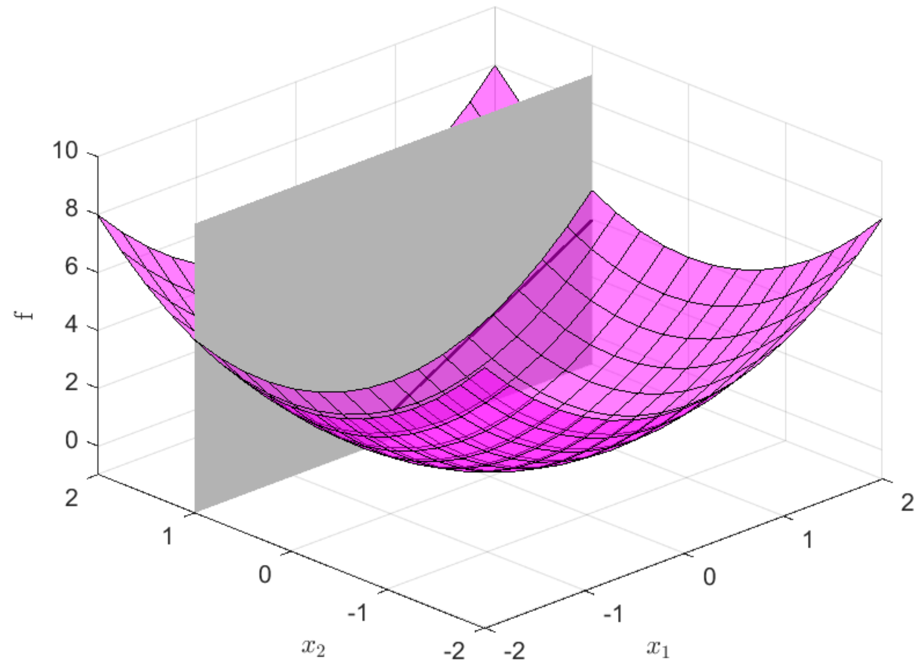
$$x_1, x_2 \geq 0, \textcolor{red}{całkowitoliczbowe}$$

Zadanie optymalizacji nieliniowej

$$\min_{\mathbf{x} \in \boldsymbol{Q}} \mathrm{f}(\mathbf{x})$$

$$\boldsymbol{Q} = \{\mathbf{x} \in \mathbb{R}^n: \quad \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\}$$

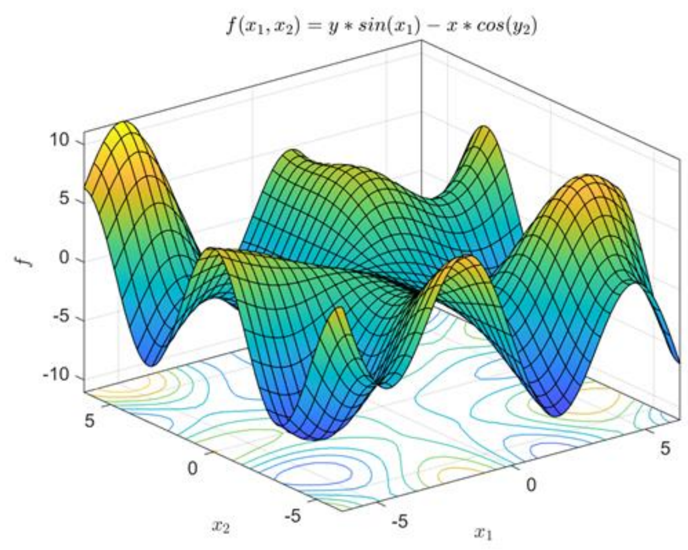
$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$



Poziomica lub warstwica funkcji

Poziomica lub warstwica

to zbiór punktów dziedziny funkcji rzeczywistej wielu zmiennych, dla których przyjmuje ona tę samą wartość.

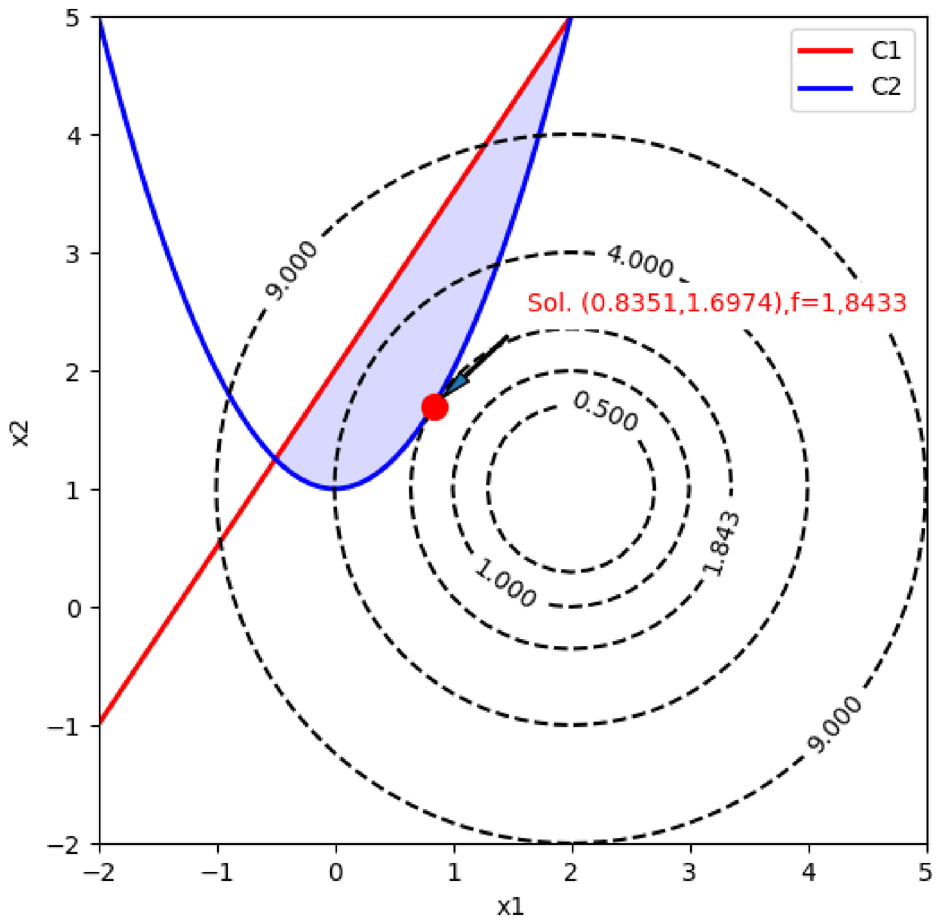


Przykład zadania optymalizacji nieliniowej

$$\max_{x_1,x_2} (x_1-2)^2 + (x_2-1)^2$$

$$x_2 \le \frac{3}{2}x_1 + 2$$

$$x_2 \ge x_1^2 + 1$$



Modelowanie niepewności

- Zbiory rozmyte
- Modele stochastyczne

"Operations Research" vs optimization

- Badania operacyjne (*Operational Research*) ukierunkowane są na zastosowania w przemyśle i usługach
- Optymalizacja ma szersze zastosowania (np. sztuczna inteligencja)

Wykorzystanie modeli w procesach zarządzania

Model musi być włączony w proces biznesowy, chyba że jest to jednorazowe zadanie planowania lub analizy strategicznej

Modele w systemach zarządzania (wybrane, często stosowane)

- Modele stosowane w planowaniu i realizacji produkcji i usług
  - Planowanie
    - Problem lokalizacji i dystrybucji (z punktu widzenia planowania)
  - Zadania operacyjne
    - Problem pakowania

- Harmonogramowanie i szeregowanie zadań

- Modele dystrybucji (operacyjne)
  - Najkrótsza ścieżka
  - Sieci przepływowe
  - Zadanie transportowe

Sarker, Ruhul & Newton, C.S.. (2007). Optimization modelling: A practical approach.

Narzędzia do modelowania i rozwiązywania zadań optymalizacji

- Narzędzia do modelowania
  - AMPL (A Mathematical Programming Language)
  - AIMMS (Advanced Interactive Multidimensional Modeling System)
  - GAMS (General Algebraic Modeling System)
  - GNU MathProg (podobny do AMPL, Interfejs Gusek)
  - Pyomo
- Solwery optymalizacyjne
  - MINOS
  - CPLEX
  - GUROBI
  - GLPK

Przykład modelowania- sformułowanie problemu

Firma wytwarza 2 produkty (P1,P2) i wykorzystuje do tego celu 3 maszyny (M1,M2,M3). Maszyna M1 jest dostępna w planowanym okresie 4 godziny, maszyna M2 – 24 godziny i maszyna M3 - 35 godzin. Wyprodukowanie jednego produktu P1 zajmuje 1 godz. maszyny M1, 3 godziny maszyny M2 i 10 godzin maszyny M3. Wyprodukowanie jednego produktu P2 zajmuje 1 godz. maszyny M1, 8 godzin maszyny M2 i 7 godzin maszyny M3. Produkt P1 przynosi 5 PLN zysku a produkt P2 7 PLN. Należy dobrać wielkość produkcji produktów P1 i P2.

Podsumowanie dostępnych danych

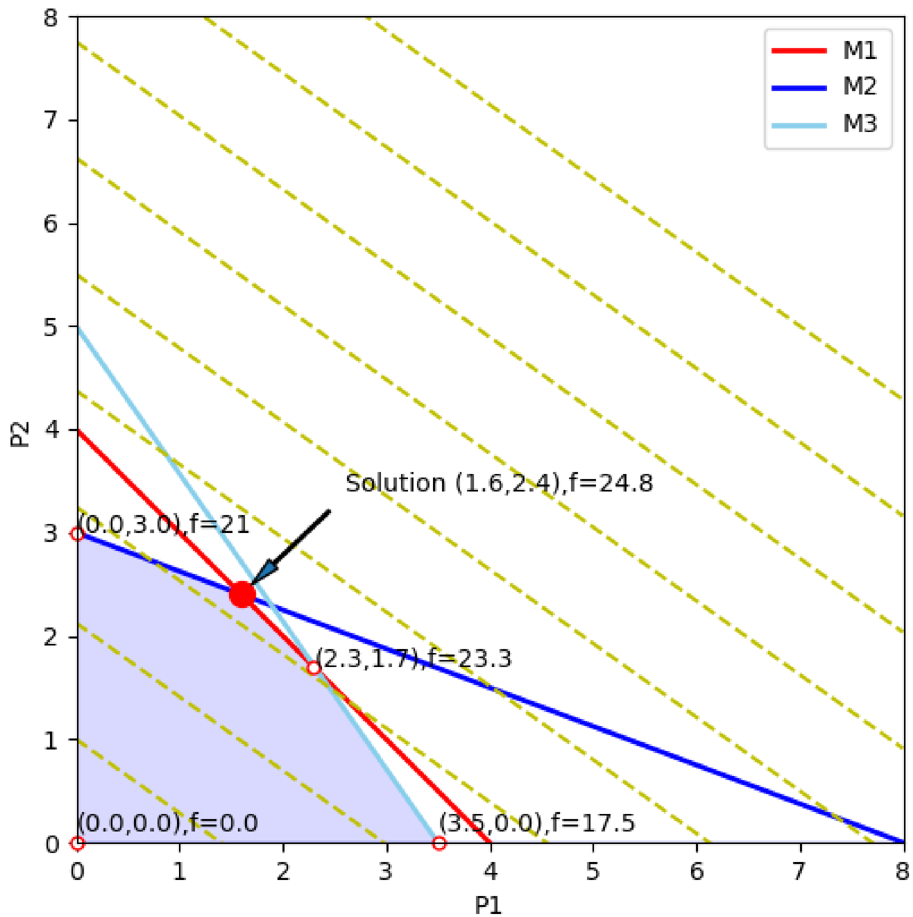
Maszyny	Czas potrzebny na wyprodukowanie produktu (h)		Dostępność maszyn (h)
	X	Y	
A	1	1	4
B	3	8	24
C	10	7	35
Zysk	5	7	

Model optymalizacyjny

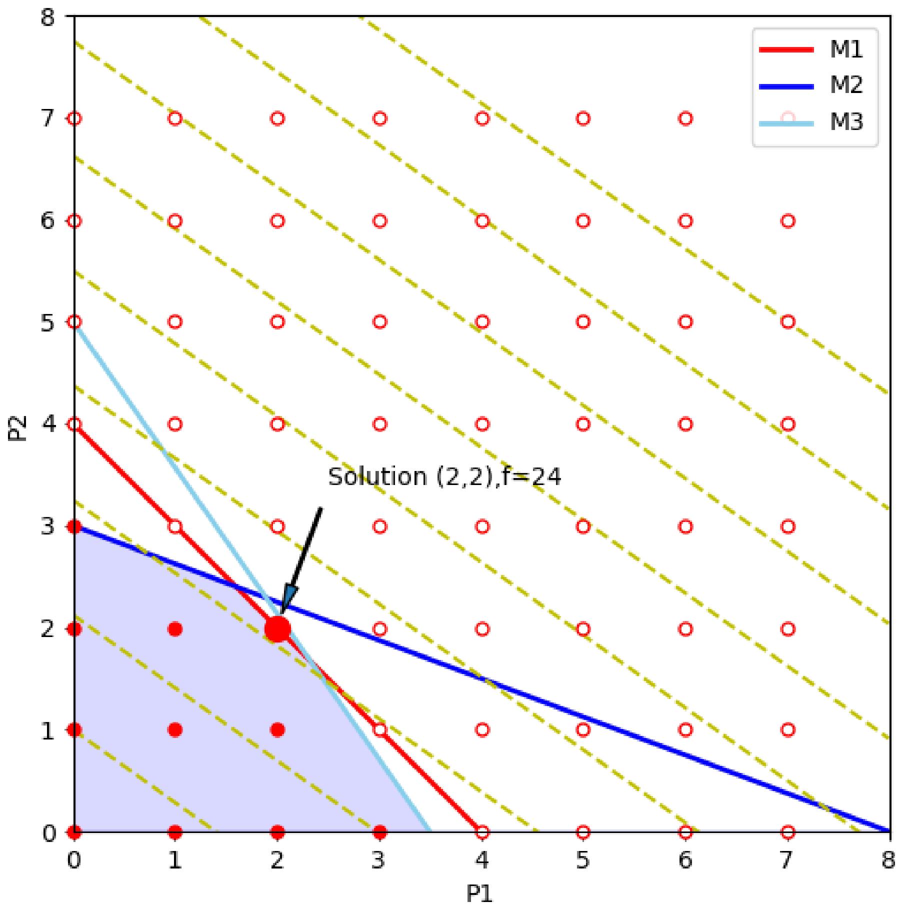
$$\max_{x,y} 5 * P1 + 7 \ * P2$$
$$P1 + P2 \ \leq 4$$
$$3 * P1 + 8 * P2 \leq 24$$
$$10 * P1 + 7 * P2 \leq 35$$
$$P1 \geq 0, \qquad P2 \geq 0$$

lub P1,P2: calkowitoliczbowe

Zbiór rozwiązań dopuszczalnych (rozwiązanie w zbiorze liczb rzeczywistych)



Zbiór rozwiązań dopuszczalnych (rozwiązanie całkowitoliczbowe)



Model

- Specyfikacja symboliczna modelu (zbiory, indeksy, parametry, zmienne decyzyjne, równania funkcji celu i ograniczeń)
- Dane
- Instancja modelu (połączenie specyfikacji modelu z wybranym zbiorem danych, dla tej samej specyfikacji i różnych zbiorów danych mamy różne instancje)

Specyfikacja modelu

```
# AMPL Specyfikacja modelu
# Deklaracja zbiorów i parametrów (sets and parameters)
set PRODUKTY;
set MASZYNY;
param czas_produkcji {MASZYNY, PRODUKTY} >= 0;
param dostepnosc {MASZYNY} >= 0;
param zysk {PRODUKTY} >= 0;
# Deklaracja zmiennych (variables)
var liczba_produktow {PRODUKTY} >= 0; #integer;
# Funkcja celu (Objective)
maximize suma_zysk: sum {k in PRODUKTY} zysk[k] * liczba_produktow[k];
#  Ograniczenia
subject to M {m in MASZYNY}: sum{k in PRODUKTY}
czas_produkcji[m,k]*liczba_produktow[k] <=dostepnosc[m];
```

Dane

```
data;
# AMPL DATA file
# Indeksy (Index sets definitions)
set PRODUKTY := P1 P2 ;
set MASZYNY := M1 M2 M3 ;
# czasy produkcji
param czas_produkcji:  P1   P2   :=
M1  1  1
M2  3  8
M3  10 7
;
```

```
# dostepnosc
param: dostepnosc  :=
M1  4
M2  24
M3  35
;
# zysk jednostkowy
param: zysk  :=
P1  5
P2  7 ;
end;
```

Wynik optymalizacji (*GNU MathProg*)

GLPSOL: GLPK LP/MIP Solver, v4.65

Parameter(s) specified in the command line:

--cover --clique --gomory --mir -m prod_dat_PL.mod -d prod_dat_PL.dat

Reading model section from prod_dat_PL.mod...

prod_dat_PL.mod:27:

27 lines were read

Reading data section from prod_dat_PL.dat...

30 lines were read

Generating suma_zysk...

Generating M...

Model has been successfully generated

GLPK Simplex Optimizer, v4.65

4 rows, 2 columns, 8 non-zeros

Preprocessing...

3 rows, 2 columns, 6 non-zeros

Scaling...

A: min|aij| = 1.000e+00 max|aij| = 1.000e+01 ratio = 1.000e+01

Problem data seem to be well scaled

Constructing initial basis...

Size of triangular part is 3

* 0: obj = -0.000000000e+00 inf = 0.000e+00 (2)

* 2: obj = 2.480000000e+01 inf = 0.000e+00 (0)

OPTIMAL LP SOLUTION FOUND

Time used: 0.0 secs

Memory used: 0.1 Mb (120672 bytes)

Display statement at line 24

liczba_produktow[P1].val = 1.6

liczba_produktow[P2].val = 2.4

Display statement at line 25

suma_zysk.val = 24.8

Model has been successfully processed

>Exit code: 0 Time: 0.238

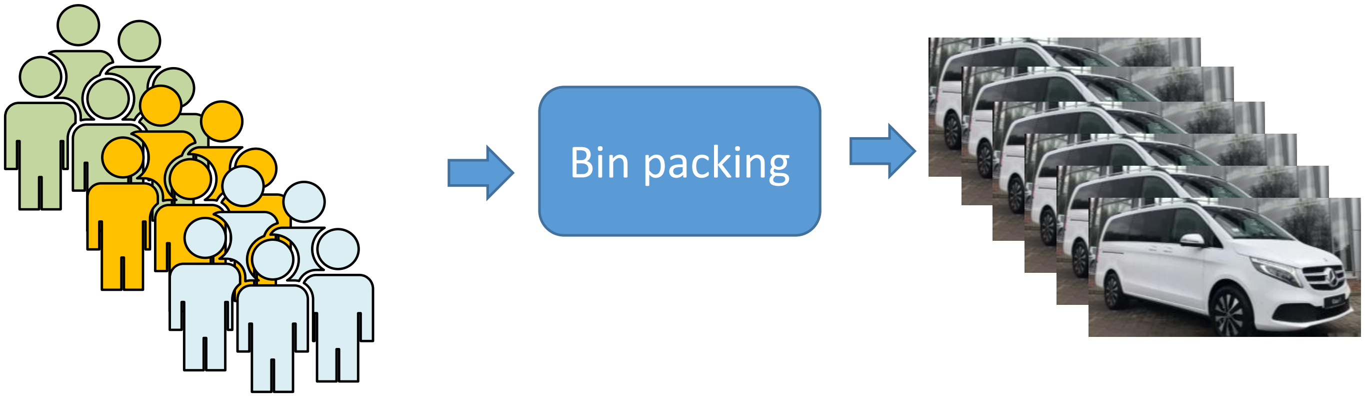
Przykład modelu stosowanego operacyjnie – zadanie pakowania pojemników (jednowymiarowe)

- $I\ - zbiór\ obiektów,\ V\ - zbiór\ pojemników$
- $i \in I -\ indeks\ obiektu,\ j \in V\ - indeks\ pojemnika$
- $B\ - pojemność\ pojemnika$
- $S(i) - rozmiar\ obiektu\ i$
- Zmienne decyzyjne
  - $y(j) = 1\ jeśli\ pojemnik\ j\ jest\ użyty, 0\ w\ przeciwnym\ razie$
  - $x(i,j) = 1\ jeśli\ obiekt\ i\ jest\ przypisany\ do\ pojenika\ j, 0\ w\ przeciwnym\ razie$
- Ograniczenia
  - $\sum_{i \in I} s(i)x(i,j)\ \le B\ y(j), \forall j \in V$ (obiekty w pojemniku nie mogą przekroczyć jego pojemności)
  - $\sum_{i \in I, j \in V} x(i,j) = 1\ \ (obiekt\ jest\ przypisany\ tylko\ raz\ do\ pojemnika)$
- Kryterium

$$\textbf{minimize} \sum_{j \in V} y(j)$$

Problem do rozwiązania

- Mamy 6 7-osobowych vanów
- Mamy grupy zwiedzające Warszawę o liczebności w przedziale [1,7] -> G1 - 2 , G2 - 3, G3 – 4 G4 – 7, G5 – 5
- Należy znaleźć optymalny przydział grup od vanów (kryterium minimalizacja zajętych samochodów)

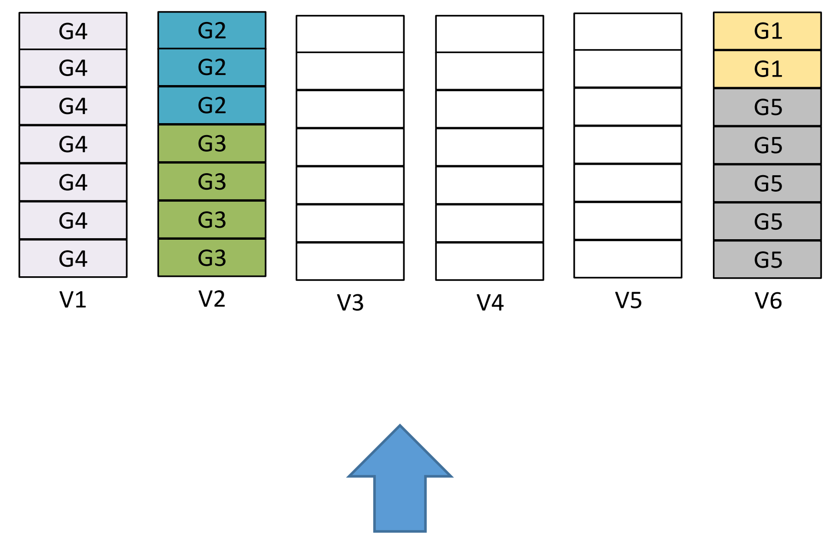


Model w AMPL i wynik

```
# zbiory
set I; #grupy pasażerów
set V; # samochody typu van B osobowe
# parametry
param B; # ilosc osob w vanie
param s {I} >=0;
# zmienne
var x {i in I, j in V} binary;
var y {V} binary;
# ograniczenia
subject to PERSON {j in V}:
sum {i in I} s[i]*x[i,j] - B*y[j] = 0;
subject to EXCLUDE {i in I}:
sum {j in V} x[i,j] = 1;

# funkcja celu
minimize K:
sum {j in V} y[j];
```

```
#dane
data;
param B:=7;
set V := V1 V2 V3 V4 V5 V6;
param: I: s :=
  G1 2
  G2 3
  G3 4
  G4 7
  G5 5;
```



Wynik optymalizacji

Pyomo vs AMPL

```
# zbiory
set I; #grupy pasażerów
set V; # samochody typu van B osobowe
# parametry
param B; # ilosc osob w vanie
param s {I} >=0;
# zmienne
var x {i in I, j in V} binary;
var y {V} binary;
# ograniczenia
subject to PERSON {j in V}:
sum {i in I} s[i]*x[i,j] - B*y[j] = 0;
subject to EXCLUDE {i in I}:
sum {j in V} x[i,j] = 1;

# funkcja celu
minimize K:
sum {j in V} y[j];
```

```
from pyomo.environ import *
model = AbstractModel()
# zbiory
model.I = Set() #grupy pasażerów
model.V = Set() # samochody typu van B osobowe
# parametry
model.B = Param() # ilosc osob w vanie
model.s = Param(model.I)
# zmienne
model.X = Var(model.I, model.V, within=Binary)
model.Y = Var(model.V, within=Binary)

# funkcja celu
def objective_rule(m):
   return summation(m.Y)
model.Total_Vans = Objective(rule=objective_rule,
                             sense=minimize )
# Person
def c1_rule(model, j):
   return sum(model.s[i]*model.X[i,j] for i in model.I)-model.B*model.Y[j]  == 0
model.c1 = Constraint(model.V, rule=c1_rule)
# Exclude
def c2_rule(model, i):
   return  sum(model.X[i, j] for j in model.V) == 1
model.c2 = Constraint(model.I, rule=c2_rule)
data = DataPortal()
data.load(filename='bin_packing.dat')
instance = model.create_instance(data)
opt = SolverFactory('glpk')
opt.solve(instance)
#display results
instance.display()
```